Realization of Delay Locked Loop using VCDL in 180 nm CMOS Technology

A Project Report is Submitted in Partial Fulfillment of the Requirement for the

MAJOR PROJECT 2

Under the guidance of

Dr. Neeta Pandey,

Associate Professor,

Department of Electronics and Communication Engineering,

Delhi Technological University

Submitted by

Himanshu Daharia

Roll No. : 2K13/VLSI/09

M.Tech. (VLSI Design & Embedded Systems)

(2013-2015)

Department of Electronics and Communication Engineering,
Delhi Technological University
Shahbad Daulatpur,
Main Bawana Road,
Delhi-110042.

CERTIFICATE

This is to certify that the. major project entitled **" Realization of Delay Locked Loop using VCDL in 180 nm CMOS Technology**" is submitted by Mr. **Himanshu Daharia (2K13/VLSI/09) ,** in partial fulfilment of the requirements for the award of degree of **Masters in Technology** in **VLSI Design and Embedded Systems,** Delhi Technological university, Delhi**.**

Date:

Dr. Neeta Pandey
Associate Professor
Delhi Technological University

DECLARATION

This is to certify that the. major project entitled **" Realization of Delay Locked Loop using VCDL in 180 nm CMOS Technology**" is submitted by me in partial fulfillment of the requirements for the award of degree of **Masters in Technology** in **VLSI Design and Embedded Systems,** Delhi Technological university, Delhi comprises of only my own work and due acknowledgement been made to all other materials used.

I, hereby, further declare that in case of any legal dispute in relation to my M.Tech. Major Project, will be solely responsible for the same.

Date:

Signature
Himanshu Daharia
(2K13/VLSI/09)

ACKNOWLEDGEMENT

It gives me a great pleasure to express my profound gratitude to my supervisor and project guide Dr. Neeta Pandey, Associate Professor, Electronics and Communication Engineering Department, Delhi Technological University, for her invaluable guidance, encouragement and patient and thorough reviews throughout the progress of this dissertation. It has been a great experience to do research under her rich experience.

I would also like to extend my heartfelt gratitude to Prof. P. R. Chadha, Head of the department and all faculty members of Electronics and Communication Engineering Department, Delhi Technological University, for keeping the spirits high and clearing the visions to work on the project.

A note of heartiest gratitude goes to my friends and classmates for helping me out whenever I was stuck during my research work.

I am also thankful to my family for their constant support and motivation during this work.

Himanshu Daharia

University Roll no: 2K13/VLSI/09
M.TECH. (VLSI Design and Embedded System)
Department of Electronics & Communication Engineering
Delhi Technological University Delhi – 110042

ABSTRACT

A proper operation of synchronous circuits requires tight control on clock signals which necessitated the use of clock synchronization circuits. Delay Locked Loop (DLL) is one among these circuits which has superior stability. The DLL uses a phase frequency detector to detect phase error between input reference clock and the output clock, a combination of charge pump and loop filter to generate control voltage and a Voltage Controlled Delay Line (VCDL) to delay the reference clock so that there is no skew between reference clock and output clock. The PFD block has been realized using NAND gates and inverters and C^2MOS based register. Two modifications are also suggested in C^2MOS based realization. This work aims at implementing a DLL and studies its performance.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviations	Full - Form	Abbreviations	Full - Form
CMOS	Complementary Metal Oxide Semiconductor	LF	Loop Filter
C^2MOS	Clocked - CMOS	PD	Phase Detector
CML	Current Mode Logic	PFD	Phase Frequency Detector
CP	Charge Pump	PLL	Phase Locked Loop
CSI	Current Starving Inverter	RC	Resistor - Capacitor
DFF	D Flip - Flop	VCDL	Voltage Controlled Delay Line
DLL	Delay Locked Loop	VCO	Voltage Controlled Oscillator

CHAPTER 1

1.1 INTRODUCTION

There are a number of digital ICs which require the on – chip periodic signal generation. Specially, the synchronous circuits need a global clock for synchronizing various events. Modern day microprocessors and other high- performance digital circuits work on very high frequencies that are in the range of gigahertz. Generally, crystal oscillators are accurate and provide less jitter, they can be used to generate reference clock signals for high performance digital circuits over a frequency range of 10's of MHz to 200's of MHz only. To generate higher frequency clocks from the low frequency crystal oscillator, a phase locked loop (PLL) is used. PLL takes the low frequency clock signals and multiplies it with a rational number N; along with generating high frequency clock signal, PLL also performs a function of synchronizing clocks among different chips. This is essential in communications between the chips. Application of PLL is shown in fig. 1.1.

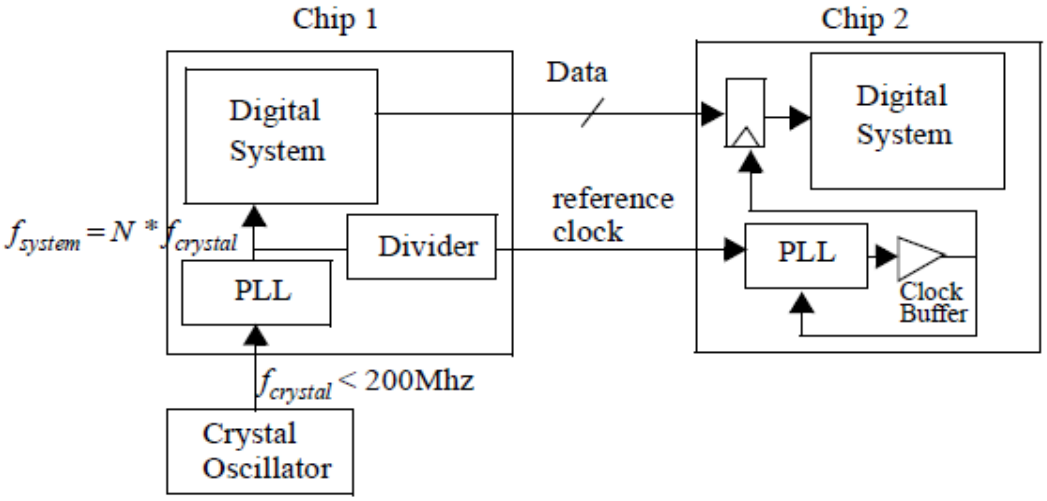


Fig. 1.1 Application of phase locked loops (PLL) [6]

As shown in above fig. 1.1, a reference clock is being sent from chip 1 to chip 2 along with the parallel data during communication. The parallel data transfer from chip 1 to chip 2 requires lower clock reference as compared to system clock; so a reference clock signal is generated by dividing the system clock by a rational number N yet keeping it in phase with the system clock, and sent with the data to synchronize the input flip-flops present in chip 2. In case of wide data buses these flip-flops can give rise to clock loading. In such cases clock buffers are to be used for

avoiding the loading. But the problem with the clock buffers is that they introduce skew between the data and reference clock. A PLL is then used to synchronize the data and output of clock buffers. If any part in the chip 2 requires frequency higher than incoming reference clock, the PLL can provide it as well as by multiplying the incoming reference clock with a rational factor.

A PLL is complex, nonlinear feedback circuit. The basic working mechanism can be better understood by the following fig. 1.2 :

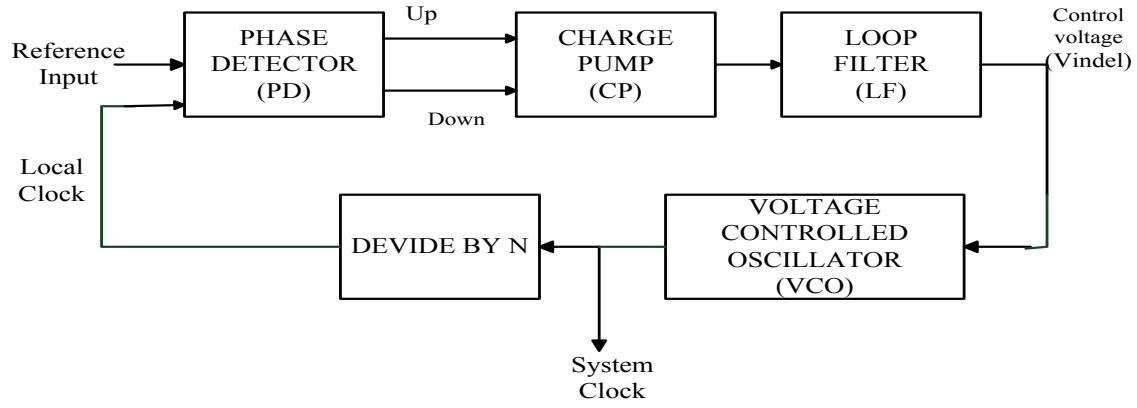


Fig. 1.2: Functional composition of a phase locked loop (PLL) [6]

As shown in the above fig. 1.2, a PLL mainly comprises of four blocks these are:

1. Phase Detector (PD)
2. Charge Pump (CP)
3. Loop Filter (LF) and;
4. Voltage Controlled Oscillator (VCO).

Note that an additional block "Divide by N" is placed also in feedback path, for multiplication of the reference signal. When a control voltage is applied to the VCO, it generates clock signal of desired frequency. The relation between control voltage (V_{indel}) and output frequency is non-linear; hence to synthesize a particular frequency for system clock it is essential to set the control voltage at an appropriate value. This function is done by the rest of the blocks and the feedback loop in the PLL.

The reference clock is taken from outside the chip and generated by a crystal oscillator. Then it is compared with divided version of system clock (local clock), using a phase detector. The Phase detector generates its outputs depending upon the phase difference between the signals. It produces a high signal at '*Up*' when reference clock leads the local clock and a high

signal at '*Down*' when reference clock lags the local clock. These outputs are given to the charge pump which converts digitally encoded control information into an analog voltage. The value of this analog signal increases when *Up* signal goes high and it speeds up the VCO. On the other hand when '*Down*' is high, the value of analog voltage decreases which slows down the VCO. Providing this analog control voltage directly from charge pump to VCO can introduce jitters at the output of VCO. To avoid this situation a low pass filter is used which removes the high frequency components present in the control voltage and smoothens out the VCO output [6].

Two types of jitter occur in PLL. These are the input exhibited and the VCO generated jitter. PLL transfer function shows low- pass characteristic for the input exhibit jitter, it means fast variations in phase of the input is not tracked at output but slow variations in phase of the input is tracked at the output. In other words, fast jitter gets attenuated but slow jitter is propagated at output. At the same time for VCO generated jitter PLL transfer function shows high- pass characteristic which means slow jitter will be attenuated but fast one will be propagated [7].

There are some techniques which can be used for reduction in jitter of PLL such as reducing the gain of the VCO and PD or reduction in the bandwidth of the loop filter, but it results in decrement of the overall gain of the forward loop. The problem associated with using small loop gain is reduction in lock-range and pull-in range along with increased pull-in and lock-range time. The alternative way to solve above problem is to use a Voltage Controlled Delay Line (VCDL) in place of VCO shown in fig. 1.3.

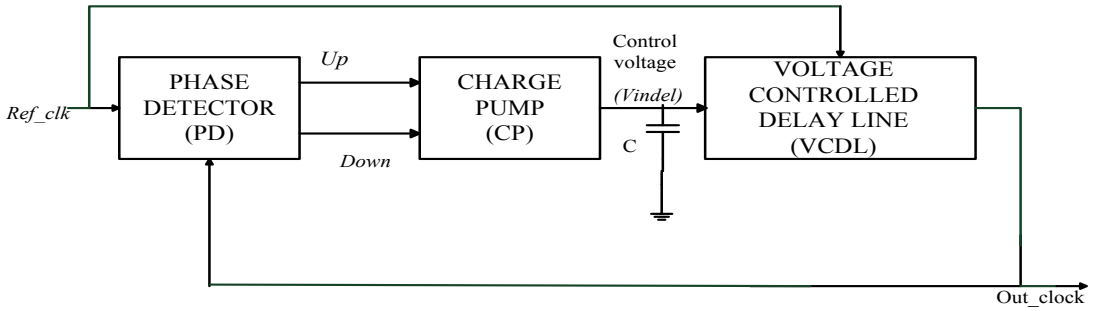


Fig. 1.3 Functional composition of a delay locked loop (DLL) [6]

The jitter in PLL output from the VCO output frequency, changing with a constant input voltage has led to the concept of DLL. At present, Delay Locked Loop (DLL), a variation of PLL structure is widely used in high- performance clocking. The working mechanism of DLL is similar to PLL as both employs feedback mechanism in their operation, but having some

differences as DLL consists of VCDL in place of VCO along with Loop Filter having a capacitor only. The main idea behind this VCDL is to delay the output clock such that it lines up perfectly with the reference. Since no signal generation takes place in VCDL as in VCO, it does not possess the problem of jitter as is the case with PLL [8].

1.2 DLL versus PLL

Some of the basic differences between DLL and PLL are as given below:

- The oscillator employed in PLL introduces phase errors and instability. This lowers the performance of the PLL when it is used for reduction in skew in clock distribution network. On the other hand, the unconditionally stable DLL architecture does not introduces phase errors.
- The DLL's closed loop transfer function is a single pole function. Thus, it is a naturally stable system. On the other hand PLL's closed loop transfer function may have two or three poles and sometimes might pose stability problems.
- At a given operating clock frequency a DLL can delay its input clock by an amount bounded by a minimum and maximum delay. However there is no hard boundary over the capture range of a PLL [9].

1.3 THESIS ORGANIZATION

The primary goal of this thesis is to demonstrate a circuit level approach for a Delay Locked Loop which is used for clock synchronization. This thesis is organized into seven chapters as follows:

CHAPTER 2: DELAY LOCKED LOOPS

This chapter explains the operating principle of DLL. Here various architectures of DLL and its specific applications along with the dynamic loop behavior of DLL is explained. Apart from this some relevant literature search is presented.

CHAPTER 3: PHASE DETECTOR AND PHASE FREQUNCY DETECTOR

One of the most important components of a DLL is Phase Detector (PD). This chapter introduces the basic concept of working mechanisms of PD and PFD.

CHAPTER 4: CHARGE PUMP AND LOOP FILTER

The PD generates Phase error and this error information has to be converted into a control voltage to control the delay of VCDL. This is done by using CP and LF. Working mechanism of CP and LF has been explained in this chapter

CHAPTER 5: VOLTAGE CONTROLLED DELAY LINE

The heart of the DLL is Voltage Controlled Delay Line (VCDL). This Chapter explains the realization of delay elements and then an eight stage VCDL has been realized.

CHAPTER 6: RESULTS AND DISCUSSIONS

This Chapter discusses all about performance summary of the DLL. All the realizations have performed in 180 nanometer CMOS process Technology and simulations performed on OrCAD PSPICE Capture.

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

This chapter summarizes the major accomplishments of this thesis and presents the scope for the future scope.

CHAPTER 2

DELAY LOCKED LOOP

In this chapter working principle and systemic study of closed loop components of DLL is described. Thereafter, different DLL architectures and literature available on this is briefly reviewed.

2.1 INTRODUCTION

The DLL is a digital circuit used for clock synchronization in digital high-performance circuits. Similar to a PLL, DLL also employs feedback mechanism in its basic operation with only one difference i.e. DLL uses VCDL in place of VCO as in PLL. A DLL is desired to produce minimum possible phase difference phase difference between the reference clock and the output clock from last stage of the VCDL. Each VCDL delay stage is able to provide a clock signal which is phase shifted from the given reference input clock when the phase is matched between reference clock and output clock.

Nevertheless, today's rapidly improving technology offers difficulty in phase matching which works on very high frequencies. To accommodate this we need DLLs which locks the output clock to reference clock within less lock-in time, and have wide lock – in range. The increasing level of integration of integrated circuits introduces the problem of clock skew on chip. The clock skew is a time difference between the rising edges of global and local clocks on the chip. It is caused due to mismatch in clock path and variations in clock loading. This phenomenon strongly affects the function and performance of the sequential circuits. A DLL is used for clock distribution in order to remove the clock skew on chips [6].

The application area of DLL includes memory, clock synthesis, clock distribution, clock and data recovery and microprocessors in order to reduce on chip clock buffer delay.

2.2 WORKING PRINCIPLE OF DLL

A DLL is a nonlinear negative feedback system. Figure 2.1 below shows the block diagram of DLL. As shown in fig. 2.1 it consists of mainly four basic components:

(i) Phase Detector (PD)

(ii) Charge Pump (CP)

(iii) Loop Filter (LF)

(iv) Voltage Controlled Delay Line (VCDL)

It works on an assumption that a reference clock of desired frequency is available and it is delayed until the reference clock and its delayed version is synchronized.

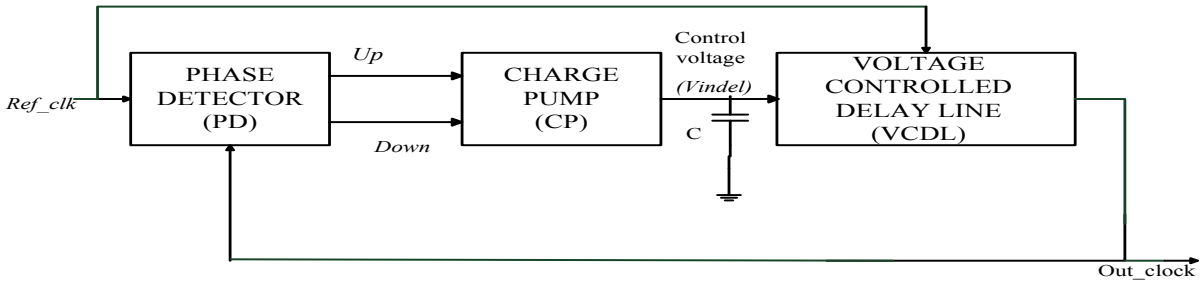
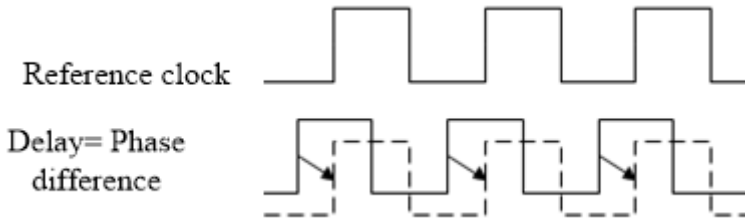


Fig. 2.1: Block diagram of DLL [6]

The overall mechanism of a DLL as follows:

PD takes reference clock and the output of VCDL as its inputs and provides information to CP either to charge or discharge the capacitor of loop filter is based on the relative phase between the two signals. The output of PD is then used to drive the charge pump whose function is to charge or discharge the capacitor of LF. If reference clock leads the VCDL's output which has been termed as *Out_Clk*, *Up* signal goes high and CP charges the capacitor of LF and the value of V_{indel} increases. On the other hand if VCDL's output leads the reference clock, *Down* signal goes high and CP discharges the capacitor of LF and the value of V_{indel} decreases. The VCDL delays the input data in accordance with control voltage (V_{indel}) and we get the required signal locking property. Owing to such a negative feedback mechanism, the phase error gradually decreases until it finally becomes zero.

A DLL finds its main application in removing clock skew in high- performance circuits. The DLL tries to remove the clock skew in between two clock signals by adding additional time delay as shown in fig. 2.2.



Output clock synchronized by adding delay (phase correction)

Fig. 2.2 Basic DLL timing

For a DLL, Locking time, static phase error and lock range are the most important performance parameters. Locking time refers to the time duration a DLL requires to attain a stable locking state from an initial state. Generally, the amount of charging and discharging current flowing in Charge Pump, the overall delay loop bandwidth and the speed of the PD determines the lock time. The maximum and minimum delay of VCDL, is referred as Lock range within which the delay of the VCDL can be varied. A DLL is able to attain lock only in this range. The operating frequency range of a DLL is directly affected by Lock range. Static Phase error refers to the phase difference between the output signal of the last of the VCDL and the input reference signal. After a DLL is locked, the phase of these two signals should be perfectly matched in the ideal case.

2.2 SYSTEMIC STUDY OF CLOSED LOOP COMPONENTS OF DLL

To analyze the dynamic behavior of DLL, its model shown in fig. 2.1 can be realized in terms of ϕ_{in} and ϕ_{out} ,where ϕ_{in} and ϕ_{out} represent the phase of input and output clock respectively as shown below in fig. 2.3.

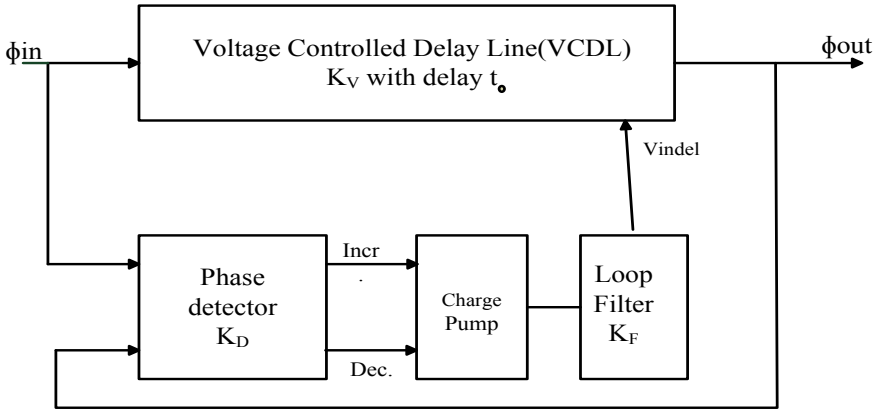


Fig. 2.3: DLL control model [9]

This model can be simplified by realizing the relationship between ϕ_{in} and ϕ_{out} .The phase (in radian) of the input clock is related to the phase of the output clock by following relation

$$\phi_{out} = \phi_{in} + t_0.\ (2\pi/T_{clock}) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.1)$$

Where T_{clock} is the period of the reference clock and t_0 represents the time delay generated by the VCDL and is related to output voltage V_{indel} of loop filter by

$$t_0 = K_V \,.\, V_{indel} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.2)$$

here K_V denotes the gain of the VCDL. Its unit is seconds/V.
The expression 2.1 can be written as

$$\phi_{out} = \phi_{in} + t_0.2\pi f \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots...(2.3)$$

where $f = 1/\ T_{clock}$.

Since one of the performance metrics of a DLL is jitter, so for a good jitter performance a DLL should meet the following condition:
"Bandwidth of DLL (the rate at which the DLL will respond to a change in the input phase) should be less below the bandwidth (the rate at which a signal will propagate around the entire loop) by the factor of 10".

If above condition is met then the DLL model of fig. 2.3 can be further simplified as shown below in fig. 2.4.

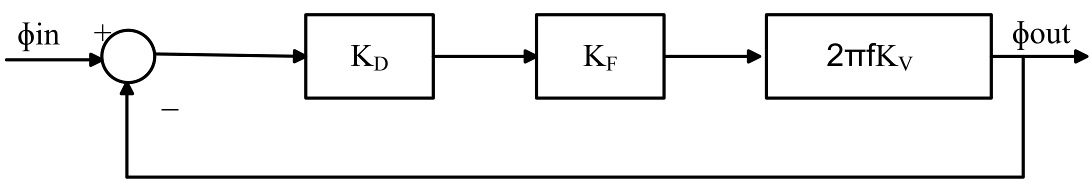


Fig. 2.4 Simplified control model of DLL [9]

the overall transfer function is computed as

$$\phi_{out} = (\phi_{in} - \phi_{out})\ K_D\,.K_F\,.K_V\,.2\pi f\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.4)$$

$$\frac{\phi out}{\phi in} = \frac{K_D K_F K_V\,.2\pi f}{1+K_D K_F K_V\,.2\pi f}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.5)$$

$$\frac{\phi out}{\phi in} = \frac{1}{1+\frac{1}{K_D K_F K_V .2\pi f}} \quad \text{.....(2.6)}$$

Since the term K_F is gain of combination of charge pump and loop filter which can be expressed as

$$K_F = \frac{K_L}{s} \quad \text{.....(2.7)}$$

Putting the value of K_F in expression 2.6 we will obtain

$$\frac{\phi out}{\phi in} = \frac{1}{1+\frac{s}{\omega_{Loop}}} \quad \text{.....(2.8)}$$

where $\omega_{Loop} = K_D .K_L .K_V .2\pi f$ .....(2.9)

From equation 2.8 we can observe that the transfer function of DLL has only one pole in s-domain, which suggests high stability of the system [9].

Generally the transfer function of PLL involves two poles that lead to less stability as compared to DLL which is a single pole system.

2.3 DLL ARCHITECTURES

The DLLs can be categorized on the basis of their phase shift generation and these are as follows:

2.3.1 Analog DLLs

Figure 2.5 below is depicting the block diagram of Analog type DLL which comprises of Phase Detector (PD), Charge Pump (CP), first order Loop Filter (LF) and Voltage Controlled Delay Line (VCDL). The VCDL is a cascade connection of several variable delay elements. In order to maintain phase alignment in input and output clock signals, PD detects phase error by comparing their rising edges and the combination of CP and LF generates control voltage proportional to this phase error. The VCDL takes two inputs, one is reference input clock and the second is control voltage. This control voltage is utilized to produce delay in reference clock through VCDL. The DLL forces the phase of its input and output signal to be aligned. In locked condition the delay between input and output should be equal to one clock period of input signal.

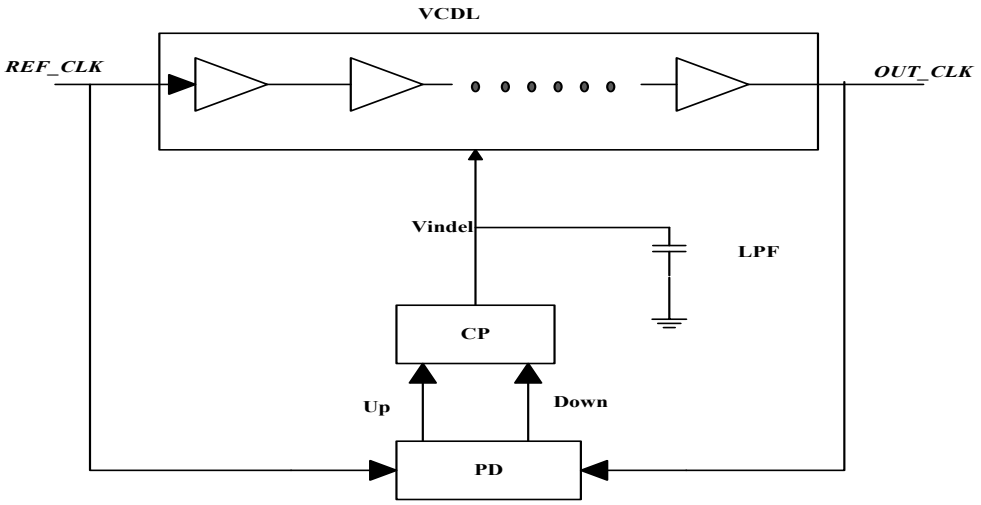


Fig. 2.5 Analog type DLL [12]

Analog type DLLs are most suitable where fine - grain delay variations are required. This type of architecture accumulates less jitter over digital DLLs because of its fine delay adjustments and this is important advantage of using analog DLLs but small delay steps limits locking range of the DLL.

2.3.2 Digital DLLs

As in case of Analog type DLL, the delay was controlled by control voltage generated by the combination of CP and LF which was analog in nature. In Digital type DLLs delay is provided in fixed and quantized steps. The block diagram of Digital DLLs has been depicted in fig. 2.6 which consist of Phase Detector (PD), Phase selector (PS), Finite State Machine (FSM) and Digitally Controlled Delay Line (DCDL). The DCDL is composed of variable delay elements chain having different lengths. The amount of delay is determined by the number of delay elements in chain. The PS is realized as multiplexer. At the output of PS, pulse of defined phase – shift is selected. The amount of delay to be given is decided by the FSM. In this way the generation of phase shift takes place in digitally controlled manner.

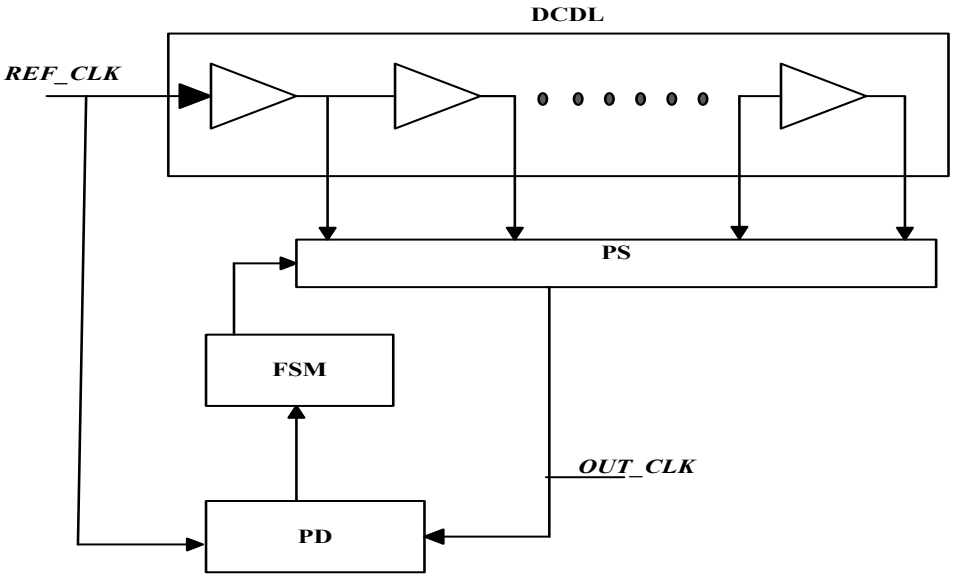


Fig. 2.6 Digital DLL [12]

Time delay generated by delay elements are fixed and quantized, hence these are used where delay in steps of course – grain is required to lock a wide range of frequencies. Since the fixed and quantized time delay is used here, it leads to large jitter problem.

2.3.3 Dual- Loop Delay Locked Loops

The architecture of Dual – Loop type DLLs consist of cascaded connection of Digital DLL and Analog DLL which has been depicted in fig. 2.7 below:

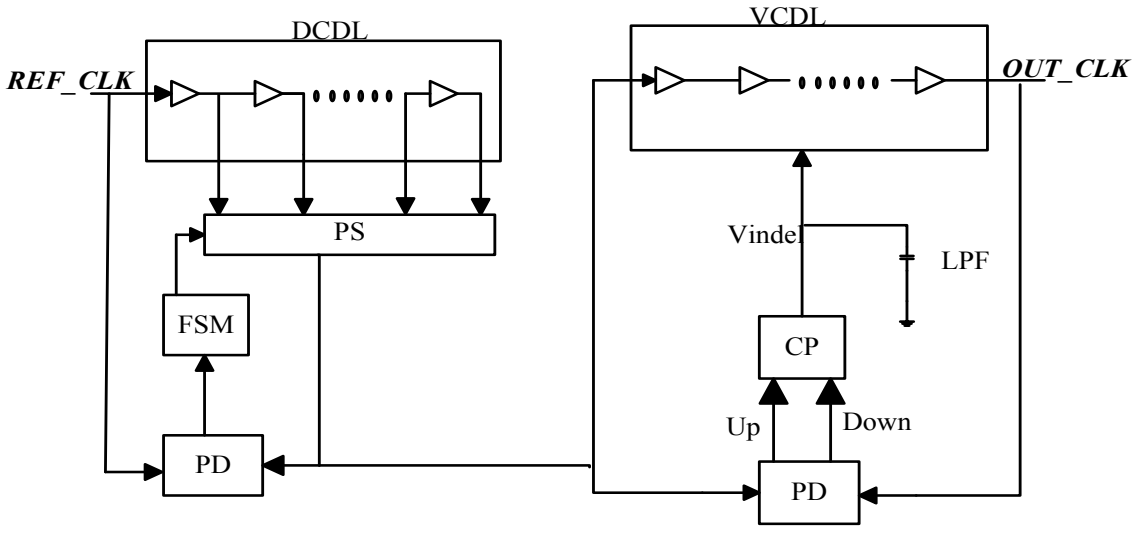


Fig. 2.7 Dual-loop type DLL [12]

On the basis of its architecture, these types of DLLs are also termed as hybrid DLLs. This type of DLLs can be used over wide range of frequencies. Jitter performance is not good as clock propagation involves two path cycles consisting of large number of delay elements. Implementation of this type of architecture is very complex and also requires more power dissipation [12].

2.4 LITERATURE REVIEW

1. Analog DLLs

Most of the recent works have already been done in Analog type DLLs. A selection of the relevant literature has been presented below, where main focus was to expand operating range of the conventional DLL, DLL as frequency multiplier and to linearize the delay of VCDL etc. have been proposed.

Yongsam Moon, Jongsang Choi, Kyeongho Lee, Deog-Kyoon Jeong and Min-Kyu Kim [1] in their work proposed a method to expand the operating frequency range of the conventional DLL by attaching a replica delay line in parallel to it. The advantage of this architecture is that it keeps the benefits of conventional DLL such as good jitter performance and multiphase clock generation with small increments in chip area and power consumption. To provide duty cycle correction capability of dell cells which can be controlled by multiple voltages have been used here. The peak to peak jitter of this architecture is less than 30 ps in frequency range between 62.5 and 250 MHz with supply jitter sensitivity of 0.11ps/mV at 250 MHz. Implementation has been done in 0.35µm CMOS technology having chip area of 0.2 mm^2 with power dissipation of 42mW.

Hsiang-Hui Chang, Jyh-Woei Lin, Ching-Yuan Yang and Shen-Iuan Liu [2] have proposed an approach to dynamically change the number of delay cells to widen the operating frequency range of the DLL. This architecture uses a phase selection circuit to decide what number of delay cells should be employed. This phase selection circuit also controls the number of capacitors, in this way it can shift the bandwidth of the overall DLL. A new start controlled circuit has also been proposed to avoid false locking. Operating frequency range of this architecture is between 6MHz and 130 MHz with maximum r.m.s. jitter 25ps. Implementation has been done in 0.35µm CMOS technology having chip area of 880µm×515µm and maximum power dissipation is 132mW.

Hsun-Hsiang Chen, Zih-Hsiang Wong and Shen-Li Chen[3] have utilized the phase selection Circuit [7] but they used Frequency to Voltage Converter for the selection of capacitor. The operating frequency range of this architecture extends from 106 MHz ~ 151 MHz to 54 MHz ~ 250 MHz, and the power dissipation increases from 2.47 mW ~ 3.33 mW to 6.7 mW ~ 14 mW.

Kuo-Hsing Cheng, Yu-Lung Lo, Wen-Fang Yu and Shu-Yin Hung [4] have also proposed phase selection circuit based on Time to Digital conversion which reduces the locking time of this DLL architecture to only one clock cycle of reference signal's time period. For expansion of operating frequency range, multi-controlled VCDL has been employed in this architecture. This architecture does not possess the problem of false locking and accumulates less jitter. Simulations were based upon TSMC 0.25 μm 1P5M N-well CMOS process. The locking range was 50MHz - 280MHz.

Chulwoo Kim, In-Chul Hwang, and Sung-Mo (Steve) Kang [5] have proposed DLL based clock generator. In this work they have proposed a phase detector with a reset circuitry and a new frequency multiplier to overcome the limited locking range and frequency multiplication problems of the conventional DLL – based systems. The operating frequency range of this architecture is from 120MHz to 1.1GHz with cycle to cycle jitter ±7.28ps at 1GHz. Implementation has been done in 0.35μm CMOS technology having chip area of $0.7mm^2$ and the power dissipation is 42.9mW at 1GHz.

Ramin Farjad-Rad, William Dally, Hiok-Tiaq Ng, Ramesh Senthinathan, M.-J. Edward Lee, Rohit Rathi, and John Poulton [11] have proposed multiplying DLL by inserting a multiplexer and selected logic in conventional DLL for high speed on – chip clock generation. This architecture possesses characteristic of PLL for multi-rate frequency multiplication on the other hand it removed all other drawbacks of PLL – frequency multiplier such as jitter accumulation. Implementation has been done in 0.18μm CMOS technology having chip area of $0.05\ mm^2$. The operating frequency range is 200MHz – 2GHz with multiplying ratios of 4, 5, 8 and 10. Its Power dissipation at 2GHz is 12mW. Peak to peak and rms jitters are 1.73ps and 13.1 ps respectively.

Goran Jovanovic,Mile Stojcev and Dragisa Krstic [12] have proposed a method to linearize the delay elements of conventional DLL. They modified the conventional Charge Pump and Bias circuit from single ended to differential input output kind of structure. This modified structure made DLL immune to power, process and temperature variations. Simulation has been done in 1.2μm double – poly and double – metal CMOS Technology.

Soh Lip-Kai, Mohd-Shahiman Sulaiman, and Zubaida Yusoff [13] have proposed a fast locking DLL with less jitter. They employed dual Charge Pump as Coarse charge pump and Fine Charge pump. Coarse charge pump made locking fast and Fine charge pump allowed fine adjustments

by reducing jitter. A modified Phase Frequency Detector with reduced reset time is also used in this structure. Implementation is done on the Silterra 0.18 µm 1P6M CMOS process. The active area of the proposed DLL circuit is 327.46µm× 116.16µm.

Pin-Tseng Chen, Chia-Chen Chang, Han-Ying Liu, and Yu-Lung Lo [14] have proposed Dual slope technique to enhance the speed of DLL. As in the previous case, they used Dual charge pumps, here two types Phase Detectors have been employed. Accordingly, a modified charge pump has also been used in this architecture keeping small area cost and better jitter performance. Implementation has been done in 0.35 µm CMOS process having core area of 0.069 mm^2. Operating frequency range is 140 MHz – 270MHz. Peak to peak jitter is 3.2 ps and static phase error is 4.1ps at 230MHz. This architecture takes 88 cycles for locking and dissipates power of 10.8 mW.

B.Venkataramani, Karutharaja.V. and M.Bhaskar [15] have proposed a modified DLL which requires less cycles for locking. They employed charge pump with symmetric load and VCDL made of current starving inverters. Implementation has been done in UMC180 nm CMOS technology. Operating frequency range was 650MHz – 1.2GHz with a locking period of 4 to 8 cycles. Jitter of this structure was 10ps.

Pierre Maillard, W. Timothy Holman, T. Daniel Loveless, Bharat L. Bhuva and Lloyd W. Massengill[16] have proposed a radiation-hardened-by-design (RHBD) VCDL for single-event mitigation in DLLs. Single-event transients (SETs) have been identified as the primary failure mechanism behind several spacecraft malfunctions. A modified differential delay cells have been used to harden the VCDL that results in reduction of missing pulses generated by DLL after ion strike in space. The single event errors can be significantly reduced by increasing the W/L ratio of feedback transistor in delay cells of VCDL while keeping high operating frequency, wide tuning range and small area penalty.

Menka Sukhwani, V. B. Chandratre, Megha Thomas, C. K. Pithawa and Vangmayee Sharda [17] have proposed a 128-bit deep switched capacitor based analog waveform sampling memory ASIC *"Anusmriti"*. The *Anusmriti* ASIC is designed specifically for pulse profile analysis of fast single shot events, occurring typically in High Energy Physics, Astronomy and Laser or accelerator based experiments without using high frequency sampling clocks. It samples and stores the randomly occurring fast exponential signals. The stored memory samples can be read and digitized subsequently at lower rate thereby relieving the need of high speed

digitization. Hence to provide accurate sampling interval this design is incorporated with 500 MHz DLL .Implementation is done in 0.7 μm CMOS process occupying chip area of 5mm × 3.5mm.

Heedon Jang, Hyunggun Ma, Dongwook Seo, Jaeho Lee, Hyungsoo Lee and Franklin Bien[18] have proposed a low power receiver for medical implantable communication system (MICS) using DLL. As compare to its counterpart MICS using PLL, the MICS using DLL offers lesser power dissipation. Two channel DLL is employed in this design to generate 400MHz from 40MHz input signal. Power Dissipation of DLL is 4.8 mW and total occupied area by overall MICS is $3mm^2$.

2. Digital DLLs

Ko-Chi Kuo and Chung-Yuan Chang and Si-Hsien Li [19] have proposed a DLL incorporated with Successive Approximation Register (SAR) in its architecture to make its locking fast. To reduce the power dissipation significantly Loop State Controller (LSC) is proposed which switches the mode of operation from tracking mode to sleeping mode and vice – versa. Phase Error Comparator (PEC) keeps the track for any variation in phase, supply, process, temperature and load. If PEC finds any error it informs the LSC to run in tracking mode and when it finds no error it informs the LSC to run in sleep mode deactivating PD and SAR. Implementation has been done in TSMC 180nm. It requires 6 cycles for locking and operating range is 150MHz – 900 MHz. The power dissipation is 15mW in tracking mode and 9mW in sleeping mode.

Bo Ye, Tianwang Li, Xingcheng Han, and Min Luo [20] have proposed Digital DLL incorporated with Delay Compensation Circuit (DCC). In this architecture, Shift Register (SR) has been employed to control the delay. It shifts the stream of bits either right or left according to input provided by the Phase Detector. This shifted bit stream decides the amount of delay to be generated. The function of DCC is to reset bits of SR when circuit is powered on. When some changes occur in SR due to any variation in supply, temperature and process, DCC takes care of it. The advantage of using DCC is that it has only one stable state. This design takes 16 cycles for locking.

Kai Huang, Zhikuang Cai, and Jun Yang [21] have proposed an all-digital delay-locked loop (ADDLL) with “reset in every step” (RES) delay line. Because of this reset mechanism in delay line this architecture has fast – locking and harmonic – free property. The locking time of

this architecture is N+1 where N is number of control bits for delay line. Simulation has been done on SMIC 180nm CMOS technology. The operating range was from 50 MHz to 250 MHz.

S. Moorthi, D. Meganathan, N. Krishna Prasad and J. Raja paul perinbam [22] have proposed a Modified Variable Successive Approximation Register controlled (MVSAR) algorithm based Digital DLL. This algorithm was used for fast-locking property, closed-loop operation and performing binary search without harmonic-locking issue. Implementation has been done in TSMC 0.18μm, six-metal technology. The operating frequency range of the ADDLL is 14MHz - 170 MHz and occupied an area of 142×142 μm^2.

3. Dual Loop Delay Locked – Loops

Eunseok Song, Seung-Wook Lee, Jeong-Woo Lee, Joonbae Park, and Soo-Ik Chae [23] have proposed a DLL architecture incorporated with Replica Delay line and Cycle Period Detector (CPD) to remove the false lock problem of conventional DLLs. The CPD is placed in auxiliary loop of DLL. Whether the main loop of DLL is in correct locked state or not is decided by the CPD by estimating the cycle period of input clock. The auxiliary loop doesn't affects the main loop as it doesn't require any external signal. Duty Cycle Correction circuit has also been used. This was implemented in 0.25- μm CMOS technology and its operating frequency ranges from 30 to 200 MHz. Its cycle-to-cycle rms jitter is 12.8 ps at 133 MHz, and it dissipates 30 mW at 2.5 V supply.

Jinyeong Moon and Hye-young Lee [24] have proposed a dual-loop DLL for GHz DRAMs. This architecture has involved a conventional analog DLL in its reference loop and digital DLL in its main loop. Its main advantage is that it employs Multi – Digitally Controlled Delay Line (DCDL) which reduces the maximum operating frequency dependence of DLL on its unit time delay provided by Delay Line. The function of this reference loop in this architecture is to provide multi – phase input to the multiplexer present in main loop. For better jitter reduction, an extra invalidation system has also been employed in this architecture. The operating frequency range of this architecture is 500MHz- 3GHz. Its power dissipation is 12.5mW and the total jitter is 64.6ps at 1.6GHz.

Jacek Jasielski, Stanis_aw Kuta, Witold Machowski, Wojciech Ko_odziejski [25] have proposed an Analog Dual locked loop incorporated with cascaded course delay line and fine programmable delay line. The coarse delay line is to push the loop close to lock state, and a fine

programmable delay line to make small adjustments to the output phase**.** Both the delay lines have employed cascaded variable-delay elements based on single-ended Schmitt triggers. For the correction of a duty cycle of the delayed output signal, a duty cycle correction (DCC) has been used. It has been simulated using Spectre and BSIM3V3 device models for 1.8V 180nm CMOS technology from UMC.

S. Senthilnathan, S. Balaji and R. Prithviraj [26] have proposed a dual – loop DLL architecture for high accuracy Positron Emission tomography imaging with low power consumption. This proposed architecture has two control voltages to control delay of VCDL, in this way it has reduced jitter significantly. Two similar loops consisting of PD, CP and LF have been used to provide control voltages. This architecture is free from false – locking and further static D flip-flops have been employed for less power dissipation. It is operated at 1GHz with acquisition time of 19.5 μs. Also its Capture range is 14 – 19.5 μs and Lock – in range is 20 – 100 μs.

CHAPTER 3

PHASE DETECTOR AND PHASE FREQUENCY DETECTOR

In this chapter working mechanism of the PD and PFD is detailed. After that realization of PFD using C^2MOS – register is proposed and their operation is verified with the help of simulations.

3.1 INTRODUCTION

The main purpose of the Phase Detector (PD) or Phase Frequency Detector (PFD) is to detect the phase difference between its input signals. The detected phase error signal is indicated as *Up* and *Down* signals. The input signals to PD or PFD are reference clock and output clock which is taken back from VCDL. The error signals *Up* and *Down* are voltage signals proportional to the phase error. This PD or PFD generates high on *Up* signal when reference signal leads to output signal. On the other hand *Down* signal goes high when reference signal lags behind output signal. When *Up* signal goes high, the value of control voltage increases through charge pump. On the contrary, if *Down* signal goes high it indicates the value of control voltage need to be decreased.

For detecting the phase error following is used:

(1) XOR gate based Phase Detector

(2) Flip- Flop based Phase Detector (also called Phase Frequency Detector)

3.2 XOR based Phase Detector (PD)

A XOR based PD is an example of a basic phase detector. It produces error pulses on both falling and rising edges.

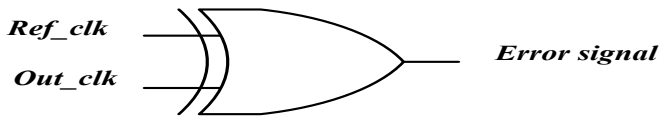


Fig. 3.1: XOR Phase Detector

A detailed analysis of the XOR PD when the reference and output signals are out of phase by 0, $\pi/2$, and π respectively is shown in fig. 3.2.

In fig. 3.2, the phase difference between the two signals is locked phase. The average output voltage is 0 in this case.

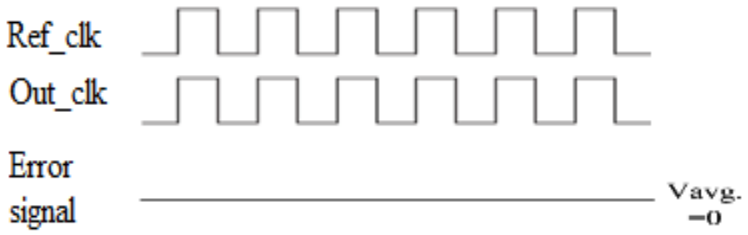


Fig. 3.2 when phase difference between the two signals is zero.

In fig. 3.3, the phase difference between the two signals is π/2. The average output voltage is VDD/2 in this case.

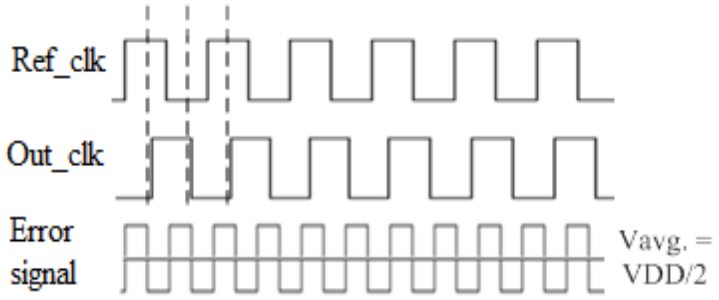


Fig. 3.3 when phase difference between the two signals is π/2.

In fig. 3.4, the phase difference between the two signals is π. The average output voltage is VDD in this case.

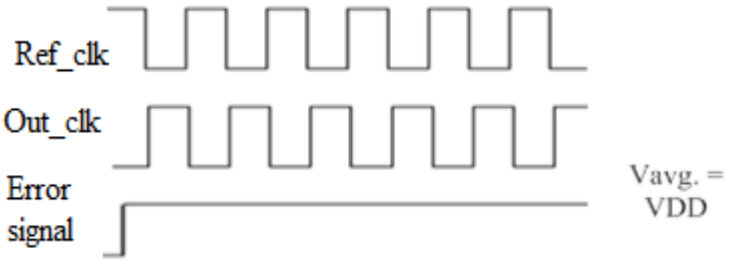


Fig. 3.4 when phase difference between the two signals is π.

Hence we can conclude that the DC value of the XOR based PD is linearly proportional to the difference between its two input signals. The XOR PD characteristic plot in fig. 3.5:

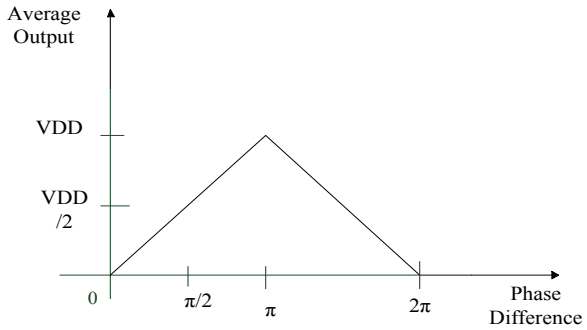


Fig. 3.5: XOR PD characteristic

XOR based PD has several drawbacks as

1. Only one output from the XOR PD makes it difficult to connect with subsequent circuit.
2. It is essentially a signal level detector, i.e. the output of the gate is dependent on duty cycle of its two input signals. Consequently XOR PD may generate incorrect phase error information unless duty cycle correction circuits are used.

The duty cycle problem of XOR phase detector can be understood by following fig. 3.6:

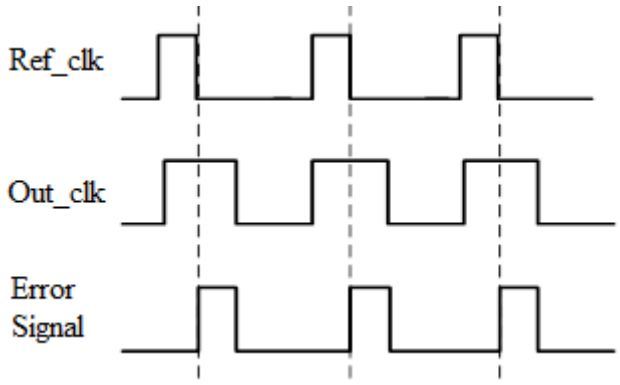


Fig. 3.6 Output of XOR PD when inputs having different duty cycle

As shown in the fig. 3.6, reference clock of 25% duty cycle and output clock with 50% duty cycle is given to XOR PD as inputs. Since both inputs have same frequency the output should be a zero level signal but XOR gate gives a high when both signals are of different levels, hence the output as shown in fig. 3.6 is obtained.

3.3 Flip-Flop based PD or PFD

An improved PD is based on flip-flops which are also called Phase Frequency Detector (PFD). Since flip-flops works on the edges of clocks, the duty cycle dependence problem with XOR gate PD can be avoided. Figure 3.7 shows a flip-flop based PFD.

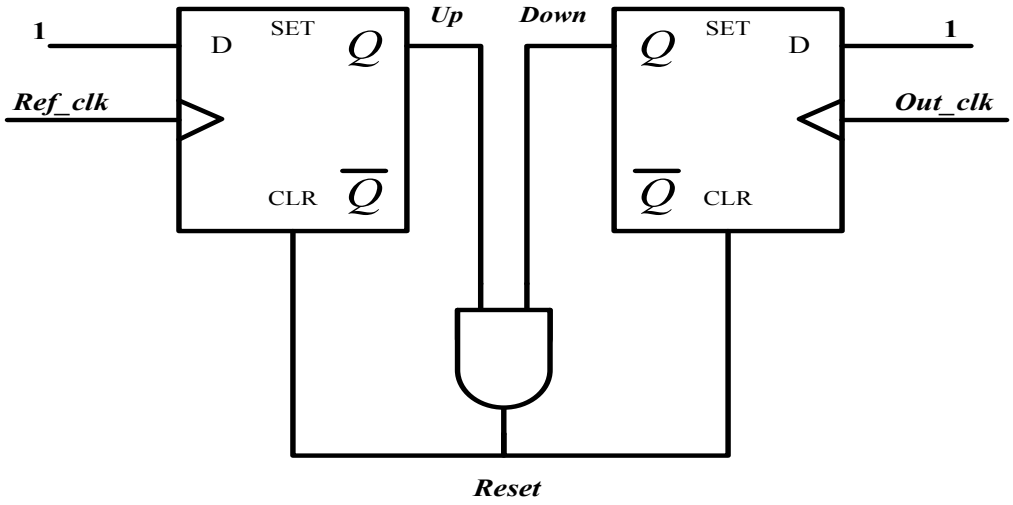


Fig. 3.7 Block Diagram of Phase Frequency Detector [8]

The PFD shown in fig. 3.7 consists of two resettable, edge triggered D-flip flops. The input D is kept at logic 1. The inputs, reference clock (R*ef_clk*) and output clock (*Out_clk*) are being provided as clock signals for these flip flops. The outputs have been denoted as *Up* and *Down*. At any point of time, the outputs of phase detector can be in one of the four states which are tabulated as:

Table 3.1: Outputs of D flip flop based PFD

Up	*Down*	*Reset*
0	0	0
0	1	0
1	0	0
1	1	1

Here, the information related to phase error is coded at the output of a PFD with the use of two D flip-flops (DFF). The DFF outputs are denoted as *Up* and *Down*, individually. Expecting both outputs are at first low, a rising edge of reference clock causes the *Up* signal to go high. This

shows that the VCDL's delay should be expanded to match the reference signal. In any case, if Output of the VCDL leads the reference clock, *Down* signal goes high showing that the output of the VCDL should be delayed to match the reference signal. At the point when both outputs go high, the AND gate generates a reset signal giving back the PFD to the zero state. In this way, the "*Up* = *Down* = high" state is removed by the reset, and the PFD is basically a three-state device.

In any case, as the reference and output clock approach one another, the DFFs get reset before any charge is fed onto the capacitor by the CP; this characterizes the dead zone. The vicinity of the dead zone causes jitter in DLL. It can be reduced by inserting inverters into the reset path to increase the reset delay. On the other hand, enlarging the reset path delay expands the time duration in which the both PMOS and NMOS transistors turned on charge pump paths are simultaneously conducting. This short circuit alters the VCDL control voltage and results in increased phase noise. The DFFs also need to be matched. If one DFF resets earlier than the other, it can cause the reset signal to go to zero before the other DFF resets.

3.4 CMOS REALIZATION OF PFD

The PFD demonstrated in fig. 3.6 can be realized using inverters and NAND gates as shown in fig. 3.7.

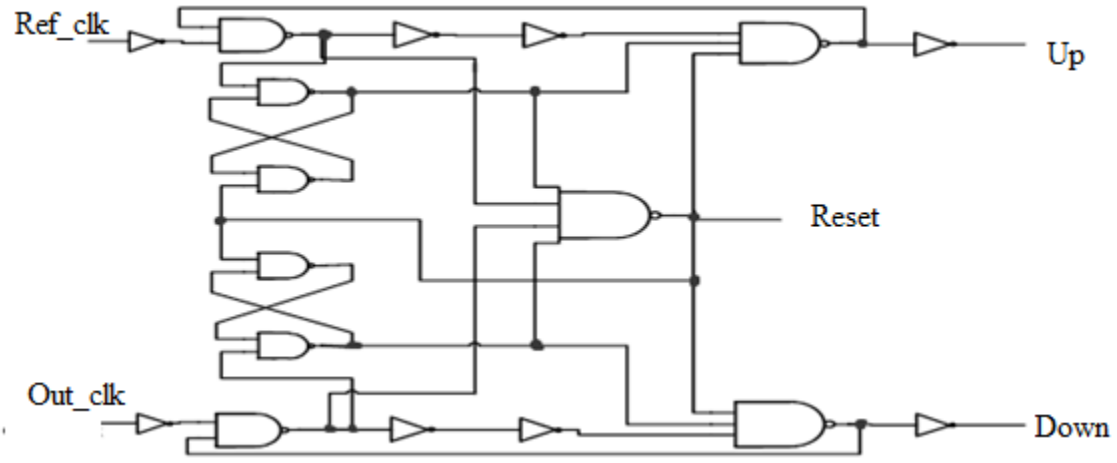


Fig. 3.8 Realization of PFD using NAND based D latch [7]

3.5 SIMULATION OF CMOS PFD CIRCUIT

Realization of circuit shown in fig. 3.8 has been done in TSMC 180 nm CMOS process technology keeping device dimensions as $W_p = 3600$ nm ,$W_n = 1800$ nm and $L_p = L_n = 180$ nm. Simulation Results for circuit shown in fig. 3.8 has been shown in fig. 3.9, fig. 3.10 and fig. 3.11. In fig. 3.9 and fig. 3.10 the phase difference between reference and output clock is $\pi/2$. Figure 3.9 shows the result when reference clock leads the output clock and in fig. 3.10 reference clock lags behind output clock.

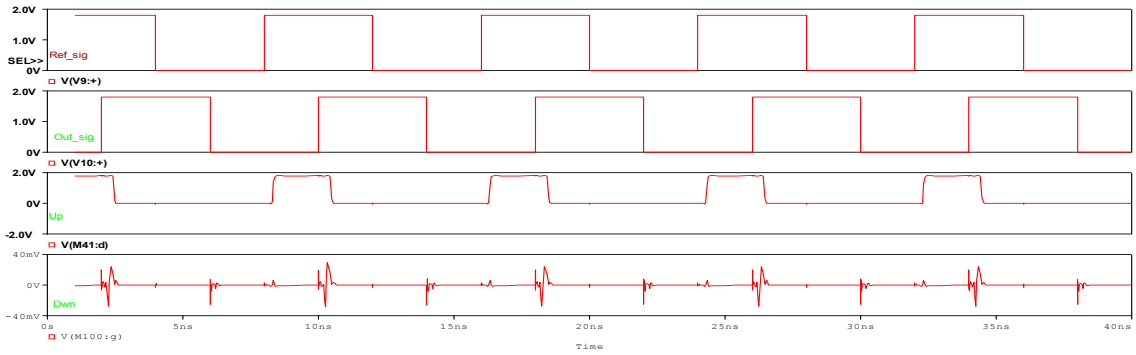


Fig. 3.9 Simulation result of PFD when reference clock leads the output clock

From above simulation it can be observed that the width of the *Up* error signal is approximately equal to the duration between rising edges of both the pulses and the *Down* signals are glitches.

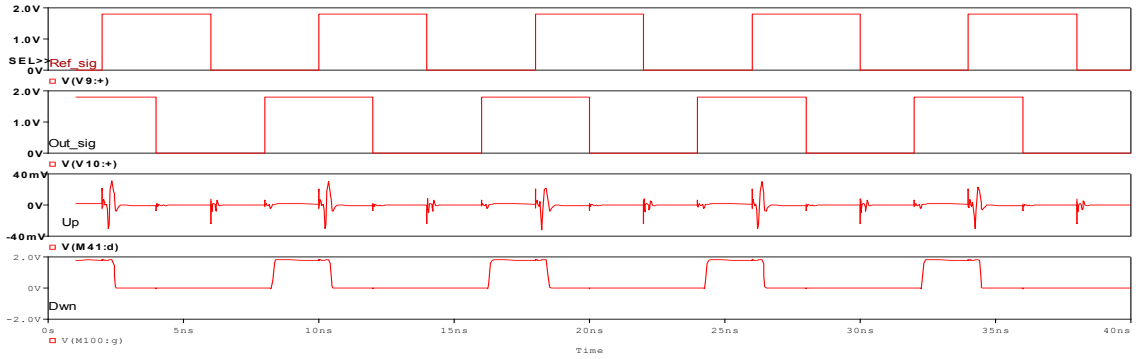


Fig. 3.10 Simulation result of PFD when reference signal lags the output signal

Here simulation has been done in opposite to previous one as reference clock lags the output clock. As a result, *Down* signal has width approximately equal to duration of rising edge of both the signals.

At the last in fig. 3.11 the situation when both clocks are synchronized or having phase difference of 0 or 2π has been simulated.

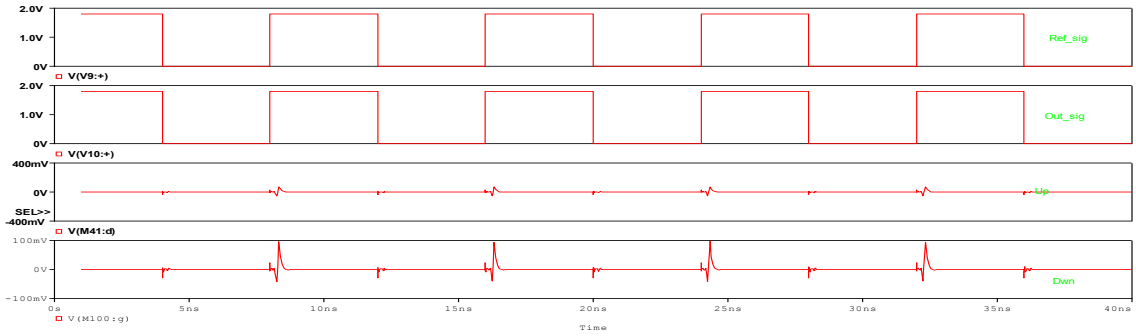


Fig. 3.10 Simulation result of PFD when phase difference is 0 or 2π.

On the rising edges of both signals approaches each other or synchronizes the AND gate in reset path resets both the outputs of D flip-flops. Here some ripples are present at this moment, it's duration in actual is time taken by the overall circuit to give a stable zero at outputs of D flip-flops. In following fig. 3.11 enlarged view of *Up* and *Down* signals have been shown in to measure the reset time of the PFD. Reset time of CMOS-PFD is 0.133ns.

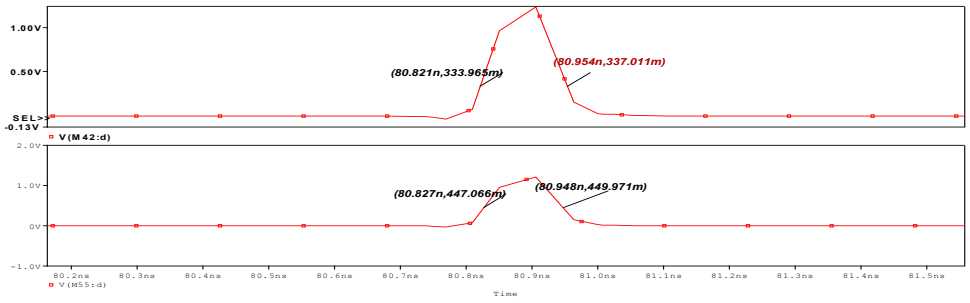


Fig. 3.11 Enlarged view of *Up* and *Down* signals

3.6 *C²MOS* – Register based PFD

In order to design a PFD using C²MOS (Clocked CMOS) – Register, this register should have a reset arrangement in itself. Figure 3.12 below depicts the circuit of positive edge triggered C²MOS – Register based on master – slave configuration with an active low reset. Before going to design a PFD let's have a brief overview on working mechanism of *C²MOS* – Register with an active low reset.

This C^2MOS (Clocked CMOS) – Register works on two phases:

1. $CLK = 0(\overline{CLK} = 1)$: Tristate driver in first stage is turned on and the master stage becomes inverter and the inverted version of D is stored at node X. At this moment master stage is in evaluation mode and slave stage is in hold mode. Transistors M_7 and M_8 are off, disconnecting input to output. Hence previous value of Q is hold at the output.
2. $CLK = 1(\overline{CLK} = 0)$: This time roles of master and slave stages are interchanged. The master stage is in hold (M_3 - M_4 off) and the slave stage is in evaluation mode (M_7 - M_8 on). The value stored at node X is propagated through second stage which is acting as inverter.

This circuit can be reset only in this second phase (CLK = 1, $\overline{CLK}$ =0) on applying active low at reset. When reset = 0 is applied the output Q immediately becomes 0. This positive edge-triggered master – slave register offers benefit of insensitivity to clock overlap, provided rise and fall times of the clock edges are sufficiently small [6].

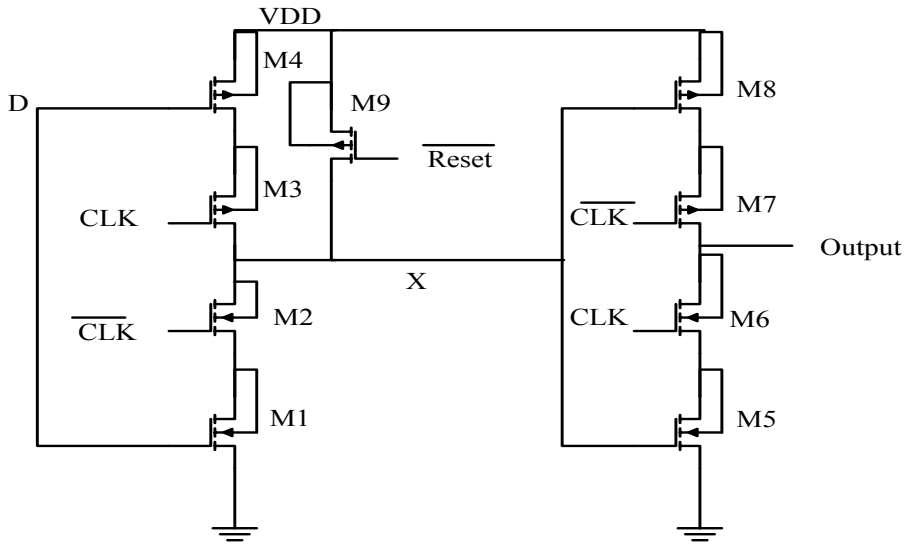


Fig. 3.12 C^2MOS– Register [8]

3.6.1 PFDs employing C^2MOS – Register

To propose PFDs which make use of C^2MOS – Register, we have used the concept of D flip-flop based PFD which has been described earlier in fig. 3.2. Here we simply replaced the D flip-flops with C^2MOS – Registers and the output of NAND is connected to reset of both registers. Figure 3.13 below is depicting the proposed PFD and it is termed as type-1.

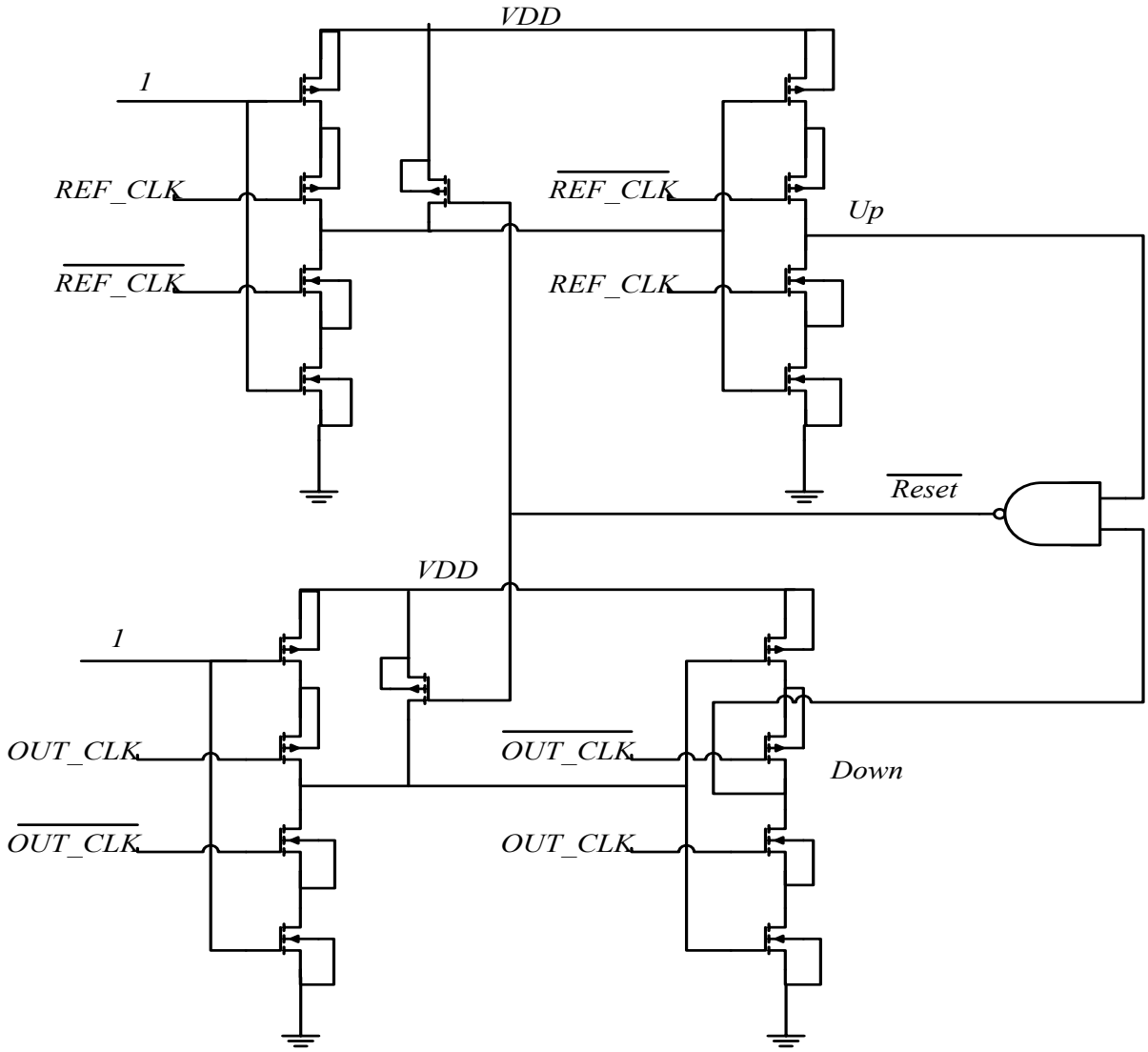


Fig. 3.13 Proposed C^2MOS – Registers based PFD (Type-1)

From the truth table1 of PFD we can conclude that reset is only activated when both the outputs *Up* and *Down* are high. So by utilizing this fact we can change the reset path of PFD shown in fig.3.13 as shown in fig.3.14. As evident in fig.3.14 we have employed two PMOS transistors for

reset in both C^2MOS – Registers. The inputs for these PMOS transistors are denoted as $\overline{Up}$ and $\overline{Down}$ which are inverted versions of outputs *Up* and *Down*. The motive behind using this method is to reduce the number of transistors and we have effectively reduced two units of transistors in this circuit as compare to circuit shown in fig.3.13. Since $\overline{Up}$ and $\overline{Down}$ require two inverters (total four transistors) for its generation and also NAND requires four transistors, so one can raise doubt whether we had reduced the number of transistors effectively. In DLL the PFD is followed by Charge Pump which requires four inputs these are *Up*, $\overline{Up}$, *Down* and $\overline{Down}$. Hence we can say that the architecture of Charge Pump employs two inverters to generate *Up* and $\overline{Down}$ that can also be utilized to reset the PFD shown in fig. 3.13 and there is no need of extra inverters to be provided in design of PFD to reset. We have termed the PFDs of fig.3.13 and fig. 3.14 as type-1 and type-2 respectively. From PFD type-2 we have further reduced one more transistor which has been termed as type-3 shown in fig. 3.15. Three PMOS transistors have been employed here for reset effectively reducing the three transistors as compare to type-1 PFD.

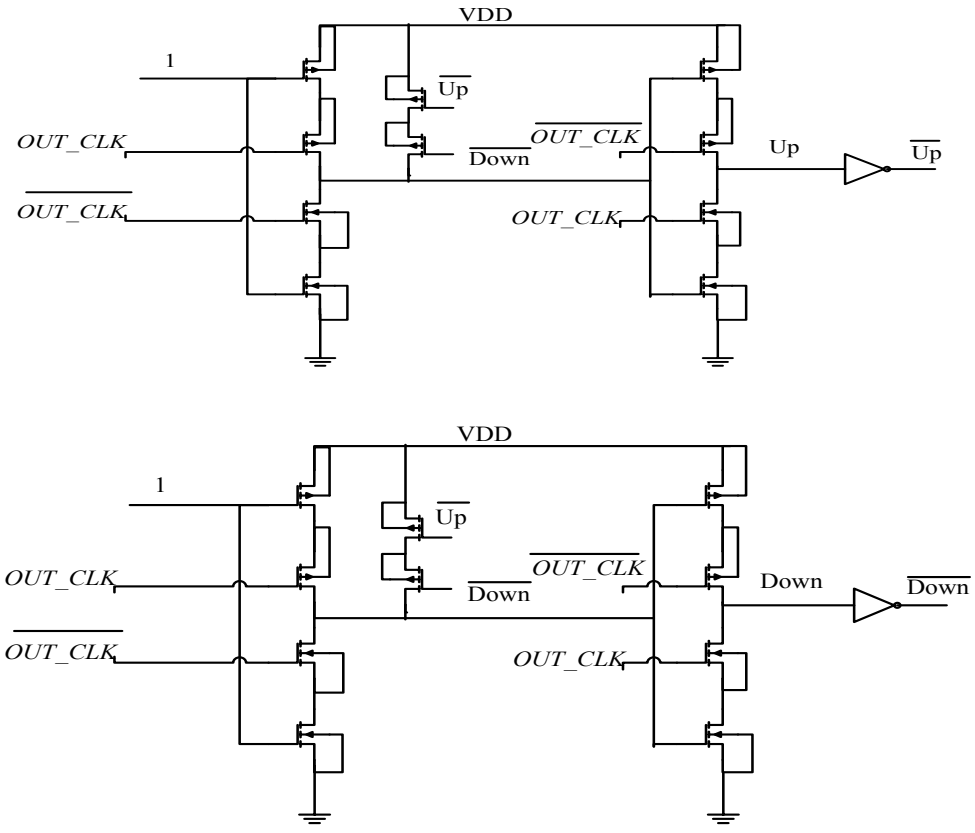


Fig. 3.14 Proposed Type - 2 PFD

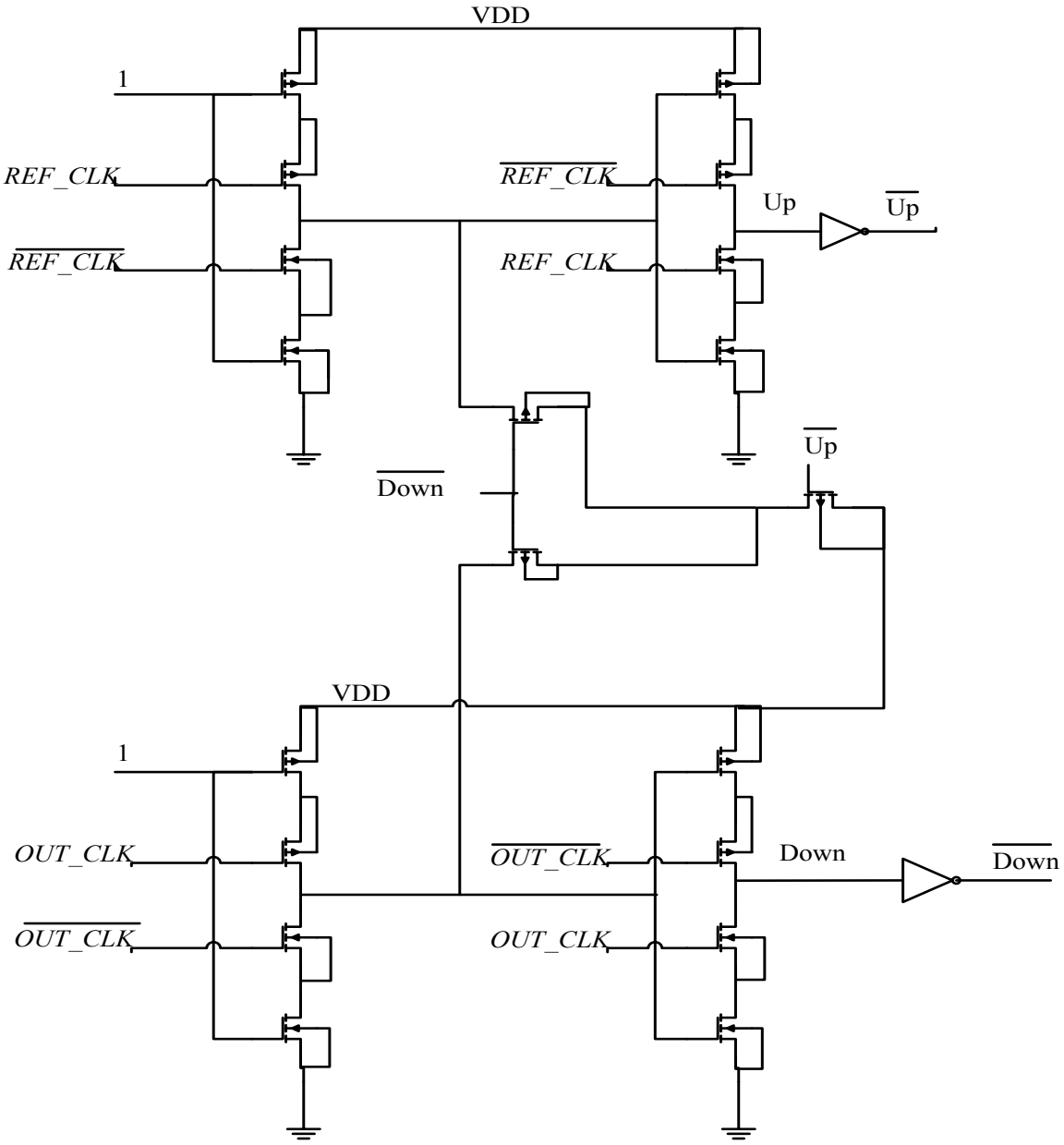


Fig. 3.15 Proposed Type - 3 PFD

3.7 SIMULATIONS OF C^{2}MOS – PFD CIRCUIT

Realization of circuits shown in fig. 3.13, 3.14 and 3.15 has been done in TSMC 180 nm CMOS process technology keeping device dimensions as W_p = 3600 nm ,W_n = 1800 nm and L_p = L_n = 180 nm. Simulations which were performed for CMOS-PFD, is performed for all three types of C^{2}MOS – PFD.

3.7.1 Simulation results of Type 1 PFD

The functionality of the circuits of fig. 3.13 – 3.15 have been verified through time domain SPICE simulations using TSMC 180 nm while widths of 1800 nm and 3600 nm are used uniformly for all NMOS and PMOS transistors.

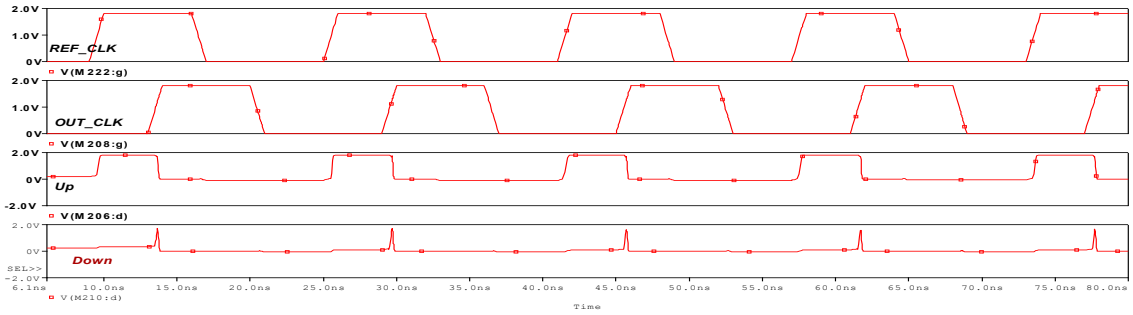


Fig. 3.16 Simulation result of Type-1 PFD when reference clock leads the output clock

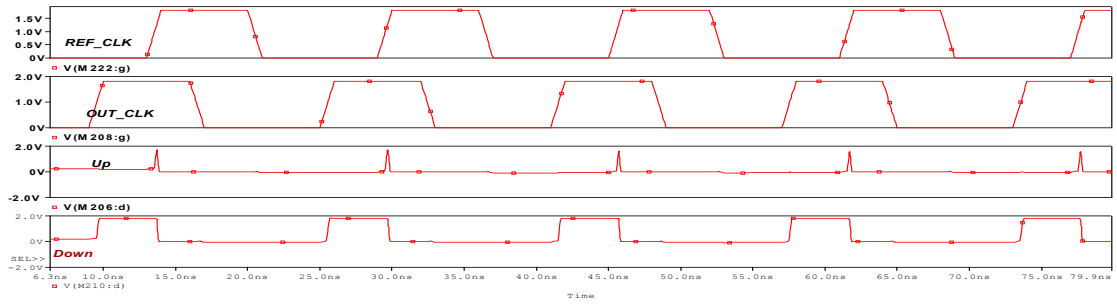


Fig. 3.17 Simulation result of Type-1 PFD when output clock leads the reference clock

At the last in fig. 3.18 the situation when both clocks are synchronized or having phase difference of 0 or 2π has been simulated.

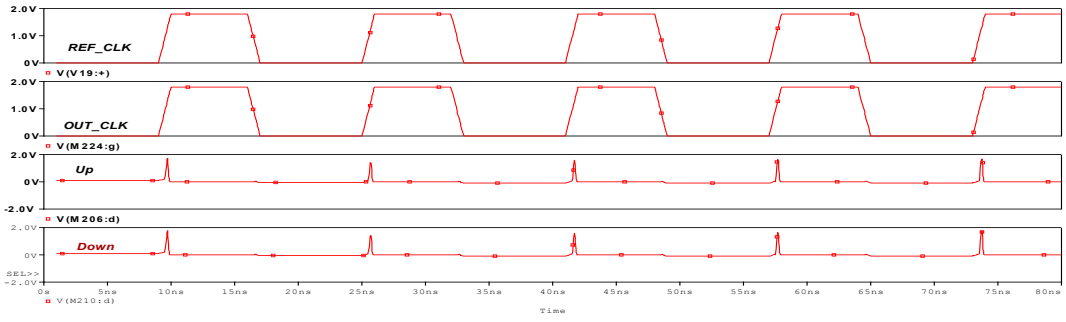


Fig. 3.18 Simulation result of Type-1 PFD when phase difference is 0 or 2π .

In all three simulations shown in fig. 3.16, 3.17 and 3.18, results show similarity with the simulation results of CMOS-PFD. Further in fig. 3.19 enlarged view of *Up* and *Down* signals has been shown to measure the reset timing. Reset time of Type-1 PFD is 0.257ns.

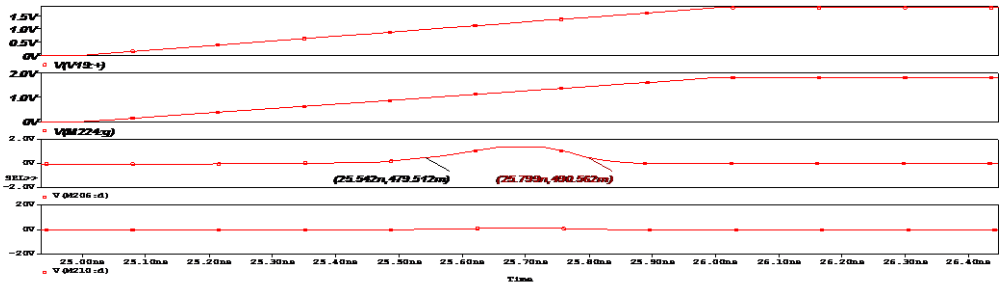


Fig. 3.19 Enlarged view of *Up* and *Down* signals

3.7.2 Simulation of Type 2 PFD circuit

Simulations in all three cases which was done for Type-1 PFD, same has been performed for circuit shown in fig. 3.14 have been shown in from fig. 3.20 to fig. 3.23. Simulation results have found similar to simulation results of Type 1- PFD and CMOS-PFD except for reset time.

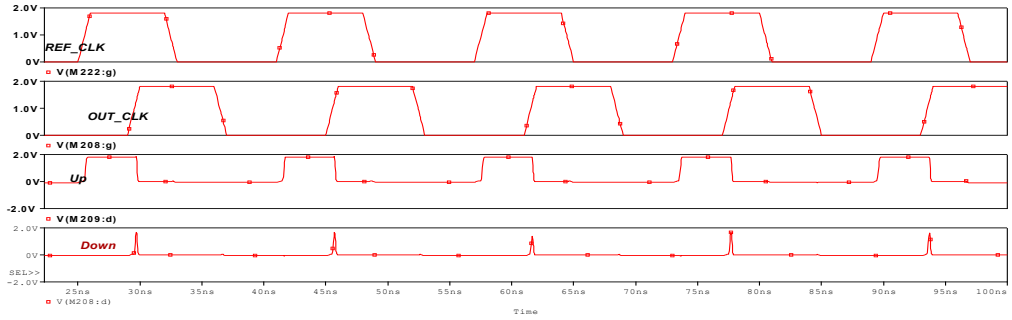


Fig. 3.20 Simulation result of Type-2 PFD when reference clock leads the output clock

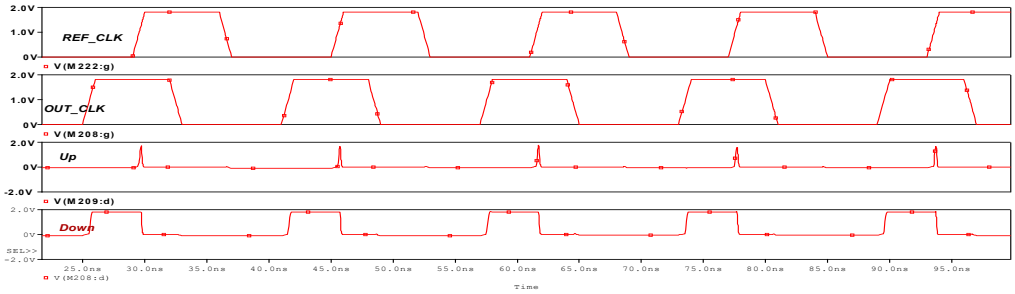


Fig. 3.21 Simulation result of Type-2 PFD when reference clock lags the output clock

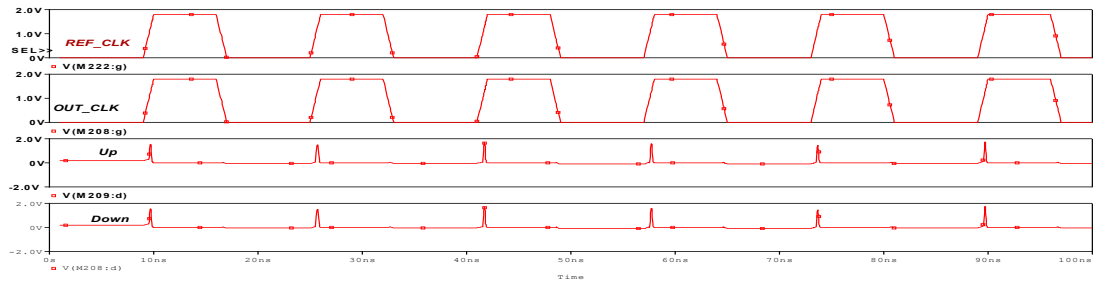


Fig. 3.22 Simulation result of Type-2 PFD when phase difference is 0 or 2π.

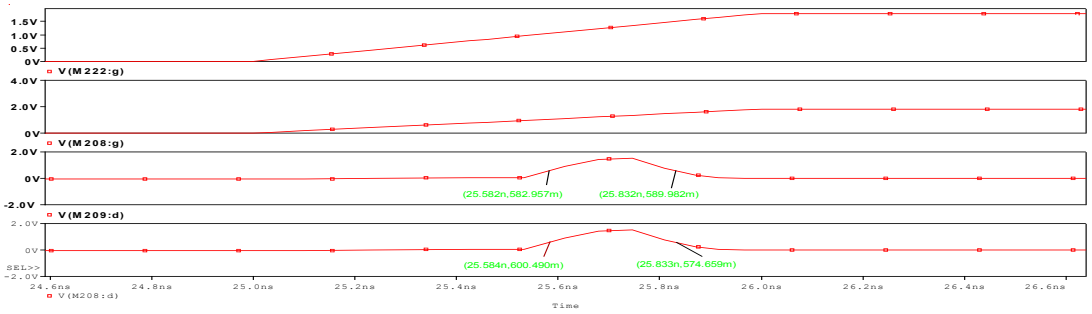


Fig. 3.23 Enlarged view of *Up* and *Down* signals

Reset time of Type 2 PFD is 0.250ns.

3.7.3 Simulation of Type 3 PFD circuit

Simulations in all three cases which was done for Type-1 PFD, same has been performed for circuit shown in fig. 3.14 have been shown in from fig. 3.24 to fig. 3.27. Simulation results have found similar to simulation results of Type 1- PFD and CMOS-PFD except for reset time.

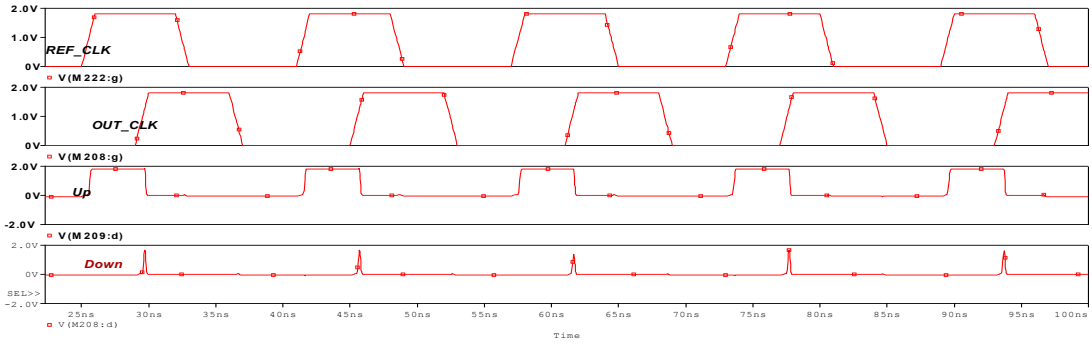


Fig. 3.24 Simulation result of Type-3 PFD when reference clock leads the output clock

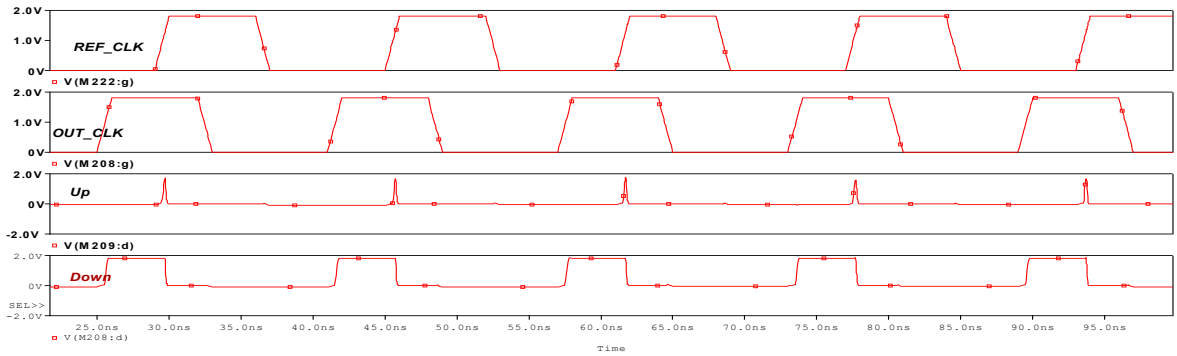


Fig. 3.25 Simulation result of Type-3 PFD when reference clock lags the output clock

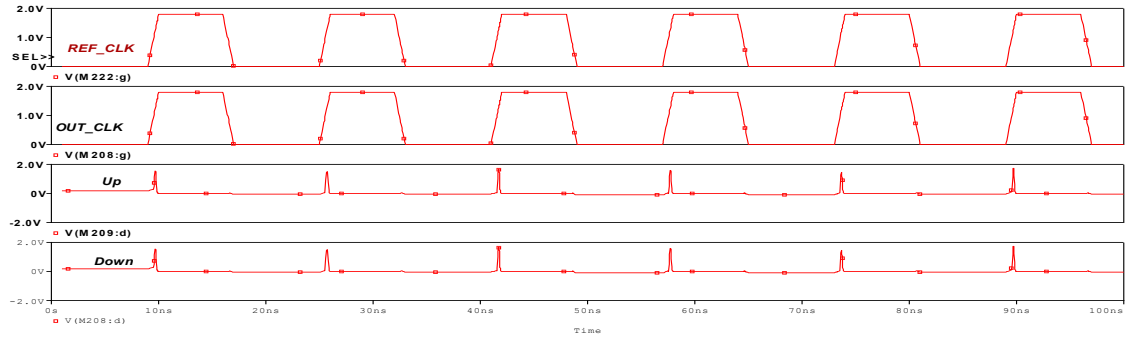


Fig. 3.26 Simulation result of Type-2 PFD when phase difference is 0 or 2π

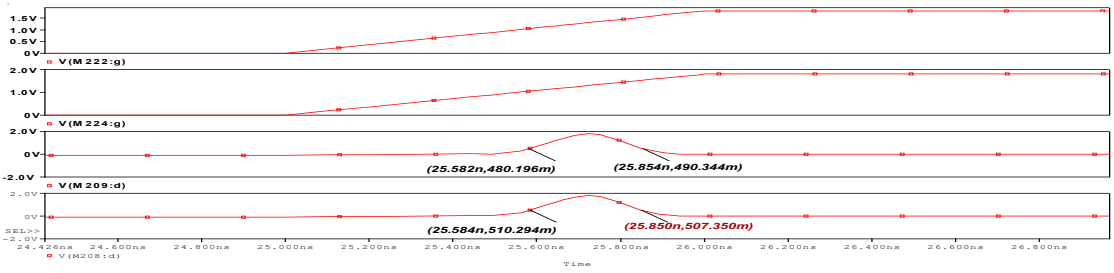


Fig. 3.27 Enlarged view of *Up* and *Down* signals

Reset time for Type-3 PFD is 0.272ns.

3.8 Performance Comparison among CMOS – PFD and all three types C^2MOS - PFD

The performance of all PFDs presented in this chapter is compared on the basis of transistor count, power consumption and reset time. The results are summarized in table 3.2.

Table 3.2. Performance Comparison among CMOS – PFD and all three types C^2MOS – PFD.

Sr. No.	Parameter	CMOS- PFD	C^2MOS-PFD Type-1	C^2MOS-PFD Type-2	C^2MOS-PFD Type-3
1.	Transistor Count	60	26	24	23
2.	Power dissipation	3.64×10^{-10} W	8.41×10^{-5} W	8.41×10^{-5} W	8.41×10^{-5} W
3.	Reset time	0.133 ns	0.257 ns	0.250ns	0.272ns

It is found that CMOS-PFD shows most power efficient characteristic. On the other hand, C^2MOS-PFDs dissipate more power. Benefit involved in using the C^2MOS-PFDs is its less transistor count as compared to CMOS PFD's transistor count.

CHAPTER 4

CHARGE PUMP AND LOOP FILTER

This chapter focuses on the operation of charge pump and the role of capacitor in DLL. Since current mirrors are integral parts of the charge pump, these are also explained here.

4.1 INTRODUCTION

The VCDL requires only one control signal but error signals generated by the PD or PFD are two in number, hence there is a requirement of combining the phase error signals into a single signal. This is accomplished by Charge Pump and a Loop filter. In a Delay Locked Loop, the phase error between the input reference signal and the VCDL output signal is detected by the PFD and transferred to the charge pump in the form of voltage pulses. The charge pump performs the function of adjusting the voltage of the loop filter and thereby altering the VCDL delay according to the phase error information provided from the Phase Frequency Detector.

As Charge Pump plays vital role in holding the control voltage of the VCDL in accordance with the phase error its proper designing is also important. We can provide higher values of current to loop filter through Charge Pump so that charging and discharging become fast, and this leads to better speed of the overall loop. The high value of current , however leads to higher power dissipation. Hence there are trade-offs, so a careful design approach is required here. Current mirror circuits are used as current sink and source in Charge Pump circuit depending upon *Up* and *Down* signals.

4.2 CHARGE PUMPs

The outputs of a Phase Frequency Detector are combined into a single input to drive the Loop Filter through a Charge Pump. There are two methods of doing this:
The first method is called a tri-state output which is shown in fig. 4.1.

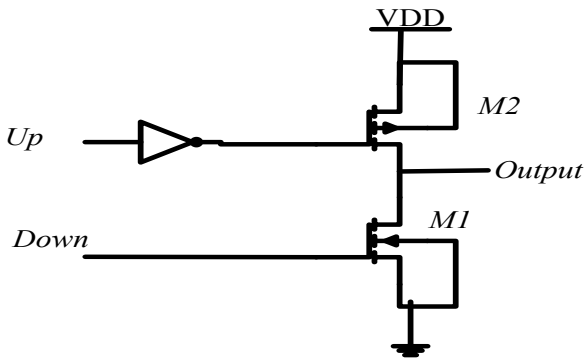


Fig. 4.1 Tri –state charge pump [8]

Both MOS devices in fig. 4.1 are functioning as switch which is used to connect the output terminal to either power supply V_{DD} or to the ground. When both *Up* and *Down* signals are low the output terminal will remain in high impedance state. A high signal at *Up* connects the output to the power supply and high at *Down* connects the output terminal to ground. The power supply variations affect its working as it modulates the output voltage.

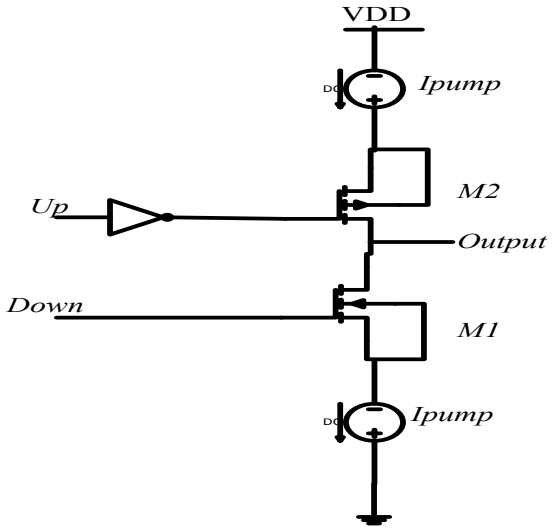


Fig. 4.2 Charge Pump [8]

The second configuration shown is in fig. 4.2 is referred as Charge-Pump. MOS current sources are placed in series with NMOS and PMOS transistors of fig. 4.1. In this configuration both MOSFETS are working as switches, that connects or disconnects, the current sources to the output terminal. Because the current source can be made insensitive to supply variation, modulation of the VCDL control voltage doesn't take place [7].

4.3 PRACTICAL IMPEMENTATION OF THE CHARGE PUMP

The circuit of fig. 4.3 demonstrates the idea of the PFD with charge pump. Here in this circuit we come across some design issues that is when the sources of transistors M1 and M2 charge to *Ground* and *VDD* respectively, as when they are switched off. For example, assume that the source of M1 is released to ground when the *Down* signal goes high. At the point when M1 turns on, it doesn't supply the charge set by I_{pump} to the loop filter but instead, until the voltage over the current sink increases, carries on like a switch connecting the input of the filter to ground, which indicates we are not controlling the signal that is connected to the loop filter.

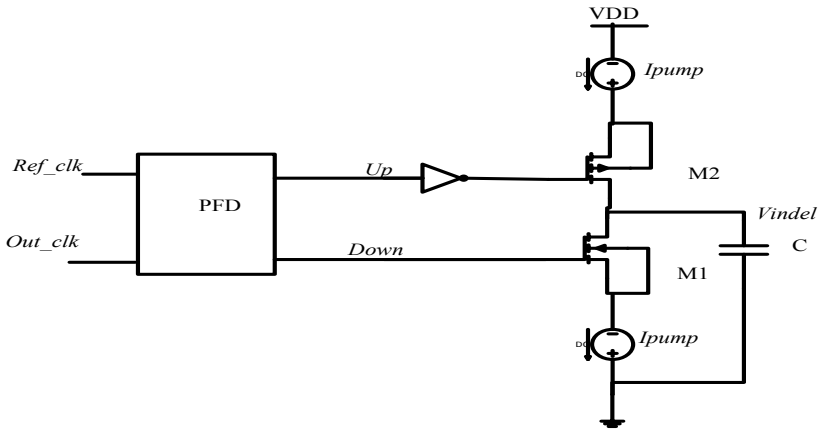


Fig. 4.3 PFD with Charge Pump [8]

To solve this problem, consider the circuit which is indicated in fig. 4.4.

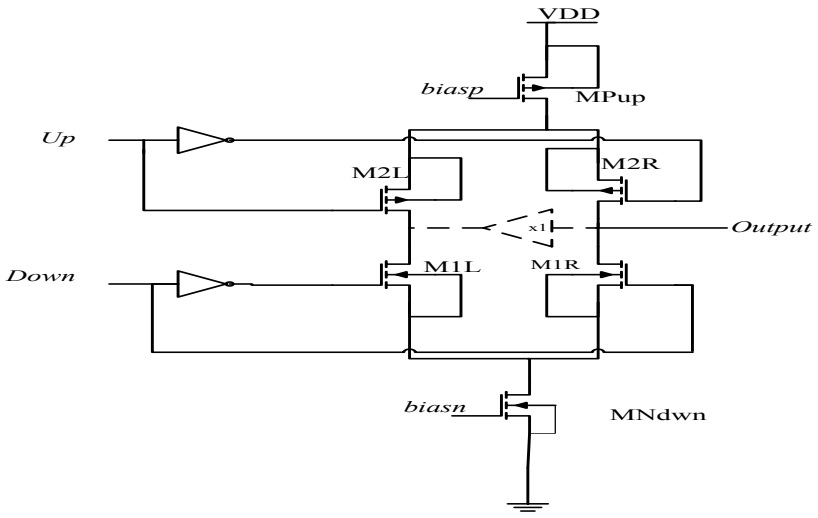


Fig. 4.4 Practical Implementation of the Charge Pump [8]

At the point when *Up* and *Down* signals are low, M1L and M2L are on. The pump-up current source, MPup, is driving the pump-*Down* current source, MNdwn. At the point when either *Up* or *Down* signals goes high, the output is connected with one of the current sources. The MOSFETs M1L, M1R, M2L and M2R just direct the current to the loop filter. Another issue which we have with this topology is that when M1L and M2L are both on, the voltage on their drains will not be matched correctly with the voltage at the loop filter (V_{indel}). It means, the voltage over each of the current sources (MPup and MNDwn) will not be the same as it is the point at which they are connected with the output node. It results in charge sharing between the parasitic capacitance on the drains of MPup and MNdwn and the capacitance utilized as loop filter causes a static phase error or jitter. To handle this difficulty, an amplifier (demonstrated in dashed lines in the fig. 4.4) can be inserted to keep the drain voltage of M1L and M2L to the same value V_{indel} [8].

4.4 CURRENT MIRRORS

Current mirrors are integral part of the charge pump as they are used in this architecture to provide the bias voltages i.e. are *biasn* and *biasp* .

The Current Mirrors are active devices which are operated based on "copying" currents from a reference, with the assumption that one precisely – defined current source is already available. This "copying" of current from a reference current source to an active device is called "mirroring". The conceptual meaning has been shown in fig. 4.5.

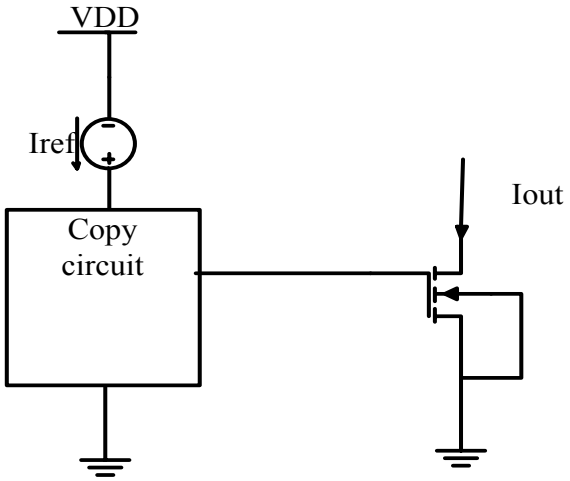


Fig. 4.5 conceptual means of copying currents [7]

$I_{out} = I_{REF}$ is done in following way:

For a MOSFET the value of drain current I_D can be expressed as

$$I_D = f(V_{GS}) \quad \text{.....(4.5)}$$

Where $f(.)$ denotes the functionality of I_D versus V_{GS} (gate to source voltage).

Then $$V_{GS} = f^{-1}(I_D) \quad \text{.....(4 .6)}$$

It means if a transistor is biased at I_{REF}, then it gives $V_{GS} = f^{-1}(I_{REF})$ as shown in fig. 4.6.

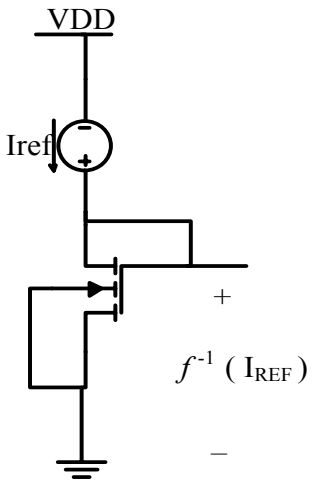


Fig. 4.7 Diode- connected device providing inverse function [7]

Hence, if same voltage is applied to the gate and source terminal of another MOSFET,the same current can be obtained in the second MOSFET. From fig. 4.7, $I_{out} = f f^{-1}(I_{REF}) = I_{REF}$.

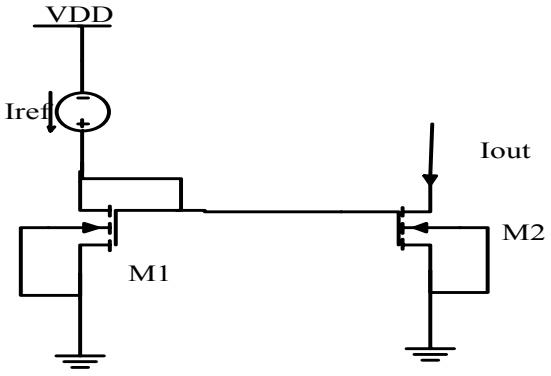


Fig. 4.7 NMOS current mirror [7]

In a simple way we can state the above phenomenon as two identical MOSFETs, are given equal gate to source voltages will have equal currents if they are operated in saturation (considering the channel length modulation is zero i.e. λ=0).

From fig. 4.7, we can write

$$I_{REF} = \frac{1}{2} \mu_n C_{ox} \left(\frac{W}{L}\right)_1 (V_{GS} - V_{TH})^2 \quad \text{.....(4.7)}$$

$$I_{out} = \frac{1}{2} \mu_n C_{ox} \left(\frac{W}{L}\right)_2 (V_{GS} - V_{TH})^2 \quad \text{.....(4.8)}$$

While writing above equations of currents we have ignored the effect of channel length modulation means λ = 0.

From the equations 4.7 and 4.8 we can obtain the relation between I_{REF} and I_{out} , as

$$I_{out} = \frac{(W/L)2}{(W/L)1} I_{REF} \quad \text{.....(4.9)}$$

This topology allows the copying of current in a precise manner without depending upon process and temperature. The ratio of reference current and output current is completely in terms of device dimensions which can be controlled with reasonable accuracy [7].

The PMOS version of fig. 4.7 is depicted in fig. 4.8

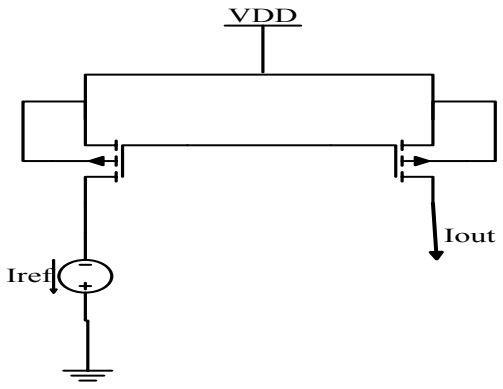


Fig. 4.8 PMOS current mirror [7]

4.5 LOOP FILTER

The output from the combined blocks of PFD and charge pump has a large periodic content due to the repetitive phase detection. To remove the high frequency component, at the output of the

charge pump a loop filter is connected. This filter output is then used as control voltage for the VCDL

Typically a low pass filter is connected at the outputs of PFD as loop filter in PLL. A DLL uses simply a capacitor to integrate the phase error, which subsequently increases control voltage to VCDL. Though the lock time increases with increasing capacitor values, it improves bandwidth and ripples on control voltage.

A decrease in capacitor values result in lower lock time at the expense of increased ripples on control voltage.

4.6 SIMULATION RESULTS

Simulations for NMOS current mirror (fig. 4.7), PMOS current mirror (fig.4.8) and the combination of charge pump (CP) and loop filter (LF) is shown in from fig.4.9 to fig. 4.10. All the circuits have been realized in TSMC 180 nm CMOS technology taking device dimensions as W_p = 3600 nm, W_n = 1800 nm and $L_p = L_n$ =180 nm.

4.6.1 SIMULATION OF CURRENT MIRROR CIRCUITS

In fig. 4.9 the simulation for PMOS current mirror which is shown in fig. 4.8 has been shown. This simulation shows the variation of I_{out} with respect to change in supply voltage.

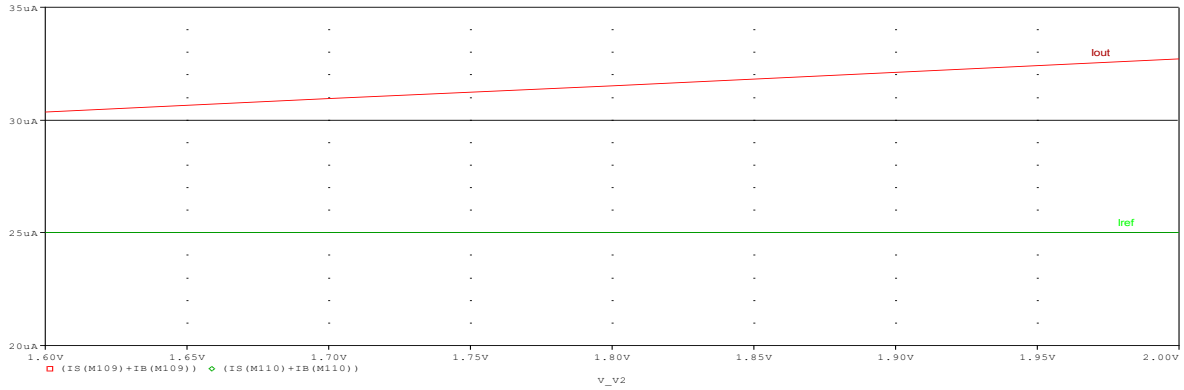


Fig. 4.9 Simulation of PMOS current mirror shown in fig. 4.6

Current I_{ref} = 25µA has been shown with green line and I_{out} with the red one which is trying to copy I_{ref} . It is evident from above simulation that the variation in supply doesn't affect too much the output I_{out} .

In fig. 4.10 the simulation for NMOS current mirror which is shown in fig. 4.7 has been shown. Here result is same as it was in previous case.

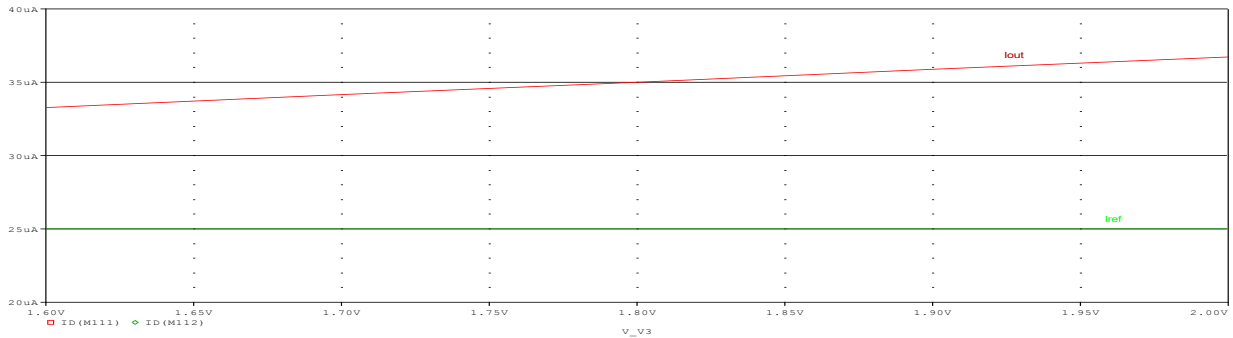


Fig. 4.10 Simulation of NMOS current mirror shown in fig. 4.7

4.6.2 PFD and CHARGE PUMP

For simulation circuit shown in fig. 4.11 has been used.

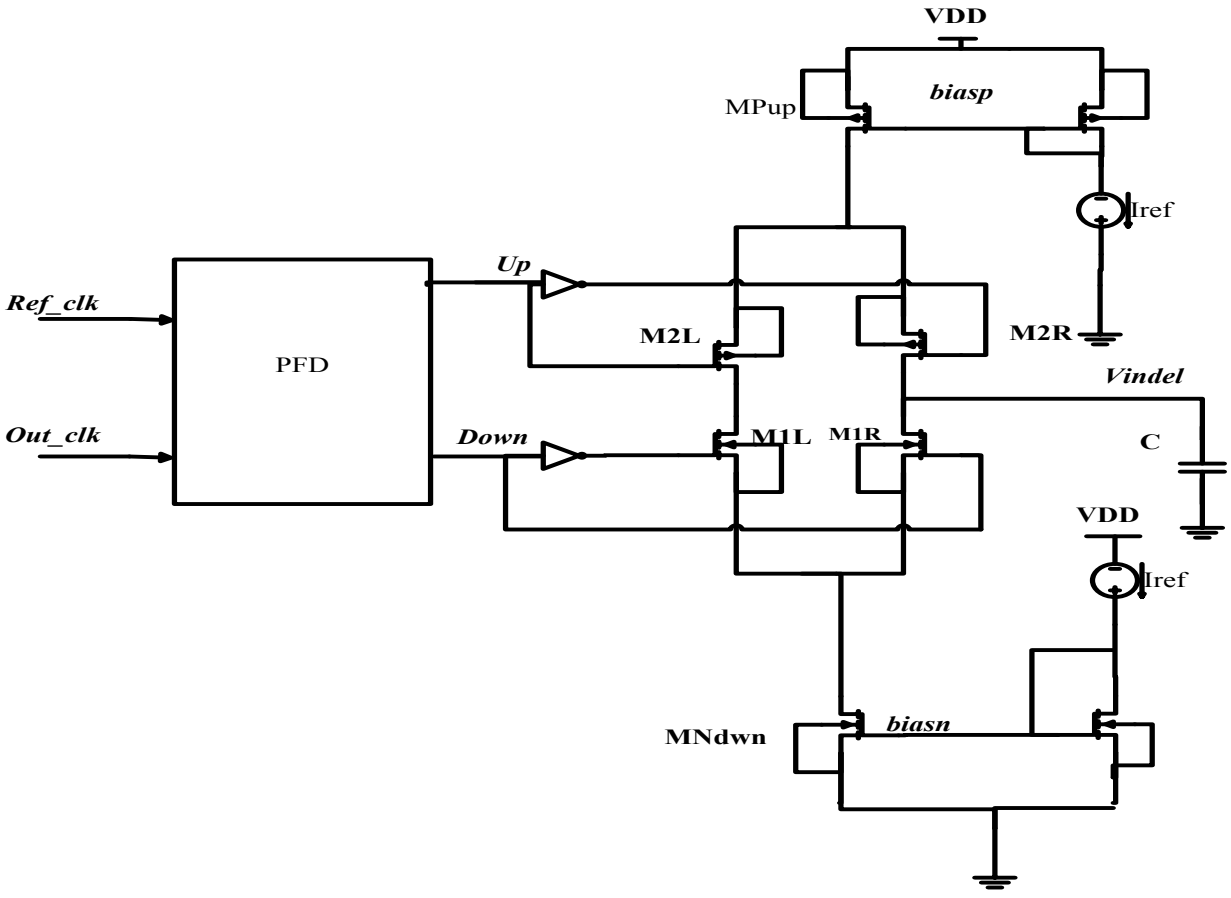


Fig. 4.11 Circuit used for the simulation of Charge Pump with PFD[8]

This arrangement is done to observe what happens at the voltage of capacitor when signals possess various phase differences.

In fig. 4.12 and fig. 4.13 the phase difference between reference and output signal is π/2. Figure 4.12 shows the result when reference signal leads the output signal and in fig. 4.13 reference signal lags behind output signal.

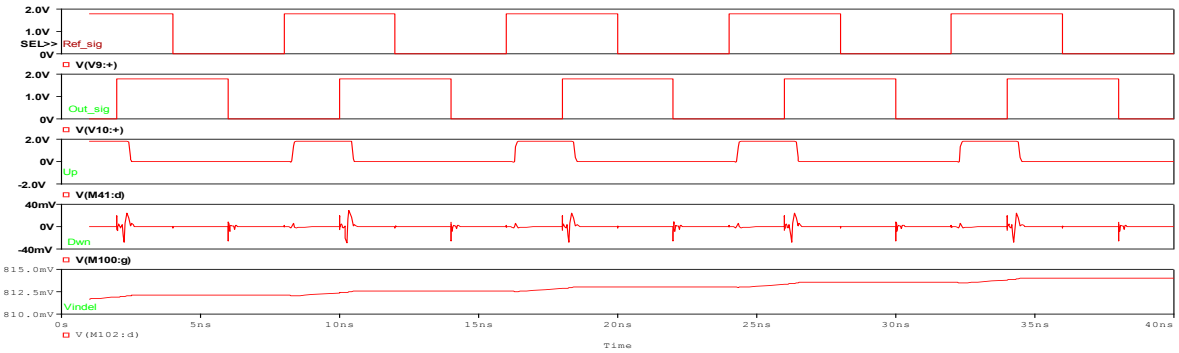


Fig. 4.12 Simulation result of circuit shown in 4.11when reference clock leads the output clock

Simulation shown in fig. 4.12, signal at node *Up* goes high. Hence the voltage on capacitor keeps on increasing.

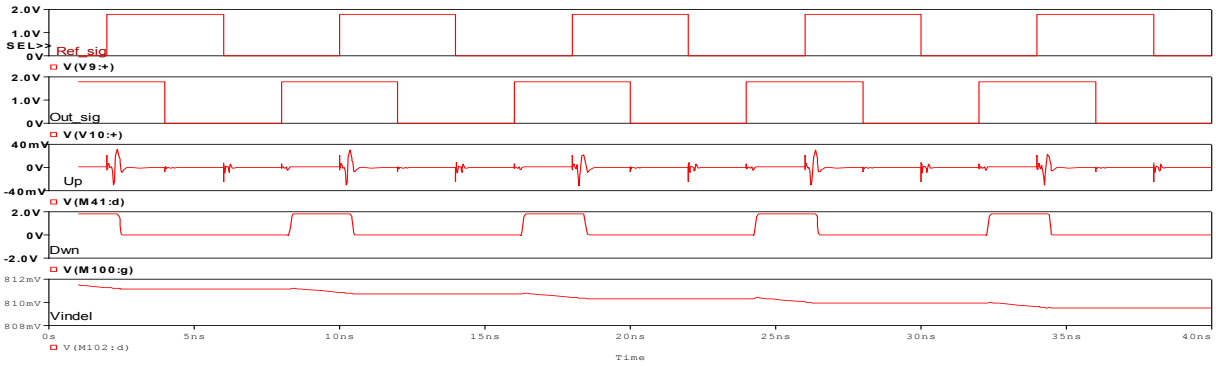


Fig. 4.13 Simulation result of circuit shown in 4.11when reference clock lags the output clock

Simulation shown in fig. 4.13, it is observed that the voltage on capacitor continuously decrease as only *Down* signal is high.

In fig. 4.14, case when the phase difference between the clocks is 0 or 2π has been taken

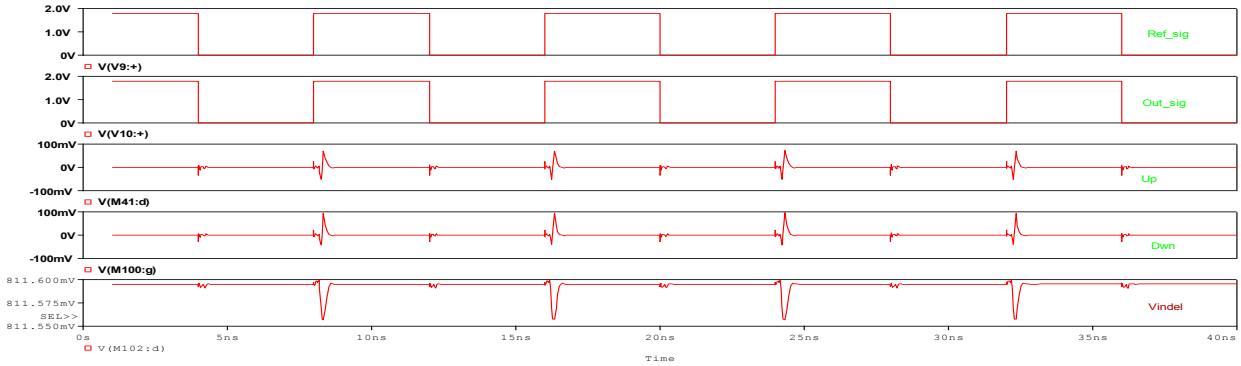


Fig. 4.14 Simulation result of circuit shown in 4.11when reference clock and the output clock is synchronized.

Simulation shown in fig. 4.14 neither *Up* nor *Down* signal goes high hence loop filter maintains a constant voltage.

CHAPTER 5

VOLTAGE CONTROLLED DELAY LINE

In this chapter different types of delay cells are described which are used to realize the VCDL. Thereafter realization of fully differential VCDL is described. Transfer characteristic of different VCDL is plotted .

5.1 INTRODUCTION

One of the most critical blocks that a DLL consists of is Voltage Controlled Delay Line as the output signal of the DLL is directly taken from it. The function of Voltage Controlled Delay Line (VCDL) is to delay the reference clock so that there is no skew between the output clock and reference clock. The lock range and stability of a DLL mainly depend upon VCDL. A typical VCDL consist of a number of delay stages, connected in series. The VCDL is different from the voltage controlled oscillator (VCO) in a DPLL (Digital Phase Locked Loop) as it is an open loop configuration, so it does not oscillate.

Fig. 5.1 A typical VCDL configuration [10]

As the VCDL uses identical delay stages, the combination of individual delay stage is T_{ref}/n for an n stage VCDL.

5.2 VARIABLE DELAY ELEMENTS

Figure5.2 (a) shows the realization method of a VCDL employing adjustable delay inverters. At the last stage of the VCDL two extra inverters has been inserted so that output signal would have sharp and well defined edges. Figure 5.2 (b) and 5.2(d) show the circuit schematics for possible delay elements.

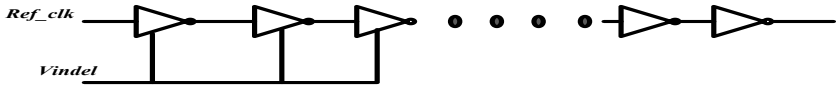


Fig. 5.2(a) VCDL made using inverter delay cells [8]

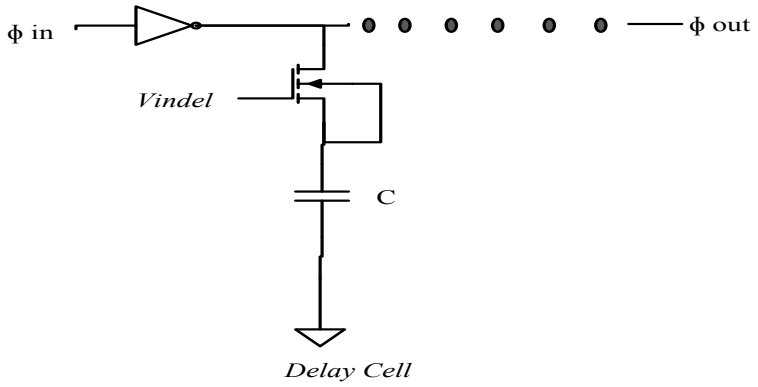


Fig. 5.2(b) R –C delay cells [8]

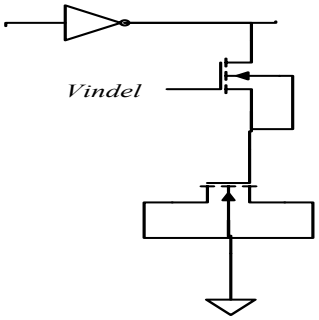


Fig. 5.2(c) Realization of C using NMOS in delay cell [26]

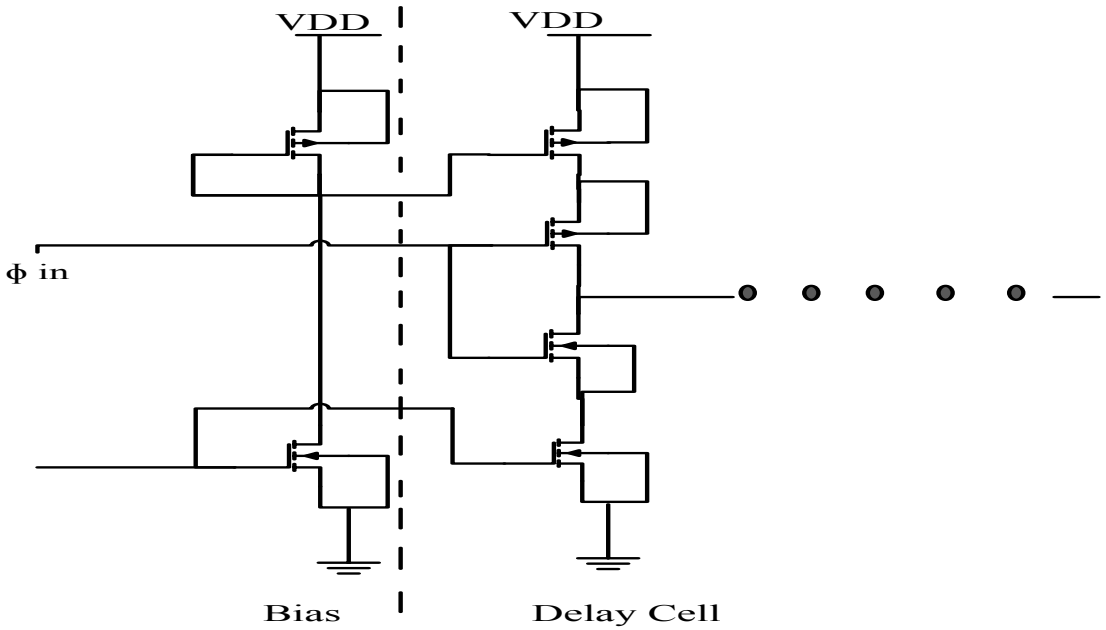


Fig. 5.2(d) Current Starving Delay cell [8]

Figures 5.2(b) and 5.2(d) are showing two types of VCDL implementations. In fig. 5.2(b) RC (Resistor- Capacitor) type delay cells have been employed to realize VCDL where resistor is voltage -controlled resistor that has been realized from NMOS and capacitor has also been realized from NMOS by shorting their source and drain terminals. Since resistors are voltage controlled, the delay can be made to vary with respect to control voltage V_{indel} . In fig. 5.2(c), the first stage of VCDL is a bias circuit used to control current in subsequent stages. In subsequent stages the current starving inverter delay cells have been cascaded to form delay chain. Since V_{indel} controls the current in inverters through bias circuit, charging and discharging of output terminals of inverters can be made variable with respect to change in control voltage V_{indel} .

Drawback: these types of delay cells are susceptible to noise and power supply variation. Due to this reason VCDL which are made of fully Differential delay elements are used.

5.3 FULLY DIFFERENTIAL VCDL

Fig. 5.3 shows the configuration of the fully differential VCDL.

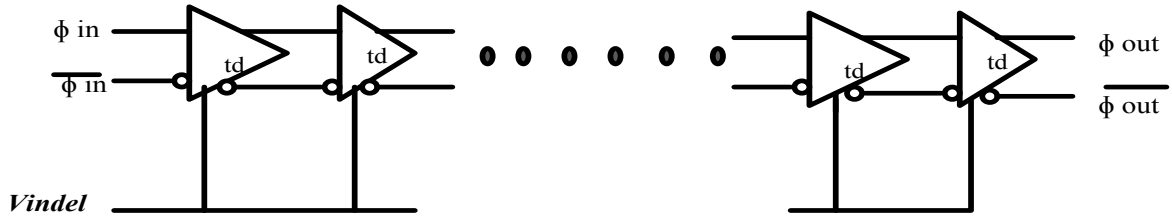


Fig. 5.3 Block Diagram of a fully differential VCDL [8]

Figure 5.4 shows the practical delay stage of the fig. 5.3 VCDL. A linear voltage to current converter is used to generate a control voltage that controls the delay of VCDL. Basically, all the delay stages in the VCDL are designed to be identical. Therefore, each delay stage contributes a time delay of T_{REF}/n for an n stage VCDL because in locked state the total delay generated by VCDL should be equal to time period of reference input i.e. is T_{REF} . As number of stages increases the phase resolution also increases. The control voltage given by charge Pump through Loop Filter controls the amount of delay of each element that constitutes VCDL [10].

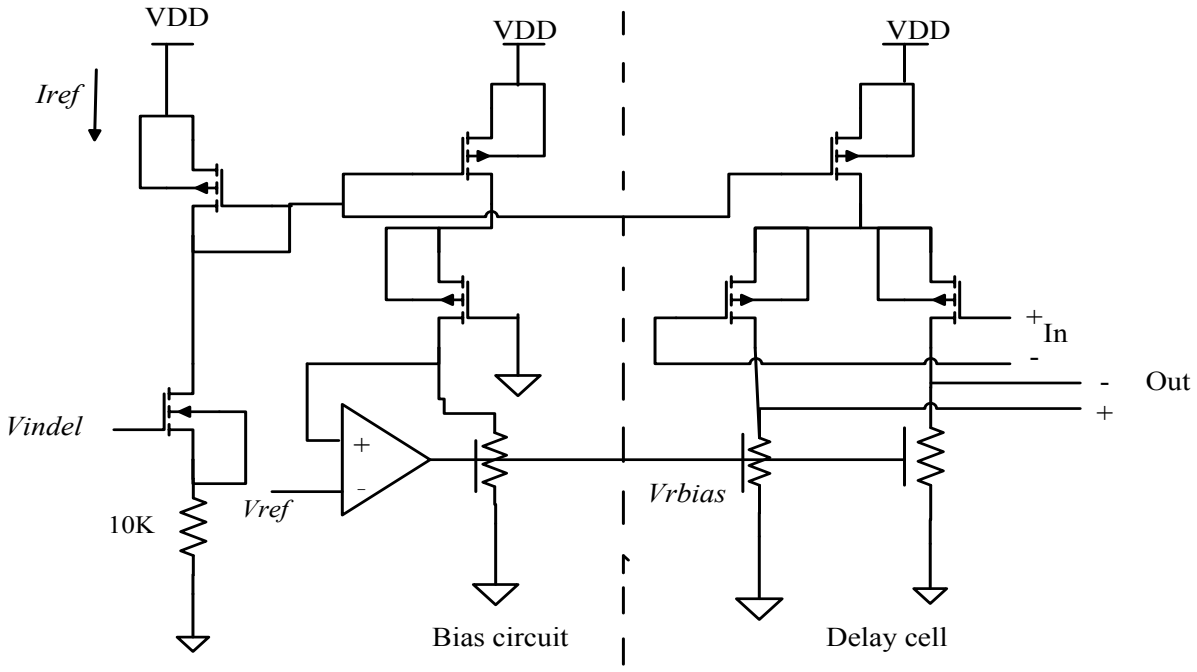


Fig. 5.4 A differential delay element based on a voltage-controlled resistor[8]

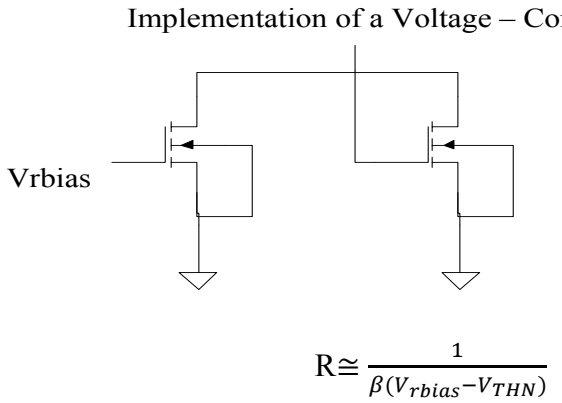


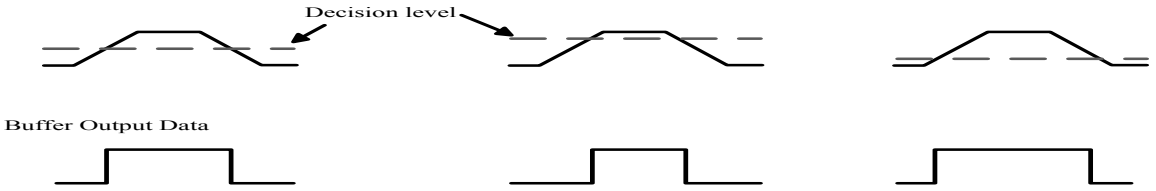
Fig. 5.5 Voltage controlled resistor [8]

The reason behind this circuit is labeled practical can be understood by first considering what happens where there is noise on VDD. The noise causes a voltage variation across the PMOS current source, MPC. Ideally, this doesn't change the current through MPC. The output voltages of the delay cell swing between V_{REF} and ground. If there is noise on ground it feeds equally into each output. This common mode noise is then, ideally rejected by the differential amplification action of the next stage. The bias circuit is a replica of the delay stage and is used to bias the

delay elements so that the output voltage swing is up to V_{REF} when the gate of one of the PMOS switches is at ground.

5.4 INPUT BUFFER

Input Buffers are used to convert a chip's imperfect signals into clean digital signal for on chip use. Here imperfect signals refer to signals that can have too slow or too high, rise and fall times. If the buffer doesn't "slice" the data in the correct position, timing errors can occur. For example, consider the waveforms seen in the fig. 5.6. If the input signal is sliced too high or too low, the output signal's width is incorrect. In high speed system this can result in errors.



(a) Correct pulse width (b) Decision level too high (c) Decision level too high

Fig. 5.6 Timing errors in regenerating digital data [8]

In order to slice input data precisely, a reference voltage may be used. In this case a differential amplifier is required. A differential amplifier input buffer amplifies the difference between its two inputs. Let one of the input of the difference amplifier is tied at fixed d.c. voltage. When the other input goes either above the d.c. level or below the d.c. level, the output of the buffer changes states as

$$V_{inp} > V_{inm} \longrightarrow \text{out='1'} \quad \text{.....(5.2)}$$

$$V_{inp} < V_{inm} \longrightarrow \text{out='0'} \quad \text{.....(5.3)}$$

The circuit shown in above fig. 5.7 is self-biased because no external references are required here to fix the current in the circuit. When V_{inp} is greater than V_{inm} , the current in M2 is greater than current in M1($V_{GS}2 > V_{GS}1$). The current in M1 flows through M3 and is mirrored by M4 and so M4's current is less than M2's current. This causes the differential amplifier's output diverted towards ground until the current in M2 becomes equal to the current in M4. An inverter is connected at the output of the differential amplifier which gives high output. This PMOS input buffer is used as differential amplifier in bias circuit of the differential delay cell VCDL

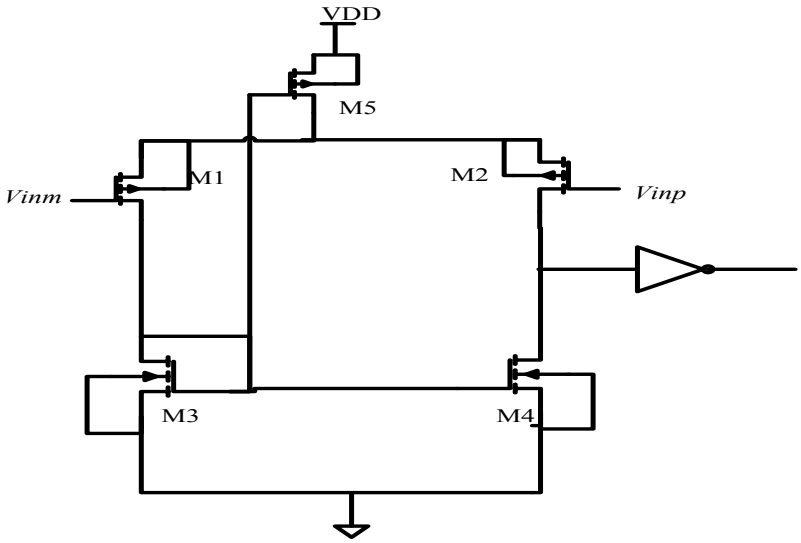


Fig. 5.7 A PMOS input buffer [8].

The transient response for the buffer above in fig. 5.7, is shown in fig. 5.8.

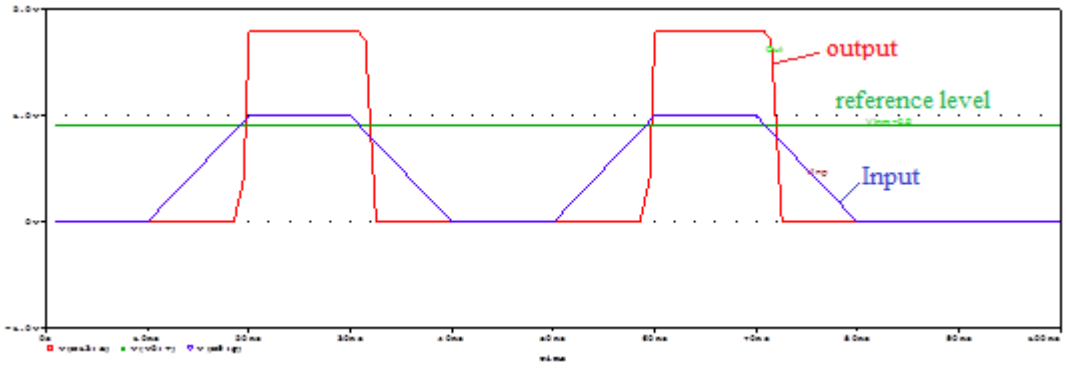


Fig. 5.8 Transient response for buffer shown in fig. 5.7.

In above simulation we can see high output is obtained whenever V_{inp} crosses the level of reference d.c. voltage (V_{inm}), otherwise the output remains low.

5.5 IMPLEMENTATION OF VCDL WITH EIGHT STAGE DELAY ELEMENTS

The main concerns involve in the realization of a practical VCDL are effects due to noise, power supply and temperature variations. Hence the designing approach for realization of a practical VCDL should involve such arrangements which minimize the effects of above concerns.

5.5.1 V to I converter and Bias Circuit

Figure 5.9 depicts the combination of voltage to current converter and bias circuit, where first stage is voltage to current converter and second stage is bias circuit. This combination is used to generate biasing voltages $V_{outfilter}$ and V_{rbias} for delay cells.

The important concerns involved in this circuit are:

1) The linearity should be maintained between V_{indel} Voltage and the generated I_{ref}.
2) The sensitivity of I_{ref} with respect to changes in V_{DD}, and
3) Well regulation of node n2 to v_{ref} by the amplifier

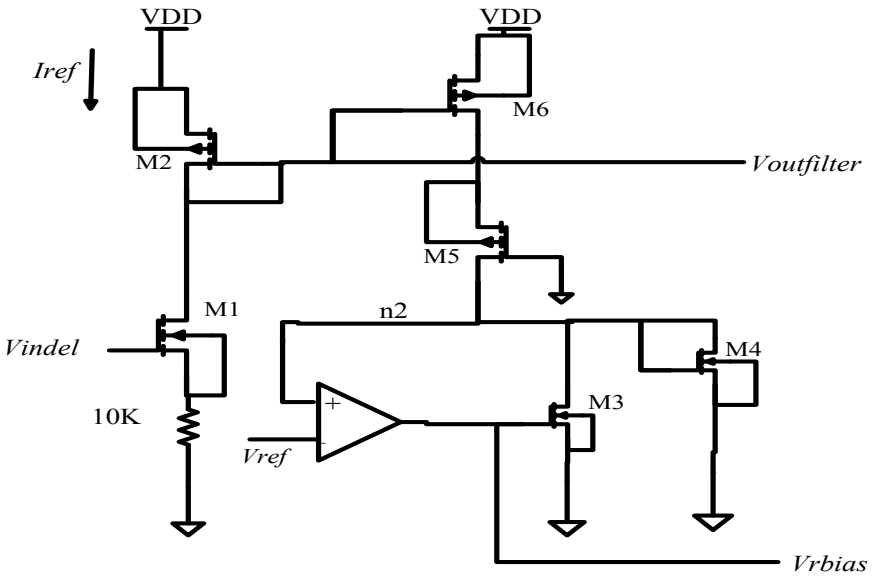


Fig. 5.9 V to I converter and Bias circuit [8]

5.5.2 Performance of V to I converter and Bias Circuit

Simulation results of fig. 5.10 and fig. 5.11 show the performance of the circuit of fig. 5.9. In fig. 5.10 the variation in current I_{ref} against V_{indel} and V_{DD} has been plotted.

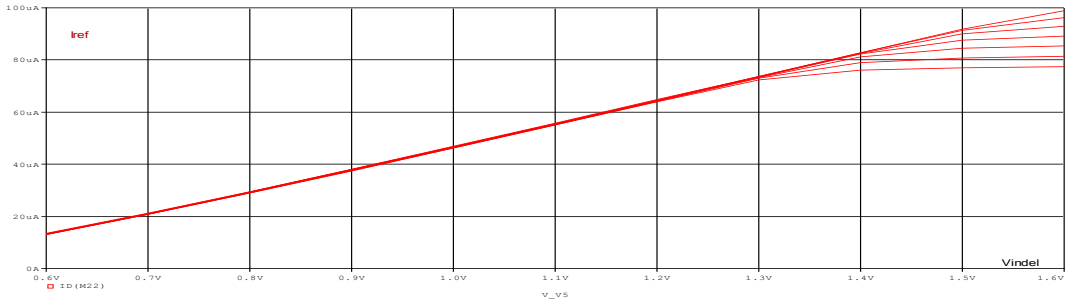


Fig. 5.10 Plot of I_{ref} vs. V_{indel} and V_{DD}

In above simulation the supply voltage changes from 1.6 to 2 V and V_{indel} from 0.6 to 1.6V. Till 1.3 V of V_{indel} , I_{ref} maintains almost same values for change in supply voltage. After 1.3 V of V_{indel} , I_{ref} increases in accordance with supply voltage.

In fig. 5.12 simulation for V_{rbias} and node voltage n2 against variation in V_{indel} and V_{DD} has been shown.

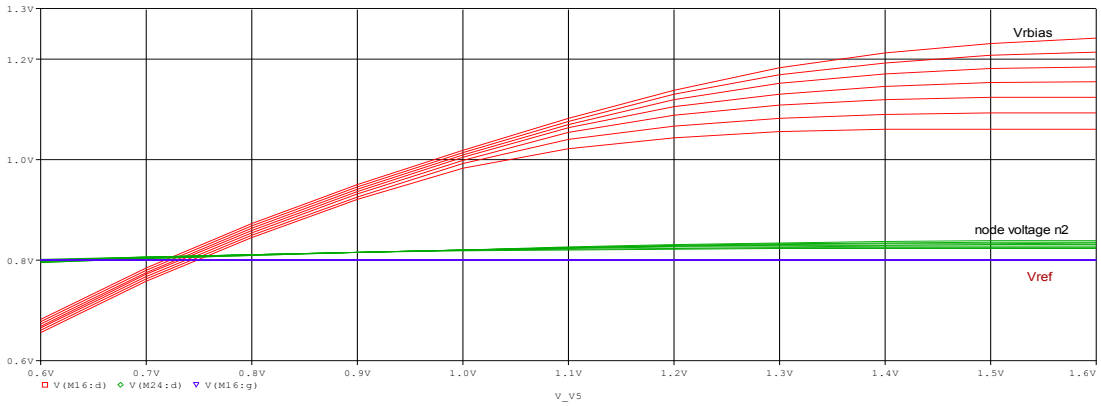


Fig. 5.11 Plot of V_{rbias} and node voltage n2 vs. V_{indel} and V_{DD}

In above simulation it can be observed the node voltage n2 almost follows the V_{ref} =0.9 V. The nature of this voltage is not affected by change in supply voltage. From simulation results of fig. 5.10 and 5.11 we can conclude that the functioning of the circuit shown in fig. 5.9 is not affected too much by variation in supply voltage.

5.5.3 An eight stage VCDL

Figure 5.12 is depicting an eight stage VCDL. The architecture of delay cells in VCDL has been implemented by using current mode logic and voltage – controlled resistors. Current mode logic implies that the current is constant and doesn't depend on the output whether it is static and switching. The amount of current is fixed by the uppermost PMOS, and by fixing the current it also determines the delay. The other two PMOS devices receive differential input to steer the current through one leg or the other. The uppermost PMOS has been sized up by factor of 10 as compare to the other two PMOS. The NMOS pairs in each leg have been implemented as voltage – controlled resistors and these are used to set maximum output swing. The precise bias voltage is fixed by differential amplifier keeping its positive terminal connected with the output of half replica delay line and inverting terminal at V_{ref} for those NMOS so that maximum output swing will be limited to V_{ref} regardless of control voltage V_{indel} . Limiting the output swing below V_{DD}

reduces power supply sensitivity. The differential signals provide common – mode noise rejection [9].

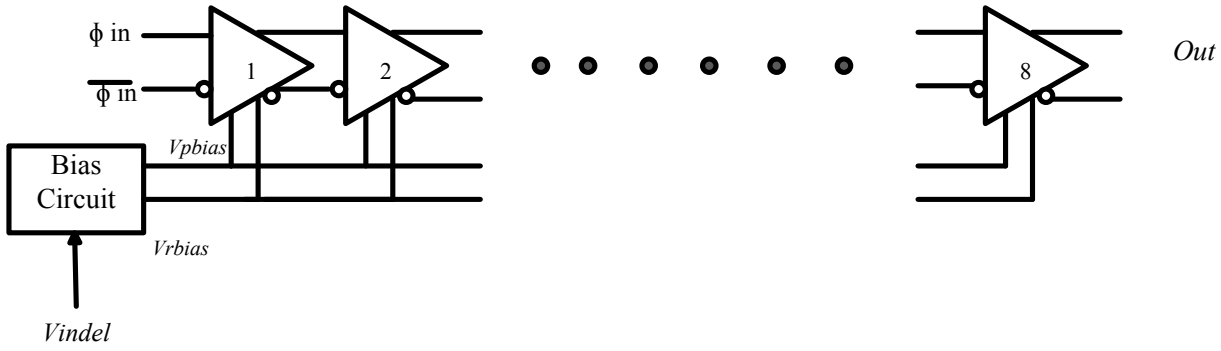


Fig. 5.12 An eight stage VCDL [8]

In above fig. 5.12 VCDL has been realized by cascading eight differential delay cells denoted by numbers. The architecture of this differential delay cell has been depicted further in fig. 5.13.

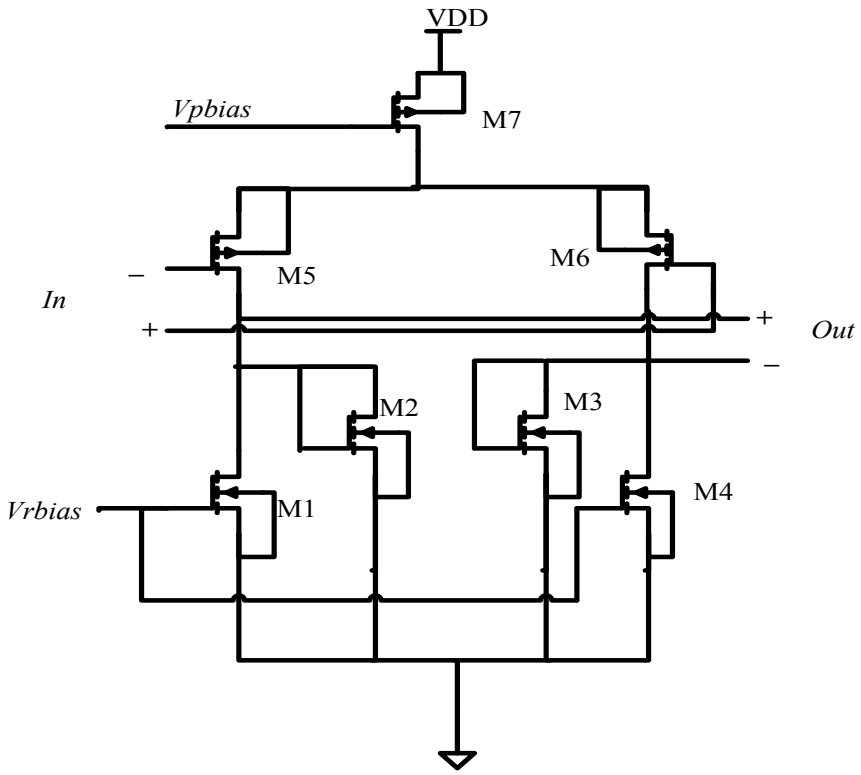


Fig. 5.13 Differential Delay Cell [8]

The simulation of fig. 5.12 circuit has been shown in fig. 5.14. In this simulation control voltage V_{indel} was fixed at $V_{DD}/2$ while the differential inputs were generated using an inverter and a

transmission gate in order to equalize the delays that the inputs signal seen to the VCDL. The outputs only swing up to V_{REF} (=0.9v).

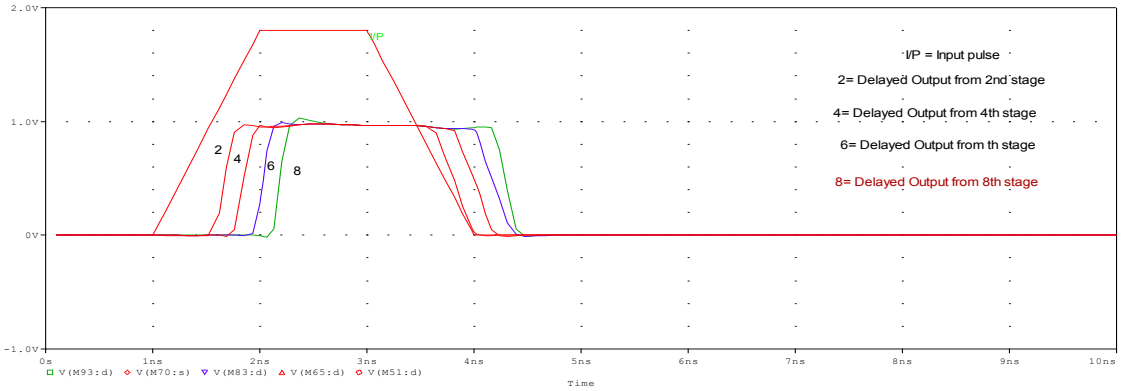


Fig. 5.14 simulation of VCDL of fig. 5.12

As the reference voltage was fixed at 0.9 volts, because of this the output voltage swings up to this value. An important concern here is how we regenerate the full logic levels at the outputs without introducing skew in this signal. Here some modification is required which has been shown in fig. 5.15. The PMOS input buffer, discussed in section 5.4 has been modified to realize as differential amplifier shown in fig. 5.16, has been connected here as last stage of VCDL. The modification is, we have connected the gate of the uppermost PMOS with *Vpbias* in order to have same amount of current that the subsequent delay stages have. The delay provided by this amplifier is less as compared to other stages of VCDL, but it is not an important concern as we are passing the output of differential amplifier through an inverter which also adds some delay. Simulation result of fig. 5.17 verifies that the differential amplifier is capable to generate full logic levels at the output of VCDL.

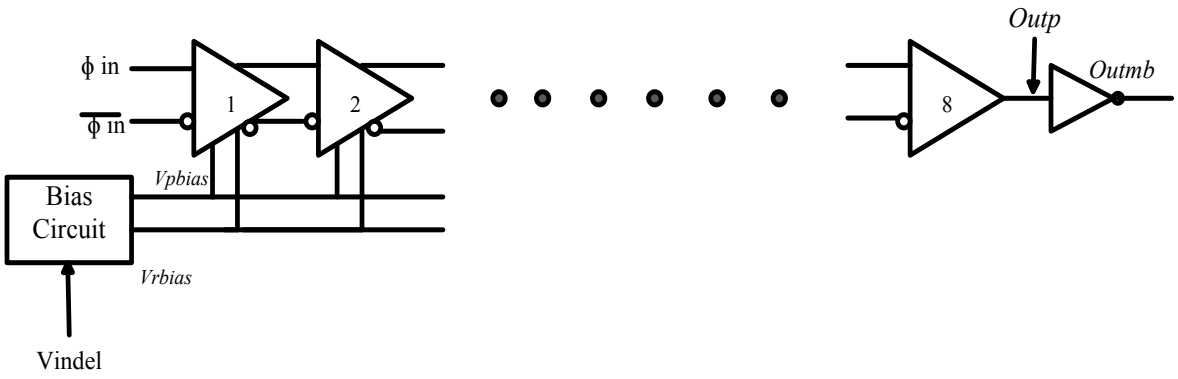


Fig. 5.15 Modified VCDL to generate full output logic levels [8]

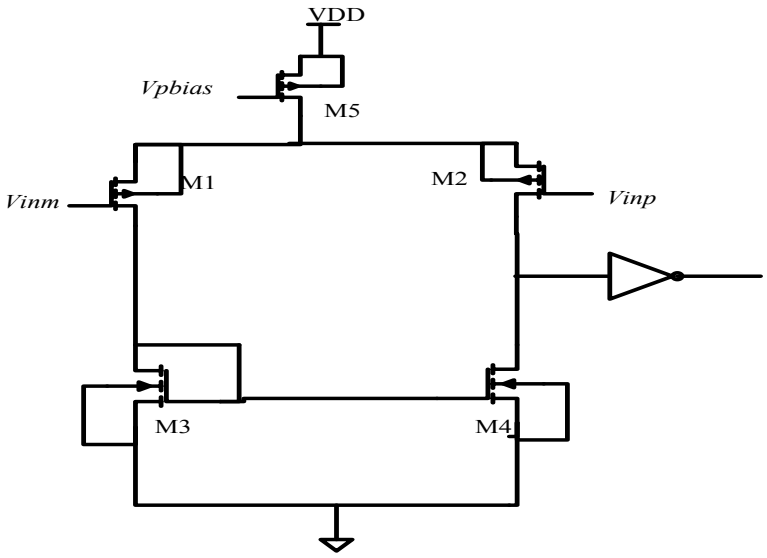


Fig. 5.16 Differential Amplifier used to generate full logic levels at output of the VCDL [8]

The simulation result for circuit 5.15 has been shown in fig. 5.17.

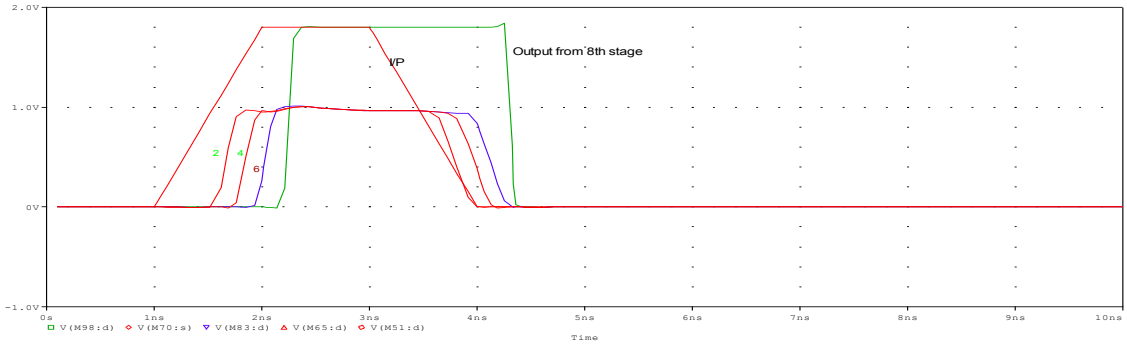


Fig. 5.17 Simulation for VCDL of fig.5.15

Enhanced performance of this VCDL comes with expense of chip area because each delay stage require more numbers of transistors as compare to current starving inverter delay cell. Since current is constant in this architecture the power consumption is also more. Apart from these the level converter is also require at the last stage of VCDL.

5.6 TRANSFER CHARACTERISTIC OF VCDL

The transfer characteristic of a VCDL is plotted between control voltage V_{indel} and generated delay by VCDL. It shows the relation between generated delay and the Control voltage.

5.6.1 Transfer Characteristic of RC delay cell VCDL

Different values of delays generated by RC delay cell VCDL shown in fig. 5.2(b) upon changing the value of V_{indel} , have been tabulated below

Table 5.1. Values of Delay with change in V_{indel} of RC delay cell VCDL

Serial No.	V_{indel} (in Volts)	Delay in RC delay cell VCDL (ps)
1.	0.4	363
2.	0.5	357
3.	0.6	360
4.	0.7	358.7
5.	0.8	366.3
6.	0.9	365.6
7.	1.0	369.2
8.	1.1	377.7
9.	1.2	384.0
10.	1.3	390.1
11.	1.4	399.4
12.	1.5	400.9
13.	1.6	409.7
14.	1.7	417.3
15.	1.8	418.9

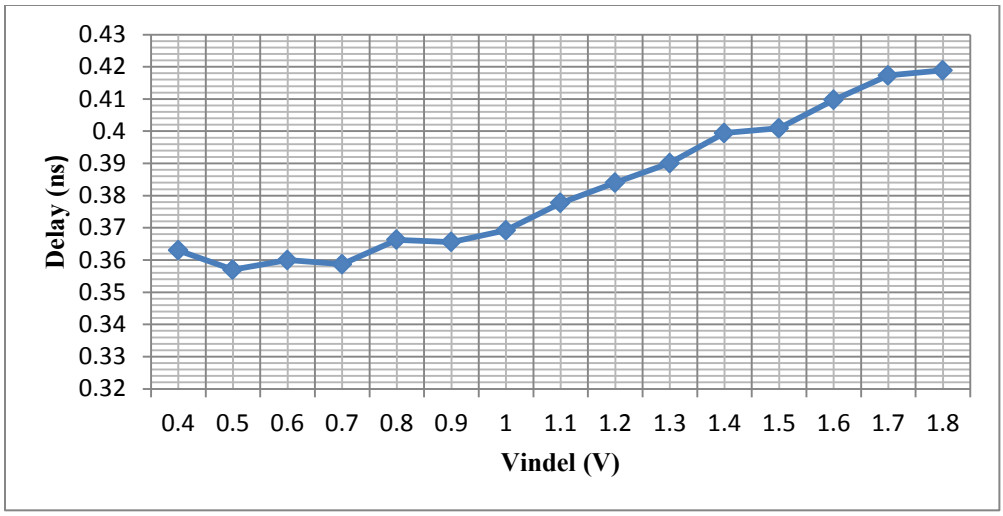


Fig. 5.18 Plot between Delay and V_{indel} of RC delay cell VCDL

To realize RC delay cell VCDL the device dimensions have been chosen as W_p =3600 nm, W_n =1800 nm and $L_p = L_n$ = 180 nm.

5.6.2 Transfer Characteristic of CSI delay cell VCDL

Different values of delays generated by Current Starving Inverter (CSI) delay cell VCDL shown in fig. 5.2(d) upon changing the value of V_{indel}, have been tabulated below

Table 5.2. Values of Delay with change in V_{indel} of CSI delay cell VCDL

Serial No.	V_{indel} (in Volts)	Delay in CSI VCDL (ns)
1.	0.7	25.487
2.	0.8	14.662
3.	0.9	10.472
4.	1.0	8.734
5.	1.1	7.675
6.	1.2	7.204
7.	1.3	6.98
8.	1.4	6.732
9.	1.5	6.687
10.	1.6	6.427
11.	1.7	6.463
12.	1.8	6.315

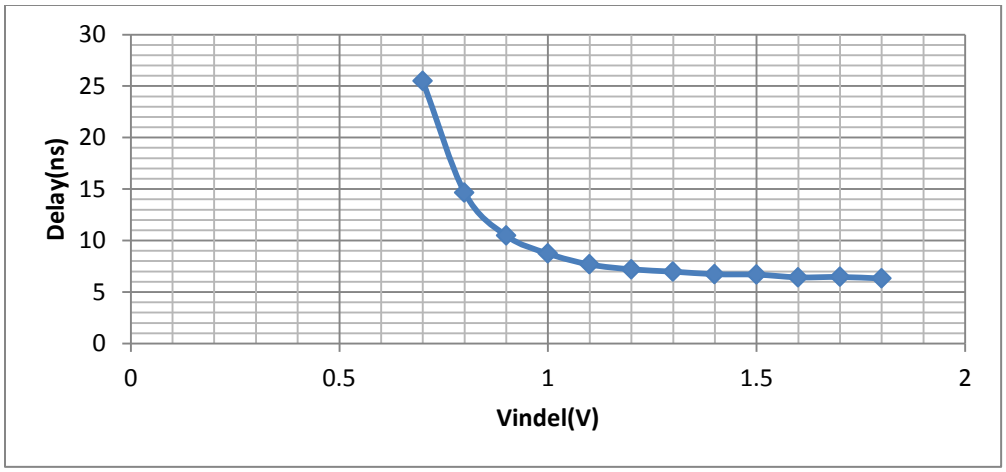


Fig. 5.19 Plot between Delay and V_{indel} of CSI delay cell VCDL

To realize RC delay cell VCDL the device dimensions have been chosen as W_p =32000 nm, W_n =16000 nm and $L_p = L_n$ = 1000 nm [9].

5.6.3 Transfer Characteristic of differential delay cell VCDL

Different values of delays generated by differential delay cell VCDL upon changing the value of V_{indel} , have been tabulated below

Table 5.3. Values of Delay with change in V_{indel} of differential delay cell VCDL

Serial No.	V_{indel} (in Volts)	Delay (in Pico seconds)
1.	0.7	753.8
2.	0.8	535.5
3.	0.9	437.7
4.	1.0	383.0
5.	1.1	346.1
6.	1.2	321.8
7.	1.3	307.3
8.	1.4	286.3
9.	1.5	274.9
10.	1.6	273.5
11.	1.7	273.1
12.	1.8	272.9

From table 5.3 it is evident that the minimum and maximum delays which can be generated by this differential VCDL are 272.9 ps and 753.8ps respectively. Hence Lock – in range of this VCDL would be 1.32 GHz to 3.66GHz.

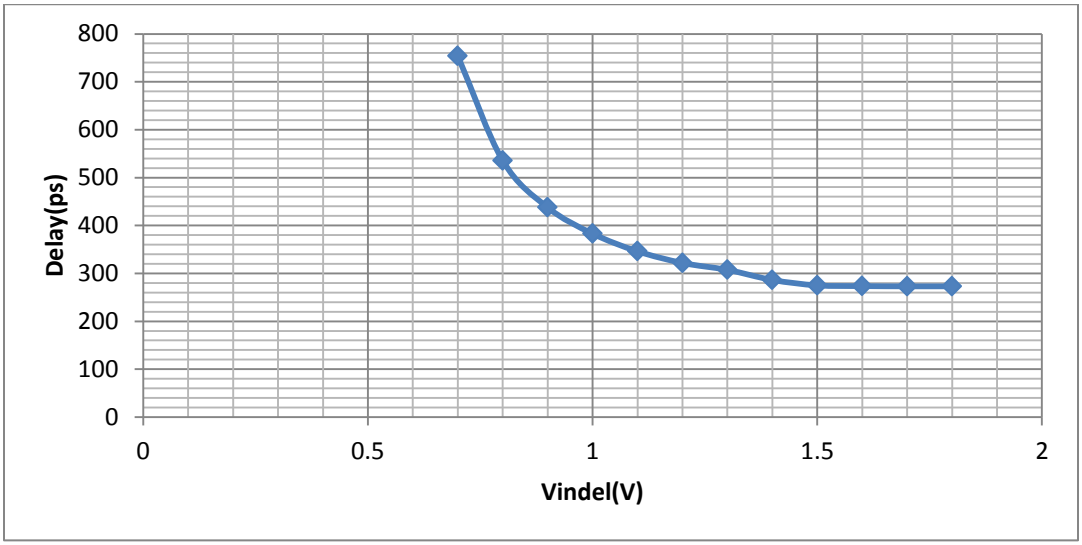


Fig. 5.20 Plot between Delay and V_{indel} of differential delay cell VCDL

CHAPTER 6

RESULTS AND DISCUSSIONS

In previous chapters, the functioning of individual blocks of DLL has been described and the operation has also been verified. These blocks are connected together to realize a DLL. In this chapter all performance specifications of finally realized DLL have to be measured.

6.1 SIMULATION CIRCUIT

Figure 6.1 shows the circuit diagram of DLL that we have realized in our work (current mirrors have not been shown that are used to provide bias voltages in CP). This circuit of DLL is simulated to find out its locking range, lock- in time and to verify its jitter performance. All the realizations is performed in 180 nanometer CMOS process technology and simulated in OrCAD PSPICE 16.5 simulation tool.

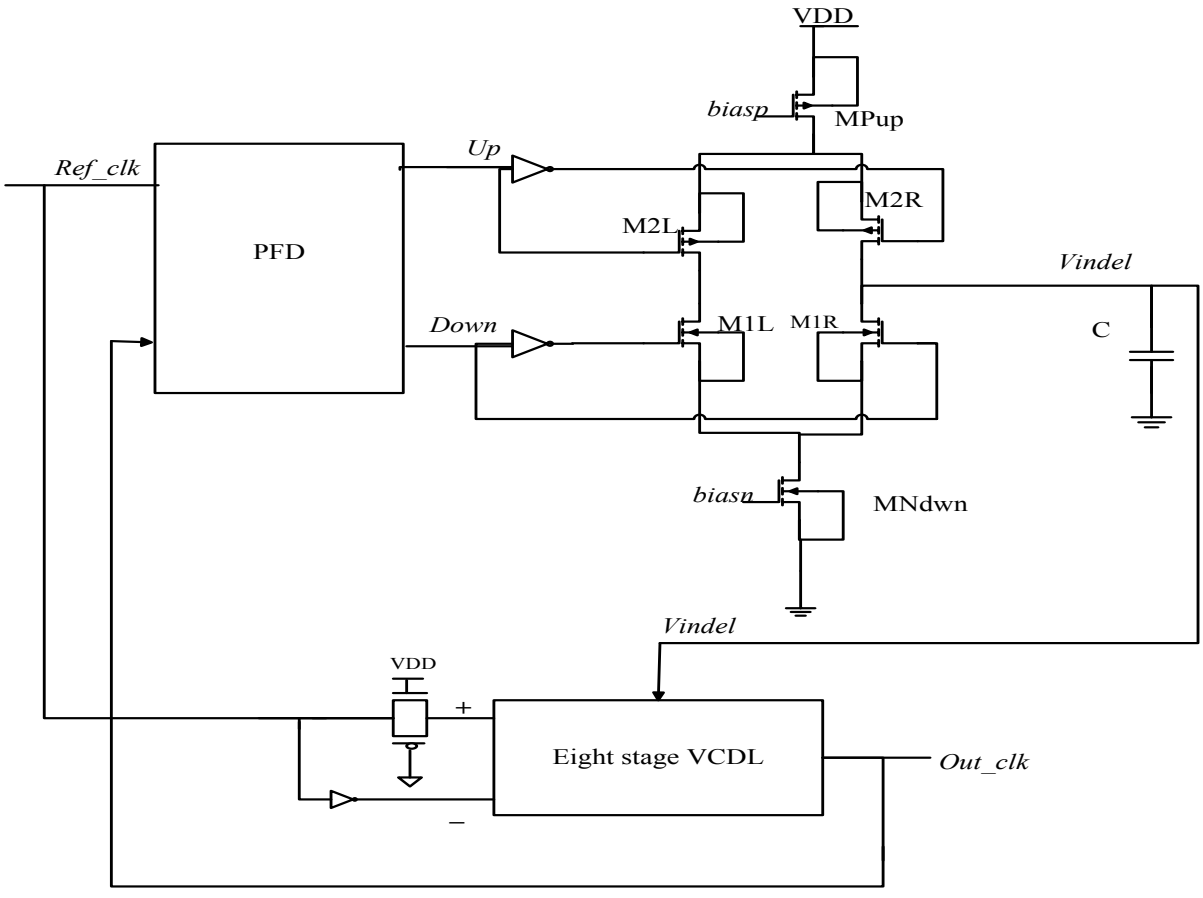


Fig. 6.1 Complete DLL circuit for simulation [7]

6.2 OPERATION OF THE DLL

In following simulations, from fig. 6.2 to fig. 6.9 we have used the DLL shown in fig. 6.1 with all the four types of PFDs.

6.2.1 DLL with CMOS – PFD

The CMOS – PFD that has been depicted in fig.3.8 is connected with CP and an eight stage VCDL to realize the DLL here. Simulations shown in fig. 6.2 and 6.3 are depicting the lock – in process and lock – in time for this DLL at 1.33 GHz and 2.85 GHz.

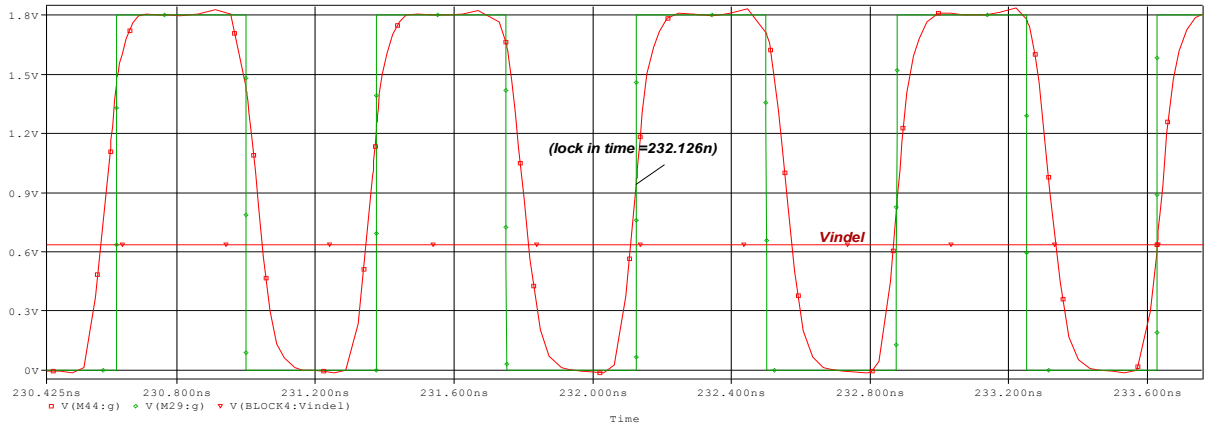


Fig. 6.2 Locking of DLL with CMOS – PFD at 1.33GHz

From the result of above simulation shown in fig. 6.2, we observed that the lock-in-time is 232.126ns. The value of capacitor was taken 4pF.

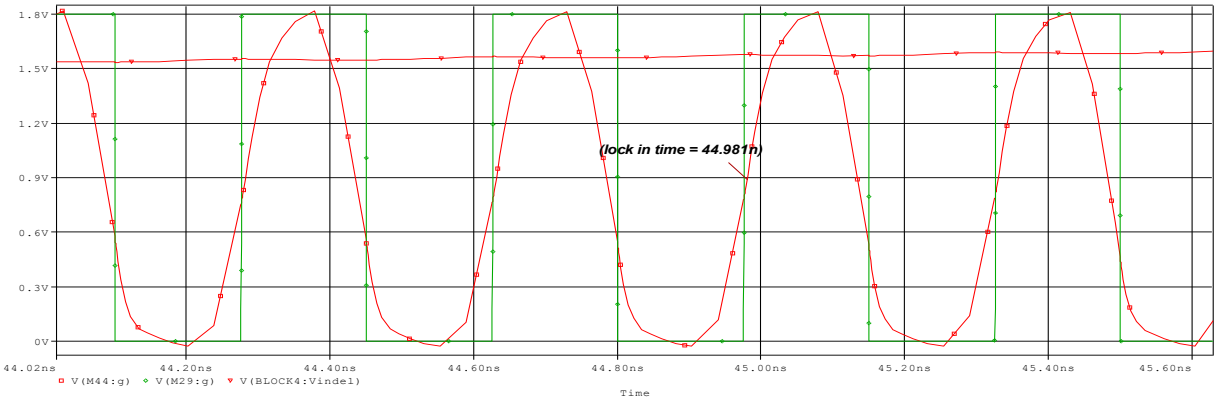


Fig. 6.3 Locking of DLL with CMOS – PFD at 2.85 GHz

From the result of above simulation shown in fig. 6.3, we observed that the lock-in-time is 44.98ns. The value of capacitor was taken 0.2pF.

6.2.2 DLL with type-1 PFD

The CMOS – PFD that has been depicted in fig.3.13 is connected with CP and an eight stage VCDL to realize the DLL here. Simulations shown in fig. 6.4 and 6.5 are depicting the lock – in time for this DLL at 1.8GHz and 2.08 GHz.

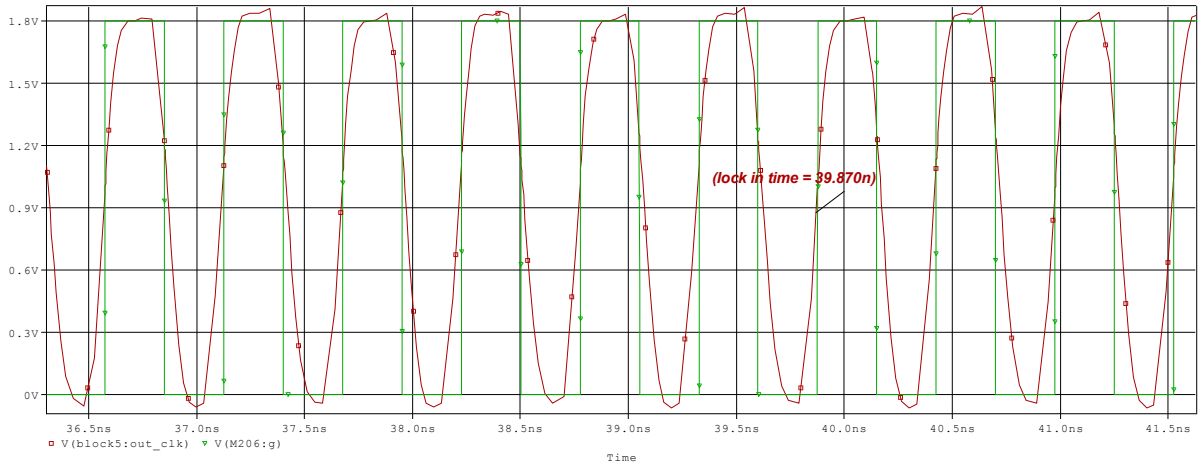


Fig. 6.4 Locking of DLL with PFD type-1 at 1.8 GHz

From the result of above simulation shown in fig. 6.4, we observed that the lock-in-time is 39.87ns. The value of capacitor was taken 1pF.

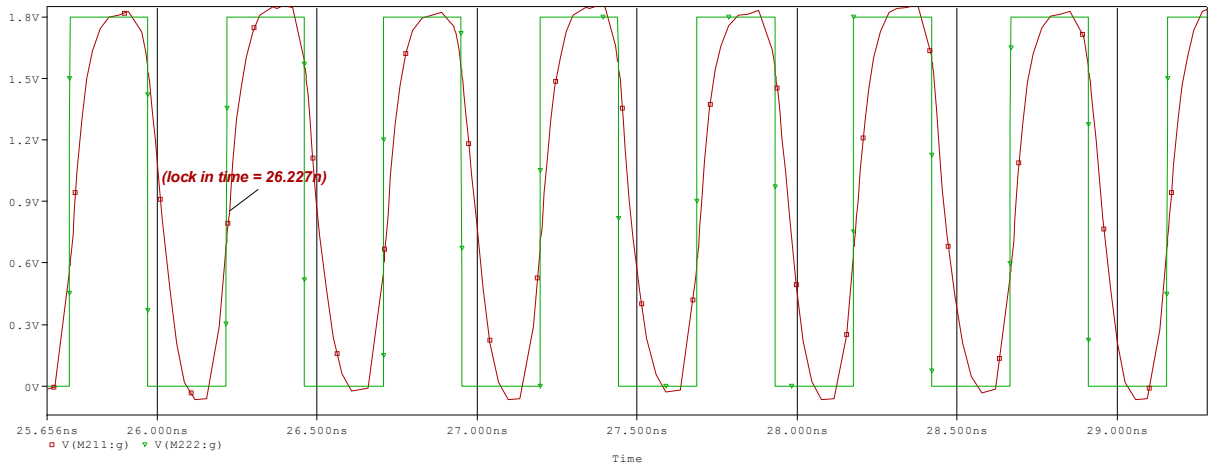


Fig. 6.5 Locking of DLL with PFD type-1 at 2.08 GHz

From the result of above simulation shown in fig. 6.5, we observed that the lock-in-time is 26.227ns. The value of capacitor was taken 0.1pF.

6.2.3 DLL with type-2 PFD

The C^2MOS – PFD type 2 that has been depicted in fig.3.14 is connected with CP and an eight stage VCDL to realize the DLL here. Simulations shown in fig. 6.6 and 6.7are depicting the lock – in time for this DLL at 1.4GHz and 2.2 GHz.

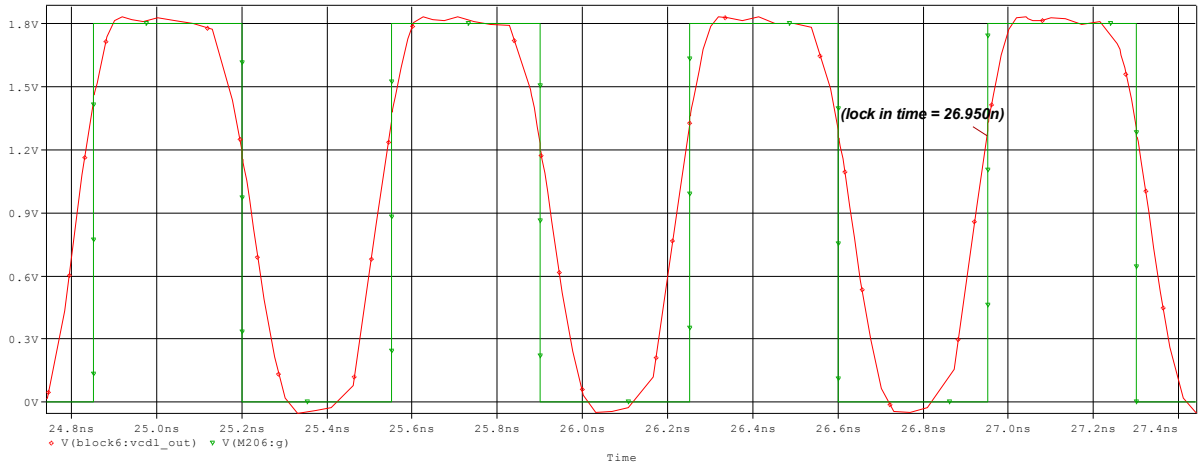


Fig. 6.6 Locking of DLL with PFD type-2 at 1.4 GHz

From the result of above simulation shown in fig. 6.6, we observed that the lock-in-time is 26.95ns. The value of capacitor was taken 1pF.

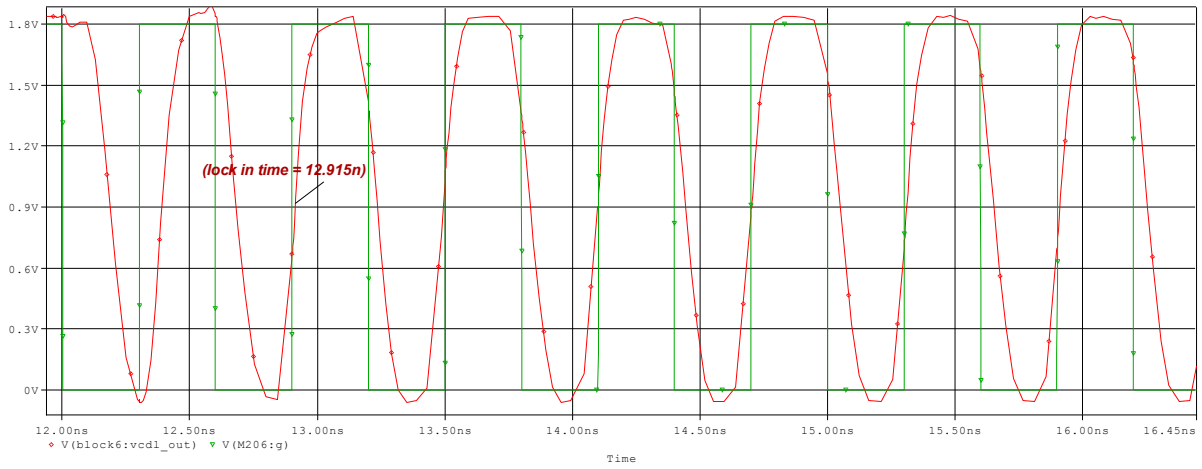


Fig. 6.7 Locking of DLL with PFD type-2 at 2.2 GHz

From the result of above simulation shown in fig. 6.7, we observed that the lock-in-time is 26.95ns. The value of capacitor was taken 1pF.

6.2.4 DLL with type-3 PFD

The C^2MOS – PFD type 2 that has been depicted in fig.3.15 is connected with CP and an eight stage VCDL to realize the DLL here. Simulations shown in fig. 6.8 and 6.9 are depicting the lock – in process and lock – in time for this DLL at 1.4 GHz and 2.2 GHz.

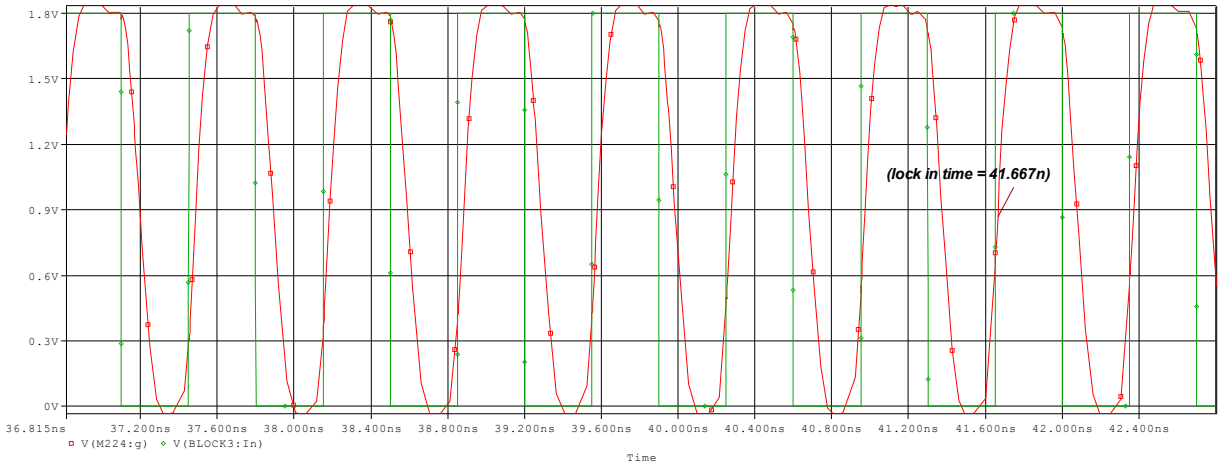


Fig. 6.8 Locking of DLL with PFD type-3 at 1.4 GHz

From the result of above simulation shown in fig. 6.8, we observed that the lock-in-time is 41.667ns. The value of capacitor was taken 1pF.

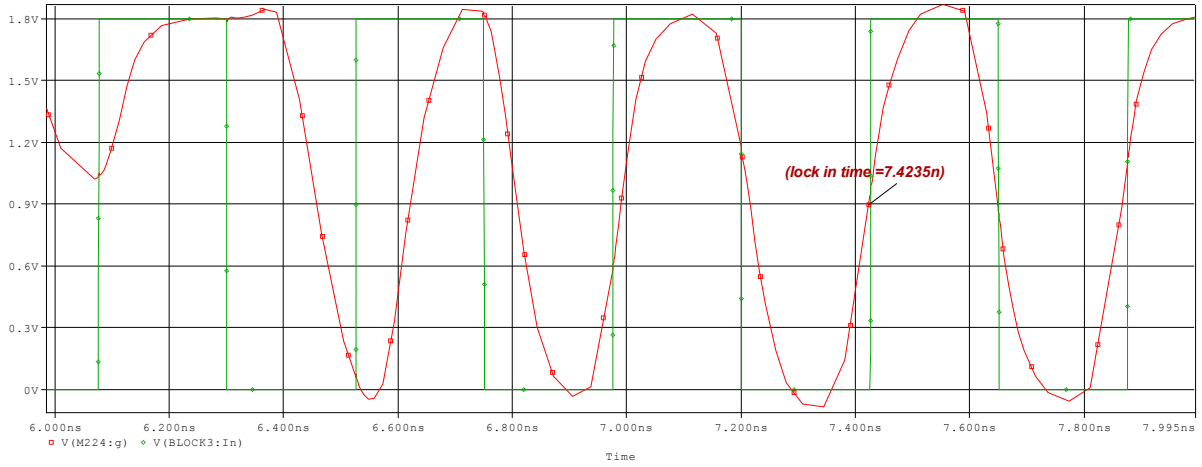


Fig. 6.9 Locking of DLL with PFD type-3 at 2.2 GHz

From the result of above simulation shown in fig. 6.9, we observed that the lock-in-time is 41.667ns. The value of capacitor was taken 0.1pF.

6.3 JITTER PERFORMANCE

One of the most important performances metric for a DLL is jitter. Jitter is measured by using eye diagram plot [13],[27]. The fig. 6.10 to 6.13, are depicting the eye diagram plots of the DLLs. Width of rising edge at $V_{DD}/2$ in this plot is measure of jitter. Jitter has been calculated at 2GHz frequency for all four DLLs.

6.3.1 Jitter in DLL with CMOS-PFD

Figure 6.10 depicts the eye diagram plot for DLL with CMOS-PFD.

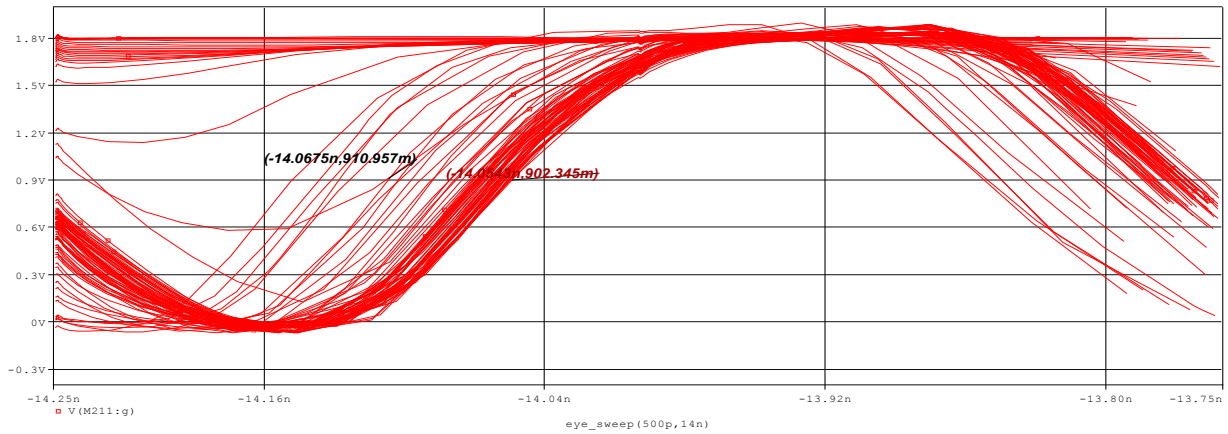


Fig. 6.10 Eye diagram plot for DLL with CMOS-PFD

From the result of above simulation shown in fig. 6.10, we observed that the width of rising edge at 0.9V is 13ps. Hence the value of jitter is 13ps.

6.3.2 Jitter in DLL with PFD type-1

Figure 6.11 depicts the eye diagram plot for DLL with PFD type-1.

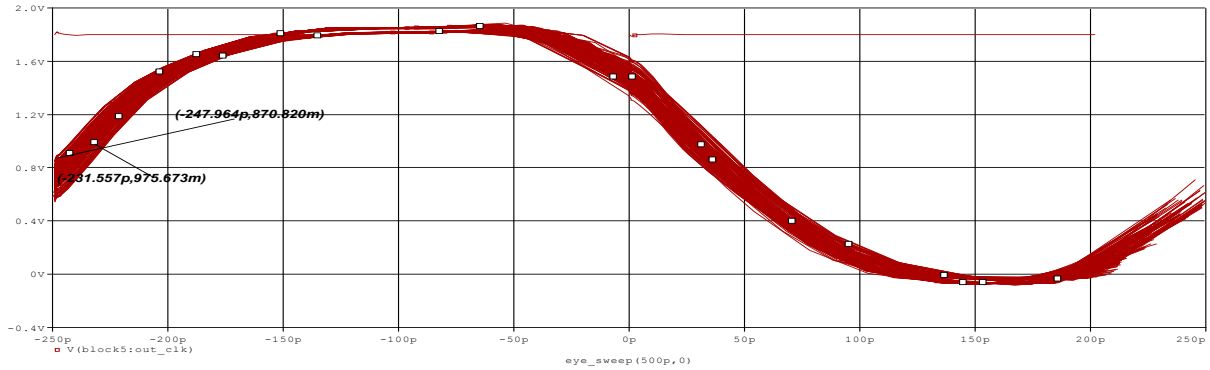


Fig. 6.11 Eye diagram plot for DLL with PFD type-1

From the result of simulation shown in fig. 6.11, we observed that the width of rising edge at 0.9V is 16.4ps. Hence the value of jitter is 16.4ps.

6.3.3 Jitter in DLL with PFD type-2

Figure 6.12 depicts the eye diagram plot for DLL with PFD type-2

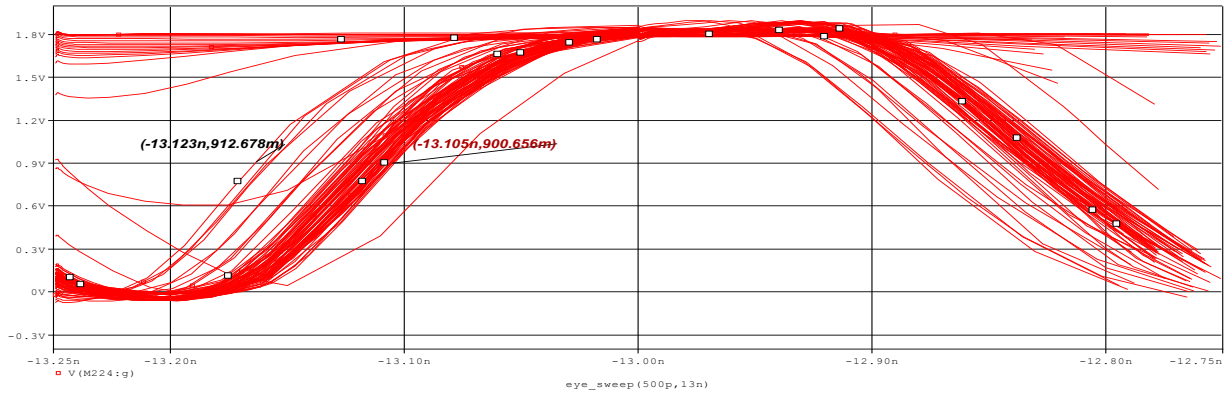


Fig. 6.12 Eye diagram plot for DLL with PFD type-2

From the result of above simulation shown in fig. 6.12, we observed that the width of rising edge at 0.9V is 18ps. Hence the value of jitter is 18ps.

6.3.4 Jitter in DLL with PFD type-3

Figure 6.13 depicts the eye diagram plot for DLL with PFD type-3

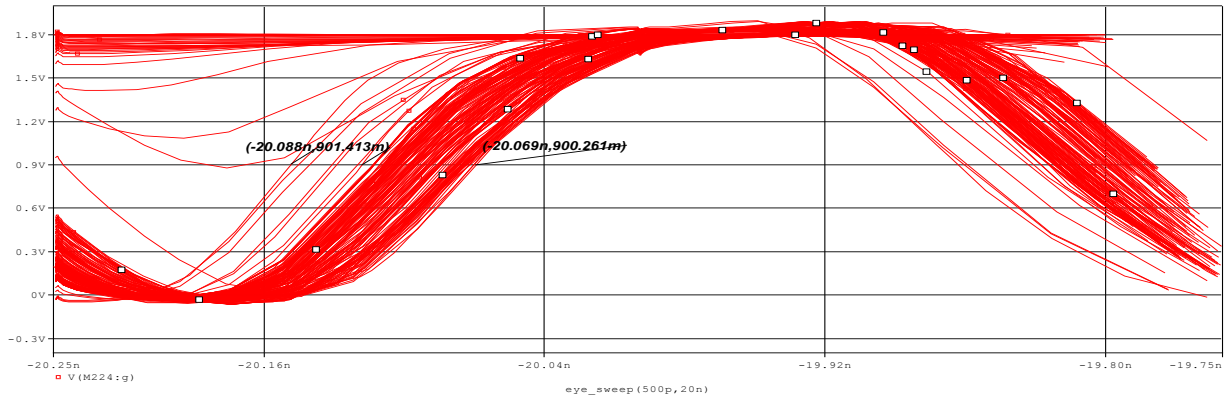


Fig. 6.13 Eye diagram plot for DLL with PFD type-3

From the result of simulation shown in fig. 6.13, we observed that the width of rising edge at 0.9V is 18ps. Hence the value of jitter is 18ps.

6.4 PERFORMANCE SUMMARY AND COMPARISONS

All the performance parameters of DLLs are tabulated in following table.

Table 6.1 Performance Summary and Comparisons

Sr. No.	Performance Parameters	DLL with CMOS – PFD	DLL with PFD type-1	DLL with PFD type-2	DLL with PFD type-3
1.	Supply	1.8 V	1.8 V	1.8 V	1.8 V
2.	Lock-in time at minimum Operating Frequency	232.12ns At 1.33GHz	39.87ns at 1.8GHz	26.95ns at 1.42GHz	41.667 At 1.42GHz
3.	Lock-in time at maximum Operating Frequency	45ns at 2.85 GHz	26.227ns at 2.08GHz	12.95ns at 2.22GHz	7.4235ns at 2.22GHz
4.	Jitter (at 2GHz)	13ps	16.4ps	18ps	18ps
5.	Power Dissipation (at 2GHz)	3.06×10^{-4} W	5.14 mW	2.76mW	2.17 mW

6.5 DISCUSSION

From table 4, it can be observed that the DLL with CMOS-PFD shows wider range of maximum as well as minimum operating frequencies but time lock – in time is more as compared to other DLLs listed in the table, which represents its slow nature. At higher frequencies all the DLLs require less time to lock the input reference clock meanwhile duty cycle of output gets affected. The capacitor, plays important role in deciding the lock in time as charging and discharging occurs in it through charge pump, controls the delay of the VCDL. At the locked state, voltage across capacitor becomes constant. The capacitors of less value take less time for charging and discharging, so it can lock high frequency clocks quickly but can't minimize the ripples across it which leads to high jitter at the output. Capacitors of large values can be employed to reduce the jitter but it requires long time for charging and discharging. So in this case lock time increases by slowing down the speed of the DLL. On the other hand, the value of current in charge pump can also be increased for faster locking but high values of current lead to high power dissipation. Hence there are some trade-offs which need to be considered carefully while choosing the values of current in charge pump and capacitors.. All the DLLs showed good jitter performance.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

In this report we have discussed the working mechanism and designing of each component of DLL such as PFD, CP, LF and VCDL. At the end we realized the DLL by assembling these components. Realization of PFD is discussed employing C^2MOS – Registers as D flip Flops. Two another variants of this PFD structure are designed by changing their reset path. By performing this operation we saved some units of transistors and compared the performances of CMOS-PFD and C^2MOS – Registers based PFDs with the help of simulations in OrCAD - PSPICE. The results of these comparisons enabled us to utilize this C^2MOS-PFD in DLL realization. The roles of CP and LF in locking of reference input clock and output clock, are also discussed in this report. We found some trade-offs in choosing the values of current in CP and capacitor in LF as large currents can speed up the locking of the clock but at the cost of power dissipation however, capacitors of less value can lock high frequency clocks but at the cost of poor jitter performance. In our designing we tried to maintain both the requirements of good speed (for locking) as well as good jitter performance in DLL.

Different types of delay cells have been discussed such as RC delay cell, Current Starving Delay Cell and Differential delay cell. We chose differential delay cell for realization of VCDL due to its robustness against power and temperature variations. Current Mode Logic has been utilized for realization of differential delay cells. The design of VCDL discussed in this report can produce the delay in the range of picoseconds which enables the DLL to lock the clock frequencies in the range of GHz.

In this report, the design of DLL showed the characteristics of good jitter performance at its maximum operating frequency range. We also found that our designed DLLs with C^2MOS-PFD are faster to lock the clock signals as compared to DLL with CMOS-PFD in their operating frequency range.The primary focus of this report is to study the scope of DLL as clock synchronizers in modern day clock distribution circuits in digital communication systems.

The duty cycle of output clock is affected during higher frequencies of operation, this is another concern which needed to be addressed. The current mode logic is utilized to design differential delay cell, where delay cell can have good immunity to power supply variations. It requires transistors with large width. A transistor with large width possesses the problems of

wide area for fabrication. Since wide transistors draw large currents the power dissipation of the circuit will be also high. So to overcome the problem of large current and area, an alternative method or design of delay cell can be explored as a futuristic addition to this work.

REFERENCES:

1. Yongsam Moon; Jongsang Choi; Kyeongho Lee; Deog-Kyoon Jeong; Min-Kyu Kim, "A 62.5-250 MHz multi-phase delay-locked loop using a replica delay line with triply controlled delay cells," in *Custom Integrated Circuits, 1999. Proceedings of the IEEE 1999* , vol., no., pp.299-302, 1999.
2. Hsiang-Hui Chang; Jyh-Woei Lin; Shen-Iuan Liu, "A wide-range and fixed latency of one clock cycle delay-locked loop," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on* , vol.3, no., pp.675-678, 2002.
3. Hsun-Hsiang Chen; Zih-Hsiang Wong; Shen-Li Chen, "Design of delay-locked loop for wide frequency locking range," in *SoC Design Conference (ISOCC), 2013 International* , vol., no., pp.302-305, 17-19 Nov. 2013..
4. Kuo-Hsing Cheng; Yu-Lung Lo; Wen-Fang Yu; Shu-Yin Hung, "A mixed-mode delay-locked loop for wide-range operation and multiphase clock generation," in *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on* , vol., no. ,pp.90-93 , 30 June-2July 2003.
5. Chulwoo Kim; In Chul Hwang; Sung-Mo Kang, "A low-power small-area ±7.28-ps-jitter 1-GHz DLL-based clock generator," in *Solid-State Circuits, IEEE Journal of* , vol.37, no.11, pp.1414-1420, Nov2002.
6. Jan M.Rabey, Anantha Chandrakasan and Borivoje Nicoloc , “Digital Integrated Circuits”, PHI Learning Pvt. Ltd. New Delhi-2011.
7. B. Razavi, “Design of Analog CMOS Integrated Circuits, McGraw-Hill”, New York, 2001.
8. R. Jacob Baker, Stuart K. Teskbury and Joe E. Brewe *, “*CMOS Circuit Design, Layout, and Simulation”, IEEE Press Series on Microelectronic Systems, 2005.
9. Eric R.booth, “Wide range, low jitter Delay-Locked Loop using a graduated digital delay line and phase interpolator”, Electrical Engineering, November 2006, Boise state University, M.S. Thesis, pp. 19 – 22, 2006.
10. C. Jia, “A Delay-Locked Loop for multiple clock phases/delays generation” Electrical and computer engineering, December 2005, Georgia Institute of Technology, PhD thesis, pp 25, 2005.

11. Farjad-Rad, R.; Dally, W.; Hiok-Tiaq Ng; Senthinathan, R.; Lee, M.-J.E.; Rathi, R.; Poulton, J., "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," in *Solid-State Circuits, IEEE Journal of* , vol.37, no.12, pp.1804-1812, Dec 2002.
12. Jovanovic, G.; Stojćev, M.; Krstic, D., "Delay locked loop with linear delay element," in *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2005. 7th International Conference on* , vol.2, no., pp.397-400 vol. 2, 28-30 Sept. 2005.
13. Soh Lip-Kai; Sulaiman, M.-S.; Yusoff, Z., "Fast-Lock Dual Charge Pump Analog DLL using Improved Phase Frequency Detector," in *VLSI Design, Automation and Test, 2007. VLSI-DAT 2007. International Symposium on* , vol., no., pp.1-5, 25-27 April 2007.
14. Pin-Tseng Chen; Chia-Chen Chang; Han-Ying Liu; Yu-Lung Lo, "A Fast-Lock Analog Multiphase Delay-Locked Loop Using a Dual-Slope Technique," in *Computer, Consumer and Control (IS3C), 2012 International Symposium on* , vol., no., pp.954-957, 4-6 June 2012.
15. Karutharaja, V.; Bhaskar, M.; Venkataramani, B., "Synchronization of on-chip serial interconnect transceivers using Delay Locked Loop (DLL)," in *Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on* , vol., no., pp.213-216, 21-22 July 2011.
16. Maillard, P.; Holman, W.T.; Loveless, T.D.; Bhuva, B.L.; Massengill, L.W., "An RHBD Technique to Mitigate Missing Pulses in Delay Locked Loops," in *Nuclear Science, IEEE Transactions on* , vol.57, no.6, pp.3634-3639, Dec. 2010,
17. Sukhwani, M.; Chandratre, V.B.; Thomas, M.; Pithawa, C.K.; Sharda, V., "500 MHz Delay Locked Loop Based 128-bin, 256 ns Deep Analog Memory ASIC "Anusmriti"," in *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on* , vol., no., pp.72-77, 4-6 July 2011.
18. Heedon Jang; Hyunggun Ma; Dongwook Seo; Jaeho Lee; Hyungsoo Lee; Bien, F., "Low power receiver for medical implantable communication system using delay locked loop," in *RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-Bio), 2014 IEEE MTT-S International Microwave Workshop Series on* , vol., no., pp.1-3, 8-10 Dec. 2014.

19. Ko-Chi Kuo; Chung-Yuan Chang; Si-Hsien Li, "Low power delay locked loop with all digital controlled SAR delay cell," in *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on* ,vol., no., pp.120-123, 2-5 Dec. 2012.
20. Bo Ye; Tianwang Li; Xingcheng Han; Min Luo, "A fast-lock digital delay-locked loop controller," in *ASIC, 2009. ASICON '09. IEEE 8th International Conference on* , vol., no., pp.809-812, 20-23 Oct. 2009.
21. Kai Huang; Zhikuang Cai; Jun Yang, "A Harmonic-Free and Fast-Locking Delay-Locked Loop Adopting a Resettable Delay Line," in *Computer Research and Development, 2010 Second International Conference on* , vol., no., pp.807-810, 7-10 May 2010.
22. Eunseok Song; Seung-Wook Lee; Jeong-Woo Lee; Joonbae Park; Soo-Ik Chae, "A reset-free anti-harmonic delay-locked loop using a cycle period detector," in *Solid-State Circuits, IEEE Journal of* , vol.39, no.11, pp.2055-2061, Nov. 2004.
23. Jinyeong Moon; Hye-young Lee, "A dual-loop delay locked loop with multi digital delay lines for GHz DRAMs," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on* , vol., no., pp.313-316, 15-18 May 2011.
24. Jasielski, J.; Kuta, S.; Machowski, W.; Kolodziejski, W., "An analog dual delay locked loop using coarse and fine programmable delay elements," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference* , vol., no., pp.185-190, 20-22 June 2013 .
25. Senthilnathan, S.; Balaji, S.; Prithiviraj, R., "Clock generation using delay lock loop with dual control for PET/CT," in *Advanced Electronic Systems (ICAES), 2013 International Conference on* , vol., no., pp.98-102, 21-23 Sept. 2013.
26. Shen-Iuan Liu; Jiunn-Hwa Lee; Hen-Wai Tsao, "Low-power clock-deskew buffer for high-speed digital circuits," in *Solid-State Circuits, IEEE Journal of* , vol.34, no.4, pp.554-558, Apr 1999.
27. Gholami, M.; Rahimpour, H.; Ardeshir, G.; MiarNaimi, H., "Digital delay locked loop-based frequency synthesiser for Digital Video Broadcasting-Terrestrial receivers," in *Circuits, Devices & Systems, IET* , vol.8, no.1, pp.38-46, Jan. 2014.

APPENDIX

- W/L values for circuit shown in fig.5.7

Sr. No.	Transistor	W (nm)
1.	M1	3600
2.	M2	3600
3.	M3	1800
4.	M4	1800
5.	M5	1800
6.	M6	3600

- W/L values for circuit shown in fig.5.9

Sr. No.	Transistor	W (nm)	L (nm)
1.	M1	18000	180
2.	M2	3600	180
3.	M3	1800	180
4.	M4	1800	360
5.	M5	3600	180
6.	M6	36000	180

- W/L values for circuit shown in fig. 5.13

Sr. No.	Transistor	W (nm)	L (nm)
1.	M1	1800	180
2.	M2	1800	360
3.	M3	1800	360
4.	M4	1800	180
5.	M5	3600	180
6.	M6	3600	180
7.	M7	3600	180

- W/L values for circuit shown in fig. 5.16

  $L_n = L_p = 180$ nm

Sr. No.	Transistor	W (nm)
1.	M1	3600
2.	M2	3600
3.	M3	1800
4.	M4	1800
5.	M5	36000

- **Model Parameters**

```
* PSPICE TSMC180nm.lib file  RWN  04/18/2010
* library  file for transistor parameters for TMSC 0.18 micron process
* uses BIM parameters added 01/15/98
* can configure and attach to Nbreak and Pbreak transistors in PSpice
****
*****************  180nm TSMC parameters  *************
*T14B SPICE BSIM3 VERSION 3.1 PARAMETERS
* downloaded from MOSIS 04/18/10
*http://www.mosis.com/cgi-bin/cgiwrap/umosis/swp/params/
* tsmc-018/t92y_mm_non_epi_thk_mtl_params.txt
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
* DATE: Jun  8/01
* LOT: T14B                WAF: 06
* Temperature_parameters=Default
*$
.MODEL TSMC180nmN NMOS (                          LEVEL   = 7
+VERSION = 3.1           TNOM    = 27             TOX     = 4.1E-9
+XJ      = 1E-7          NCH     = 2.3549E17      VTH0    = 0.354505
+K1      = 0.5733393     K2      = 3.177172E-3    K3      = 27.3563303
+K3B     = -10           W0      = 2.341477E-5    NLX     = 1.906617E-7
```

+DVT0W = 0 DVT1W = 0 DVT2W = 0

+DVT0 = 1.6751718 DVT1 = 0.4282625 DVT2 = 0.036004

+U0 = 327.3736992 UA = -4.52726E-11 UB = 4.46532E-19

+UC = -4.74051E-11 VSAT = 8.785346E4 A0 = 1.6897405

+AGS = 0.2908676 B0 = -8.224961E-9 B1 = -1E-7

+KETA = 0.021238 A1 = 8.00349E-4 A2 = 1

+RDSW = 105 PRWG = 0.5 PRWB = -0.2

+WR = 1 WINT = 0 LINT = 1.351737E-8

*+XL = -2E-8 XW = -1E-8

+ DWG = 1.610448E-9

+DWB = -5.108595E-9 VOFF = -0.0652968 NFACTOR = 2.4901845

+CIT = 0 CDSC = 2.4E-4 CDSCD = 0

+CDSCB = 0 ETA0 = 0.0231564 ETAB = -0.058499

+DSUB = 0.9467118 PCLM = 0.8512348 PDIBLC1 = 0.0929526

+PDIBLC2 = 0.01 PDIBLCB = -0.1 DROUT = 0.5224026

+PSCBE1 = 7.979323E10 PSCBE2 = 1.522921E-9 PVAG = 0.01

+DELTA = 0.01 RSH = 6.8 MOBMOD = 1

+PRT = 0 UTE = -1.5 KT1 = -0.11

+KT1L = 0 KT2 = 0.022 UA1 = 4.31E-9

+UB1 = -7.61E-18 UC1 = -5.6E-11 AT = 3.3E4

+WL = 0 WLN = 1 WW = 0

+WWN = 1 WWL = 0 LL = 0

+LLN = 1 LW = 0 LWN = 1

+LWL = 0 CAPMOD = 2 XPART = 0.5

+CGDO = 7.7E-10 CGSO = 7.7E-10 CGBO = 1E-12

+CJ = 1.010083E-3 PB = 0.7344298 MJ = 0.3565066

+CJSW = 2.441707E-10 PBSW = 0.8005503 MJSW = 0.1327842

+CJSWG = 3.3E-10 PBSWG = 0.8005503 MJSWG = 0.1327842

+CF = 0 PVTH0 = 1.307195E-3 PRDSW = -5

+PK2 = -1.022757E-3 WKETA = -4.466285E-4 LKETA = -9.715157E-3

+PU0 = 12.2704847 PUA = 4.421816E-11 PUB = 0

```
+PVSAT   = 1.707461E3     PETA0   = 1E-4          PKETA   = 2.348777E-3     )
*
.MODEL TSMC180nmP PMOS (                          LEVEL   = 7
+VERSION = 3.1            TNOM    = 27            TOX     = 4.1E-9
+XJ      = 1E-7           NCH     = 4.1589E17     VTH0    = -0.4120614
+K1      = 0.5590154      K2      = 0.0353896     K3      = 0
+K3B     = 7.3774572      W0      = 1E-6          NLX     = 1.103367E-7
+DVT0W   = 0              DVT1W   = 0             DVT2W   = 0
+DVT0    = 0.4301522      DVT1    = 0.2156888     DVT2    = 0.1
+U0      = 128.7704538    UA      = 1.908676E-9   UB      = 1.686179E-21
+UC      = -9.31329E-11   VSAT    = 1.658944E5    A0      = 1.6076505
+AGS     = 0.3740519      B0      = 1.711294E-6   B1      = 4.946873E-6
+KETA    = 0.0210951      A1      = 0.0244939     A2      = 1
+RDSW    = 127.0442882    PRWG    = 0.5           PRWB    = -0.5
+WR      = 1              WINT    = 5.428484E-10  LINT    = 2.468805E-8
*+XL     = -2E-8          XW      = -1E-8
+DWG     = -2.453074E-8
+DWB     = 6.408778E-9    VOFF    = -0.0974174    NFACTOR = 1.9740447
+CIT     = 0              CDSC    = 2.4E-4        CDSCD   = 0
+CDSCB   = 0              ETA0    = 0.1847491     ETAB    = -0.2531172
+DSUB    = 1.5            PCLM    = 4.8842961     PDIBLC1 = 0.0156227
+PDIBLC2 = 0.1            PDIBLCB = -1E-3         DROUT   = 0
+PSCBE1  = 1.733878E9     PSCBE2  = 5.002842E-10  PVAG    = 15
+DELTA   = 0.01           RSH     = 7.7           MOBMOD  = 1
+PRT     = 0              UTE     = -1.5          KT1     = -0.11
+KT1L    = 0              KT2     = 0.022         UA1     = 4.31E-9
+UB1     = -7.61E-18      UC1     = -5.6E-11      AT      = 3.3E4
+WL      = 0              WLN     = 1             WW      = 0
+WWN     = 1              WWL     = 0             LL      = 0
+LLN     = 1              LW      = 0             LWN     = 1
+LWL     = 0              CAPMOD  = 2             XPART   = 0.5
```

```
+CGDO    = 7.11E-10       CGSO    = 7.11E-10       CGBO    = 1E-12
+CJ      = 1.179334E-3    PB      = 0.8545261      MJ      = 0.4117753
+CJSW    = 2.215877E-10   PBSW    = 0.6162997      MJSW    = 0.2678074
+CJSWG   = 4.22E-10       PBSWG   = 0.6162997      MJSWG   = 0.2678074
+CF      = 0              PVTH0   = 2.283319E-3    PRDSW   = 5.6431992
+PK2     = 2.813503E-3    WKETA   = 2.438158E-3    LKETA   = -0.0116078
+PU0     = -2.2514581     PUA     = -7.62392E-11   PUB     = 4.502298E-24
+PVSAT   = -50            PETA0   = 1E-4           PKETA   = -1.047892E-4    )
*
.ENDS
```