ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Design of Digital Finite Impulse Response Filter Using Different Low Power Mulitipliers

M.Renuka [1], M.Nikhila [2], S.Dinesh Reddy [3]

[1] Assistant Professor, ECE, Vidya Jyothi Institute of Technology, Aziz Nagar, India
[2] B.Tech Student-ECE, Vidya Jyothi Institute of Engineering &Technology, Aziz Nagar, India
[3] B.Tech Student-ECE, Vidya Jyothi Institute of Engineering &Technology, Aziz Nagar, India
renu_404@gmail.com

Abstract

Low Multipliers and Adders are used 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 reduce power consumption and Distortion is also reduced.

The proposed Finite Impulse Response (FIR) filter consists of delays, multipliers, adders. Multiplier consumes more power to reduce the power consuming designed multiplier using four different techniques. The multiplying operation is the basic operation in the FIR filter and is reduced by using these four techniques.

In this project Finite Impulse Response (FIR) filter is design by 8-tap, I Implemented 8-tap to 12-tap filter and designed for different multipliers for 12-tap FIR filter, and also comparing powers of proposed FIR filter with implemented FIR filter. When the MAC operation is reduced in the filter automatically the power is also reduced. The proposed Finite Impulse Response (FIR) filter were synthesized and implemented using Xilinx ISE 9.2i on Spartan 3E and power is analyzed using Xilinx Power analyzer.

Keywords: FIR FILTER, SPARTAN 3E, MAC, MULTIPLIER, LOW POWER, ADDERS

I. INTRODUCTION:

In the early 1970s designing for high speed and minimum area, especially in memories, were the main design constraints. Most of the EDA tools were created to meet these criteria and papers published advertised lower delays and smaller structures. Power consumption was a part of the design process, but was less visible. However, reaching structures below 0.8 mm and having high-performance-chips work in portable devices, power dissipation has become the main design concern.

In the past, the major concerns of the VLSI designer were area, performance, cost and reliability; power consideration was mostly of only secondary importance. In recent years, however, this has begun to change and, increasingly, power is being given comparable weight to area and speed considerations. Several factors have contributed to this trend. Perhaps the primary driving factor has been the remarkable success and growth of the class of personal computing devices (portable desktops, audio- and video-based multimedia products) and wireless communications systems (personal digital assistants and personal communicators) which demand high-speed computation and complex functionality with low power consumption. In these applications, average power consumption is a critical design concern.

The projected power budget for a battery-powered, portable multimedia terminal, when implemented using off-the-shelf components not optimized for low-power operation, is about 40 W. In the absence of low-power design techniques then, current and future portable devices will suffer from either very short battery life or very heavy battery pack.

There also exists a strong pressure for producers of high-end products to reduce their power consumption. The cost associated with packaging and cooling such devices is prohibitive. Since core power consumption must be dissipated through the packaging, increasingly expensive packaging and cooling strategies are required as chip power consumption increases. Consequently, there is a clear financial advantage to reducing the power consumed in high performance systems.

In addition to cost, there is the issue of reliability. High power systems often run hot, and high temperature tends to exacerbate several silicon failure mechanisms. Every 10 °C increase in operating temperature roughly doubles a component's failure rate. In this context, peak power

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

(maximum possible power dissipation) is a critical design factor as it determines the thermal and electrical limits of designs, impacts the system cost, size and weight, dictates specific battery type, component and system packaging and heat sinks, and aggravates the resistive and inductive voltage drop problems. It is therefore essential to have the peak power under control.

Another crucial driving factor is that excessive power consumption is becoming the limiting factor in integrating more transistors on a single chip or on a multiple-chip module. Unless power consumption is dramatically reduced, the resulting heat will limit the feasible packing and performance of VLSI circuits and systems.

Low Power Multiplier Design has been an important part in low-power VLSI system design. For getting the low power low area architecture, the modifications made to the conventional architecture consist of the reduction in switching activities of the major blocks of the multiplier, which includes the reduction in switching activity of the adder and counter. This project implements three different techniques to reduce the dynamic power and area.

Multiplication is the key in arithmetic operation and multiplier plays an important role in digital signal processing. Unfortunately, the major source of power dissipation in digital signal processors is multipliers.Today every circuit has to face the power consumption issue for both portable device aiming at large battery life and high end circuits avoiding cooling packages and reliability issues that are too complex. It is generally accepted that during logic synthesis power tracks well with area. This means that a larger design will generally consume more power. The multiplier is an important kernel of digital signal processors. Because of the circuit complexity, the power consumption and area are the two important design considerations of the multiplier.

In this paper a low power low area architecture for the shift and add multiplier is proposed. For getting the low power low area architecture, the modifications made to the conventional architecture consist of the reduction in switching activities of the major blocks of the multiplier, which includes the reduction in switching activity of the adder and counter.

2. Adders:

The saying goes that if you can count, you can control. Addition is a fundamental operation for any digital system, digital signal processing or control system. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Adders are also very important component in digital systems because of their extensive use in other basic digital operations such as subtraction, multiplication and division. Hence, improving performance of the digital adder would greatly advance the execution of binary operations inside a circuit compromised of such blocks. The performance of a digital circuit block is gauged by analyzing its power dissipation, layout area and its operating speed.

2.1 Basic Adder Unit: The most basic arithmetic operation is the addition of two binary digits, i.e. bits. A combinational circuit that adds two bits, according the scheme outlined below, is called a half adder. A full adder is one that adds three bits, the third produced from a previous addition operation. One way of implementing a full adder is to utilizes two half adders in its implementation. The full adder is the basic unit of addition employed in all the adders studied here.

2.1.1 Half Adder:

A half adder is used to add two binary digits together, A and B. It produces S, the sum of A and B, and the corresponding carry out Co. Although by itself, a half adder is not extremely useful, it can be used as a building block for larger adding circuits (FA). One possible implementation is using two AND gates, two inverters, and an OR gate instead of a XOR gate as shown in Fig. 2.1

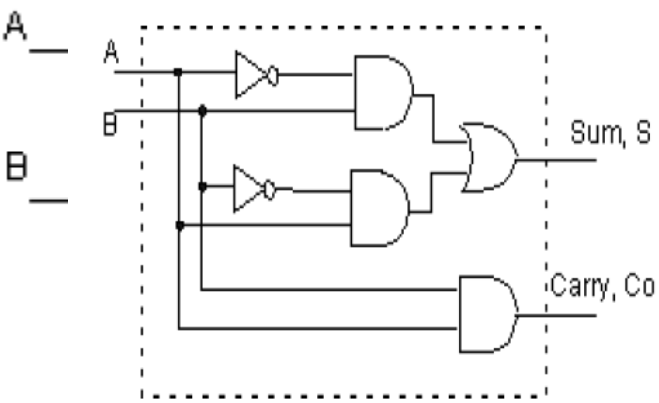


Figure. 2.1 Half Adder

A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 2.1 Half Adder truth table

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Boolean Equations:

S = A'B + AB'

Co = AB

2.1.2 Full Adder

A full adder is a combinational circuit that performs the arithmetic sum of three bits: A, B and a carry in, C, from a previous addition, Fig. 2.1. Also, as in the case of the half adder, the full adder produces the corresponding sum, S, and a carry out Co.As mentioned previously a full adder maybe designed by two half adders in series as shown below in Figure 2.2 The sum of A and B are fed to a second half adder, which then adds it to the carry in C (from a previous addition operation) to generate the final sum S. The carry out, Co, is the result of an OR operation taken from the carry outs of both half adders. There are a variety of adders in the literature both at the gate level and transistor level each giving different performances

Boolean equation:

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

$$C_o = AB + C(A \oplus B)$$

A	*B*	*C*	*S*	*Co*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

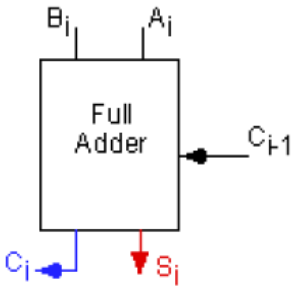


Table 2.2 Full Adder truth table

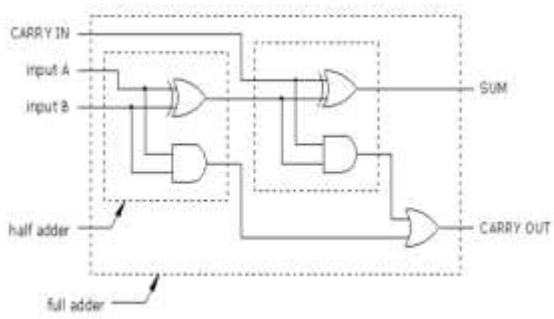


Figure. 2.2 Full adder

2.3 Parallel Adders:

Parallel adders are digital circuits that compute the addition of variable binary strings of equivalent or different size in parallel. The schematic diagram of a parallel adder is shown below in Fig. 2.3.

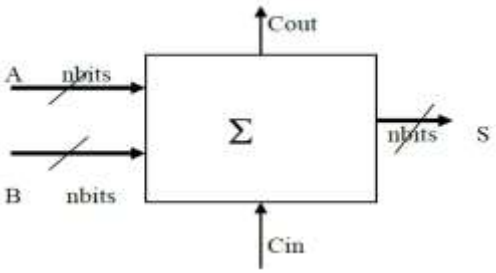


Figure. 2.3 Block diagram of parallel adder

2.4 Ripple Carry adder:

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.

A number of full adders may be added to the ripple carry adder or ripple carry adders of different sizes may be cascaded in order to accommodate binary vector strings of larger sizes. For an n-bit parallel adder, it requires n computational elements (FA). Figure 4 shows an example of a parallel adder: a 4-bit ripple-carry adder. It is composed of four full adders. The augend"s bits of x are added to the addend bits of y respectfully of their binary position. Each bit 6 addition creates a sum and a carry out. The carry out is then

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

transmitted to the carry in of the next higher-order bit. The final result creates a sum of four bits plus a carry out (c4).

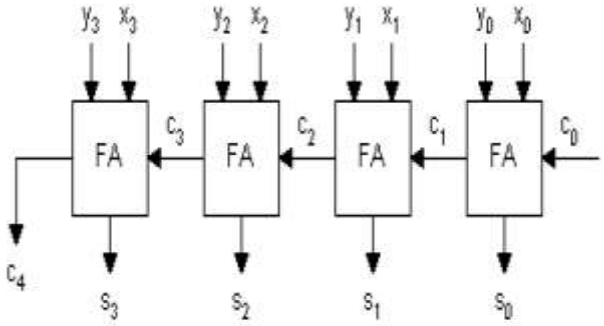


Figure. 2.4 Ripple carry adder

Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used. one of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. As mentioned before, each full adder has to wait for the carry out of the previous stage to output steady-state result. Therefore even if the adder has a value at its output terminal, it has to wait for the propagation of the carry before the output reaches a correct value as shown in Fig. 5. Taking again the example in figure 4, the addition of x4 and y4 cannot reach steady state until c4 becomes available. In turn, c4 has to wait for c3, and so on down to c1. If one full adder takes Tfa seconds to complete its operation, the final result will reach its steady-state value only after 4.Tfa seconds.

Its area is n AfaA (very) small improvement in area consumption can be achieved if it is known in advance that the first carry in (c_0) will always be zero. (If so, the first full adder can be replace by a half adder). In general, assuming all gates have the same delay and area of NAND-2 then this circuit has 3n Tgate delay and 5nAgate.Generally speaking, the worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$t=(n-1)t_c+t_s$$

where tc is the delay through the carry stage of a full adder, and ts is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly proportional to n, the number of bits, therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as a compact layout giving smaller chip area.To design a larger adder ripple carry adders are cascaded.

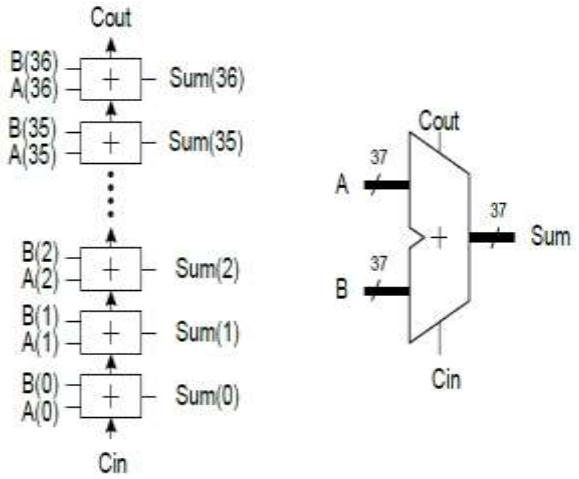


Fig. 2.5 cascaded ripple carry adder

As of today standards, it is a common philosophy that area can be traded off in order to achieve higher speed. This will be shown in the next sections by presenting alternative methods that are based on pre-determining the carry signal of a number of stages based only on the input values.

3.Multipliers:

3.1Existingmethod:

3.1.1 Conventional Multiplier: A Binary multiplier is an electronic hardware device used in digital electronics or a computer or other electronic device to perform rapid multiplication of two numbers in binary representation. It is built using binary adders. The rules for binary multiplication can be stated as follows

1. If the multiplier digit is a 1, the multiplicand is simply copied down and represents the product.
2. If the multiplier digit is a 0 the product is also 0.

For designing a multiplier circuit we should have circuitry to provide or do the following three things
1. It should be capable identifying whether a bit is 0 or 1.
2. It should be capable of shifting left partial products.
3. It should be able to add all the partial products to give the products as sum of partial products.
4. It should examine the sign bits.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

If they are alike, the sign of the product will be a positive, if the sign bits are opposite product will be negative. The sign bit of the product stored with above criteria should be displayed along with the product. From the above discussion we observe that it is not necessary to wait until all the partial products have been formed before summing them. In fact the addition of partial product can be carried out as soon as the partial product is formed.

3.1.2 Multiplying unsigned numbers: Multiplying unsigned numbers in binary is quite easy. Recall that with 4 bit numbers we can represent numbers from 0 to 15. Multiplication can be performed done exactly as with decimal numbers, except that you have only two digits (0 and 1). The only number facts to remember are that 0*1=0, and 1*1=1 (this is the same as a logical "and").Multiplication is different than addition in that multiplication of an n bit number by an m bit number results in an n+m bit number. Let's take a look at an example where n=m=4 and the result is 8 bits.

Example 1:

11 1 0 1 1
multiplicand (4 bits)

X 13 X 1 1 0 1
multiplier (4 bits)

33 1 0 1 1

11 0 0 0 0

143 1 0 1 1

1 0 1 1

1 0 0 0 1 1 1 1 Product (8 bits)

3.1.3 Multiplying signed numbers: Multiplication can be performed done exactly as with decimal numbers, except that you have only two digits (0 and 1). The only number facts to remember are that 0*1=0, and 1*1=1 (this is the same as a logical "and").Recall that with 4 bit numbers we can represent numbers from -8 to 7.If the MSB bit in the multiplier is '1', the last partial product is 2's compliment of multiplicand.

Example 2:

-5 1 0 1 1 multiplicand (4 bits)

X -3 X 1 1 0 1 multiplier (4 bits)

15 **1 1 1** 1 0 1 1

0 0 0 0 0 0

1 1 0 1 1

0 1 0 1

1 0 0 1 1 1 1 Product (8 bits)

1

Carry discarded

3.2 Proposed Methods: There are three designs to reduce the power consumption.

1) MAC FIR Filter Based on Booth Multiplier
2) MAC FIR Filter Based on Low Power Serial Multiplier and Serial Adder
3) FIR Filter Based on Shift and Add Multiplier

3.2.1 MAC FIR Filter Based on Booth Multiplier:

Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation.

3.2.1.1 Booth algorithm:

Booth's algorithm examines adjacent pairs of bits of the N-bit multiplier Y in signed two's complement representation, including an implicit bit below the least significant bit, $y_{-1} = 0$. For each bit y_i, for i running from 0 to N-1, the bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator P remains unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times $2^i$is added to P; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from P. The final value of P is the signed product.

The representation of the multiplicand and product are not specified; typically, these are both also in two's complement representation, like the multiplier, but any number system that supports addition and subtraction will work as well. As stated here, the order of the steps is not determined. Typically, it proceeds from LSB to MSB, starting at $i = 0$; the multiplication by 2^i is then typically replaced by incremental shifting of the P accumulator to the right between steps; low bits can be shifted out, and subsequent additions and subtractions can then be done just on the highest N bits of P.[1] There are many variations and optimizations on these details.

The algorithm is often described as converting strings of 1's in the multiplier to a high-order +1 and a low-order –1 at the ends of the string. When a string runs through the MSB, there is no high-order +1, and the net effect is interpretation as a negative of the appropriate value.

3.2.1.2 A Typical Implementation:

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values *A* and *S* to a product *P*, then performing a rightward arithmeticshift on *P*. Let m and r be the multiplicand and multiplier, respectively; and let *x* and *y* represent the number of bits in m and r.

1. Determine the values of A and S, and the initial value of P. All of these numbers should have a length equal to (x + y + 1).
    1. A: Fill the most significant (leftmost) bits with the value of m. Fill the remaining (y + 1) bits with zeros.
    2. S: Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining (y + 1) bits with zeros.
    3. P: Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant (rightmost) bit with a zero.
2. Determine the two least significant (rightmost) bits of P.
    1. If they are 01, find the value of P + A. Ignore any overflow.
    2. If they are 10, find the value of P + S. Ignore any overflow.
    3. If they are 00, do nothing. Use P directly in the next step.
    4. If they are 11, do nothing. Use P directly in the next step.
3. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.
4. Repeat steps 2 and 3 until they have been done y times.
5. Drop the least significant (rightmost) bit from P. This is the product of m and r

3.2.2 MAC FIR Filter Based On Low Power Serial Multiplier And Serial Adder:

Digit-serial implementation styles are best suited for implementation of digital signal processing systems which require reduce dynamic power consumption for design MAC fir filter we use of low power serial multiplier and low power serial adder consider the bit-serial multiplier shown in Figure 2.5 where the coefficient word length is four bits. This architecture contains four full adders, four multipliers, and some delay elements. In this multiplier the carry-out signal of every adder is feed back after a delay to the carry-in signal of the same adder.

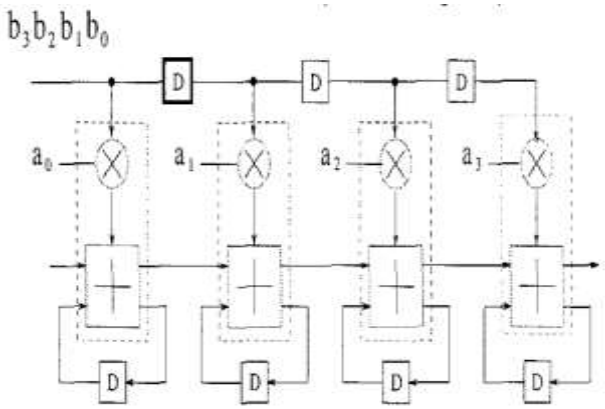


Figure: 2.5 low power serial multiplier and serial adder

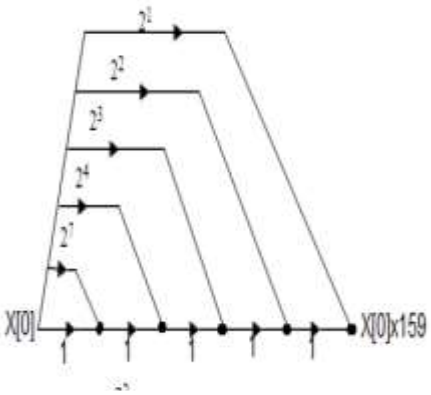
3.2.3. FIR Filter Based on Shift and Add Multiplier:

Fir filter is implemented using the shift and add method. We perform all our optimization in the multiplier block. The constant multiplications are decomposed in to additions and shifts and the multiplication complexity is reduced. It's possible to implement the design in the two forms described below:

I. The coefficients are changed to integer getting multiplied to a multiple power of 10, then we arrange these coefficients the positive power of 3. This procedure is shown below in graph form. Graph branches stand for left shift and notches stand for sum.
Design1:

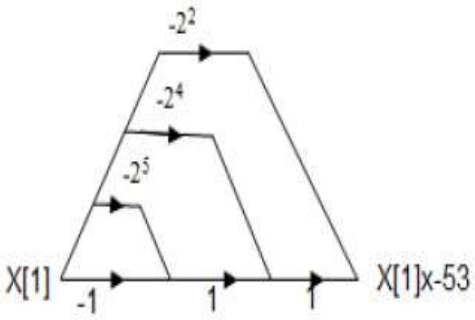
c[0]= 0.159 ,c[1]= -.053^{x1000} c[0]= 159,c[1]= -53

C [0]= 128+16+8+2+1 = $2^7+2^4+2^3+2^2+2^1+2^0$



ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

C[1]= -32-16-4-1 = $-2^5-2^4-2^2-2^0$



II. First we arrange decimal coefficients according to negative and positive power of 2 (no need to them into integer). So the filter hardware and power consumption will reduce.
Design 2:

c[0]=3.75

C[0] = $2^1+2^{-1}+2^-$

4. Digital FIR Filter:

4.1 FIR Filter Theory:

In signal processing, there are many instances in which an input signal to a system contains extra unnecessary content or additional noise which can degrade the quality of the desired portion. In such cases we may remove or filter out the useless samples. For example, in the case of the telephone system, there is no reason to transmit very high frequencies since most speech falls within the band of 400 to 3,400 Hz. Therefore, in this case, all frequencies above and below that band are filtered out. The frequency band between 400 and 3,400 Hz, which isn't filtered out, is known as the pass band, and the frequency band that is blocked out is known as the stop band.

FIR filters are one of the primary types of filters used in Digital Signal Processing. FIR filters are said to be finite because they do not have any feedback. Therefore, if you send an impulse through the system (a single spike) then the output will invariably become zero as soon as the impulse runs through the filter.The FIR filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the filter's output.

Figure shows the basic block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The h_k values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients.

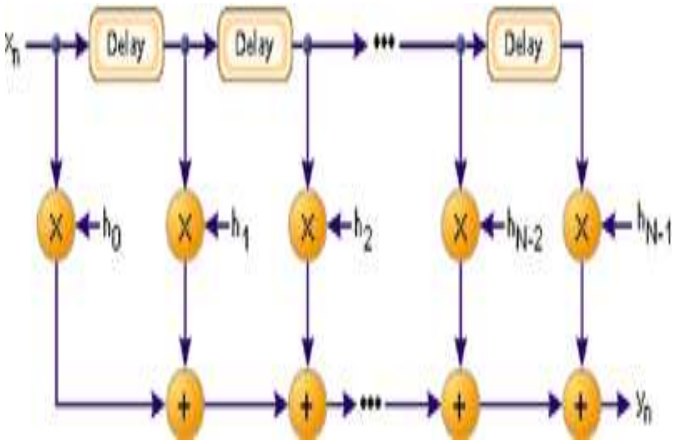


Figure 4.1:The logical structure of an FIR filter

The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stop band and pass band parameters will result from running the filter.

4.1.1 Technical Definition:

The difference equation that defines the output of an FIR filter in terms of its input is

$Y[n] = b_0 X[n] + b_1X[n-1] + \ldots + b_NX[n-N]$

WhereX[n] is the input signal

Y[n] is the output signal

b_i are the filter coefficients, also known as tap weights and

N is the filter order – an Nth-order filter has (N + 1) terms on the right-hand side.

These are commonly referred to as taps (the number of inputs), and one may speak of a "5thorder/6-tap filter", for instance. This equation can also be expressed as a convolution of the coefficient sequence b_i with the input signal:

$$y[n] = \sum_{i=0}^{N} b_i x[n-i].$$

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

That is, the filter output is a weighted sum of the current and a finite number of previous values of the input.

4.1.2 Properties:

Are inherently stable. This is due to the fact that all the poles are located at the origin and thus are located within the unit circle. An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter. FIR filters:

- Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric linear phase, or phase change proportional to frequency, corresponds to equal delay at all frequencies. This property is sometimes desired for phase-sensitive applications, for example crossover filters, and mastering.

The main disadvantage of FIR filters is that considerably more computation power is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequencies (relative to the sample rate) cutoffs are needed.

4.1.3 How to characterize digital FIR filters:

There are a few terms used to describe the behavior and performance of FIR filter including the following:

- **Filter Coefficients:** The set of constants, also called tap weights, used to multiply against delayed sample values. For an FIR filter, the filter coefficients are, by definition, the impulse response of the filter.
- **Impulse Response:** A filter's time domain output sequence when the input is an impulse. An impulse is a single unity-valued sample followed and preceded by zero-valued samples. For an FIR filter the impulse response of a FIR filter is the set of filter coefficients.
- **Tap:** The number of FIR taps, typically N, tells us a couple things about the filter. Most importantly it tells us the amount of memory needed, the number of calculations required, and the amount of "filtering" that it can do. Basically, the more taps in a filter results in better stopband attenuation (less of the part we want filtered out), less rippling (less variations in the passband), and steeper rolloff (a shorter transition between the passband and the stopband).
- **Multiply-Accumulate (MAC):** In the context of FIR Filters, a "MAC" is the operation of multiplying a coefficient by the corresponding delayed data sample and accumulating the result. There is usually one MAC per tap.

4.1.4 Impulse Response:

The impulse response h[n] can be calculated if we set X[n] = δ[n] in the above relation, where δ[n] is the Kronecker delta impulse. The impulse response for an FIR filter then becomes the set of coefficients b_n, as follows

$$H[n] = \sum_{i=0}^{L-1} bi\ \delta[n-i] = bn$$

From = 0 to N.

The Z-transform of the impulse response yields the transfer function of the FIR filter

$$H(z) = Z\{h[n]\} = \sum_{n=-\infty}^{\infty} h[n] z^{-n} = \sum_{n=0}^{N} b_n z^{-n}.$$

FIR filters are clearly bounded-input bounded-output (BIBO) stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than $\sum |b_i|$ times the largest value appearing in the input.

4.1.5 Why is the impulse response "FINITE" :

In the common case, the impulse response is finite because there is no feedback in the FIR. A lack of feedback guarantees that the impulse response will be finite. Therefore, the term "finite impulse response" is nearly synonymous with "no feedback". However, if feedback is employed yet the impulse response is finite, the filter still is a FIR. An example is the moving average filter, in which the Nth prior sample is subtracted (fed back) each time a new sample comes in. This filter has a finite impulse response even though it uses feedback: after N samples of an impulse, the output will always be zero.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

4.2 Sources of Power Consumption:

When designers recognized power consumption as a design constraint, simple models were created. Power per MHz is still a commonly used representation of a component. With a closer look at power dissipation, it becomes obvious that the subject is not that simple. Electric current is not constant during operation; peak power is an important concern.

This chapter gives an overview of the sources of power consumption. A formula for average power is given in equation 1.

$P_{avg} \square \square P_{dynamic} \square \square P_{short} \square \square P_{leakage} + P_{static}$ .....(1)

The total power consumed by conventional CMOS circuitry is composed of two sources. The first is active power, this power represents the power consumed by the intended work of the circuit to switch states and thus execute logic functions. Active power is primarily composed of the power associated with the charging or discharging of the capacitance of the switching nodes. In addition to active power, there are components of leakage power, the most dominant of which is the sub-threshold current of the transistors in the circuit.

The four components are dynamic, short-circuit, leakage and static power consumption. The share of each part depends on the application and technology. In several cases e.g. $P_{leakage}$ might be negligible, in others not.

4.2.1 Dynamic Power (Capacitive Switching Activity):

Switching activity is a measure for the number of gates and their outputs that change their bit-value during a clock cycle. Fig 2.1 shows the dynamic switching power dissipation. To toggle between logic zero and logic one capacities have to be discharged and charged. The electric current id that flows during this process causes a power dissipation Pdynamic. The current is dependent on the capacitive output load Cout and the supply voltage Vdd. In equation 2 this behavior is reproduced.

$P_{dynamic} = K\, C_{out} V^2_{dd}\, f$ .....(2)

K is the average number of rising transitions during one clock cycle and f the clock frequency. It is interesting to see, that the supply voltage has a quadratic effect on dynamic power consumption. Reducing power supply will therefore have the greatest effect on saving power, taking into account that typically $P_{dynamic}$ is responsible for 80% of P_{avg}. Unfortunately, designers don't have the freedom to choose the parameters arbitrarily. The chosen technology and the given timing constraints set a minimum and maximum range for the acceptable values.

The dynamic power dissipation, which has two components: short circuit and charge/discharge of capacitance power dissipation. The short circuit power dissipation is a function of the slew rate of the input voltage; the sharper the clock edge, the lower the short circuit power dissipation.
Short circuit power is result when both p-transistor and n-transistor is on for short duration of time. Mathematically,

$V_{dd} < |V_{tp}| + |V_{tn}|$ ..... (3)

Where: V_{tp} and V_{tn} of equation 3 are threshold voltages for PMOS and NMOS transistors, respectively. Threshold voltage is a voltage at which channel formation occurs in a metal–oxide– semiconductor field- effect transistor (MOSFET).
Power dissipation is given by

$P = \alpha C V^2 f$ ....(4)

Where: V yields to supply voltage and Voltage swing. It is tried to reduce both supply voltage and Voltage swing because voltage have highest impact on total power dissipation as show in equation 4. 'α' is switching activity, 'C' is capacitance and 'f' is operating frequency.

However, one major drawback associated with clock networks is their power dissipation. Studies have shown that the clock network can dissipate 20-50% of the total power on a chip. In the context of the growing importance of low power designs for portable electronics, it is necessary to develop strategies to significantly reduce the power dissipation of the clock network, since this will lead to a major reduction in the overall power dissipation of the chip.

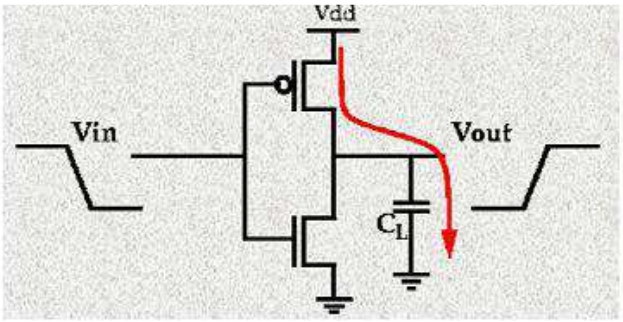


Figure.4.2 Dynamic switching power

Until now, we assumed that all charge drawn from the power supply is collected by the output capacitances. Some current flows directly from power to ground and surmises a short-circuit current during bit switching.

4.2.2 Short-Circuit Power:

Short circuit power consumption occurs during switching of both NMOS and PMOS transistors in the circuit and they conduct simultaneously for a short amount of time. Since there is a finite rise/fall time for both pMOS and nMOS, during transition, for example, from off to ON, both the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

transistors will be on for a small period of time in which current will find a path directly from V_{DD} to ground, hence creating a short circuit current. Short circuit power dissipation increases with rise and fall time of the transistors.

An additional form of power consumption became significant in the 1990s as wires on chip became narrower and the long wires became more resistive.

CMOS gates at the end of those resistive wires see slow input transitions. During the middle of these transitions, both the NMOS and PMOS logic networks are partially conductive and current flows directly from V_{dd} to V_{SS}. The power thus used is called crowbar power. Careful design which avoids weakly driven long skinny wires has ameliorated this effect, and crowbar power is nearly always substantially smaller than switching power CMOS circuits consist of a pull-up and pull-down network (see figure 2.2), which have a finite input fall/rise time larger then zero. During this short time interval, when the pull-down and pull-up network are conducting both, a current isc flows from power to ground. This current is called short-circuit current.

Reducing Vdd slows down charging of capacitances. The resulting power consumption is calculated by equation 5. The origin of the formula is carried out in. b is the gain factor of a MOS transistor, VT its threshold voltage and t is the rise/fall time of the gate inputs.

Vdd

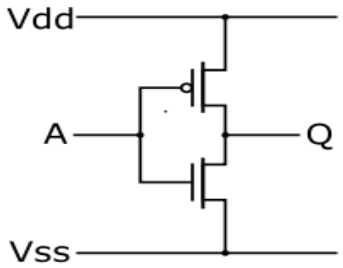


Figure. 4.3 CMOS circuit

$$P_{short} = K (\beta/12) (V_{dd}-2V_T)^3 f_r \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Since the parameters of this equation cannot be tuned by designers, Pshort is not going to be taken into consideration during low power design. This is no set-back because several authors observed that short-circuit power dissipation is usually a small fraction of total power usage, around 10%. Dynamic and short-circuit power depend on switching activity represented in the parameter K.

As an effect, no power should be lost during idleness of a CMOS circuit, when K is zero. The existence of leakage currents shows another piece of reality.

4.2.3 Static Power:

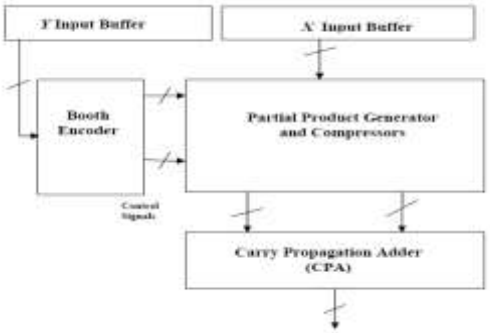
The static power component, due to leakage, that is present even when the circuit is not switching. This, in turn, is composed of two components - gate to source leakage, which is leakage directly though the gate insulator, mostly by tunneling, and source-drain leakage attributed to both tunneling and sub-threshold conduction. The contribution of the static power component to the total power number is growing very rapidly in the current era of Deep Sub-Micrometer (DSM) Design.

Static power dissipation depends on a current flow from power to ground during idle time unlike short-circuit power dissipation, which occurs only during switching activity. NMOS circuits show high static power consumptions because power is connected directly to ground when a gate's output denounces logic zero, resolving into a great short circuit current. In well designed and flawless CMOS designs static power should be zero. If not, the reason could be e.g. a bus conflict where multiple drivers attempt to drive a signal to different logic values.

Static power consumption: Static current: in CMOS there is no static current as long as

$$V_{in} < V_{TN} \text{ or } V_{in} > V_{DD} + V_{TP} \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Leakage current: determined by "off" transistor Influenced by transistor width, supply voltage, transistor threshold voltages.



V_{dd}

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

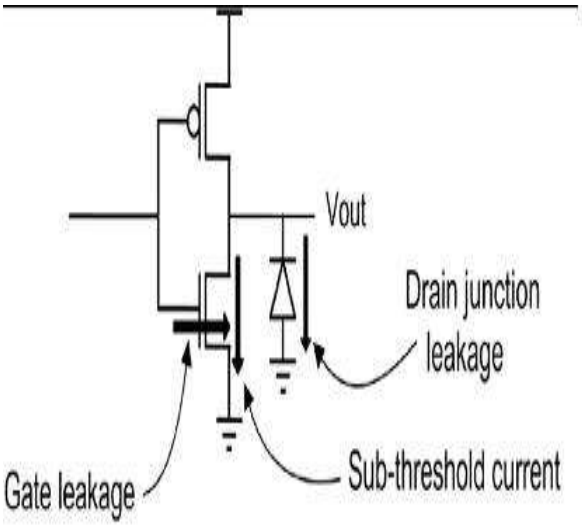


Figure. 4.4 static power dissipation model

Types of static power consumption:
1. ISUB (subthreshold current)
2. Gate leakage
3. Gate induced drain leakage
4. Reverse biase junction leakage.

5. Implementation Of The Design:

5.1 Block diagram of FIR filter: This chapter illustrates the three different techniques which are mentioned in the previous chapter which are implemented in digital fir filter which shows in below block diagram.

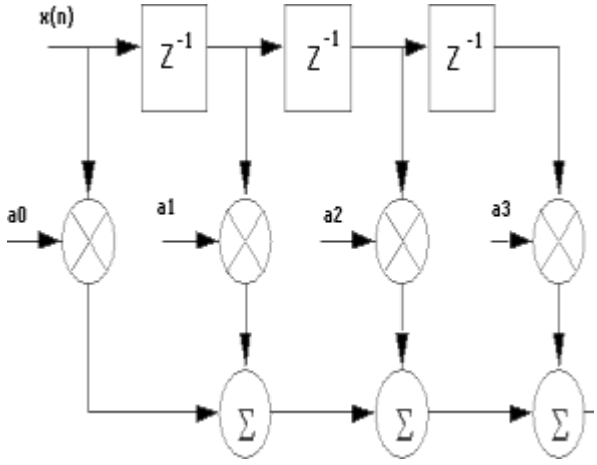


Figure. 5.1 Block diagram of FIR filter

These are

1) MAC Fir Filter Based on Booth Multiplier
2) MAC Fir Filter Based on Low Power Serial Multiplier and Serial Adder
3) Fir Filter Based on Shift and Add Multiplier

5.1 MAC Fir Filter Based on Booth Multiplier:

A multiplier has two stages. In the first stage, the partial products are generated by the booth encoder and the partial product generator (PPG), and are summed by compressors. In the second stage, the two final products are added to form the final product through a final adder.

Figure. 5.2 Block Diagram of booth Multiplier Architecture

The block diagram of traditional multiplier is depicted in Figure 3.1. It employs a booth encoder block, compression blocks, and an adder block. X and Y are the input buffers. Y is the multiplier which is recoded by the booth encoder and X is the multiplicand. PPG module and compressor form the major part of the multiplier. Carry propagation adder (CPA) is the final adder used to merge the sum and carry vector from the compressor module.

5.1.1 Booth Encoder and Partial Product Generator:

The Booth encoder was implemented using two XOR gates and the selector using three MUXs and an inverter Careful optimization of the partial-product generation can lead to some substantial delay and hardware reduction. In the normal 8*8 multiplication 8 partial products need to be generated and accumulated. For accumulation seven adders to reduce power are required but in the case of booth multiplier only 4 partial products are required to be generated and for accumulation three adders, reduced delay required to compute partial sum and reduces the power consumption.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

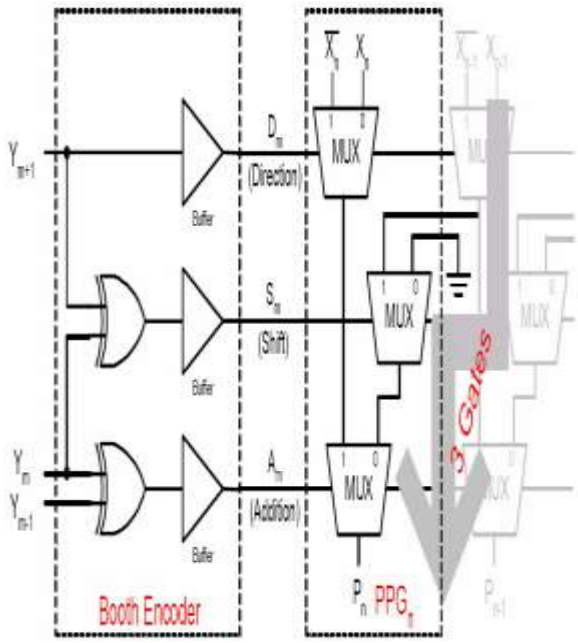


Figure. 5.3 Booth encoder and PP (m=2i)

Partial product generation is the very first step in binary multiplier. Partial product generators for a conventional multiplier consist of a series of logic AND gates as shown in Figure 3.3.

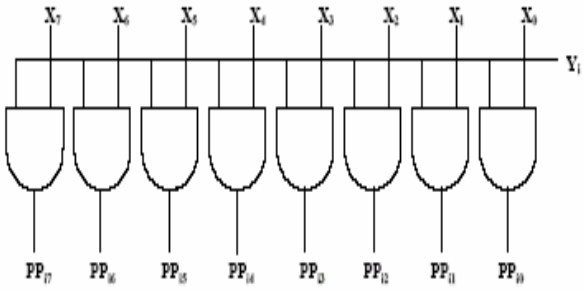


Figure. 5.4 Partial Product generator using AND gates

If the multiplier bit is '0', then partial product row is also zero, and if it is '1', then the multiplicand is copied as it is. From the second bit multiplication onwards, each partial product row is shifted one unit to the left. In signed multiplication, the sign bit is also extended to the left

5.1.2 Booth's Algorithm:

A.D. Booth proposed Booth encoding technique for the reduction of the number of partial products. This algorithm is also called as Radix-2 Booth's Recoding Algorithm. Here the multiplier bits are recoded as Zi for every i^{th} bit Yi with reference to Yi-1. This is based on the fact that fewer partial products are generated for groups of consecutive zeros and ones. For a group of consecutive zeros in the multiplier there is no need to generate any new partial product. We only need to shift previously accumulated group partial product one bit position to the right for every 0 in the multiplier.

The radix-2 algorithms results in these observations:

(a) Booth observed that whenever there was a large number of consecutive ones, the corresponding additions could be replaced by a single addition and a subtraction

$2j + 2j-1 + \ldots\ldots\ldots\ldots\ldots + 2i+1 + 2i = 2j+1 - 2i$

(b) The longer the sequence of ones, the greater the savings.

(c) The effect of this translation is to change a binary number with digit set [0, 1] to a binary signed-digit number with digit set [-1, 1].

In this algorithm the current bit is Y_i and the previous bit is Y_{i-1} of the multiplier Y_{n-1} Y_{n-2}…… Y_1 Y_0 are examined in order to generate the i^{th} bit Z_i of the recoded multiplier Z_{n-1} Z_{n-2} …….Z_1 Z_2. The previous bit Y_{i-1} serves only as the reference bit. The recoding of the multiplier bits need not be done in any predetermined order and can be even done in parallel for all bit positions.

6. Simulation Results: We used XILINX ISE 12.1 for our programming. We considered Verilog HDL as our primary language. For test bench waveforms we also used MODELSIM to write our own test benches and for simulation. Model Synthesis Map report all features in Xilinx helped us a lot. We used Xilinx's power analyzer in order to calculate power consumed in any arithmetic circuit. This analysis implemented using 4 bit width of multiplier and multiplicand in digital FIR filter.

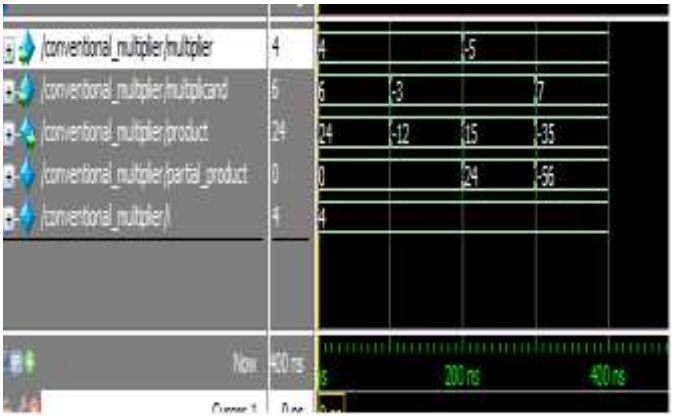


Fig. 6.1Conventional multiplier waveform

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

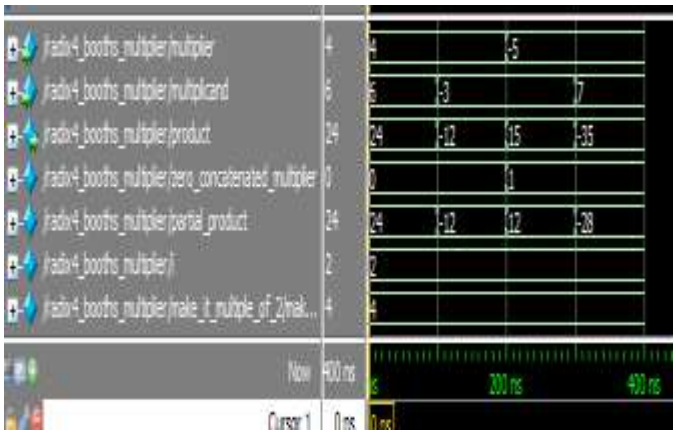


Fig.6.2 Radix-4 booth multiplier waveform

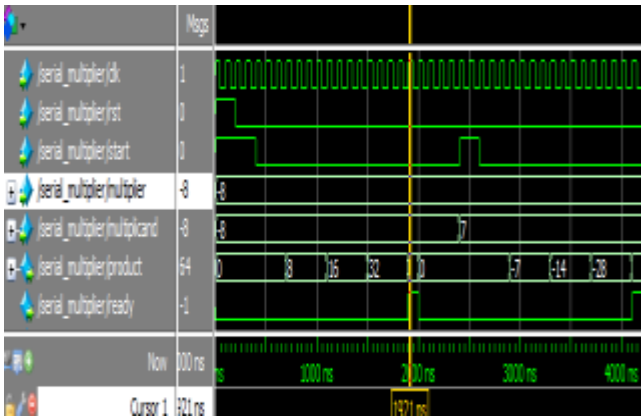


Fig. 6.3 Serial multiplier and serial adder waveform

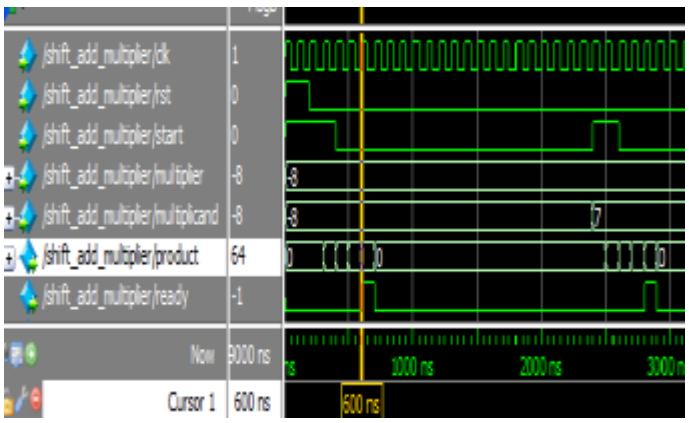


Fig. 6.4 Shift and add multiplier waveform

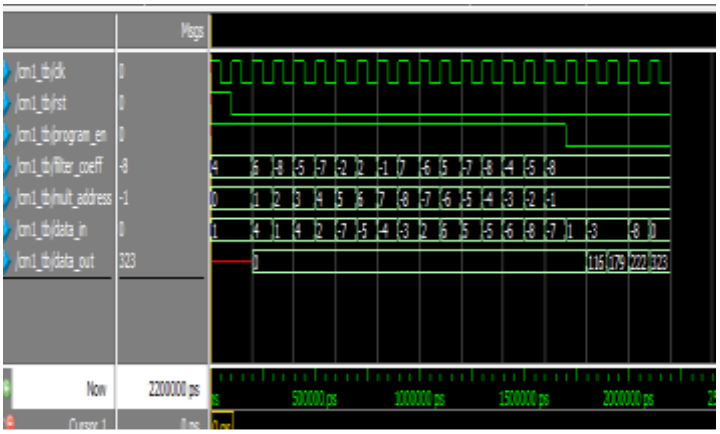


Fig. 6.5 Conventional multiplier FIR filter waveform

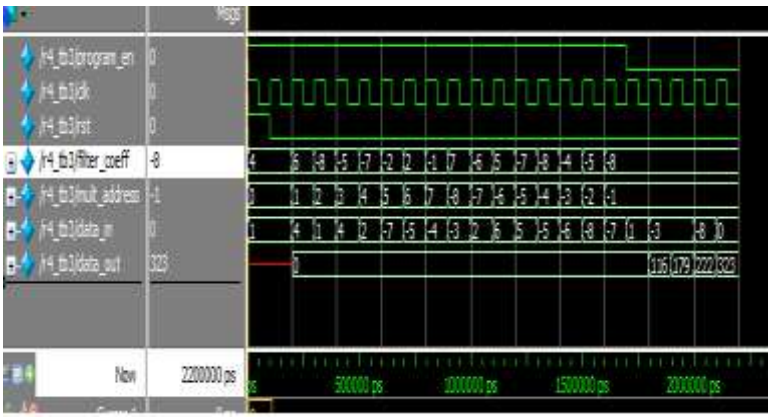
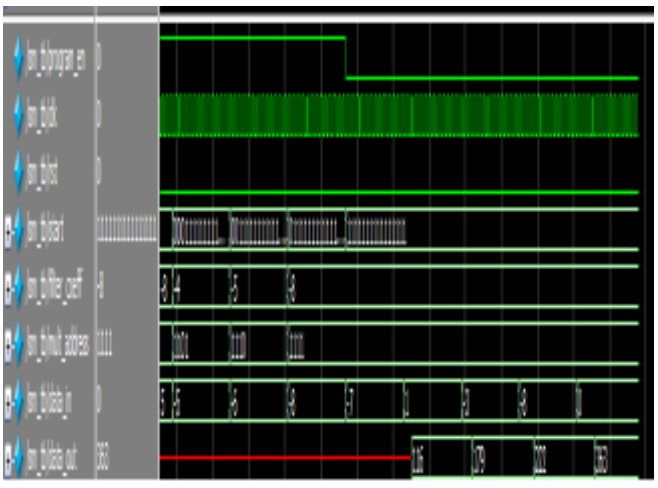


Fig. 6.6 Radix-4 Booth multiplier FIR filter waveform



ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Fig.6.7 Serial multiplier and serial adder FIR filter waveform

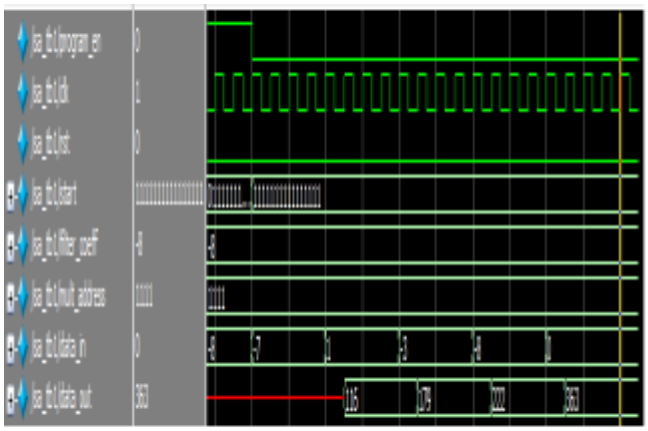


Fig. 6.8 Shift add multiplier FIR filter waveform

7. FUTURE SCOPE:

The future scope of this work includes the following:

- The optimization of the design can be done in terms of area occupied on chip.
- It can be extendable to 16 bit 32 bit up to N number of bit tab.

Multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier.

Thus making them suitable for various high speed, low power, and compact VLSI implementations. However area and speed are two conflicting constraints. So improving speed results always in larger areas. So here we try to find out the best trade off solution among the both of them. Generally as we know multiplication goes in two basic steps. Partial product and then addition. Hence in this paper we have first tried to design different adders and compare their speed and complexity of circuit i.e. the area occupied. And then we have designed Wallace tree multiplier then followed by Booth's Wallace multiplier and have compared the speed and Power consumption in them. While comparing the adders we found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which Carry Select Adders are high speed but posses a larger area.

And a Carry Look Ahead Adder is in between the spectrum having a proper tradeoff between time and area complexities. After designing and comparing the adders we turned to multipliers. Initially we went for Parallel Multiplier and then Wallace Tree Multiplier. In the mean time we learned that delay amount was considerably reduced when Carry Save Adders were used in Wallace Tree applications. Then we turned to Booths Multiplier and designed Radix-4 modified booth multiplier and analyzed the performance of all the multiplier. After that we turned to different methods of power optimization, of which we could only complete a few like we went for designing different recoding schemes and their corresponding partial product generator scheme. After that we designed these encoders and pp generators and found out the time delays and area covered and power consumed by each scheme.

We took into consideration that since all the pp generators take a huge amount of area we need to go for simplest of the designs for them and also side by side we need to ensure that we don't have much switching actions in the circuit. After this we even modified one of the recoding schemes to lower the delay and power required by the circuit. The result of our project helps us to make a proper choice of different multipliers in fabricating in different arithmetic units as well as making a choice among different adders in different digital applications according to requirements. All the programs and results have been given in the following sections.

8. REFERENCES:

[1] Jin-Gyun Chung, Keshab K. Parhi "Frequency Spectrum Based Low-Area Low- Power Parallel FIR Filter Design" EURASIP Journal on Applied Signal Processing 2002, vol. 31, pp. 944–953.
[2] AHMED F. SHALASH, KESHAB K. PARHI "Power Efficient Folding of Pipelined LMS Adaptive Filters with Applications" Journal of VLSI Signal Processing, pp. 199–213, 2000.
[3] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner, "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", IEEE, 2006.
[4] Kousuke TARUMI, Akihiko HYODO, Masanori MUROYAMA, Hiroto YASUURA,"A design method for a low power digital FIR _lter indigital wireless communication systems," 2004.
[5] " Design and Implementation of Low Power Digital FIR Filters relyingon Data Transition Power Diminution Techniques DSP

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Journal,Volume 8, pp.21-29, 2008.
[6] "Design of FIR filter" by Elena punskaya.
[7] Ronak Bajaj, Saransh Chhabra, Sreehari Veeramachaneni, M B Srinivas,"A Novel, Low-Power Array Multiplier Architecture.
[8] Bahram Rashidi, Bahman Rashidi and Majid Pourormazd "Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on Xilinx FPGA"2011 IEEE.
[9] A.Aenthikumar, A.M.Natarajan "Design and Implementation of Low Power Digital FIR Filters relying on Data Transition Power Diminution Technique" DSP Journal, Volume 8, pp. 21-29,2008.
[10] Yun-NanChang, Janardhan H. Satyana:rayanaKeshabK.Parhi" LOW-POWER DIGIT-SERIAL MULTIPLIERS", 1997 IEEE. International Symposium on Circuits and Systems, June i3-12,1997

9. Authors Biography:

M.RENUKA[1] completed **M.Tech (VLSI)** from Mahaveer institute of science and technology JNTUH. **B.TECH (ECE)** completed at vidya jyothi institute of technology in year 2007. She worked as **"Hardware Design Engineer"** in ECIL from 2007 to 2008.and also she worked as **Assistant Trainee Officer** in the year 2008 to 2009.Presently She Is Working As **Assistant professor** in **ECE** department vidya jyothi institute of technology Still up to date. Her areas of research interests include **VLSI, Embedded systems, EDC and Digital signal processing.**

MIRIYALA NIKHILA[2] is from MANDALAPALLY, ANDHRA PRADESH. Completed **B.Tech in Electronics and Communications with 72.21%** from VIDYA JYOTHI INSTITUTE OF TECHNOLOGY affiliated to JNTUH in 2013. Completed Intermediate with 91%. She has done a project in 'Image Processing'. Has good technical hold in Signal Processing, Communications and Control systems. Her areas of interests in research include **Very Large Scale Integrated Circuits,Embedded systems,network security,Digital signal Processing, Communication Systems.**

S.DINESH REDDY[3] is from HYDERABAD, ANDHRA PRADESH. Completed **B.Tech in Electronics and Communications** with 67.4% from VIDYA JYOTHI INSTITUTE OF TECHNOLOGY affiliated to JNTUH in 2013. Completed Intermediate with 76%. Has done a project in 'Image Processing'. Currently working with a few NGOs namely **'WWF-India, Andhra Pradesh State Office', 'Swecha'(FSMI) and 'avashaH - hands that help'.** Has good technical hold in **Signal Processing, Communications and Control systems**. Areas of interests in research include **Digital signal Processing, Image Processing, Telecommunications,CommunicationSystems,vlsi.**