FIR Filter IC Design Using Redundant Binary Number Systems

A Thesis Submitted In Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Technology

In

Electronics and Communication Engineering
(VLSI Design and Embedded System)

by

Prabhat Kumar Barik
Roll No: 209EC2120

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela
June 2011

FIR Filter IC Design Using Redundant Binary Number Systems

A Thesis Submitted In Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Technology

In

Electronics and Communication Engineering
(VLSI Design and Embedded System)

by

Prabhat Kumar Barik
Roll No: 209EC2120

Under the Supervision of

Prof. Girija Sankar Rath

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela

June 2011

Certificate

This is to certify that the thesis entitled, *"**FIR Filter IC Design Using Redundant Binary Number Systems**"* submitted by Mr. **Prabhat Kumar Barik** in partial fulfillment of the requirements for the award of **MASTER OF TECHNOLOGY** Degree in **Electronics and communication Engineering** with specialization in "**VLSI Design and Embedded System**" at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other Institute for the award of any degree.

Date:

Dr. Girija Sankar Rath
Professor
Department Of Electronics and Communication Engineering
National Institute of Technology
Rourkela - 769008

Abstract

Conventional number systems is the weighted fixed positive radix number systems, where signed number uses the sign as a symbol followed by the number part either in magnitude or *r'*s complement form. Addition of conventional number systems requires carry propagation (serial signal propagation) from LSD to MSD and the addition time depends on word-length, which is the main limitation of the VLSI performance.

But Redundant number systems (RNS) is to allow addition of two numbers in which no serial signal propagation is required along the adder; that is, the time duration of the operation is independent of length of the operands and is the time required for the addition of two digits. This is the advantage of RNS over conventional number systems.

Because of this advantage, in this thesis it proposed to design an FIR filter based on RNS. In order to implement FIR filter, it is necessary to design adder, multiplier and D-FF. For implementation, the structural blocks are to be designed such as PPM adder, MMP subtractor, D-FF, Digit-serial multiplier.

In this thesis, a 368.18MH_Z 3-tap FIR filter and 80MH_Z Box-car FIR filter be designed based on bottom-up design flow using CADENCE 5.1.41, cadence IC design environment. The design was based on the CMOS 90nm technology process. Bottom level transistors are used from gpdk090 library. The advantages of full custom are maximum circuit performance, minimum design size, and minimum high-volume production cost.

ACKNOWLEDGEMENTS

This project is by far the most significant accomplishment in my life and it would be impossible without people who supported me and believed in me.

I would like to extend my gratitude and my sincere thanks to my honorable, esteemed supervisor **Prof. G.S. Rath,** Department of Electronics and Communication Engineering.

He is not only a great lecturer with deep vision but also and most importantly a kind person. I sincerely thank for his exemplary guidance and encouragement. His trust and support inspired me in the most important moments of making right decisions and I am glad to work with him.

I want to thank all my teachers **Prof. K. K. Mahapatra, Prof. D. P. Acharya** for providing a solid background for my studies and research thereafter. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all my friends and especially my classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I would like to thank all those who made my stay in Rourkela an unforgettable and rewarding experience.

Last but not least I would like to thank my parents, who taught me the value of hard work by their own example. They rendered me enormous support during the whole tenure of my stay in NIT Rourkela.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Since the theory of digital signal processing (DSP) is developed and applied to the electrical engineering world, digital filtering always plays a very important role. Digital filtering techniques is used to suppress noise, enhance signal in selected frequency ranges, constrain bandwidth, remove or attenuate specific frequencies and other special operations. Digital filters are classified into finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR digital filters can have exactly linear phase response and a very regular architecture, and suffer less from the effects of finite word length as compared with IIR digital filters. This thesis presents the design and an implementation of such a filter based on redundant binary number systems. The main components of FIR filter are adder, multiplier and delay. The carry propagation delay is a limiting factor of the adder and multiplier. Based on redundant number, adders and multipliers are designed in such a way that the propagation delay is reduced of the FIR filter.

In this thesis the FIR filter is designed based on bottom-up or full custom design flow using CDS 5.1.41. The advantages of full custom are maximum circuit performance, minimum design size, and minimum high-volume production cost. Finally, designing box-car FIR filter and 3-tap FIR(multiplier coefficient 4-bit) filter can serve as a basic of IC design students to work with as a tool in their understanding of digital design. It is also a stepping-stone for students in designing other CMOS chips using the 90nm CMOS technology and to encourage them to make improvements in the design.

1.2 Motivation and goals

Area, delay (performance) and power are the three important design constraints for designing an embedded real-time digital signal processing systems. The area constraint is imposed primarily by considerations of cost. Area efficient implementation results in a smaller die size and hence becomes more cost effective. It also enables integrating more functionality on a single chip. The performance requirements of a system are driven by its data processing needs. For DSP systems,

throughput is the primary performance criterion. The performance constraint is thus dependent on the rate at which the input signals are sampled and on the complexity of processing to be performed. Low power dissipation is a key requirement for portable, battery operated systems as it extends battery life. Low power dissipation also helps reduce the packaging cost (plastic instead of ceramic), eliminate / reduce cooling (heat sinks) overhead and increase the reliability of the device.

For the requirement of high-speed and low-power applications, the development and implementation of high-speed FIR digital filters need both increased parallelism and reduced complexity in order to meet both sampling rate and power dissipation goals. In this thesis, FIR filter is designed based on RNS to achieve high speed operation. Bottom-up design flow is used for maximum circuit performance, minimum design size, and minimum high-volume production cost.

1.3 VLSI Design Flow

1.3.1 Introduction on Bottom-up and Top-down Design Flow

The designer usually follows some design phases to create his project. At the beginning the designer has to specify the functionality of the system. Basic blocks of the hardware are identified and their interfaces, composed of data and control signals, are fixed. Today, there are two principal ways to design a VLSI circuit with traditional tools that have been developed in these last years. The designer can choose at discretion an approach Bottom-up or a Top-down flow but sometimes the choice can be forced in consequence of particular design requirements or circuit structure. Top-down is a process of iterative refinements. The designer starts with a top view of the system and decomposes single blocks into smaller ones. Bottom-up flow starts with low-level building blocks and interconnects them to greater ones. In reality, these two techniques are not very incompatible and, for instance, the designer can also choose to use particular self-made cells and to do not touch their structure within a top-down approach [10]. The approach Bottom-up is preferable in digital design if the designer desires to plain a particular cell achieving specific performance with transistors full-custom designed and then he wants to replicate this structure in his project.

1.3.2 Bottom-Up Design Flow

The Bottom-Up design flow is given in Fig 1.3.1. The Bottom-Up design flow starts with a set of design specifications. The “specs” typically describe the expected functionality of the designed circuit as well as other properties like delay times, area, etc. To meet the various design specifications certain design trade offs (area verses delay) are required [10].

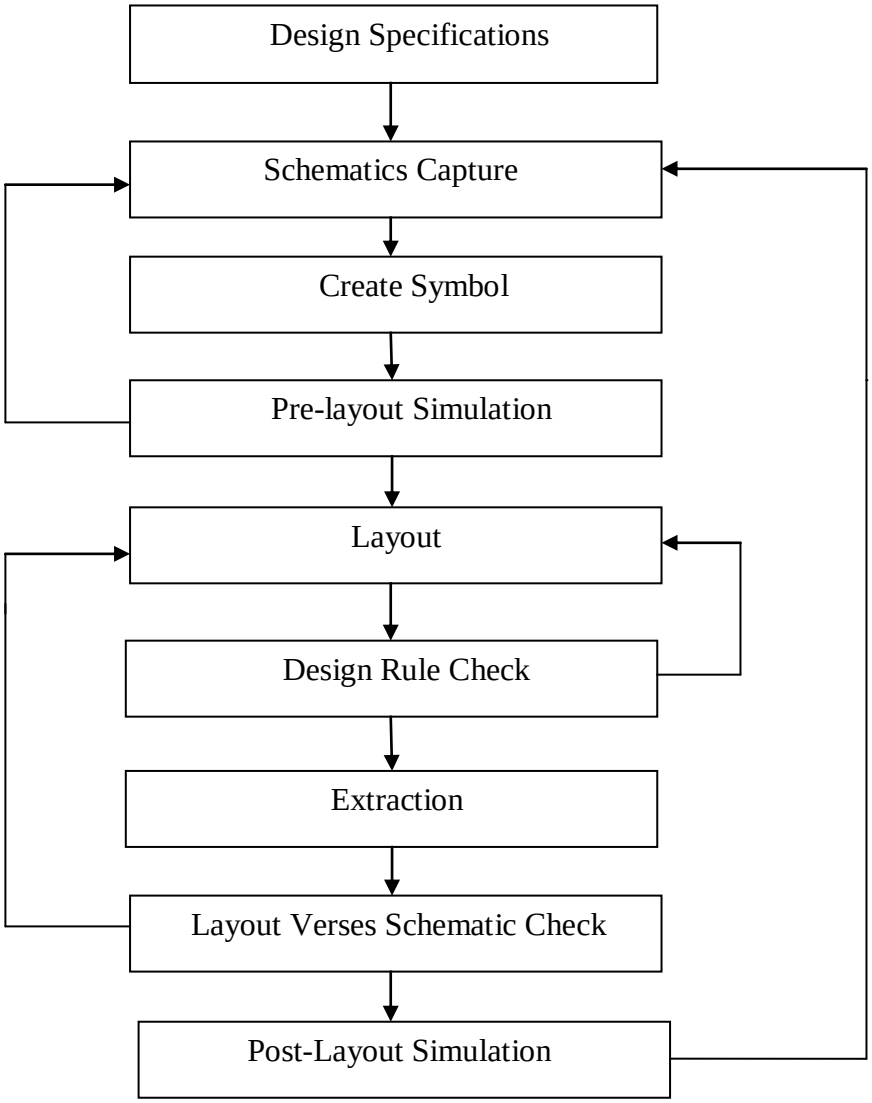


Fig 1.3.1 Bottom-Up Design Flow

A. **Schematic Capture**

A Schematic Editor is used for capturing (i.e. describing) the transistor-level design. The Schematic Editors provide simple, intuitive means to draw, to place and to connect individual components that make up the design. The resulting schematic drawing must accurately describe the main electrical properties of all components and their interconnections. Also included in the schematic are the supply connections (V_{DD} and gnd), as well as all pins for the input and output signals of the circuit. From the schematic, a netlist is generated, which is used in later stages of the design. The generation of a complete circuit schematic is therefore the first important step of the transistor-level design.

B. Symbol Creation

A symbol view of the circuit is also required for some of the subsequent simulation steps or for documentation purposes. Thus, the schematic capture of the circuit topology is usually followed by the creation of a symbol to represent the entire circuit. The shape of the icon to be used for the symbol may suggest the function of the module (logic gates – AND, OR, etc.), but the default symbol icon is a simple rectangular box with input and output pins. The symbol creation will also help the circuit designer to create a system level design consisting of multiple hierarchy level.

C. Layout

The creation of the mask layout is one of the most important steps in the full-custom design flow, where the designer describes the detailed geometrics and the relative positioning of each mask layer to be used in actual fabrication, using a Layout Editor. Physical layout design is very tightly linked to overall circuit performance since the physical structures determines the transconductances of the transistors, the parasitic capacitances and resistances, and obviously the silicon area which is used to realize a certain function. But the process is very intensive and time-consuming design effort. It is also extremely important that the layout design must not violate any of the layout design rules, in order to ensure a defect free fabrication of the design. The layout process can be a manual process, in which layout of each design is done manually or an automatic process using a CAD tool. But the quality of the layouts produced using automatic processes are still far from hand optimized layouts.

D. Design Rule Check (DRC)

The created mask layout must conform to a complex set of design rules, in order to ensure a lower probability of fabrication defects. A tool built in to the layout editor called Design Rule Checker, is used to detect any design rule violations during and after the mask layout design. If errors are detected, they should be removed from the mask layout, before the final design is saved.

E. Circuit Extraction

After the mask layout has been made free from design rule errors, circuit extraction is performed to create a detailed netlist for the simulation of the circuit. The circuit extractor identifies the individual transistors and their connections as well as the parasitic capacitances and resistances that are inevitably present. The extracted netlist can give a very accurate estimation of the device dimensions and device parasitic that ultimately determine the circuit performance. The extracted netlist are used in transistor level simulations and in Layout Verses Schematic comparison.

F. Layout Verses Schematic Check

After the mask layout design of the circuit is completed, the design should be checked against the schematic circuit description created earlier. The 'Layout Verses Schematic (LVS) Check' will compare the original network with the one extracted from the mask layout. The LVS step provides an additional level of confidence for the integrity of the design, and ensures that the mask layout is a correct realization of the intended circuit topology. Also it should be noted that a successful LVS would not guarantee that the extracted circuit would actually satisfy the performance requirements since LVS check guarantees only a topological match. If any errors show up during LVS, then it should be corrected before proceeding to post layout simulation.

G. Post-Layout Simulation

The electrical performance of a full custom design can be best analyzed by performing a post-layout simulation on the extracted circuit netlist. The detailed simulation performed using the extracted netlist will provide a clear assessment of the circuit speed and the influence of circuit parasitic. If the results of the post-layout simulation are not satisfactory, the designer should modify the transistor dimensions or the circuit topology, in order to achieve the desired circuit

performance. Thus, it may require multiple iterations on the design, until the postlayout simulation results satisfy the original design requirements.

Finally, it should be noted that a satisfactory result in post-layout simulation is still no guarantee for a completely successful product, since the actual performance of the chip can be only be verified by testing the fabricated prototype.

1.4 Thesis Organization

The organization of this thesis is as follows. In Chapter 2, a review of computer binary number systems, redundant number systems and its arithmetic operations. In Chapter 3, it describes the FIR filter theory, box-car fir filter and components of the filter. In Chapter 4, it gives the architecture, design and implementation of different digital cells and finally implemented FIR filter. Conclusion and future works are given in Chapter 5.

Chapter 2

Computer Binary Number Systems

2.1 Binary Number Systems

A number system is defined by the set of values that each digit can assume and by an interpretation rule that define the mapping between the sequences of digits and their numerical values. There are two types of number systems namely conventional (e.g. binary, decimal) and unconventional (e.g. signed-digit number).In conventional number systems, every number has a unique representations i.e. no two sequences have the same numerical value and hence these numbers are called non-redundant number systems [2]. In conventional digital computers, integers are represented as binary numbers of fixed length n having a represents the integer value by

$$X = x_i 2^i \tag{2.1}$$

The weight of the digit x_i is the i^{th} power of the 2 where 2 is the radix of the binary number system e.g. the integer X (5) can be represented as $5=1*2^2+0*2^1+1*2^0$. Because of the tradeoff between the word length and hardware size and between the propagation delay, various types of number representations have been proposed. In non-redundant number systems, carry propagation is the limitation of VLSI implementation of high speed multiplication and addition. In the addition of two conventional binary numbers the carry may propagate all the way from the least significant digit to the most significant. The addition time thus dependent on the word-length. To reduce the addition time i.e. propagation delay, we need another number systems called unconventional or signed digit number systems [2].

2.2 Signed Digit Number Systems

In an unconventional radix-r number system, a digit can take on values {0, 1, 2, …….. , r-1 } and the digit set is S = {-(r-1), -(r-2), …… , -1, 0, 1, …. , (r-1) }. For example, the digit set {-1, 0, 1} is used for radix-*2* (r =*2*) number system. A signed-digit is represented by the digits z_i and has the algebraic value

$$Z = \sum z_i r^{-i} \tag{2.2.1}$$

In this case, the number 3 can be represented as 0011 or 0101-1. Hence every number allows multiple representations in signed-digit format and these numbers are called Redundant Number Systems. Signed-digit representations limit carry propagation to one position to the left during the operation of addition and subtraction in digital computers [1-2]. Carry propagation chains are eliminated by the use of redundant representations for the operands.

2.3 Redundant Number Systems (RNS)

The class of signed-digit number or redundant number representations is derived according to four requirements which are postulated as necessary for number representations in fast parallel arithmetic.

The purpose of Redundant number representations is to allow addition and subtraction of two numbers in which no serial signal propagation is required along the adder; that is, the time duration of the operation is independent of length of the operands and is equal to the time required for the addition or subtraction of two digits. The signed-digit representation must have a unique representation of zero algebraic value of a number [3]. The redundant number is represented by $n+m+1$ digits z_i ($i=-n\ldots\ldots, -1,0,1,\ldots.., m$) has the integer value

$$Z = \sum_{-n}^{m} z_i r^{-i} \tag{2.3.1}$$

Where the values of r and z_i are such that the following requirements are satisfied:

1) The radix r is a positive integer.
2) The algebraic value Z=0 has a unique representation.
3) There exits transformations between the conventional representation and the signed-digit representation for every algebraic value Z within a specified range.
4) Totally parallel addition and subtraction is possible for all digits in corresponding positions of two representations.

2.4 Arithmetic Operations of RNS

The arithmetic operations of totally parallel addition and subtraction of two digits z_i and y_i from the corresponding positions of the representations of numbers Z and Y are defined as follows [3]:

Definition 1: Addition of digits z_i and y_i is totally parallel if the following two conditions are satisfied:

1) The sum digits s_i (ith digit of the sum S=Z+Y) is a function only of the augned digit z_i addend digit y_i and the transfer digit t_i from the (i+1)th position on the right: $s_i=f(z_i, y_i, t_i)$. The term "transfer digit" is used here instead of the commonly used terms "carry" or "borrow" for two reasons. First the transfer digit may assume both positive and negative values for their addition or subtraction; secondly unlike the "carry" or "borrow" of conventional addition or subtraction, the transfer digit is never propagated past the first adder position on the left.
2) The transfer digit t_{i-1} to the (i-1)th position on the left is a function only of the augend digit z_i and the addend digit y_i : $t_{i-1}=f(z_i, y_i)$.

Definition 2 : Totally parallel subtraction of the subtrahend digit y_i from the minuend digit z_i is performed as the totally parallel addition of the additive inverse of y_i, i.e., $z_i-y_i=z_i+(-y_i)$.

The addition of two digits is performed in two successive steps. First, an outgoing transfer digit t_{i-1} and an interim sum digit w_i are formed:

$$z_i+y_i=rt_{i-1}+w_i \qquad (2.4.1)$$

Then the sum digit s_i is formed:

$$s_i=w_i+t_i \qquad (2.4.2)$$

Definition 1 will be satisfied if the range of values which s_i may assume in (4) does not exceed the allowed the range of values for the digits z_i and y_i may assume in (4) does not exceed the allowed range of values for the digits z_i and y_i in (3). Definition will be satisfied if, every allowed nonzero value of the digit y_i;

For every y_i=a, there exits y_i=-*a* such that a+(-a)=0.The requirement for unique representations of the zero value of a number will be satisfied by the condition:

$$|z_i| \leq r\text{-}1 \qquad (2.4.3)$$

2.5 CARRY-FREE RADIX-2 ADDITION (PPM ADDER)

Redundant number representations limit the carry propagation to a few bit positions, which is usually independent of the word length *W*. This carry propagation-free feature enables fast addition.

A radix-2 signed digit number is coded using two unsigned binary numbers, one is positive and other is negative, as $X = X^+ - X^-$. Hence each signed digit is represented using 2 bits as $x_i = x_i^+ - x_i^-$, where x_i^+, $x_i^- \in \{0, 1\}$ and $x_i \in \{1^-, 0, 1\}$. In adder shown in Fig (2.5.1), one signed digit number x_i is to be added to an unsigned digit y_i. This addition can be carried out in two steps. The first step is carried out in parallel for all bit positions i ($0 \leq i \leq w\text{-}1$). An intermediate sum $p_i = x_i + y_i$ is computed, which lies in the range $\{1^-, 0, 1, 2\}$ [1], [5-7]. This addition is expressed as

$$x_i + y_i = p_i = 2t_i + u_i \qquad (2.5.1)$$

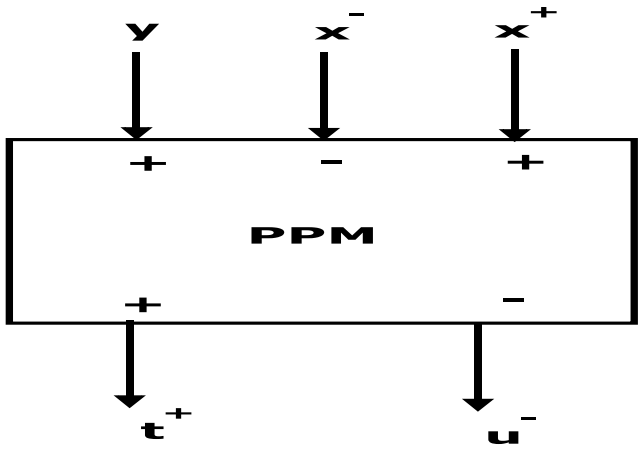


Fig. 2.5.1 PPM adder

Where t_i is the transfer digit and has the value either 0 or 1, and is denoted as t_i^+; u_i is the interim sum and has the value either 1^- or 0, and is denoted as $-u_i^-$. The least significant transfer digit t_{-1}

is assigned the zero value, the same as the most significant interim sum digit u_w. In the second step, the sum digits s_i is formed by combining t_{i-1}^{+} and u_i^{-} as 1 digit as shown in fig (2.5.2) :

$$s_i = t_{i-1}^{+} - u_i^{-} \tag{2.5.2}$$

Table 2.1 summarizes the digit sets involved in adder operation.

Then the addition operation, performed by the adder, is

$$x_i^{+} - x_i^{-} + y_i^{+} = 2t_i^{+} - u_i^{-} \tag{2.5.3}$$

This arithmetic operation can be performed by the adder known as plus-plus-minus adder (PPM).The PPM adder is also called Redundant Binary Full Adder (RBFA).

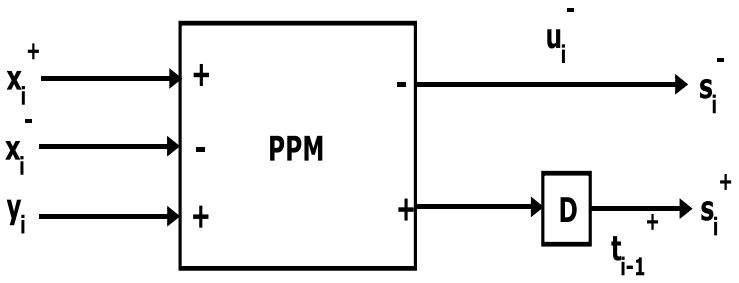


Fig 2.5.2 lsd PPM Adder

Table 2.5.1 Digit sets in addition.

Digit	Digit Set	Binary Code
x_i	{-1, 0, 1}	x_i^{+}-x_i^{-}
y_i	{0,1}	y_i^{+}
$p_i=x_i+y_i$	{-1, 0, 1, 2}	…..
u_i	{-1, 0}	-u_i^{-}
t_i	{0, 1}	t_i^{+}
$s_i=u_i+t_{i-1}$	{-1, 0, 1}	s_i^{+}-s_i^{-}

Fig 2.5.3 shows the structure of a 4-digit parallel addition. In fig. the sum has 5 digits, i.e, 1 more digit than the addends [1].

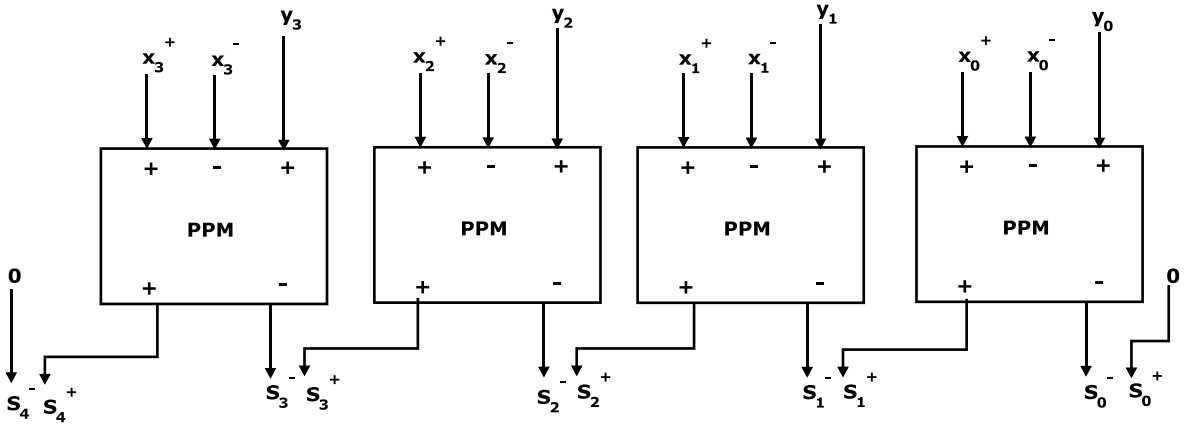


Fig 2.5.3 four-bit digit PPM adder

2.6 Radix-2 subtraction (MMP Subtractor)

The subtractor shown in fig 2.6.1 can subtract an unsigned number from a signed digit number. A radix-2 signed digit number is coded using two unsigned binary numbers, one is positive and other is negative, as $X = X^+ - X^-$. Hence each signed digit is represented using 2 bits as $x_i = x_i^+ - x_i^-$, where x_i^+, $x_i^- \in \{0, 1\}$ and $x_i \in \{1^-, 0, 1\}$. One signed digit number x_i is to be added to an unsigned digit y_i. This subtraction can be carried out in two steps. In the first step, an intermediate difference $p_i = x_i - y_i$ is computed digit independently, which lies in the range $\{2^-, 1^-, 0, 1\}$ shown in Table 2.2 and is expressed using following equation [1] :

$$x_i - y_i = p_i = 2t_i + u_i \tag{2.6.1}$$

where the transfer digit t_i has value either 1^- or 0, and is denoted as $-t_i^-$, the interim difference u_i has value either 0 or 1, and is denoted as u_i^+. In the second step, the sum digit s_i is formed by combining t_{i-1}^- and u_i^+ as 1 digit:

$$s_i = -t_{i-1}^{-} + u_i^{+} \tag{2.6.2}$$

Then the subtraction operation, performed by the subtractor, is

$$x_i^{+} - x_i^{-} - y_i^{-} = -2t^{-} + u^{+} \tag{2.6.3}$$

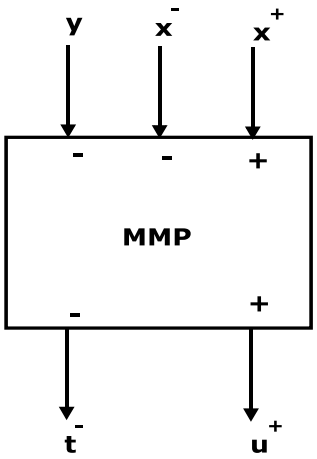


Fig 2.6.1 MMP adder

This arithmetic operation can be performed by the subtraction known as minus-minus-plus (MMP) subtractor or type-2 full adder. Fig 2.6.2 shows the structure of a 4-digit parallel radix-2 subtractor.

.

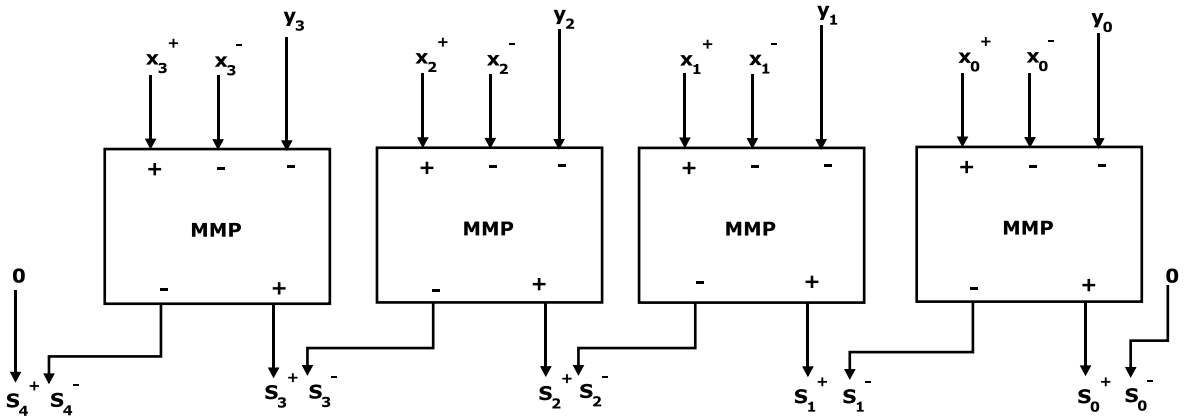


Fig 2.6.2 four-bit digit MMP subtractor.

Table 2.6.1 Digit sets in subtraction

Digit	Radix-2 Digit Set	Binary Code
x_i	{-1, 0, 1}	$x_i^+ - x_i^-$
y_i	{0, 1}	y_i^-
$p_i = x_i - y_i$	{-2, -1, 0, 1}	…….
u_i	{0,1}	u_i^+
t_i	{-1, 0}	$-t_i^-$
$s_i = u_i + t_{i-1}$	{-1, 0, 1}	$s_i^+ - s_i^-$

2.7 Digit-serial SBD redundant adder

In Digit-serial SBD adder shown in fig() , two redundant binary numbers x_i (= $x_i^+ - x_i^-$) and y_i (= $y_i^+ - y_i^-$) can be added simultaneously and gives the result as a redundant binary digit sum s_i (= $s_i^+ - s_i^-$). This adder consists of PPM adder, MMP subtractor and D-FF (delay). This adder behaves as pipelining architecture, by which critical path will be reduced and hence reduction of the propagation delays [1].

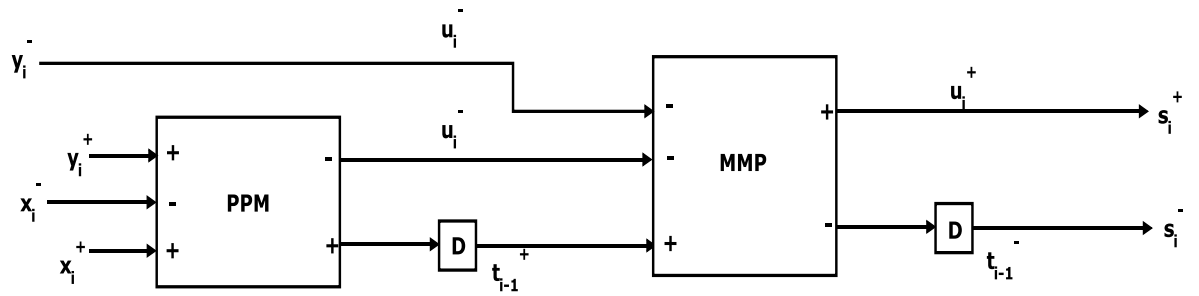


Fig 2.7 Digit-serial SBD redundant adder.

2.8 Radix-2 Redundant Binary Multiplier

Consider the bit-serial multiplication of two W-bit numbers a and b to yield a product *p* as described by the algorithm below [9]:

Algorithm

INPUT: a, b

OUTPUT: p

INITIALIZE: $a_{i,} b_i = 0$ for $i > W-1$

$c_{i,j}\ s_{i,j} = 0\ \forall\ i, j$

begin

for i=0 to W-1

begin

for j=0 to W

begin

$$a_i * b_j + c_{i,j-1} + s_{i-1,j+1} = 2c_{i,j} + s_{i,j} \qquad (2.8.1)$$

Using systolic design method, the resulting bit-serial multipliers of above equation are shown in fig (2.8.1) [1, 13,14].

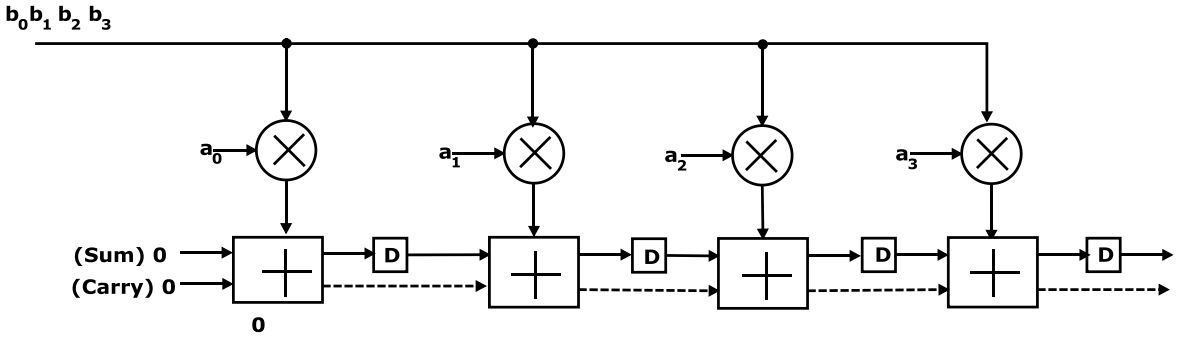


Fig 2.8.1 redundant multiplier architecture.

Consider the multiplicand $B(b_3b_2b_1b_0)$ is to be a radix-*2* redundant number, the number $A(a_3a_2a_1a_0)$ is to be in unsigned representation. Each digit $b_j = b_j^+ - b_j^-$ of a radix-*2* redundant number B is recoded (shown Table 2.8) using a sign bit and a magnitude bit as follows:

Table 2.8.1 Recoding of b_j

b_j	-1	0	1
sign(b_j)	1	0	0
$\lvert b_j \rvert$	1	0	1

:

If the input bit b_j is positive, then the adder cell corresponding to coefficient a_i, $0 \le i \le 2$ in fig 2.8.1 can be implemented as an full adder; the last adder cell, which involves the most significant sign bit of A with negative weight, carries out the following computation [1]:

$$-a_3 * b_j + carry_{in} + sum_{in} = 2* carry_{out} - sum_{out} \qquad (2.8.2)$$

Which can be implemented as a PPM adder consisting of an full adder and 2 inverters. If the input digit b_j is negative, then combining above two equations and the detailed multiplier circuit shown in fig 2.8.2.

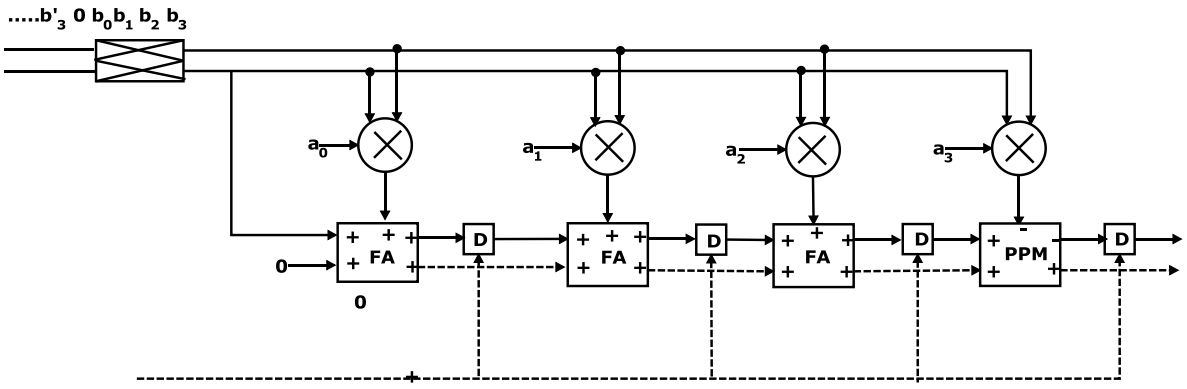


Fig 2.8.2 redundant multiplier with PPM

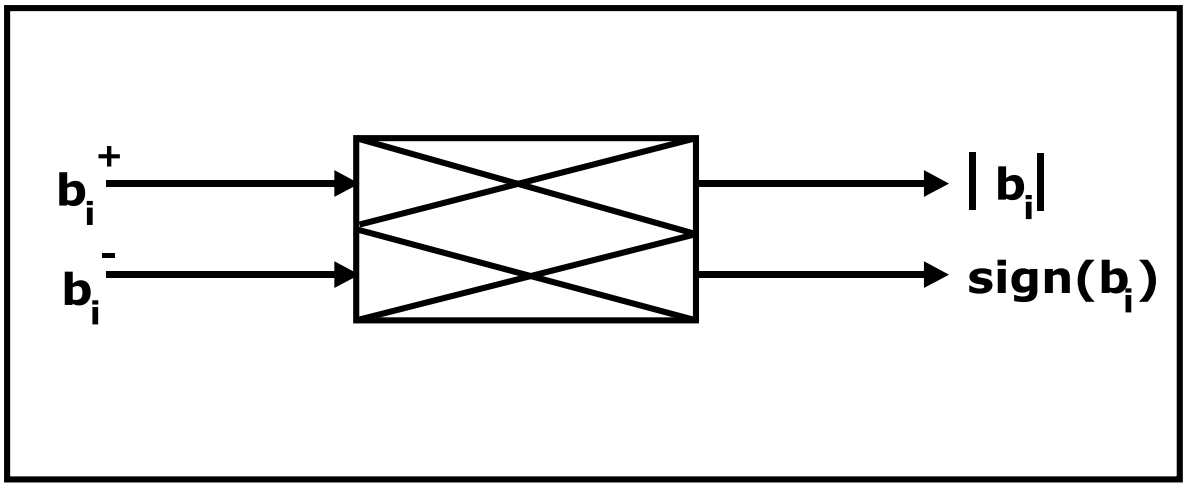


Fig 2.8.3 Recoding of b_j

2.9 Redundant Binary to Binary Conversion

The conversion process from redundant binary to binary format in lsd-format mode can be carried out by considering x^+ and x^- as 2 independent unsigned numbers and subtracting x^- from x^+ as follows:

$$x_i^+ - x_i^- - c_i = -2c_{i+1} + s_i \,, \qquad (2.9.1)$$

where an MMP adder is used at each bit position. Lsd-first redundant binary to binary conversion circuit is shown in fig 2.9.1 for a word-length of 4 bits. In this circuit carryout at any stage can be either 0 or 1^- [1, 14].

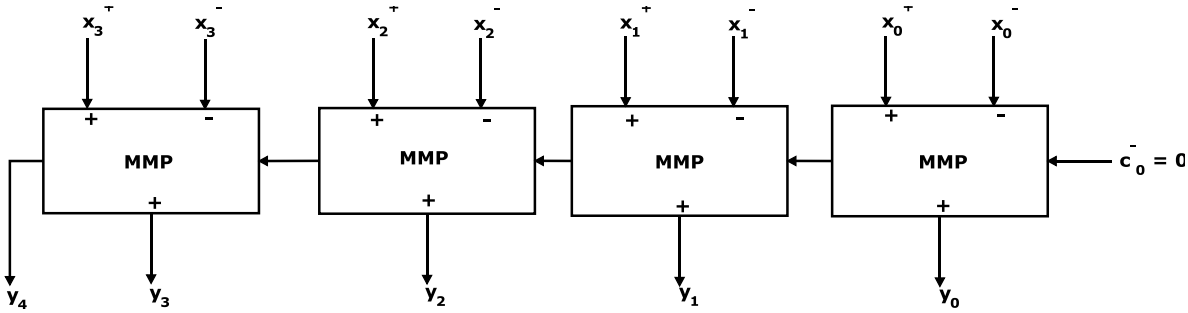


Fig 2.9 RB-to-Binary conversion.

Chapter 3

3.1 FIR Filter Theory

A filter is used to remove some component or modify some characteristic of a signal, but often the two terms are used interchangeably. A digital filter is simply a discrete-time, discrete-amplitude convolved. Basic Fourier transform theory states that the linear convolution of two sequences in the time domain is the same as multiplication of two corresponding spectral sequences in the frequency domain. Filtering is in essence the multiplication of the signal spectrum by the frequency domain impulse response of the filter [1].

A finite impulse response (FIR) filter performs a weighted average of a finite number of samples of the input sequence. The basic input-output structure of the FIR filter is a time-domain computation based on a feed-forward difference equation. Figure 3.1 shows a flow diagram of a standard 3-tap FIR filter. The filter has seven data registers. The FIR is often termed a transversal filter since the input data transverses through the data registers in shift register fashion. The output of each register (D1 to D2) is called a tap and is termed x[n], where n is the tap number. Each tap is multiplied by a coefficient c_k and the resulting products are summed. A general expression for the FIR filter's output can be derived in terms of the impulse response. Since the filter coefficients are identical to the impulse response values, the general form of a standard FIR filter can be represented as Equation 3.1.

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k] \tag{3.1}$$

When the relation between the input and the output of the FIR filter is expressed in terms of the input and the impulse response, it is called a finite convolution sum. We say that the output is obtained by convolving the sequences x[n] and h[n]. There is a simple interpretation that leads to a better algorithm for achieving convolution. This algorithm can be implemented using the tableau that tracks the relative position of the signal values. The example in Figure 2.3 shows how to convolve x[n] with h[n]. The determination of filter coefficients controls the characteristic of the FIR filter.

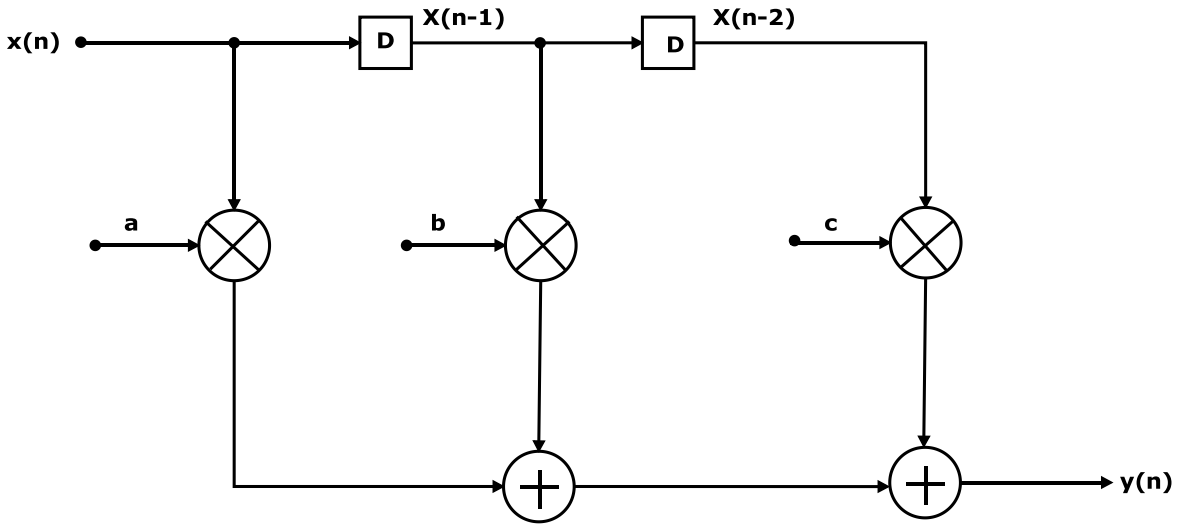


Fig 3.1 FIR filter.

3.2 Architecture of FIR filter

The speed of the filter is defined as the rate at which input samples can be processed. To increase the speed it is necessary to reduce the critical path between input and output. The critical path is defined to be the path with the longest computation time among all paths that contain zero delays. Fig 3.1 shows direct chain has an estimated delay of

$$T_{chain} = T_m + (N-1)T_a \tag{3.2}$$

The sample period (T_{sample}) is given by,

$$T_{sample} \geq T_m + (N-1)T_a \tag{3.3}$$

Therefore the sampling frequency (f_{sample}) is given by

$$f_{sample} \leq \frac{1}{Tm+2Ta} \tag{3.4}$$

For 3-tap FIR filter, the critical path delay is (T_m+2T_a). Pipelining reduces the effective critical path by introducing pipelining latches along the data path. The critical path is now reduced from T_m+2T_a to T_m+T_a shown in fig (3.2a and 3.2b). In this arrangement while the left adder initiates the computation of the current iteration the right adder is completing the computation of the previous iteration result [1].

Fig 3.2a datapath

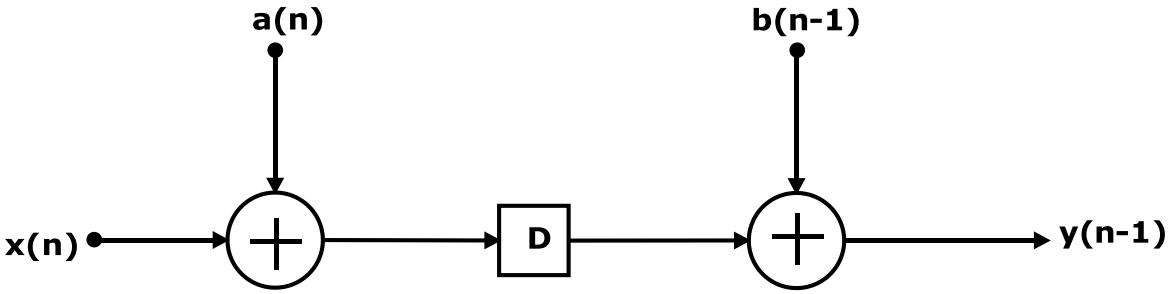


Fig 3.2b 2-level pipelined structure

Another FIR filter known as transposed or data-broadcast structured shown in fig3.3. The critical path of the filter of fig3.1 can be reduced without introducing any pipelining latches by transposing structure. Now the propagation delay is T_m+T_a.

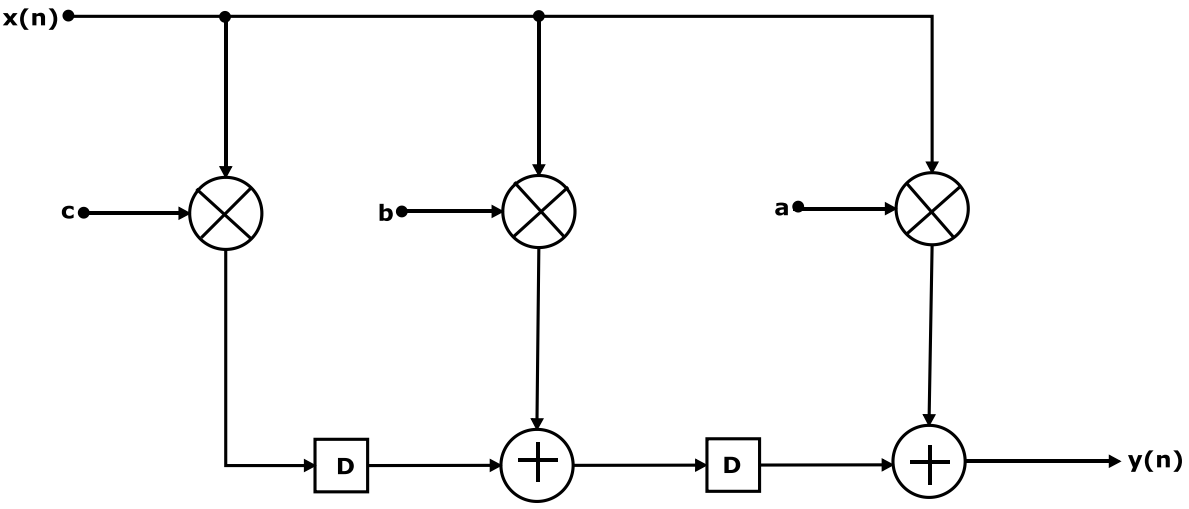


Fig 3.3 transposed FIR filter.

3.3 BOX-CAR FIR filter

If the multiplier coefficients of the filter are 1, then the filter is called box-car FIR filter. The critical path depends upon only the time needed for addition operation.

The FIR filter consists of three main components:

(1) A D-FF to implement a simple delay.
(2) A Multiplier to implement the coefficients.
(3) An Adder to sum the nodes at the end of each tap.

Chapter 4

4.1 Architecture, Design and Implementation of Different Digital Cells

Design of 3-tap FIR filter is being implemented in the 090nm CMOS technology and all architecture and simulation is done using Cadence Design Environment 5.1.41. The FIR filter IC design consists of D-FF, the multiplier, and the adder. By using functional description approach design, all individual architecture digital logic cells be designed. Because functional description utilizes the modular design making it easier to understand. Since each block will be thought out individually, the designer has intimate knowledge of how their circuit works. The downside of this method is that the circuit may be of less than optimal size. From transistor level to gate level design of different digital cells such as D-FF, adder and multiplier, first CMOS inverter is be taken as reference cell.

4.2 CMOS Inverter

For 90nm CMOS technology, power supply V_{DD} is 1.8v.The schematic of inverter shown in fig-4.2.1.

Using gpdk090nm technology library, (see APPENDIX-I)

$\mu_n C_{OX}$ = 300 $\mu A/V^2$, $\mu_p C_{OX}$ =170 $\mu A/V^2$
Width of PMOS = Wp
Width of NMOS = W_n
Length of PMOS and NMOS = $L_p = L_n$ =100nm.

For better noise margin or symmetrical inverter design, the voltage V_I is called the inverter gate threshold voltage, and is defined by the point where the voltage transfer curve intersects the unity gain line defined by $V_{out} = V_{in}$ [3].

Device transconductances value on NMOS is $\beta_n = k_n(W/L)_n$ and for PMOS $\beta_p = k_p(W/L)_p$.

(β_n/β_p) =1.083
But (β_n/β_p) =1.76(W_n/W_p)
⇨ W_p/W_n =1.63.

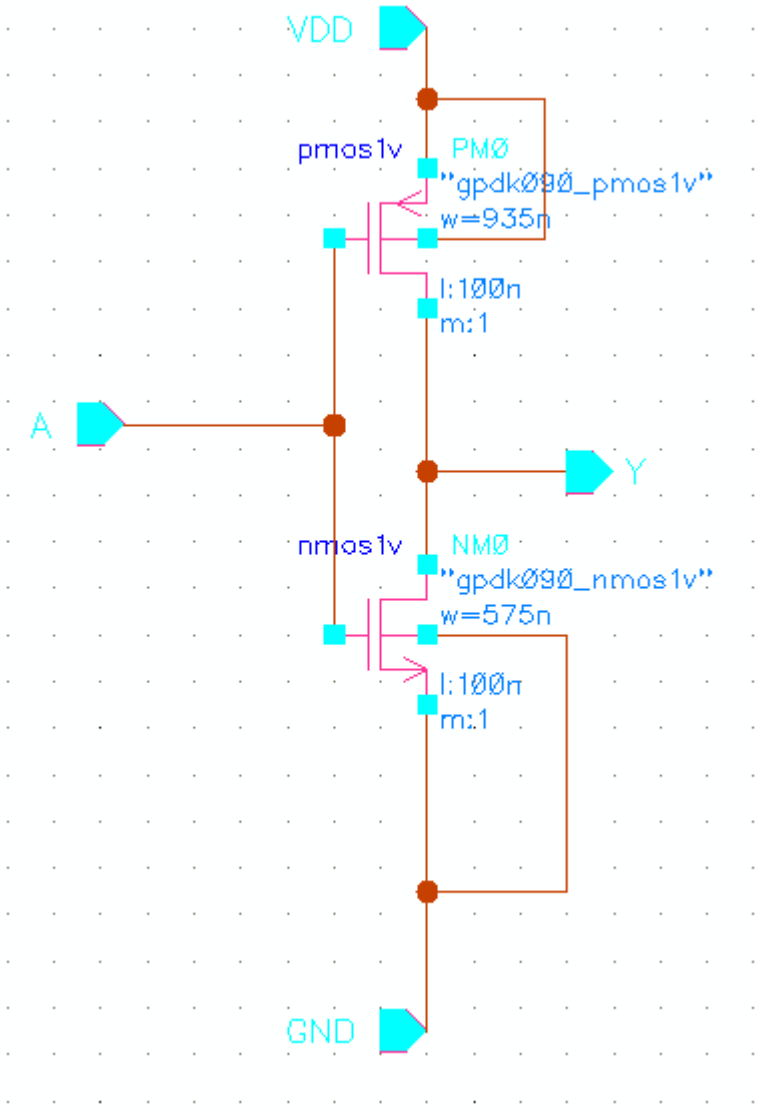


Fig 4.2.1 CMOS Inverter schematic.

Specification of CMOS inverter

Maximum switching frequency f_{max} =10 GH$_z$, rise time(t_r) = fall time(t_f)

4.2.1 Design

The sum of the transient times ($t_r + t_f$) represents the minimum time needed for a gate to undergo

a complete switching cycle, i.e, for the output to change from a logic 1 to a logic 0 voltage, and then back up to a logic 1 value. We may use this to define the **maximum switching frequency** by

$$f_{max} = 1/(t_r + t_f) \tag{4.2.1}$$

Switching performance of CMOS digital circuits are characterized by the time intervals required to charge and discharge capacitors at output nodes. CMOS inverters use transistors to provide current flow paths between the power supply (Mp) and ground (Mn). All switching times are thus set by the current levels and the value of C_{out}. The output high-to-low time represents the time interval needed for the output capacitor to discharge through the n-channel MOSFET Mn when Mp is in cutoff. is also referred to as the **fall time** (t_f) for the circuit since it gives the time needed for the output to decay from a well-defined logic 1 state to a well-defined logic 0 state. The low-to-high time also known as the rise time (t_r) represents the time interval needed for the output capacitor to charge through the p-channel MOSFET Mp. During this time interval, Mn is in cutoff while Mp is conducting from the power supply [3].

From the design specifications:

t_r =(1/20 GH_z)

$$t_r = \left\{ \frac{2(V_{Tn} - V_0)}{(V_{DD} - V_{Tn})} + \ln\left(\frac{2(V_{DD} - V_{Tn})}{V_0} - 1 \right) \right\} R_n C_{out} \tag{4.2.2}$$

$$R_n = \frac{1}{\beta_n (V_{DD} - V_{Tn})} \tag{4.2.3}$$

Where V_{Tn} : threshold voltage of NMOS.

R_n : resistance of NMOS.

C_{out} : capacitive load applied to the output of the inverter=50f.

V_0 : $0.1V_{DD}$.

V_1 : $0.9V_{DD}$.

Putting all the values together, we have

$(W/L)_n$ =5.725

W_n =5.725*100nm =572.5nm

But (W_p/W_n) =1.63

W_p =1.63*W_n =933.175nm.

Using gpdk090nm CMOS technology in cadence IC 5.1.41, W_p =935n, W_n =575n. The schematic of the inverter shown in fig 4.2.1 [11].

4.2.2 Simulation of CMOS inverter

Transistors levels are simulated by using SPECTRE simulator in the Analog Design Envirnoment. Both DC and transient analysis are done shown in fig4.2.2. Then the Affirm Analog test bench was created to test the schematics [11]. The Analog simulations will show the effects of capacitance related to transistor sizing and therefore clock skew, signal delays and setup-and-hold violations will become evident.

The propagation delay time t_P is the logic delay through a gate. Physically we interpret as the average time needed for the output to respond to a change in the input logic state. By definition,

$$t_P = \frac{(t_{PHL} + t_{PLH})}{2} \tag{4.3.1}$$

Where t_{PHL} and t_{PLH} represent the propagation delays for a high-to-low, and a low-to-high transition, respectively. Let us define the 50% voltage points as $V_{I/2} = 0.5V_{DD}$ .Then, t_{PHL} and t_{PLH} are defined by the time intervals between the input and output voltages.

From the simulation

$$t_{PHL} = 35.45\,p\text{sec}$$
$$t_{PLH} = 141.9\,p\text{sec}$$
$$t_P = 88.7\,p\text{sec}$$

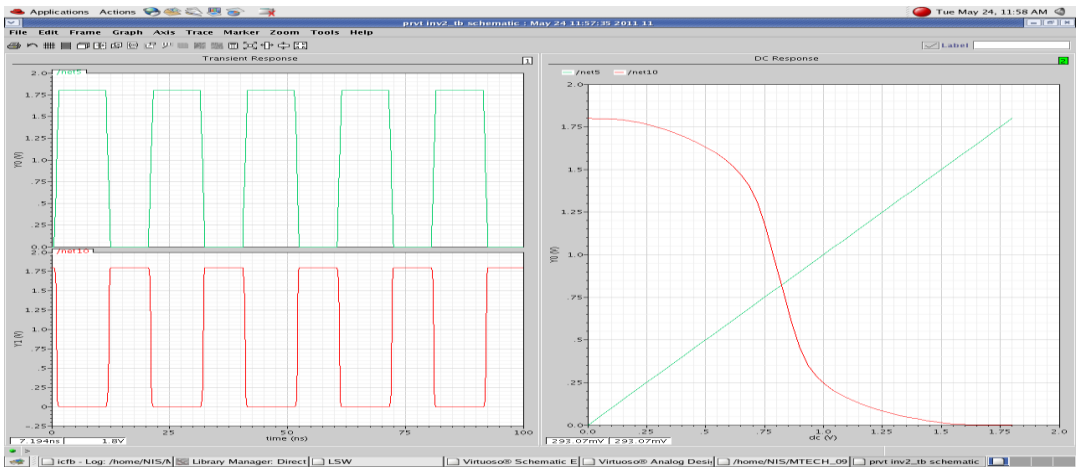


Fig 4.2.2 transient and DC analysis waveforms.

4.2.3 Layout

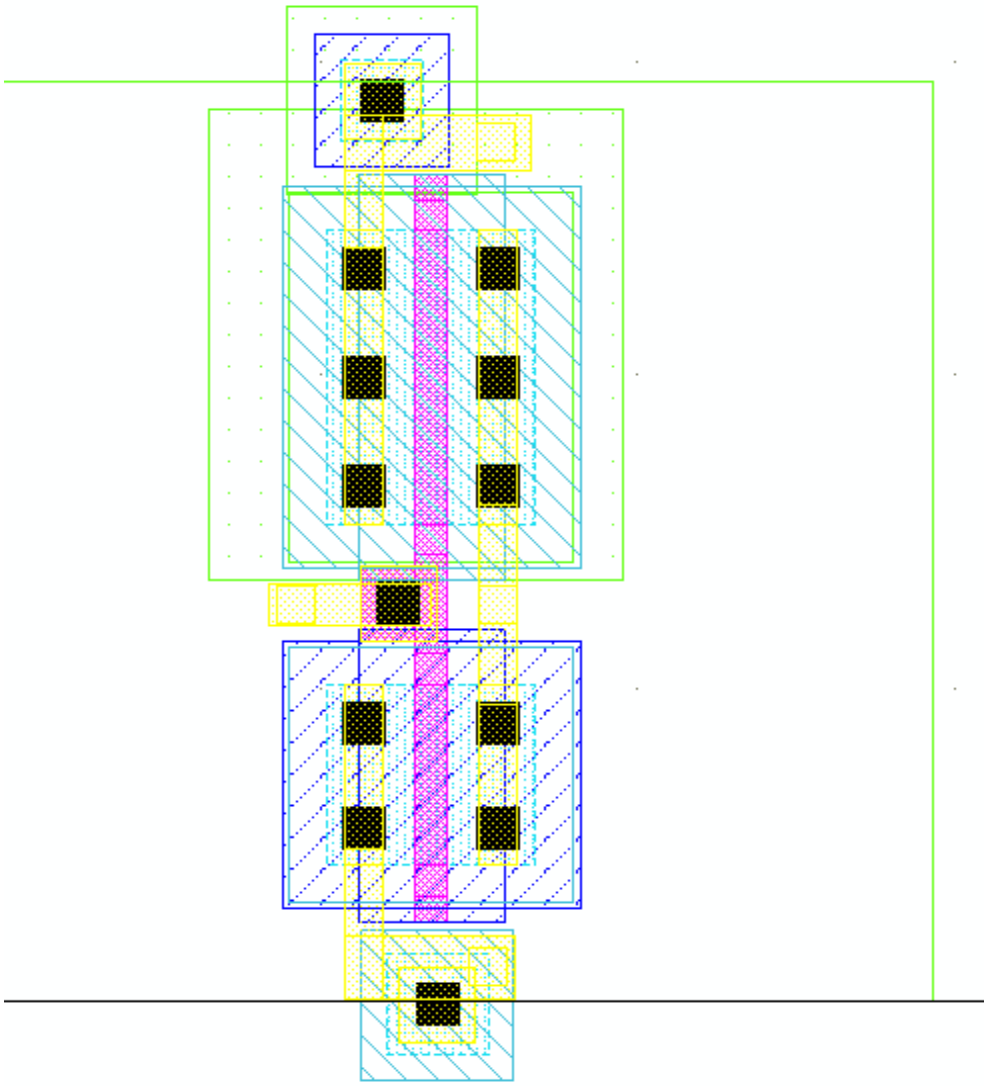


Fig 4.2.3 CMOS inverter layout

4.2.4 Parametric Extraction and Post-layout simulation

Fig 4.1.4 shows the parametric extraction of CMOS inverter. After post-layout simulation the propagation delay is 90.2 psec.

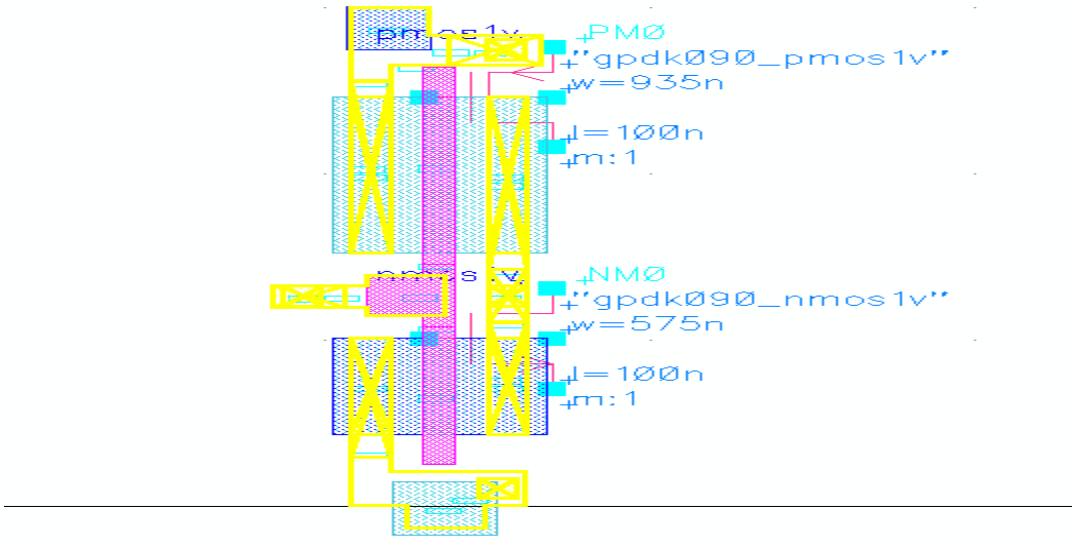


Fig 4.2.4 avs extraction of CMOS Inverter.

4.3 NAND-2

Two PMOS transistors are connected in parallel and two NMOS are connected in series shown in fig 4.3.1 [3].

$(W_n)_{NAND} = 2\ (W_n)_{INV} = 575*2 = 1150nm$

$(W_p)_{NAND} = (W_p)_{INV} = 935nm.$

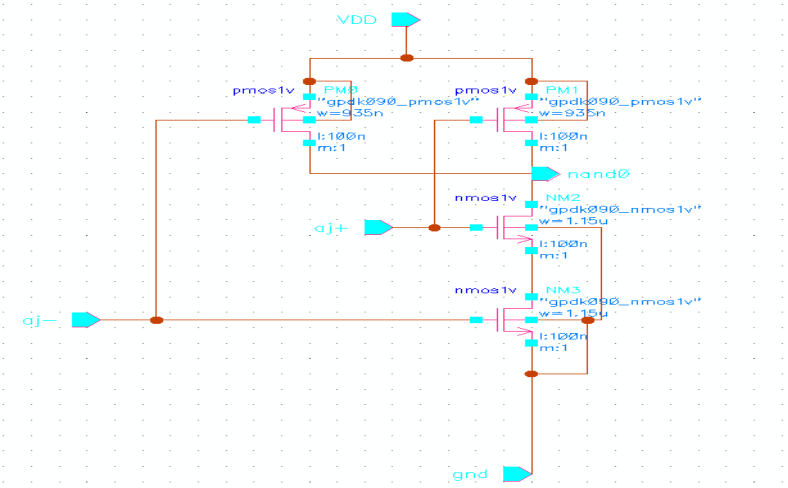


Fig 4.3.1 NAND2 schematics.

$t_{PHL} = 104.9 p\sec$

$t_{PLH} = 37.91 p\sec$

$t_P = 71.41 p\sec$

. The layout and parametric extraction are shown in fig 4.3.2 and fig 4.3.3. The propagation delay after post-layout is 71.41*psec*.

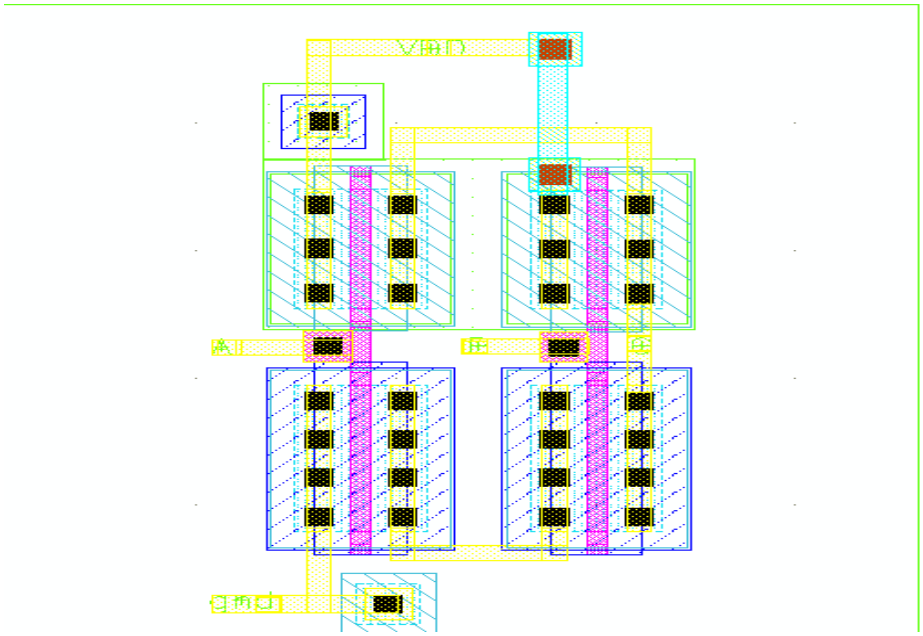


Fig 4.3.2 NAND2 Layout

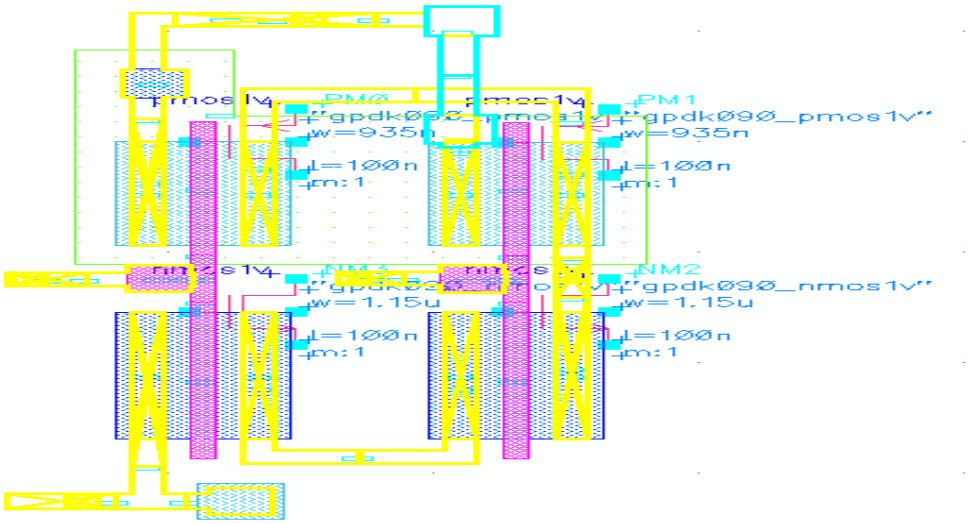


Fig 4.3.3 Extraction of NAND2

4.4 NAND3

Three PMOS transistors are connected in parallel and three NMOS are connected in series shown in fig 4.4.1 [3].

$(W_n)_{NAND} = 3\ (W_n)_{INV}$ =575*3=1725nm.

$(W_p)_{NAND} = (W_p)_{INV}$= 935nm.

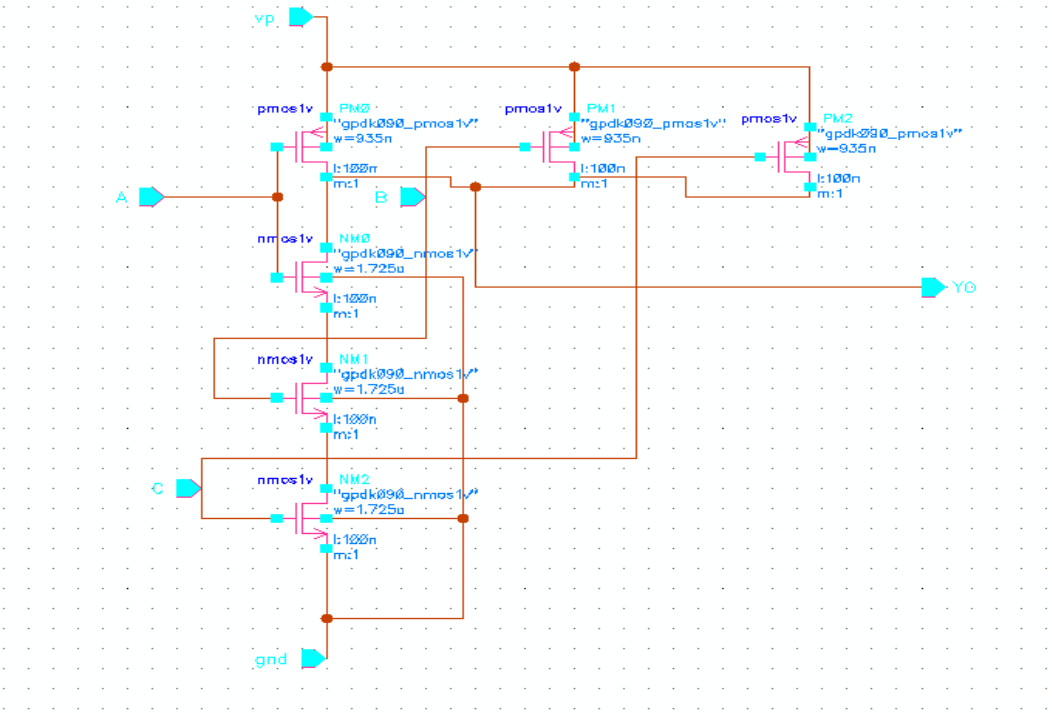


Fig 4.4.1 NAND3 schematics.

The layout and parametric extraction are shown in fig 4.4.2 and fig 4.4.3. The propagation delay in pre-layout and post-layout is measured as 56 psec.

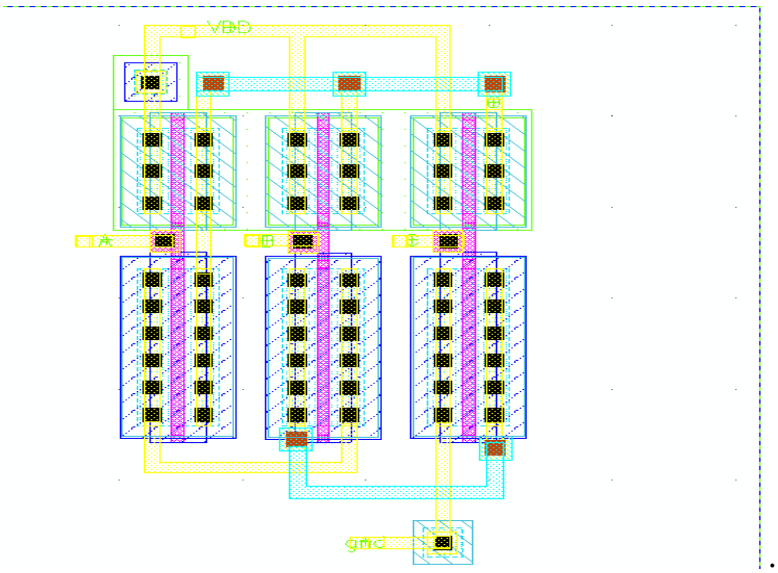
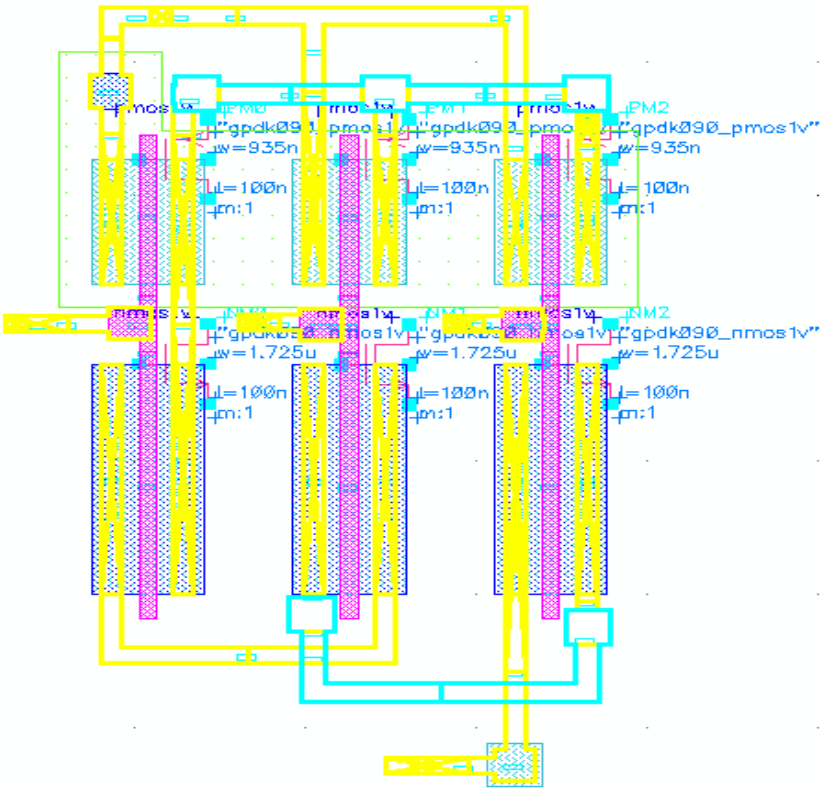


Fig 4.4.2 layout of NAND3.



4.4.3 Extraction of NAND3.

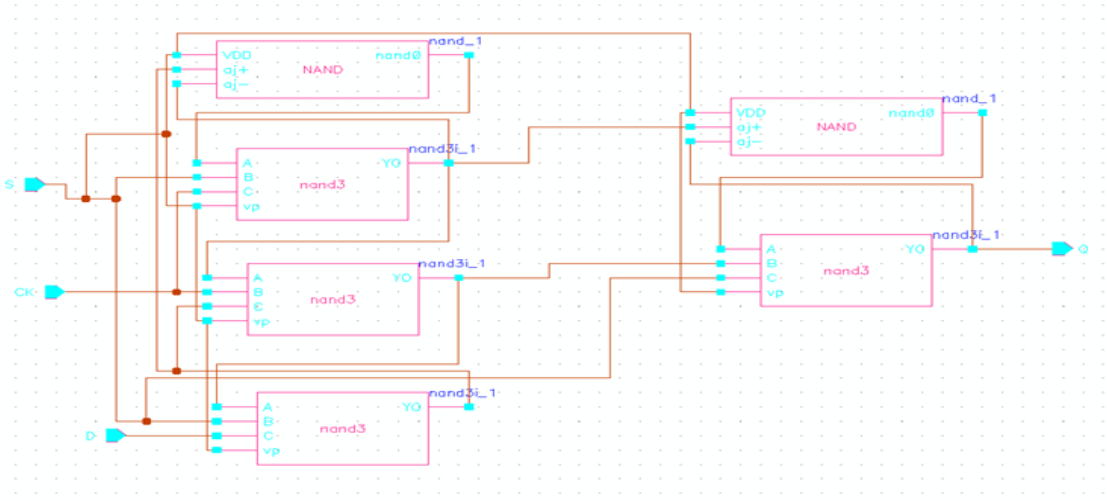
4.5 D-FF (Delay)

A D-flip-flop was made from NAND3s, NAND2s, and an Inverter shown in fig 4.5.1.

D-FF with Set

With S = 1.8V, Q = D

With S = 0V, Q = 0V



. Fig 4.5.1 D-FF schematics

After simulation of D-FF, the result gives

$t_{PHL} = 12.99ns$

$t_{PLH} = 11.96ns$

$t_P = 12.47ns$

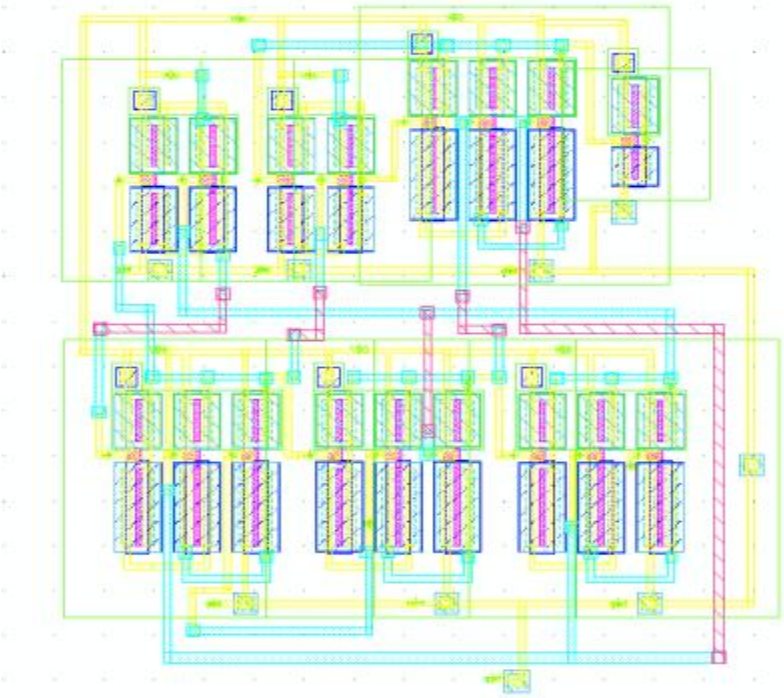


Fig 4.5.2 layout of D-FF.

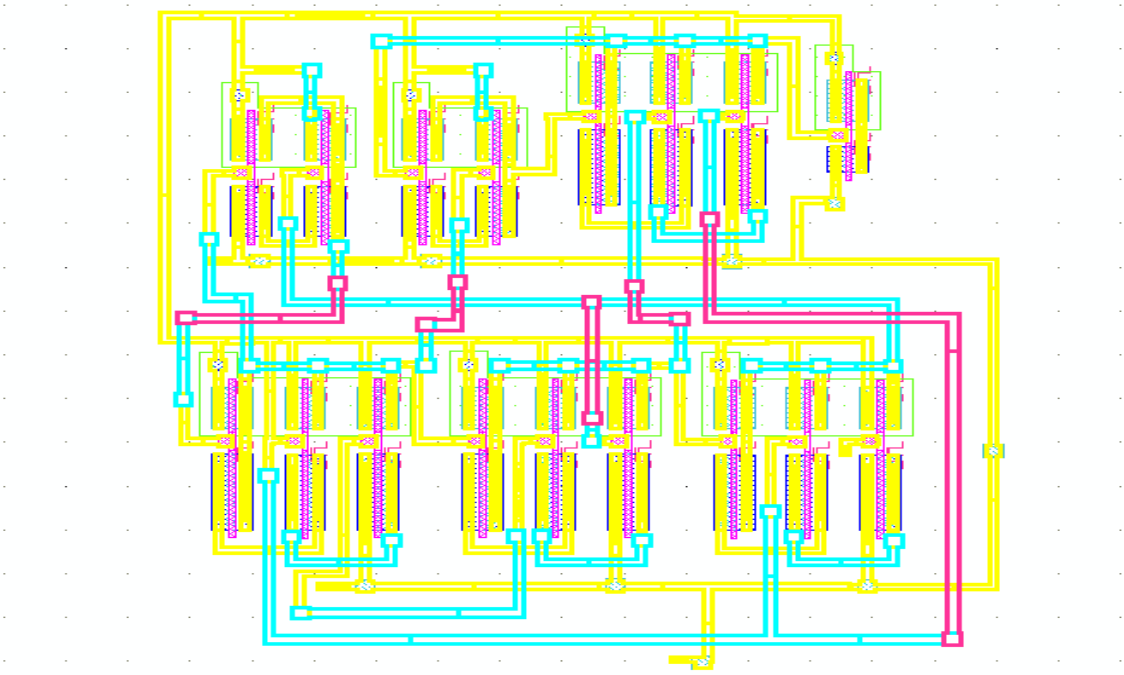


Fig 4.5.3 extraction of D-FF.

4.6 PPM Adder

The PPM Adder performs the following operation:

$$x + y = x^{+} - x^{-} + y = 2t^{+} - u^{-} \tag{4.6.1}$$

Using above equation t^{+} and u^{-} are represented in

$$u^{-} = x^{+} \overline{\oplus} x^{-} \overline{\oplus} y \tag{4.6.2}$$

$$t^{+} = x^{+}(x^{+} \oplus x^{-}) + u^{-}(x^{+} \overline{\oplus} x^{-}) \tag{4.6.3}$$

Based on equations (4.6.2, 4.6.3), u^{-} consists of two XNOR gates and t^{+} be used using XNOR, XOR and pass transistors shown in fig 4.6.1. The Adder consists of 10 transistors. From figure , using two 4-transistors XNOR gates to generate u^{-} and two pass gates to generate t^{+}. These two gates are based on pass transistor logic causing threshold voltage V_{Tn} losses for specific input sets. Because n-MOS pass transistors from voltage loss when transmitting logic 1, while p-MOS degrades the transmission of logic 0 voltage level by V_{Tp} instead of 0 [3], [5-7]. The logic 1 output voltage of the 10-transistors PPM Adder degraded to $V_{DD} - 2V_T$ instead of V_{DD} and $2V_T$ instead of 0v shown in fig 4.6.2.

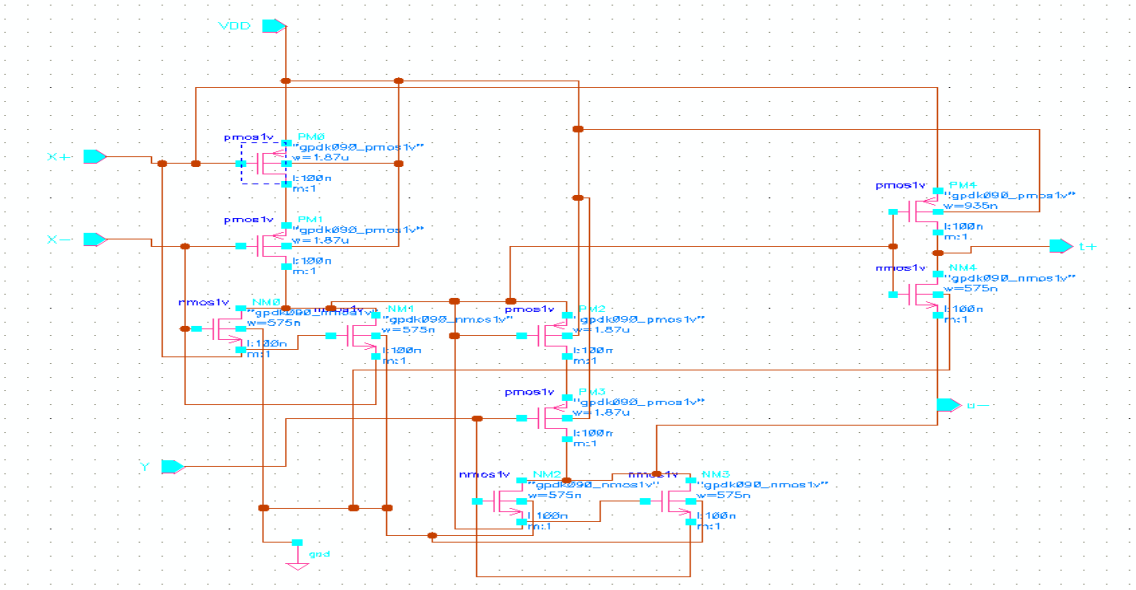


Fig 4.6.1 PPM Adder schematics.

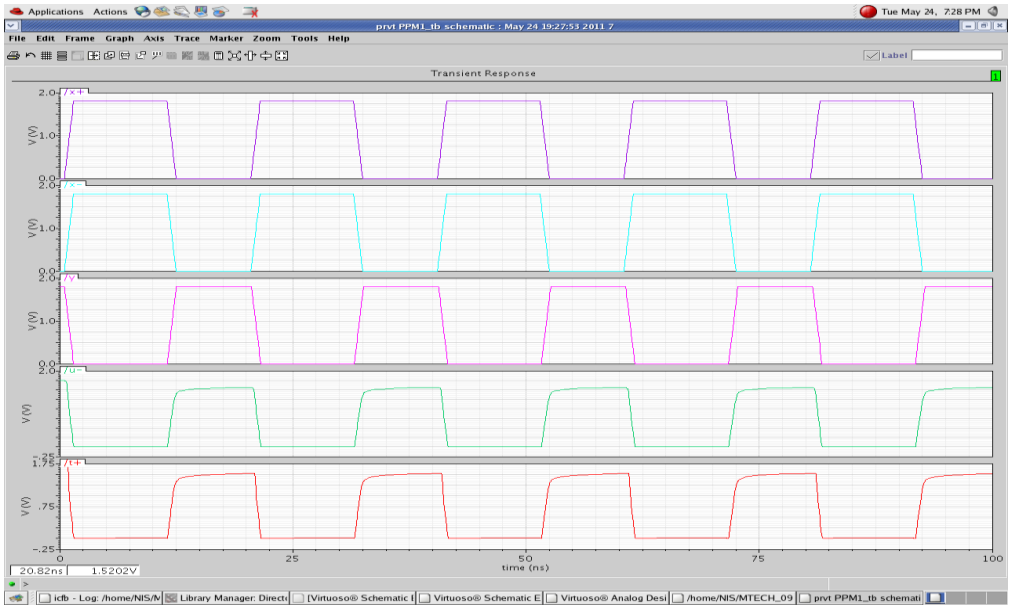


Fig 4.6.2 waveform of PPM Adder.

Layout and extractions are shown in fig 4.6.3 and 4.6.4. Comparison of t_P between pre-layout and post-layout simulation shown below:

	Pre-layout	post-layout
t_{PHL}	*75.91psec*	*83.06psec*
t_{PLH}	*79.85psec*	*81.44psec*
t_P	*77.74psec*	*82.25psec*
avg pwr	*1.968µwatt.*	*3.305µwatt.*

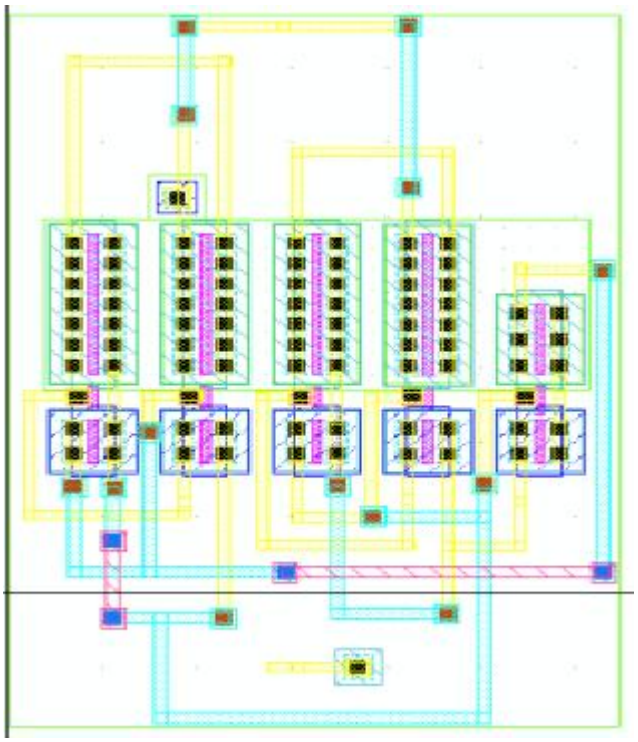


Fig 4.6.3 PPM Adder layout.

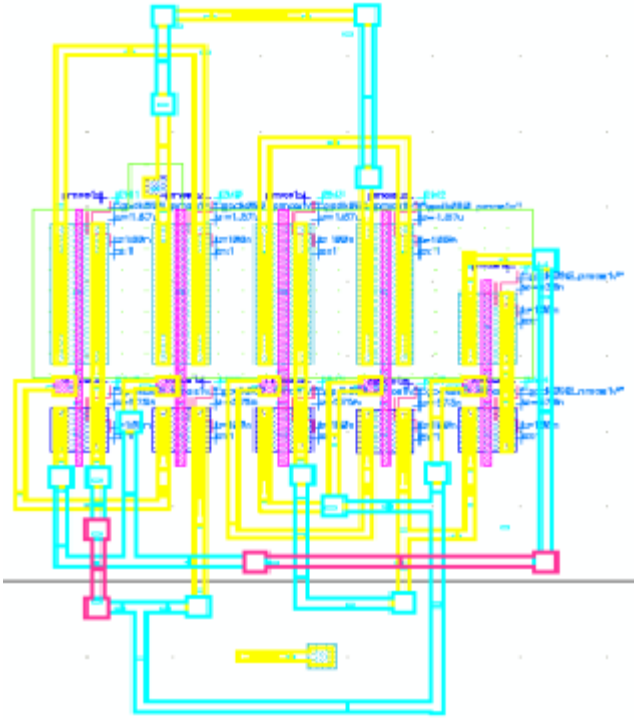


Fig 4.6.4 Extraction PPM Adder.

4.7 MMP Subtractor

The MMP subtractor performs the following operation:

$$x - y = x^{+} - x^{-} - y = -2t^{-} + u^{+} \tag{4.7.1}$$

Using above equation t^{-} and u^{+} are represented in

$$u^{+} = x^{+} \overline{\oplus} x^{-} \overline{\oplus} y \tag{4.7.2}$$

$$t^{-} = x^{-}(x^{+} \oplus x^{-}) + u^{+}(x^{+} \overline{\oplus} x^{-}) \tag{4.7.3}$$

Based on equations (4.7.2, 4.7.3), u^{+} consists of two XNOR gates and t^{-} be used using XNOR, XOR and pass transistors shown in fig 4.7.1. The subtractor consists of 10 transistors. From figure, using two 4-transistors XNOR gates to generate u^{+} and two pass gates to generate t^{-}. These two gates are based on pass transistor logic causing threshold voltage V_{Tn} losses for specific input sets. Because n-MOS pass transistors from voltage loss when transmitting logic 1, while p-MOS degrades the transmission of logic 0 voltage level by V_{Tp} instead of 0 . The logic 1 output voltage of the 10-transistors MMP subtractor degraded to $V_{DD} - 2V_T$ instead of V_{DD} and $2V_T$ instead of 0v as shown in fig 4.7.2 [3], [6].

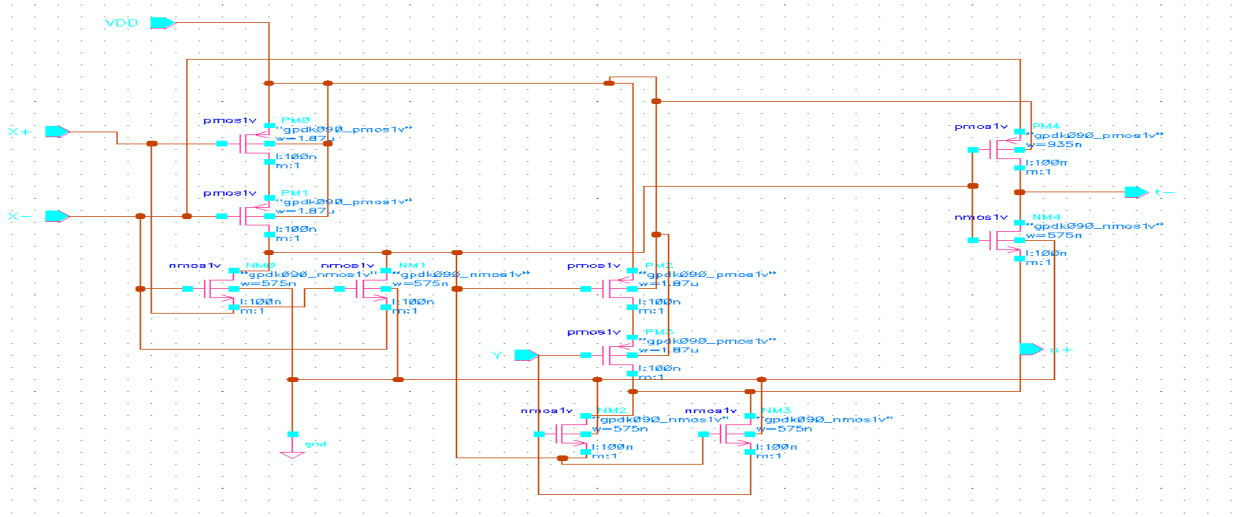


Fig 4.7.1 MMP Subtractor schematics.

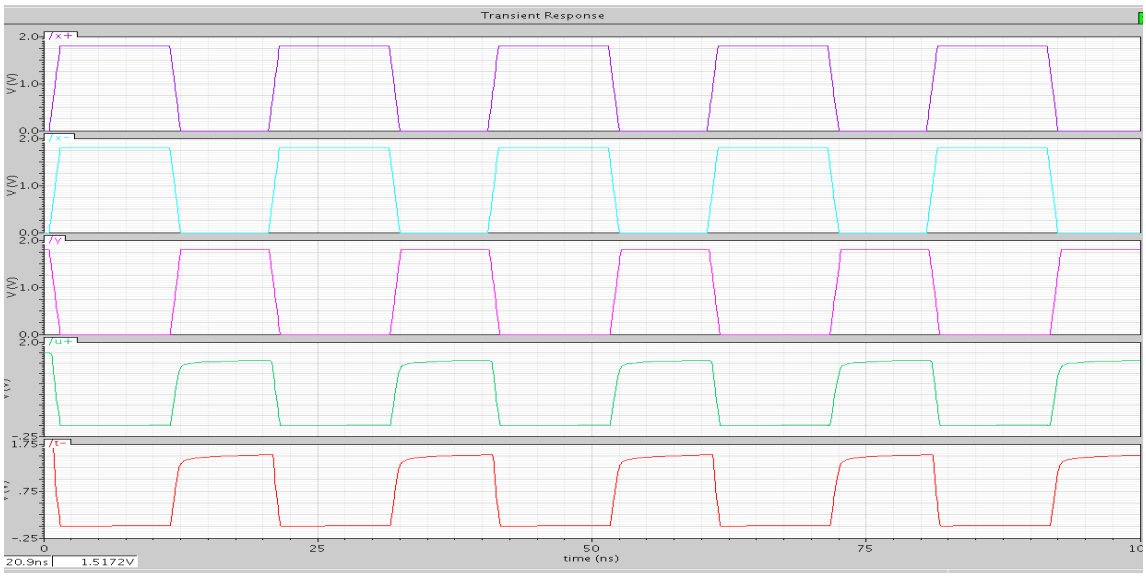


Fig 4.7.2 waveform of MMP Subtractor.

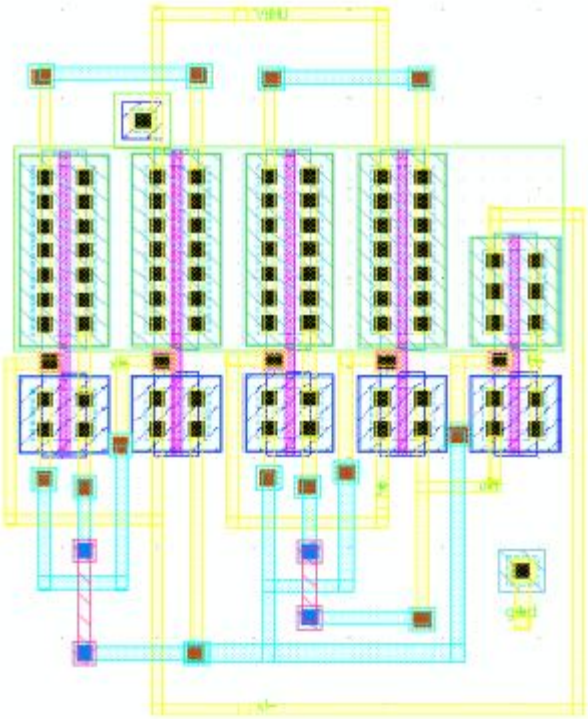


Fig 4.7.3 layout of MMP Subtractor.

The layout and extraction of MMP are shown in fig 4.7.3 and fig 4.7.4 and comparison of simulation results shown.

	Pre-layout	post-layout
t_{PHL}	*75.91psec*	*84.39psec*
t_{PLH}	*79.53psec*	*81.42psec*
t_P	*77.22psec*	*82.91psec*
avg pwr	*3.49µwatt.*	*3.299µwatt.*

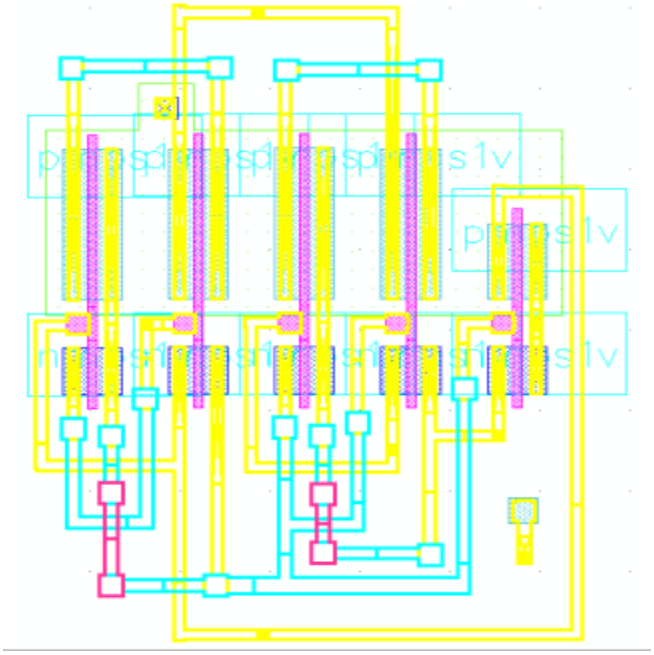


Fig 4.7.4 Extraction of MMP Adder.

4.8 Digit-serial SBD redundant adder

The digit-serial SBD redundant adder consists of three components, PPM Adder, MMP subtractor and delays shown in fig 4.8.1, simulation waveform shown in fig 4.8.2 and propagation delay and average power dissipation are measured.

$t_{PHL} = 20.99n$ $\quad\quad$ $t_P = 11.48n$

$t_{PLH} = 1.964n$ $\quad\quad$ $power_{avg} = 182.1\mu watt.$

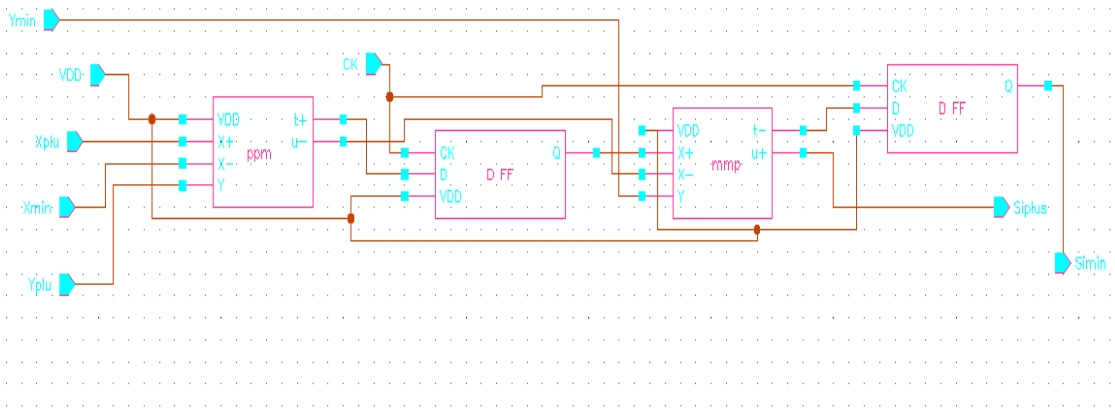


Fig 4.8.1 SBD Adder schematic.

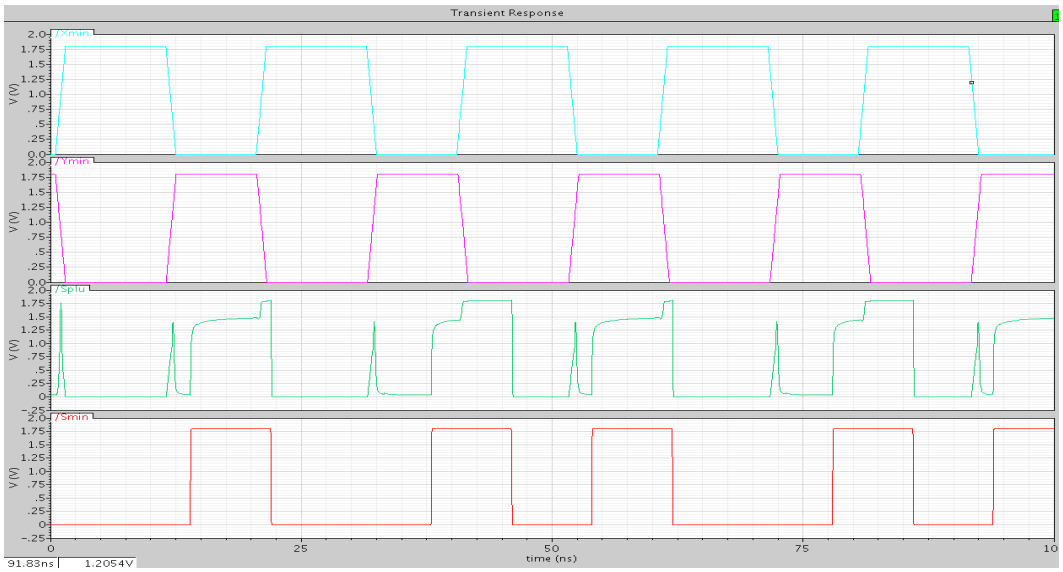


Fig 4.8.2 Simulation waveform SBD Adder.

4.9 Box-car FIR filter

4-tap, 1-bit input and 4-bit input Box-car FIR filter are shown in fig 4.9.1 and fig 4.9.2. The simulation waveform shown in fig 4.9.3, and measured propagation delay is 11.48nsec. But avg pwr for 1-bit input is 663.4µwatt, and 2.653mwatt for 4-bit input.

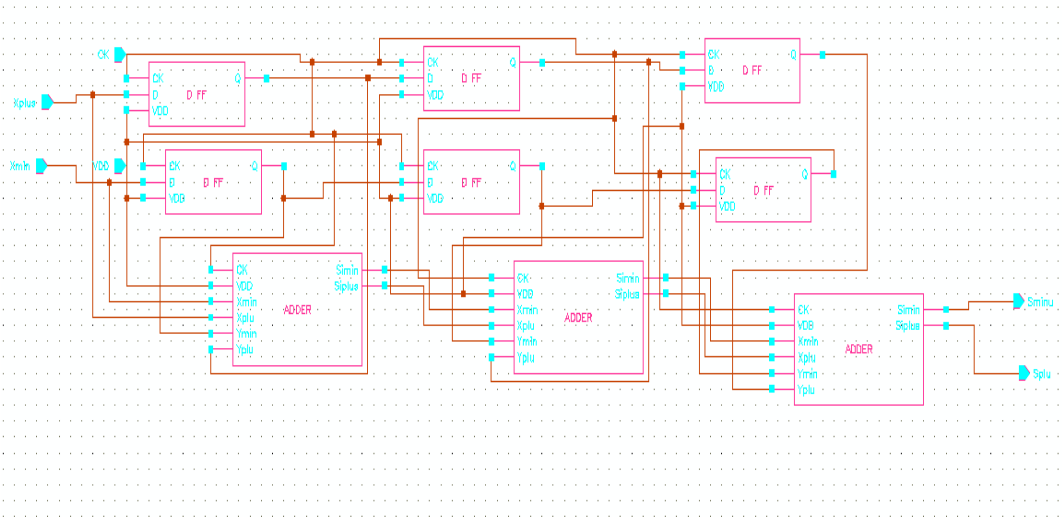


Fig 4.9.1 4-tapBoxcar FIR filter(1-bit input).

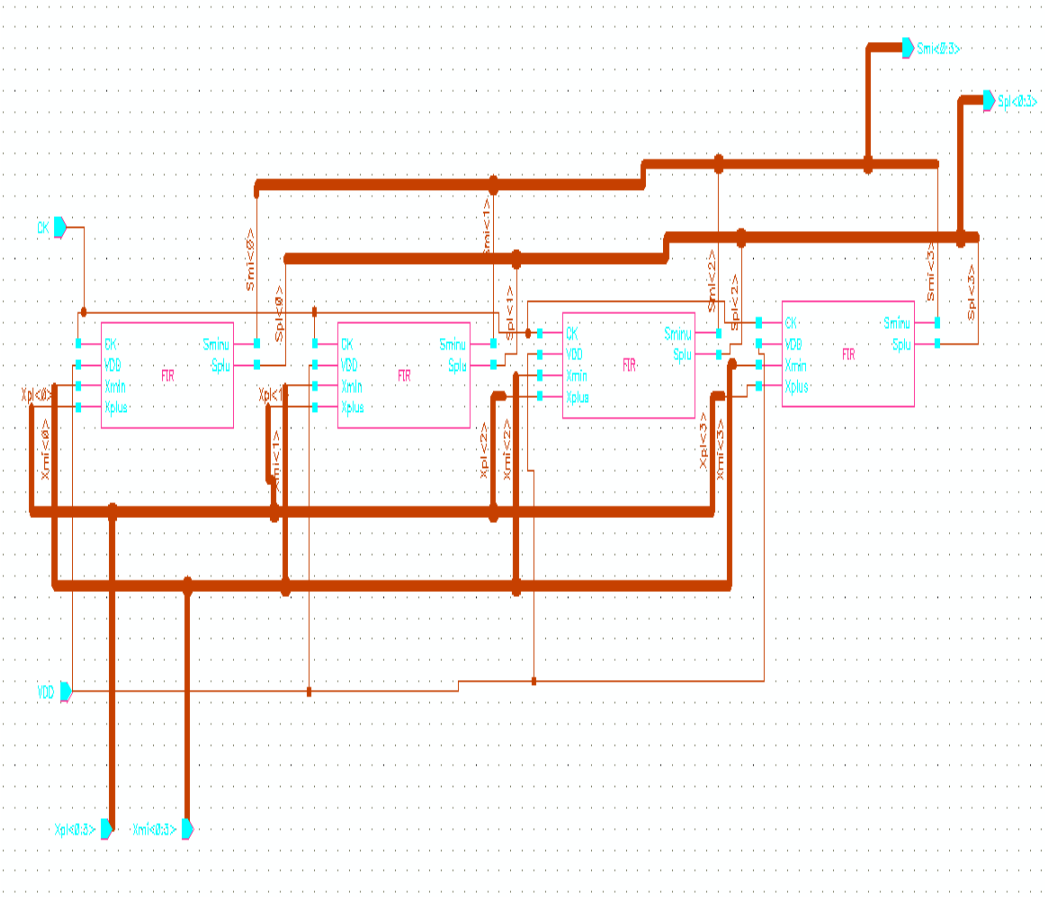


Fig 4.9.2 4-tap Box-car FIR filter(4-bit input).

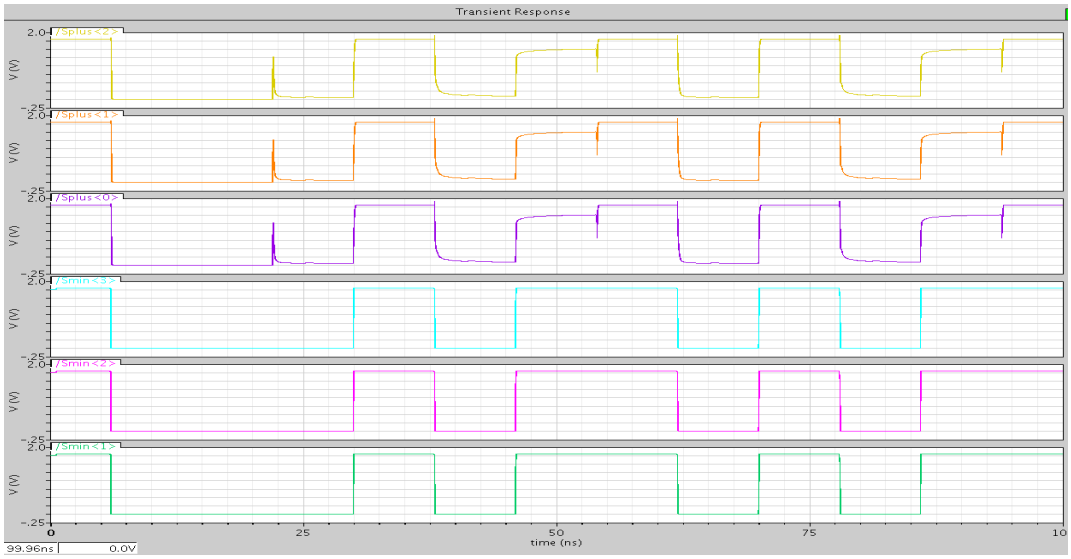


Fig 4.9.3 Simulation waveforms of Box-car FIR filter.

4.10 Digit-serial Multiplier

Multiplication is just a series of repeated addition that are shifted. Consider the following signed binary multiplication of a two 4-bit integer value. The multiplicand B ($b_3b_2b_1b_0$) is signed binary and the multiplier A ($a_3a_2a_1a_0$) is normal binary representations. Depending the value b_j, it is recoded (using Table 2.8) and its gate implementation is shown in fig 4.10.1. In Digit-serial multiplier, for every bit, starting with the most significant bit (MSB) and ending with the least significant bit (LSB), the multiplier is multiplied with the multiplicand. Every multiplication bit is just combination of X-OR and AND operation [1,9,13] shown in fig 4.10.2. For an N-bit wide multiplicand and Multiplier (an N x N multiplication), the product will have a 2N-bit wide product. The result of our desired 4 X 4 multiplication has a 8-bit product. But here it is 9-bit, because of PPM and MMP adders are used. At the final stage the 9-bit multiplier output is in normally binary form as shown in fig 4.10.3.

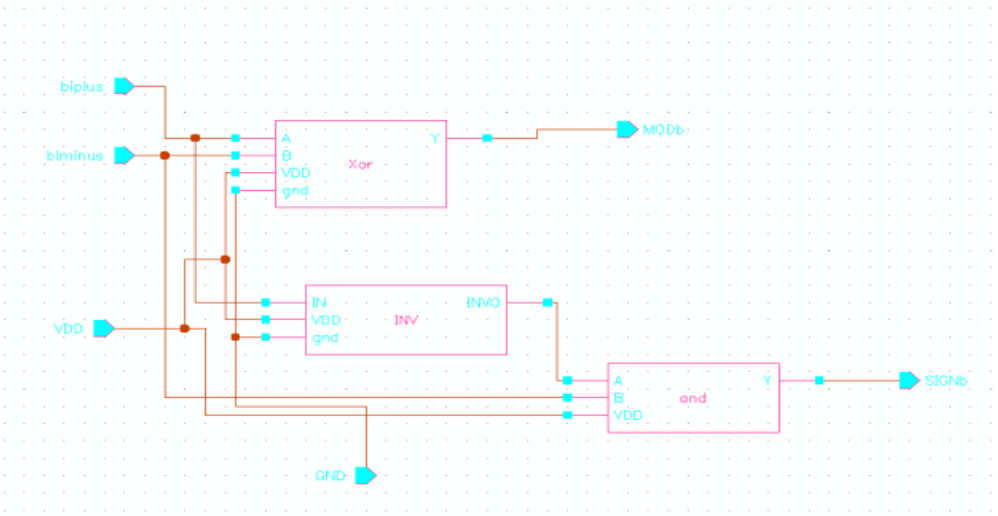


Fig 4.10.1 Recoding of b_j(schematics).

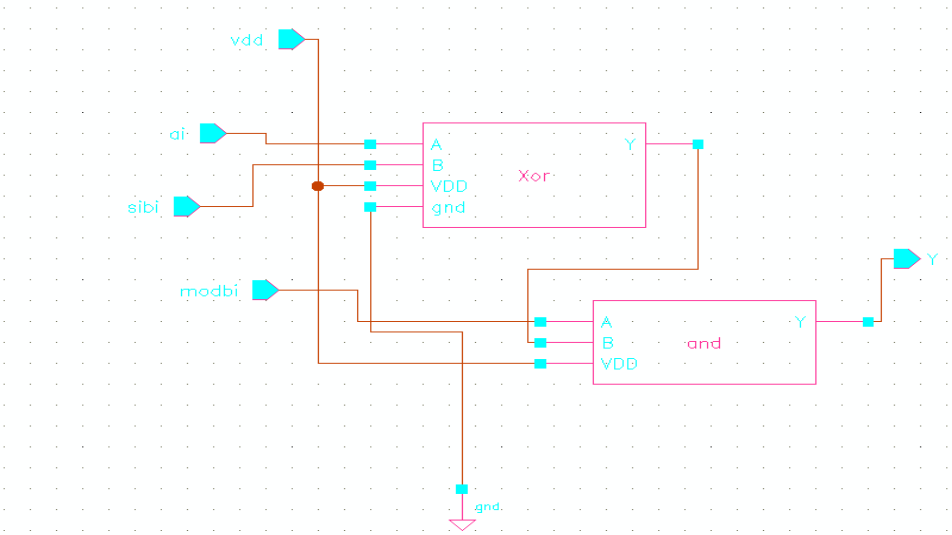


Fig 4.10.2 multiplier (schematics).

The simulation results of the Digit-serial multiplier are shown below:

$t_{PHL} = 11.99n$

$t_{PHL} = 260.5p\sec$

$t_P = 6.124n$

$pwr_{avg} = 1.06mwatt.$

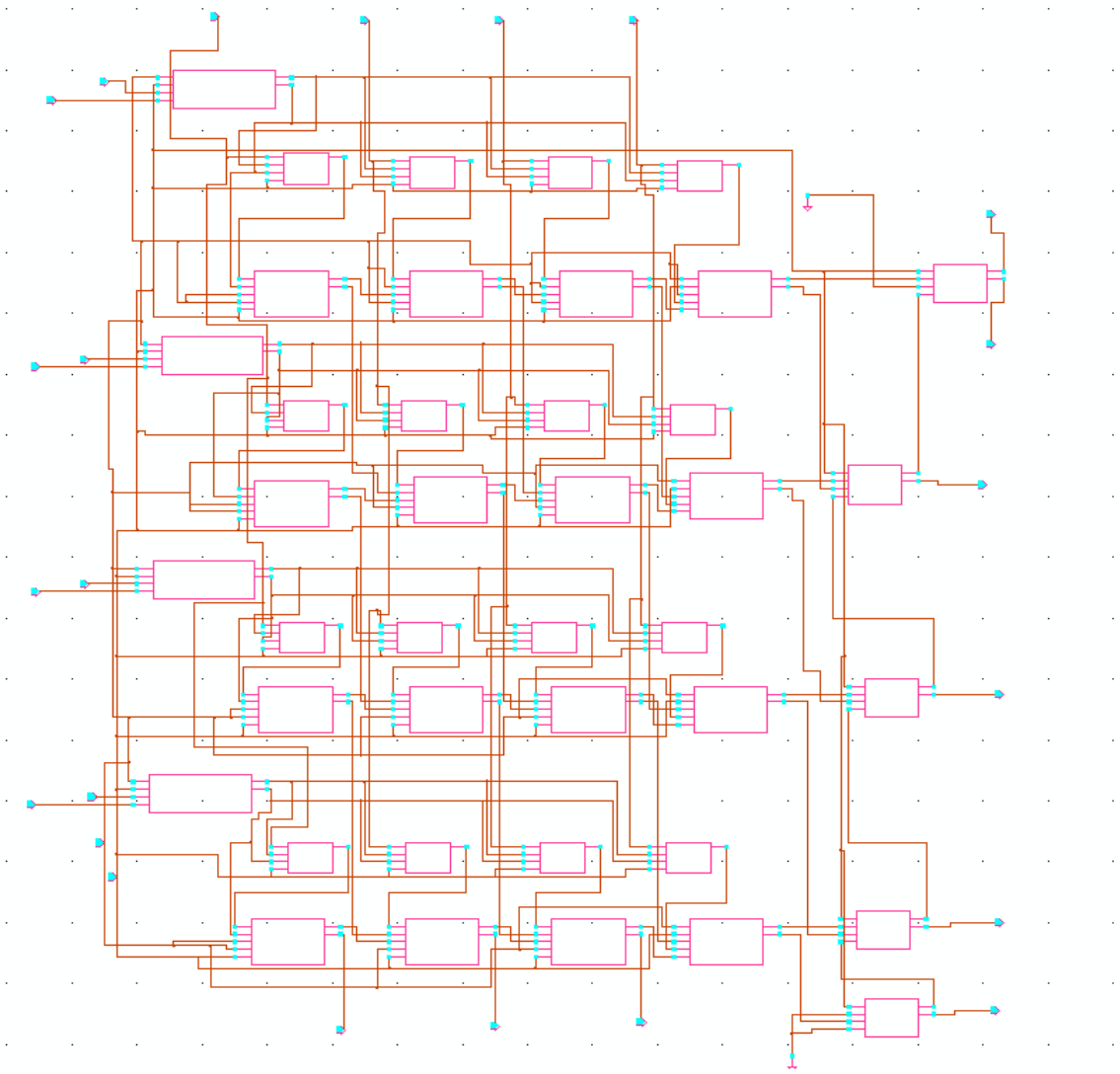


Fig 4.10.3 Digit-serial multiplier.

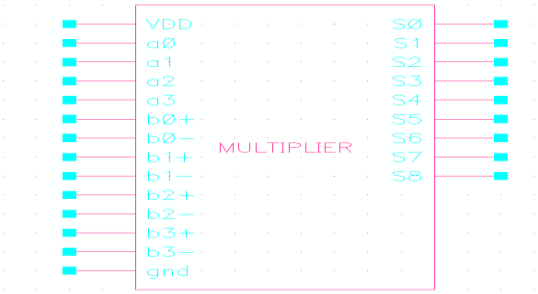


Fig 4.10.4 symbol of multiplier

4.11 3-tap FIR filter

Here we have design the data-broadcast or transposed 3-tap FIR filter. It consists of three multipliers, two adders with D-FFs. *X* (n) is the 4-bit impulse input and 4-bit multiplier coefficient having value less than 1. We choose the coefficient A is 0.125(see APPENDIX II). 9-bit output is produced at the multipliers and then these are connected to 9-FA with 9-D-FFs shown in fig4.11.1 and fig4.11.2. Finally, 10-bit ($S_{9.....}S_0$) is produced at the output of the filter.

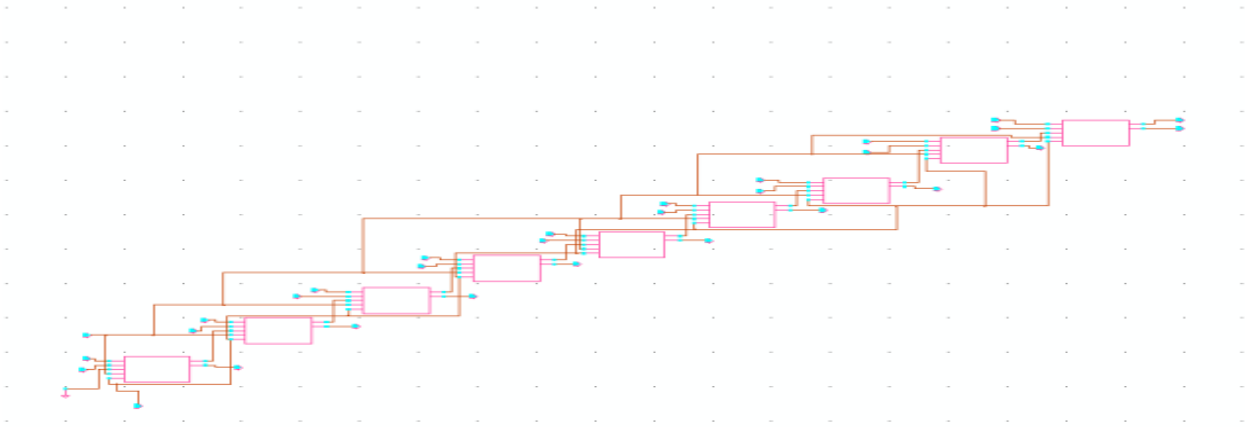


Fig 4.11.1 9-FA schematic.

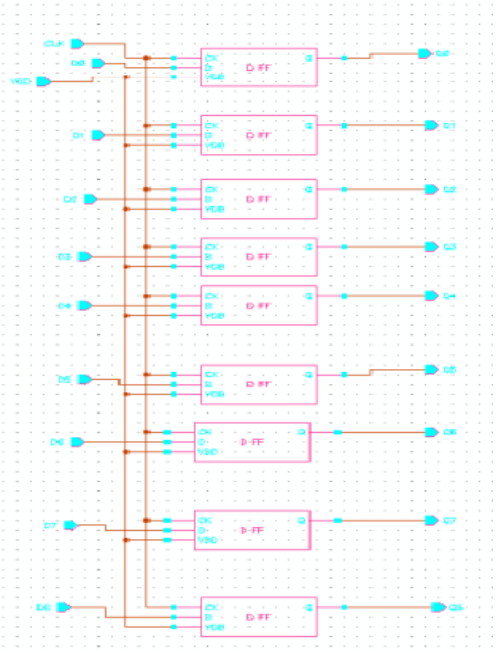


Fig 4.11.2 9-DFF schematic.

The schematic and test-bench of 3-tap FIR filter shown in fig4.11.3 and fig 4.11.4. The simulation waveforms are shown in fig 4.11.5.

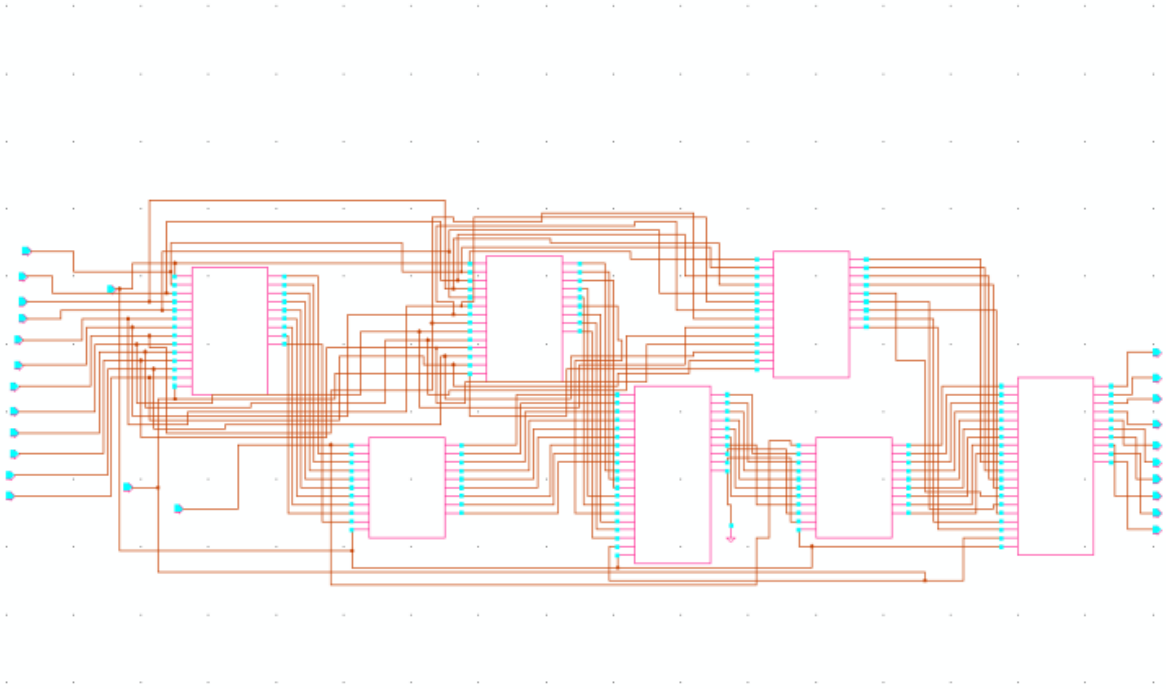


Fig 4.11.3 FIR filter schematic.

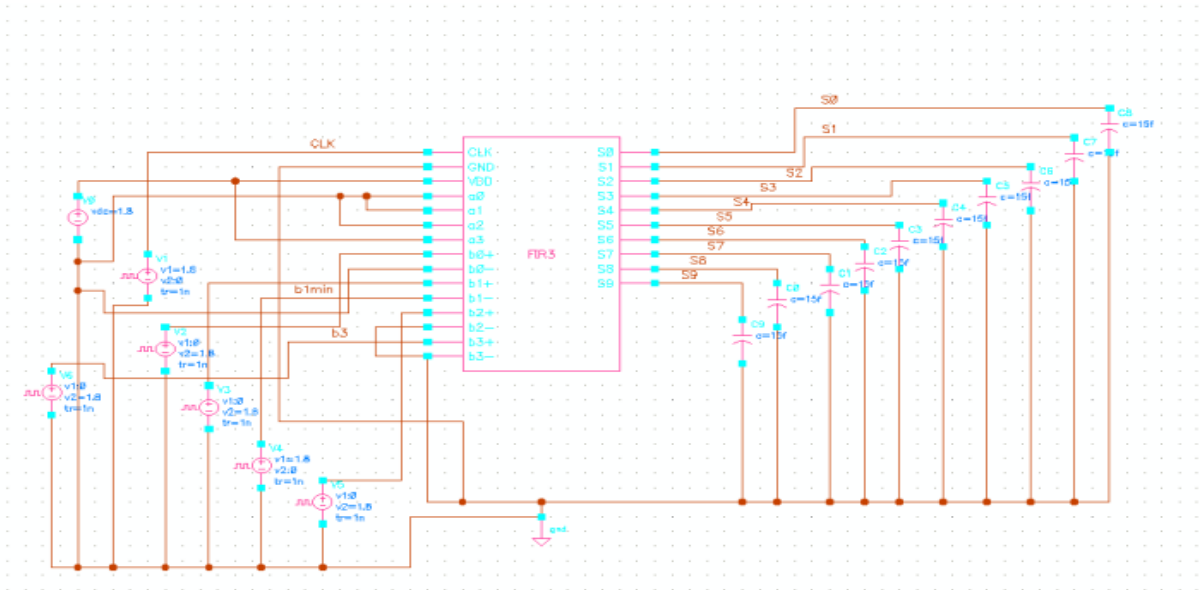


Fig 4.11.4 test-bench of FIR filter.

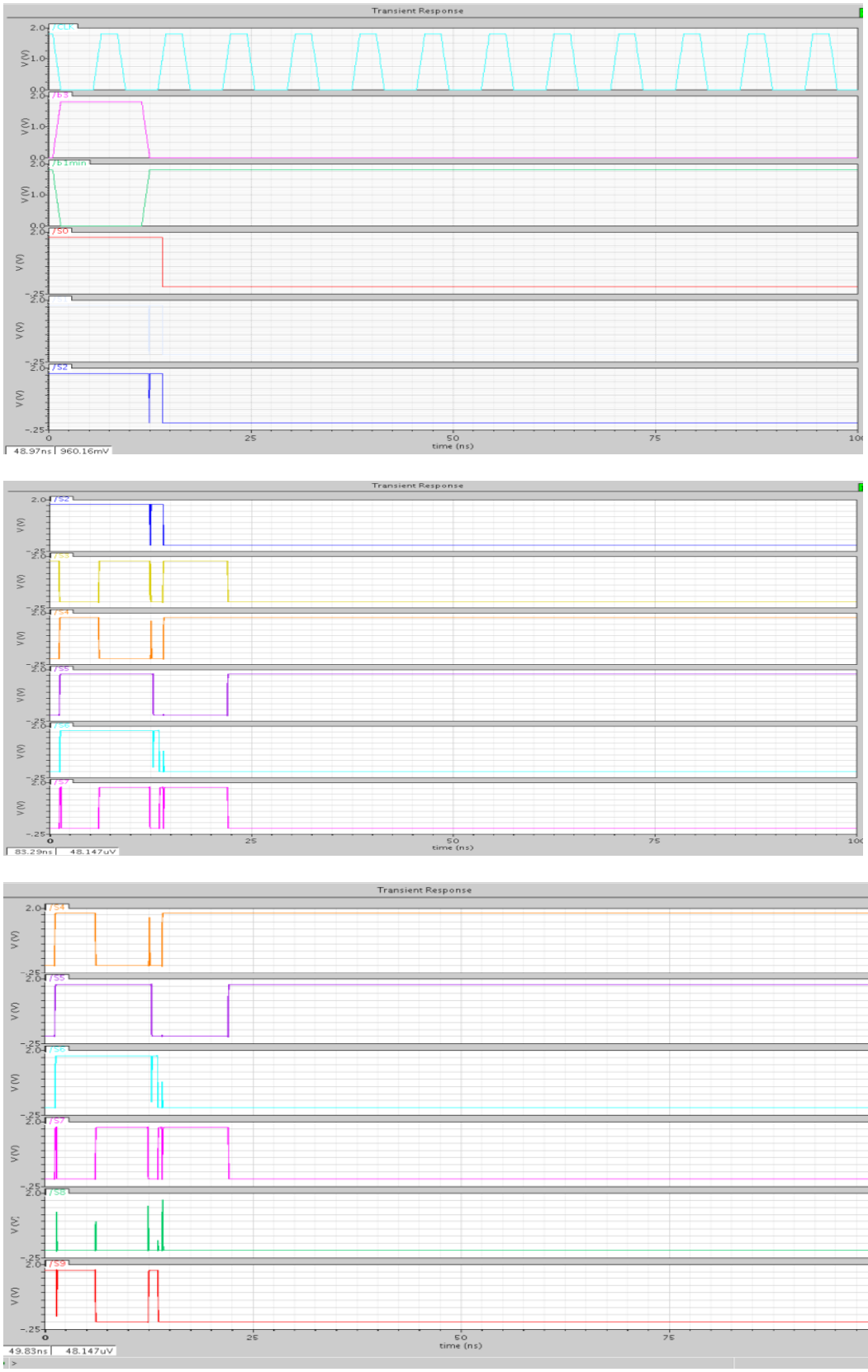


Fig 4.11.5 simulation waveforms of FIR filter.

The simulation results are summarized below:

$t_{PHL} = 5.076n$

$t_{PLH} = 357\,p\,\text{sec}$

$t_P = 2.716n$

$pwr_{avg} = 7.825mwatt.$

As the propagation delay of the designed filter is 2.716nsec, frequency of operation is 368.18MHz.The sampling frequency of the filter is 368.18MH_Z.

Chapter 5

5.1 Conclusion

1) A bottom-up 3-tap FIR filter is designed with sampling frequency 368.18MH_Z i.e. the design can run at 368.18 MH_Z and uses 7.825 mwatt per clock of power.

2) Complexity is more.

3) High performance Boxcar FIR filter was designed.

4) Because FIR Filters are such an important element of DSP design, it is beneficial to do a project like this to strengthen understanding of the concept.

5) This project is a good start for students to learn IC design flow with CDS tool.

5.2 Future Work

It is impossible to design N-tap FIR filter using bottom-up flow. A top-down ASIC design flow is used to design N-tap filter with optimization algorithm technique.

References

[1] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: John Wiley & Sons, Inc., 1999.

[2] I. Koren, Computer Arithmetic Algorithm, Prentice Hall: New York, 1993.

[3] Jhon P.Uyemura, CMOS Logic Circuit Design: Kluwer Academic Publishers, Norwell,MA. 1999.

[4] A. Avizienis, "Signed digit number for fast parallel arithmetic," *IRE Transactions on Electronic Computers,* vol. EC-10, pp. 389-400, Sept. 1961.

[5] Z. Abid, *et. al.*, "PPM adder designs," in *Proceeding of IEEE CCECE,* 2005, vol. 4, pp. 226-229, May 2005.

[6] Z. Abid, *et. al.*, "New designs of redundant binary full adders and its applications," *IEEE Conference*, 2008

[7] Ahmed M. Shams *et. al.*, "Performance Analysis of Low-Power 1-bit CMOS full adder cells," *IEEE Trans. On VLSI systems,* vol.10. No.1 Feb, 2002.

[8] H. T. Bui, Yuke Wang and Y. Jiang, "Design and analysis of lowpower 10-transistor full adders using novel XOR-XNOR gates," *IEEE Transactions CAS –II,* vol. 49, no. 1, pp. 25-30, Jan. 2002.

[9] Y.-N. Chang, J. H. Satyanarayana, and K. K. Parhi, "Systematic design of high-speed and low-power Digit-Serial Multipliers," *IEEE Transactions on circuits and systems* II: analog and Digital signa processing, vol. 45, no. 12, Dec.1998.

[10] Microelectronics Group at Worchester Polytechnic Institute, "Cadence Design Tools Tutorial", October 1999, http://www.vlsi.wpi.edu/cds

[11] David W.Parent, "A Tutorial Guide for Using CDS Tools for IC Design",*IC Design Group San Jose State University,*Tutorial 2.

[12] K. K. Parhi, "Fast low-power VLSI binary addition," in *Proc. Of 1997 IEEE Int. Conf. On Computer Design (ICCD),* (Austin, Texas), pp. 676-684, Oct. 1997.

[13] G. Privat, "A novel class of serial-parallel redudant signed-digit multipliers," in *1990 IEEE International Symposium on Circuits and Systems,* (New Orleans, LA), pp. 2116-2119, May 1990.

[14] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K.Mashiko, "An 8.8

ns 54 × 54-bit multiplier with high speed redundant binary architecture," *IEEE J. Solid-State Circuits*, vol. 31, no. 6, pp.773–783, Jun. 1996.

[15] S. Baik, K. Han, E. Yoon, "A 230MH_Z 8 tap programmable FIR filter using redundant binary number systems," *IEEE International Symposium on Circuits and Systems, ISCAS'99,* vol. 3, pp.415-418, Jul. 1999.

Appendix I

Specification parameters of NMOS and PMOS in gpdk090nm library:

NMOS parameters:

$V_{Tn} = 1.692662e-001(v)$

$t_{ox} = 2.330000e-009(m)$

$CGDO = 2.667600e-010(F/m)$

$Cjsw = 4.790122e-011(F/m)$

$Cj = 7.983537e-004$

$W_n(\min) = 120n$

$L_n(\min) = 100n$

$\mu_n = 2*10^{-2} m^2/v.\sec$

PMOS parameters

$V_{Tp} = -1.359511e-001v$

$t_{ox} = 2.480000e-009m$

$CGDO = 2.506253e-0109(F/m)$

$Cjsw = 4.747351e-011$

$Cj = 7.912252e-004$

W_p=120n,

L_n=100n

μ_p=1.2*$10^{-2}$$m^2$/v.sec.

Appendix II

Coefficient Values in Binary

a_3	a_2	a_1	a_0	value
0	**0**	**0**	**0**	**0.000000**
0	**0**	**0**	**1**	**0.015625**
0	**0**	**1**	**0**	**0.031250**
0	**0**	**1**	**1**	**0.046875**
0	**1**	**0**	**0**	**0.062500**
0	**1**	**0**	**1**	**0.078125**
0	**1**	**1**	**0**	**0.093750**
0	**1**	**1**	**1**	**0.109375**
1	**0**	**0**	**0**	**0.125000**
1	**0**	**0**	**1**	**0.140625**
1	**0**	**1**	**0**	**0.156250**
1	**0**	**1**	**1**	**0.171875**
1	**1**	**0**	**0**	**0.187500**
1	**1**	**0**	**1**	**0.203125**
1	**1**	**1**	**0**	**0.218750**
1	**1**	**1**	**1**	**0.234375**