SERIALIZER

A Project Report

submitted by

SREENIVASA RAO G

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JUNE 2019

THESIS CERTIFICATE

This is to certify that the thesis titled **SERIALIZER**, submitted by **SREENIVASA RAO G**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Saurabh Saxena
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai

Date: 10^{th} June 2019

ACKNOWLEDGEMENTS

I would like to express my earnest gratitude to my guide Dr. Saurabh Saxena whose knowledge and dedication has inspired me in developing interest towards Analog circuits and to work efficiently on the project. I thank him for motivating me, allowing me freedom to exchange ideas and flexibility while working on the project.

I would like to express my sincere gratitude towards my team members R. Gautham, Jaya Deepthi and M. Kiran Kumar who helped me in situations of need, with their experience, timely guidance and support.

This journey would not have been possible without the support of my family, professor and mentors, and friends. To my family, thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I am especially grateful to my parents, who supported me emotionally and financially.

ABSTRACT

In this modern day electronics industry, large amount of data need to be transmitted from chip-to-chip on a PCB say from One core of a Microprocessor to another, from a Microprocessor to an IO device, from a Microprocessor to memory in a computer. The day to day improvements in the existing features and incorporation of new features say for example, mobile phone has necessitated the chip designers to think of alternative ways of transmitting this large amounts of data. Routing new channels for handling this data is not a feasible thing to do, as it will not only increase the chip area which increases the cost of the chip, but also induces second order effects on the other working blocks.

This has forced the chip designers to go for Serializer/Deserializer abbreviated as SERDES where in, the data is transmitted at a higher data rate on the existing channel by introducing additional circuitry both at the sender chip (Transmitter) and at the receiver chip (Receiver).

This thesis work concentrates on the transmitter part of the SERDES Which employs a Serializer, Pre driver an Output driver. The Serializer converts data from seven channels each at a data rate fs (Time period Ts), to a single channel at a data rate 7fs (Time period $\frac{Ts}{7}$). The inputs to the Serializer are data and seven clock phases each at a spacing of $\frac{Ts}{7}$ from the consecutive phase. These clock phases are generated by a Delay Locked Loop (DLL). The Serializer inserts the data bit in the phase difference between the clock phases. The serialized data so obtained is transmitted by an LVDS driver on to the channel.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
CP	Charge Pump
PD	Phase Detector
VCDL	Voltage Controlled Delay Line
LVDS	Low Voltage Differential Signalling
DLL	Delay Locked Loop
PSNR	Power Supply Noise Rejection
PSRR	Power Supply Ripple Rejection
PSD	Phase noise Spectral Density
PCB	Printed Circuit Board
IO	Input Output
JTRAN	Jitter Transfer
f_{ugb}	Unity Gain Frequency
PRBS	Pseudo Random Binary Sequence
PLL	Phase Locked Loop
PFD	Phase Frequency Detector
VCO	Voltage Controlled Oscillator
HSS	High-speed Serializer/DeSerializer
FFE	Feed Forward Equalizer

NOTATION

r	Radius, m
α	Angle of thesis in degrees
β	Flight path in degrees

CHAPTER 1

INTRODUCTION

High-Speed Serializer/Deserailizer (HSS) devices are the dominant implementation of I/O interfaces at speeds of 2.5Gbps and higher. Such devices are differentiated from source-synchronous interfaces in that the receiver device contains a clock and data recovery (CDR) circuit which dynamically determines the optimal sampling point of the data signal based upon the transition edges of the signal. In other words, clock information is extracted directly from the data rather than relying on a separate clock.

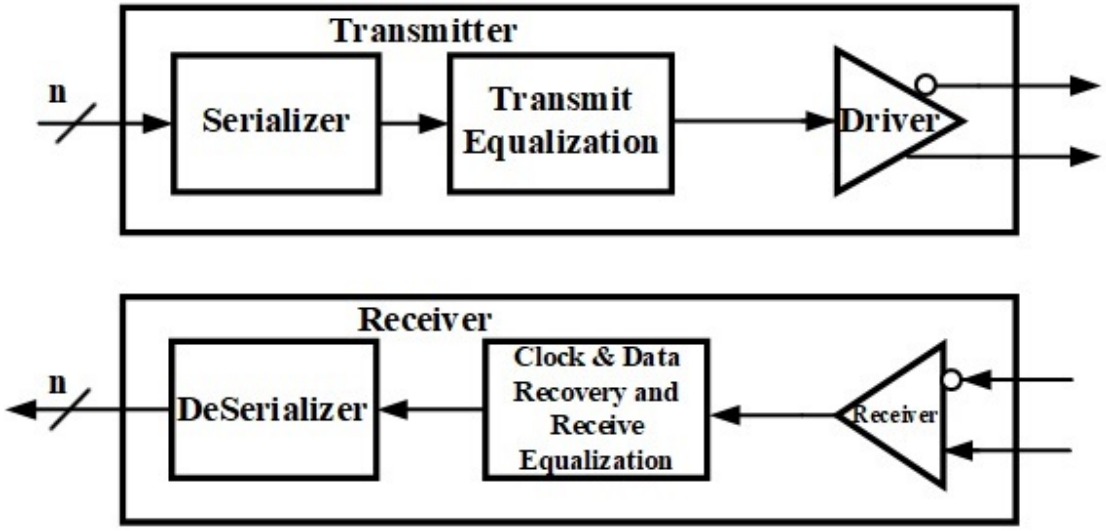

Figure 1.1: Basic block diagram of typical high speed SERDES

Figure 1.1 illustates the basic block diagram of the transmit and receive channels of an HSS device. The transmitter serializes parallel data, equalizes it and then drives the serial data onto a differential signal pair of interconnect wires. Feed Forward Equalizers (FFE) are commonly used in High speed SERDES devices. The receiver consists of a differential receiver, a CDR circuit which may also integrate an equalizer, and deserializes the data based upon the sample point established by the CDR. Peaking amplifiers and/or decision feedback equalizers (DFE) are commonly used for equalization in High speed SERDES receiver devices.

1.1 Serializer

Conceptually, the input to the serializer transmit stage is an n-bit datapath which is serialized to a one bit serail data signal for application to the FFE and driver stages. Values of n which are multiples are 8 are useful for sending uncoded and/or scrambled data bytes; values of n which are multiples of 10 are useful for protocols which use 8B/10B coding.

1.2 Driver

The differential driver stage is an analog circuit which drives the true and complement legs of the differential signal. Output data must be driven such that jitter is minimized. In some architectures, data is latched in a flip-flop clocked at the baud rate and the output of this flop is driven on to the differential output.

The block diagram of SERDES chain being implemented in this thesis is as shown below.

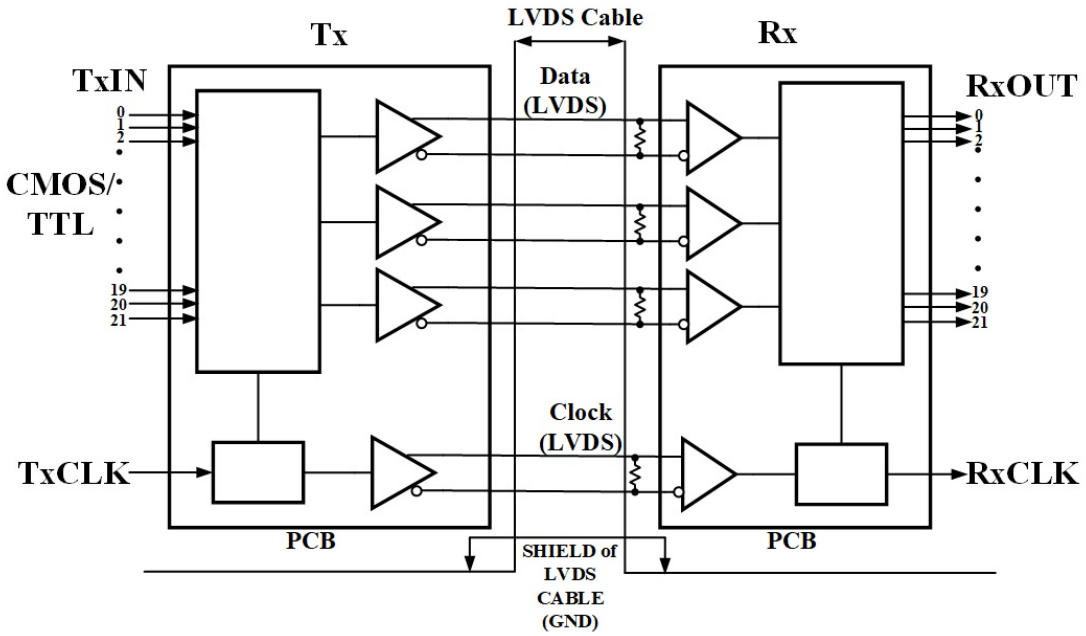

Figure 1.2: Block diagram of SERDES

This SERDES has 21 CMOS/TTL input channels each having a data rate of 15Mbps-75Mbps. These 21 channels are seggregated into 3 groups each having 7 CMOS/TTL

channels. The serialized data from each group which is at a data rate of 105Mbps-525Mbps, is transmitted over LVDS channel as differential signals. This SERDES uses a clock forward architecture and hence there is no need of clock recovery circuit at the receiver. As the clock signal is transmitted at the base rate, the receiver uses a PLL for clock multiplication.

The motivation for going towards LVDS lies in the fact that, as the signal frequency increases, the circuit impedances tend to decrease requiring more current to generate a voltage. Since, power is the product of voltage and current, maintaining or reducing power levels require a reduction in operating voltage. Thus, in LVDS data bits are transmitted as low voltage differential signals. The schematic of single chain of SERDES Block is as shown below.

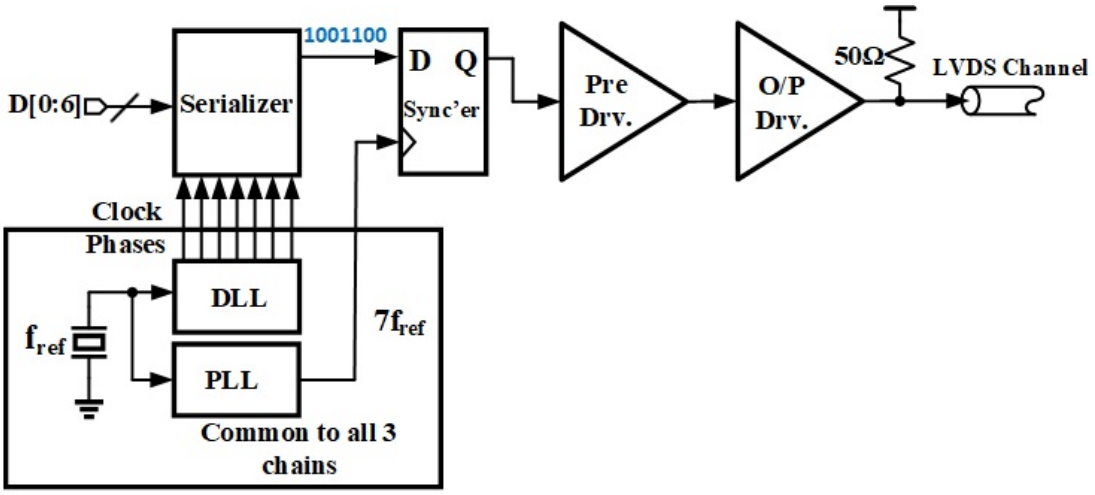

Figure 1.3: Single chain of SERDES Block

Table 1.1: The specifications of the transmitter in this thesis are tabulated as shown below.

Sl.No	Description	Specification
1	Power supply	1.8V
2	Clock	15MHz-75MHz
3	Output signal type	Differential
4	Maximum LVDS channel datarate	525Mbps
5	LVDS swing	350mV
6	LVDS common mode voltage	1.2V
7	LVDS transmitter load	C_L = 5pF and R_L = 100Ω
8	Power dissipation	Minimum

CHAPTER 2

BACKGROUND

2.1 Analysis of Delay Locked Loop

The Delay Locked Loop generates the seven clock phases needed by the serializer. The DLL uses a Phase Detector (PD), Charge Pump, Regulator loop and the Delay Line. The Delay Line generates the clock phases each with a phase difference of $\frac{Ts}{7}$. The phase difference between the successive phases is varied across the entire operating range of 15MHz-75MHz by increasing/decreasing the supply voltage to the delay line, herein after referred to as V_{VCDL}.

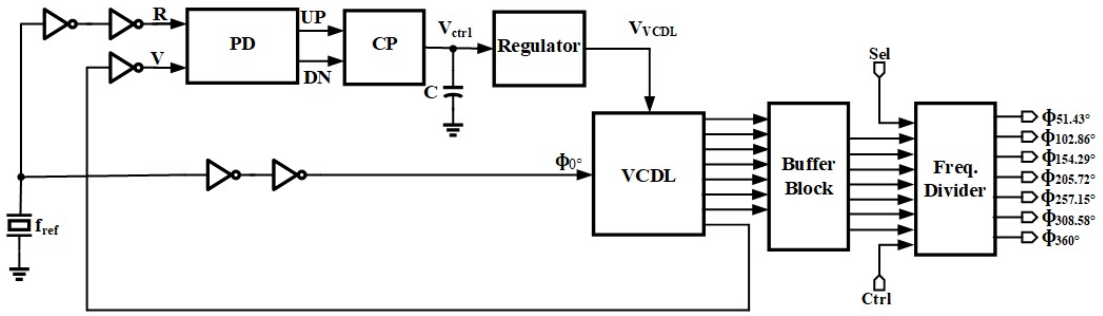

Figure 2.1: Block diagram of DLL

In the phase domain model, the delay line has a transfer function of $\phi_{out} = K_{VCDL}V_{cont} + \phi_{off}$ where K_{VCDL} is the gain of variable delay stage defined as $K_{VCDL} = \frac{\partial\phi}{\partial V_{cont}}$, ϕ_{off} is the offset phase difference . In the analysis followed, it is assumed that $\phi_{off} = 0$.

The phase domain small signal equivalent circuit of the DLL becomes

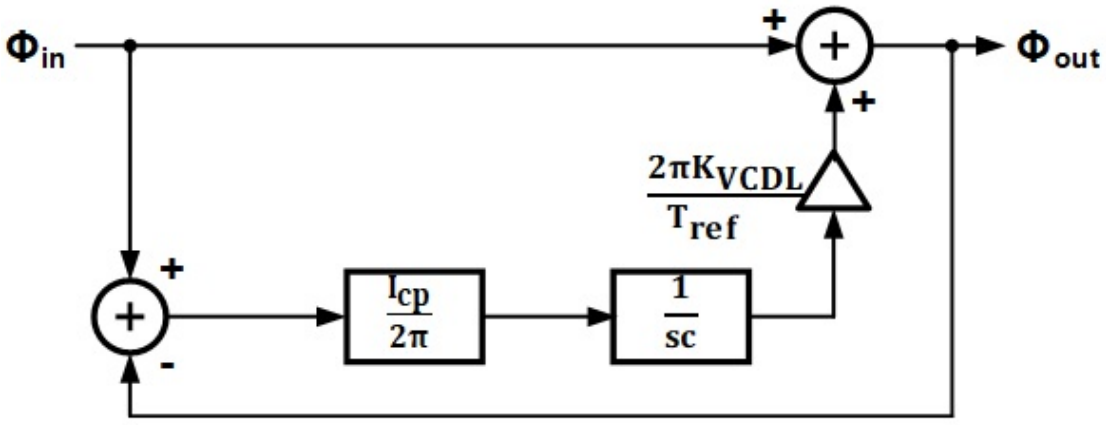

Figure 2.2: Small signal phase model of DLL

$$\phi_{out} = \phi_{in} + (\phi_{in} - \phi_{out})\frac{I_{cp}}{2\pi sc}.\frac{2\pi K_{VCDL}}{T_{ref}}$$

$$\phi_{out}[1 + \frac{I_{cp}K_{VCDL}}{T_{ref}sc}] = \phi_{in}[1 + \frac{I_{cp}K_{VCDL}}{T_{ref}sc}]$$

$$\phi_{out} = \phi_{in}$$

Thus, Jitter Transfer (JTRAN) function = $\frac{\phi_{out}}{\phi_{in}}$ = 1

$$LG(s) = \frac{\frac{2\pi K_{VCDL}}{T_{ref}sc}.\frac{I_{cp}}{2\pi}}{\frac{I_{cp}K_{VCDL}}{T_{ref}sc}}$$

Noise Transfer Functions (NTF's) of various sources

Reference source:

From Figure 2.2,

$NTF_{ref} = \frac{\phi_{n,out}}{\phi_{n,in}} = 1$

Charge Pump:

The phase model small signal equivalent circuit with charge pump noise is

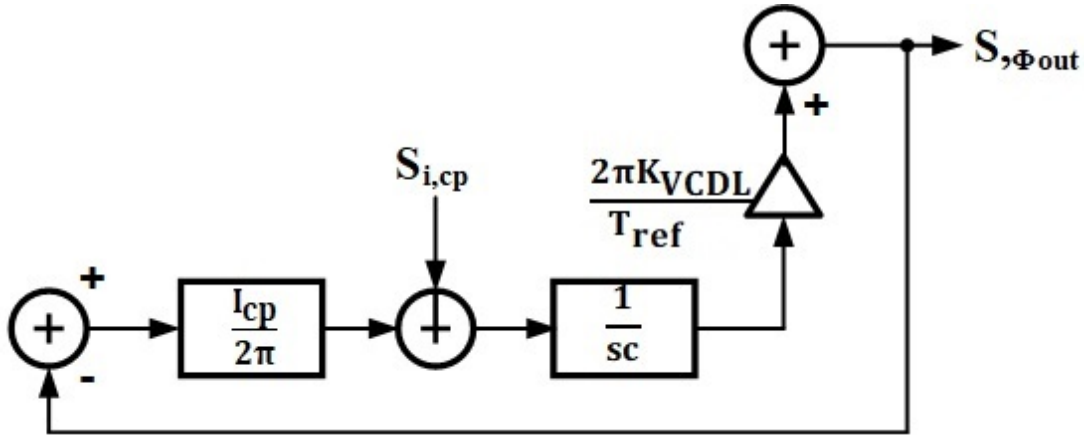

Figure 2.3: Small signal phase model of DLL with CP Noise

$$NTF_{CP} = \frac{\phi_{n,out}}{\phi_{n,cp}} = \frac{\frac{2\pi K_{VCDL}}{T_{ref} sc}}{1+\frac{I_{cp} K_{VCDL}}{T_{ref} sc}}$$

$$= \frac{2\pi K_{VCDL}}{I_{CP} K_{VCDL} + T_{ref} sC}$$

$$= \frac{\frac{2\pi}{I_{CP}}}{1+\frac{sT_{ref}C}{I_{CP}K_{VCDL}}}$$

Voltage Controlled Delay Line:

The phase model small signal equivalent circuit with VCDL noise is

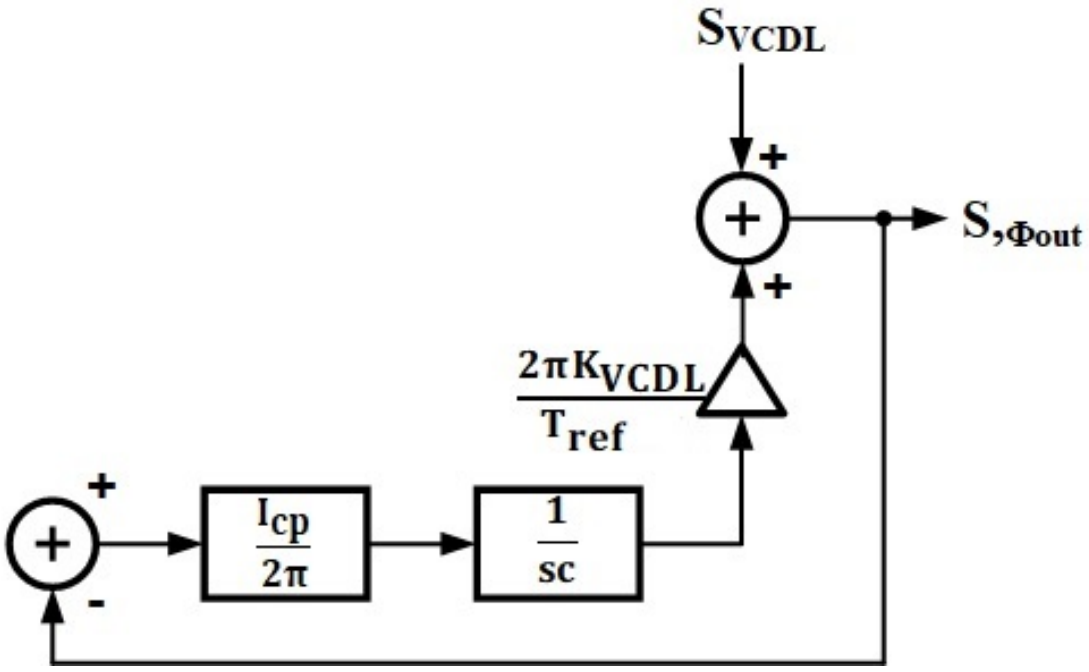

Figure 2.4: Small signal phase model of DLL with VCDL Noise

$$NTF_{VCDL} = \frac{1}{1+\frac{I_{CP} K_{VCDL}}{T_{ref} sC}}$$

The jitter from all the above sources is calculated as follows.

$$S^{out}_{n,ref} = S_{n,ref} |NTF_{ref}|^2$$

$S^{out}_{n,CP} = S_{n,CP}|NTF_{CP}|^2$

$S^{out}_{n,VCDL} = S_{n,VCDL}|NTF_{VCDL}|^2$

$S^{out}_{n,total} = S^{out}_{n,ref} + S^{out}_{n,CP} + S^{out}_{n,VCDL}$

$$\phi^2_{n,total} = \int S^{out}_{n,total} d\omega \quad \sigma_{rms} = \frac{T_{ref}}{2\pi}\sqrt{\phi^2_{n,total}} \tag{2.1}$$

Based on the matlab simulations of Jitter evaluated from all the noise sources stated above at the output of DLL, a graph of jitter Vs f_{ugb} is plotted for the DLL. Based, on the jitter specification, the required value of f_{ugb} is chosen for the DLL. The result of Matlab simulation is as shown below.

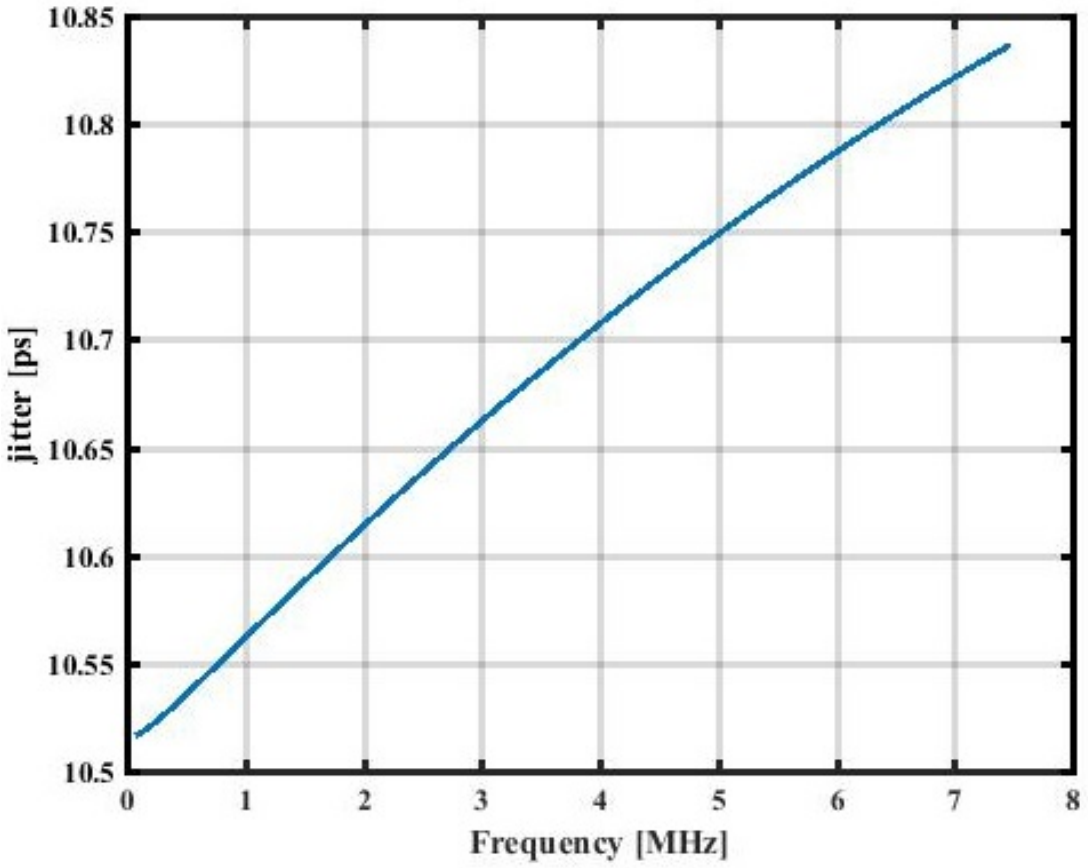

Figure 2.5: Jitter Vs f_{ugb} of DLL

2.2 Analysis of PLL

The PLL generates the clock at the rate of serialized data to be used by the synchronizer to remove possible jitter from propagating further down the transmitter.

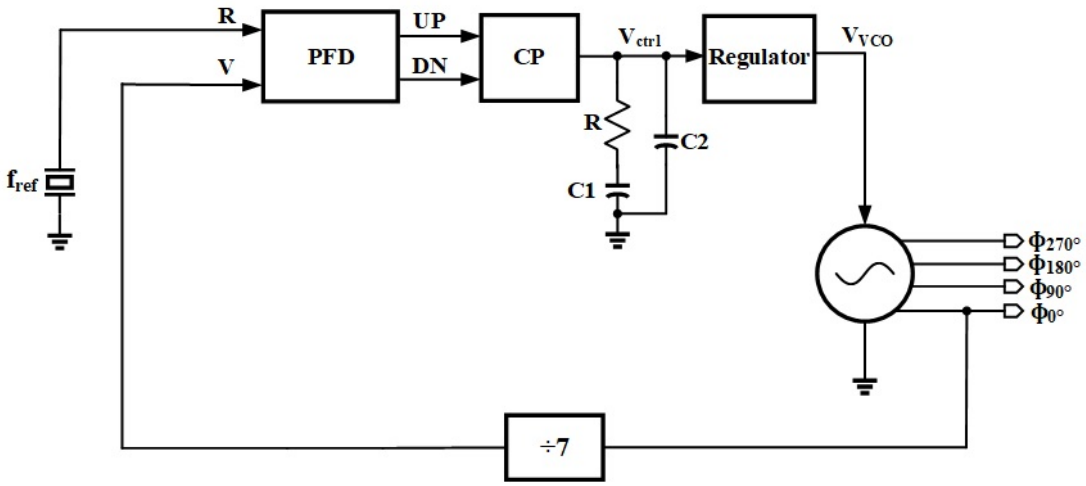

Figure 2.6: Block diagram of PLL

The phase domain small signal equivalent circuit of the PLL becomes

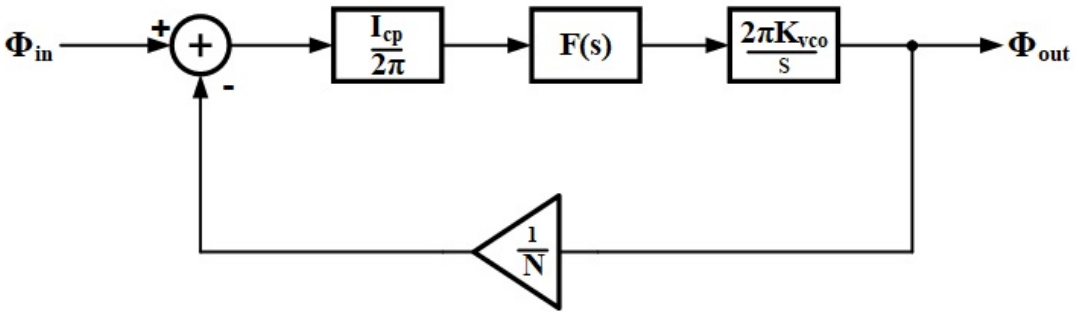

Figure 2.7: Small signal phase model of PLL

where F(s) = $\frac{1+\frac{s}{\omega_z}}{s^2(1+\frac{s}{\omega_{p3}})}$, $\omega_z = \frac{1}{RC_1}$ and $\omega_{p3} = \frac{1}{R\frac{C_1C_2}{C_1+C_2}}$

$$LG(s) = \frac{I_{cp}RK_{vco}}{N(C_1+C_2)} \times \frac{1+\frac{s}{\omega_z}}{s^2\left(1+\frac{s}{\omega_{p3}}\right)} and \frac{\phi_{out}}{\phi_{in}} = \frac{N \times LG}{1+LG}$$

CHAPTER 3

Implementation of DLL

Based on the jitter calculations in Chapter 2, the DLL is designed to work from 30MHz to 75MHz. In order, to meet the specification of 15MHz-75MHz, the operating frequency range of DLL is expanded by inserting a Divider block, which down converts the clock phases from 30MHz to 15MHz.

The DLL also has an option for selecting either divide by 2/3. When divide by 3 is chosen, the divider has an addition feature of duty cycle correction.

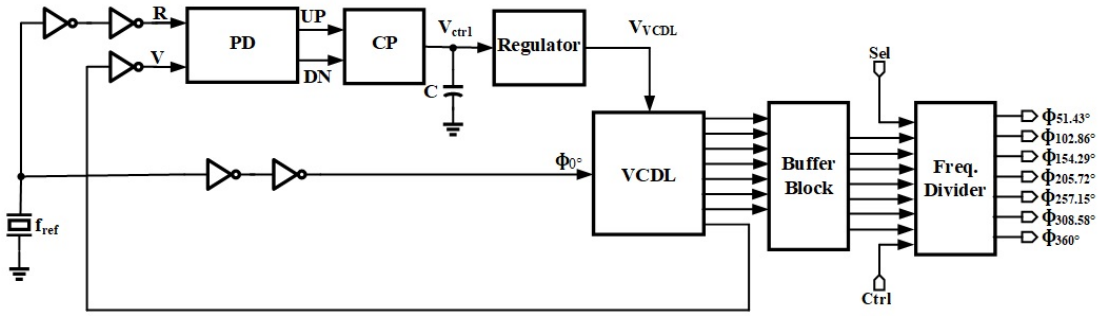

Figure 3.1: Block diagram of DLL

The RC extracted phases of the DLL at 75MHz and 15MHz respectively are as shown

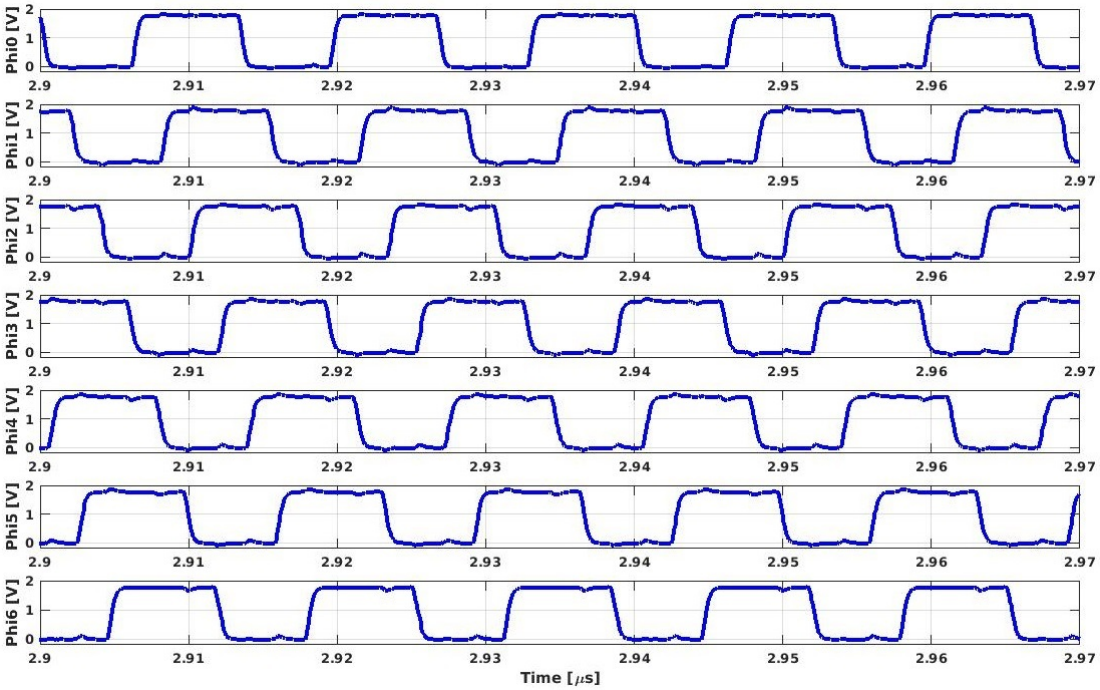

Figure 3.2: Phases of DLL @75MHz

As can be seen, the time period of each phase is same as that of reference period i.e 13.33ns

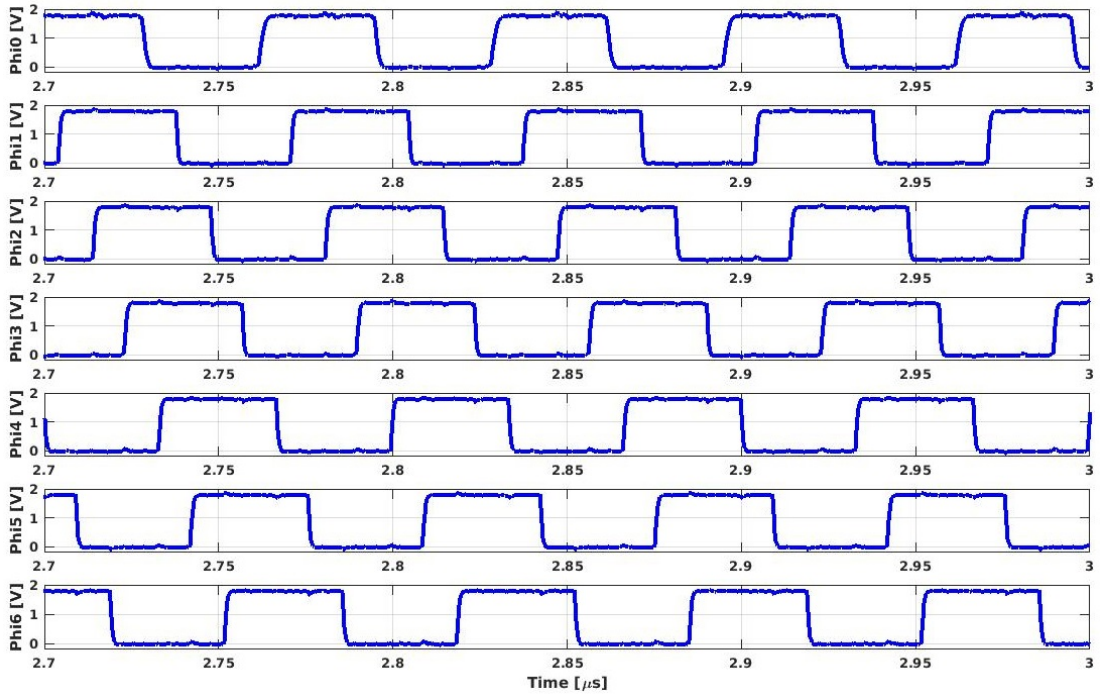

Figure 3.3: Phases of DLL @15MHz

As can be seen, the time period of each phase is same as that of reference period i.e 66.66ns

The corresponding V_{ctrl} and V_{VCDL} voltage waveforms are as shown below.

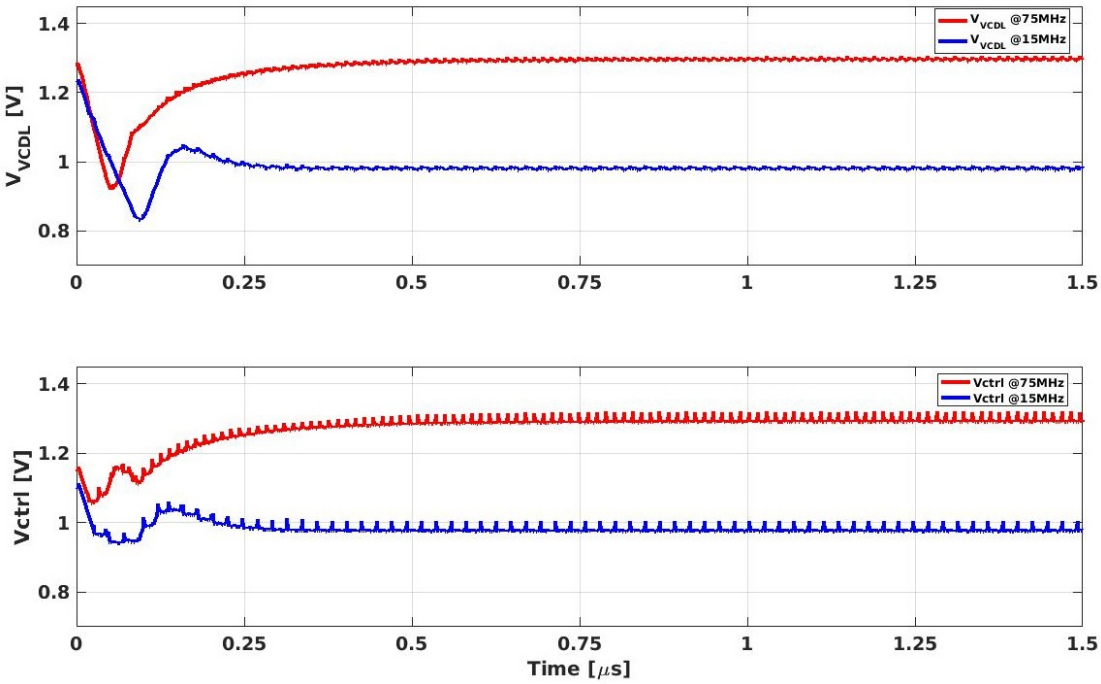

Figure 3.4: V_{ctrl} and V_{VCDL} at 75MHz and 15MHz respectively

3.1 Phase Detector (PD)

The PD block tracks the phase of the reference clock and the last phase of Delay line. It produces four pulses UP, DN, $\bar{UP}$ and $\bar{DN}$ based on the phase difference between the two input signals. The DLL achieves a lock when the phase difference between the two inputs become zero.

The PD is implemented as shown below.

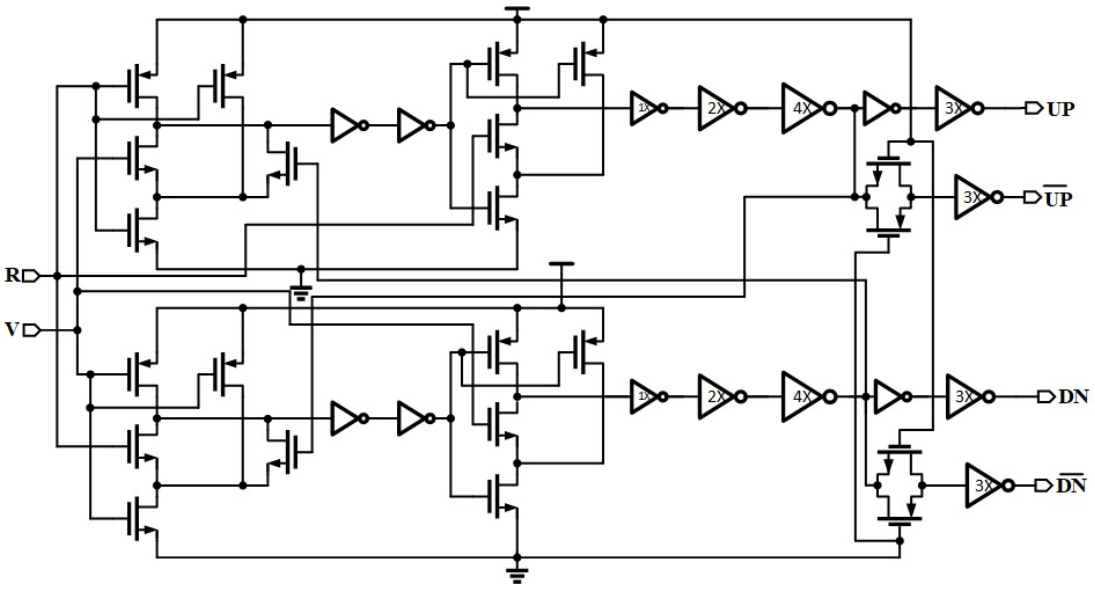

Figure 3.5: Phase Detector

The behaviour of R leading V is captured by UP and $\bar{UP}$ signals. UP goes high for the period during which R is 1 and V is 0. Now, as soon as both the input signals become i.e V becomes 1 while R is still 1, for a very short period of time UP and DN signals are high, before both becoming 0 again. Since, these UP and DN signals are fed to charge pump, they need to have sufficient pulse widths in order to activate the charge pump. If the pulse widths are insufficient, the PD-charge pump combine will not be able to detect the small phase difference, a problem called dead-zone. For this, more delay can be added to the reset path shown as two inverters in the PD. The characteristics of above PD are obtained by plotting the average value of UP-DN signals Vs phase difference between R and V on time scale and is as shown below.

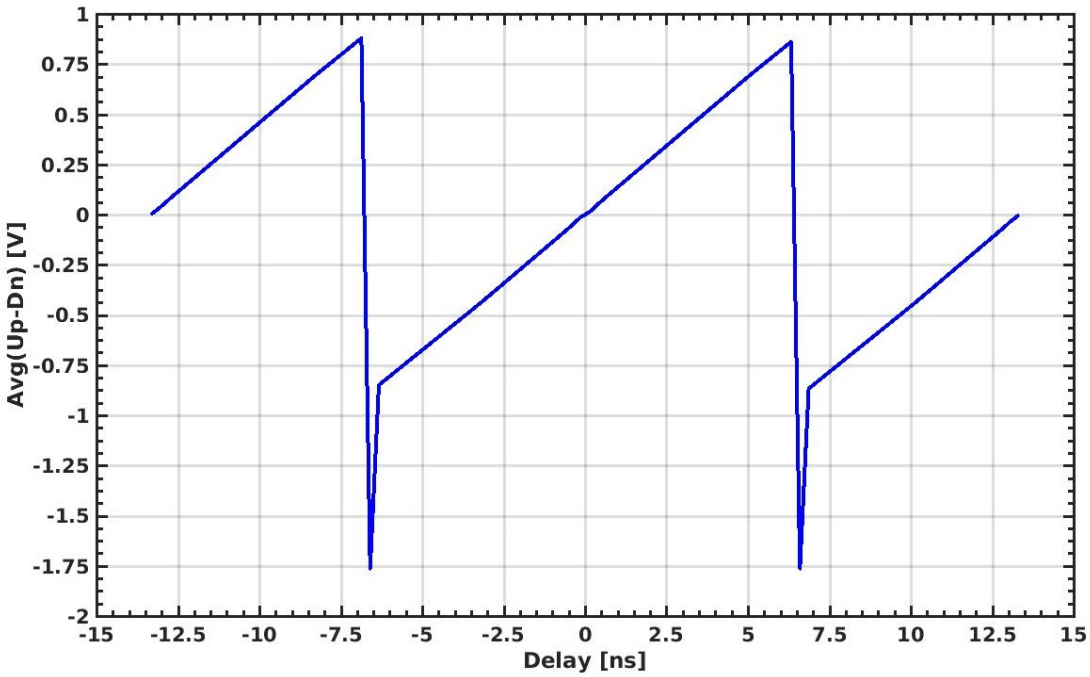

Figure 3.6: RC extracted PD Characteristics

3.2 Charge Pump CP)

The charge pump converts the UP and DN input pulses coming from PFD into current pulses that charge/discharge the loop filter capacitor to give the desired control voltage V_{ctrl}. Due to a phase difference between R and V signals, PD produces UP and DN pulses causing I_{CP} current to flow into/out of the loop filter capacitor. The UP current causes charge to be dumped on the loop filter capacitor and hence V_{ctrl} increases. On the other hand, the DN current pulse causes charge to be removed from loop filter capacitor and hence V_{ctrl} decreases. In order to maintain a good match between the UP and DN currents, boot strapped charge pump topology was used in this thesis.

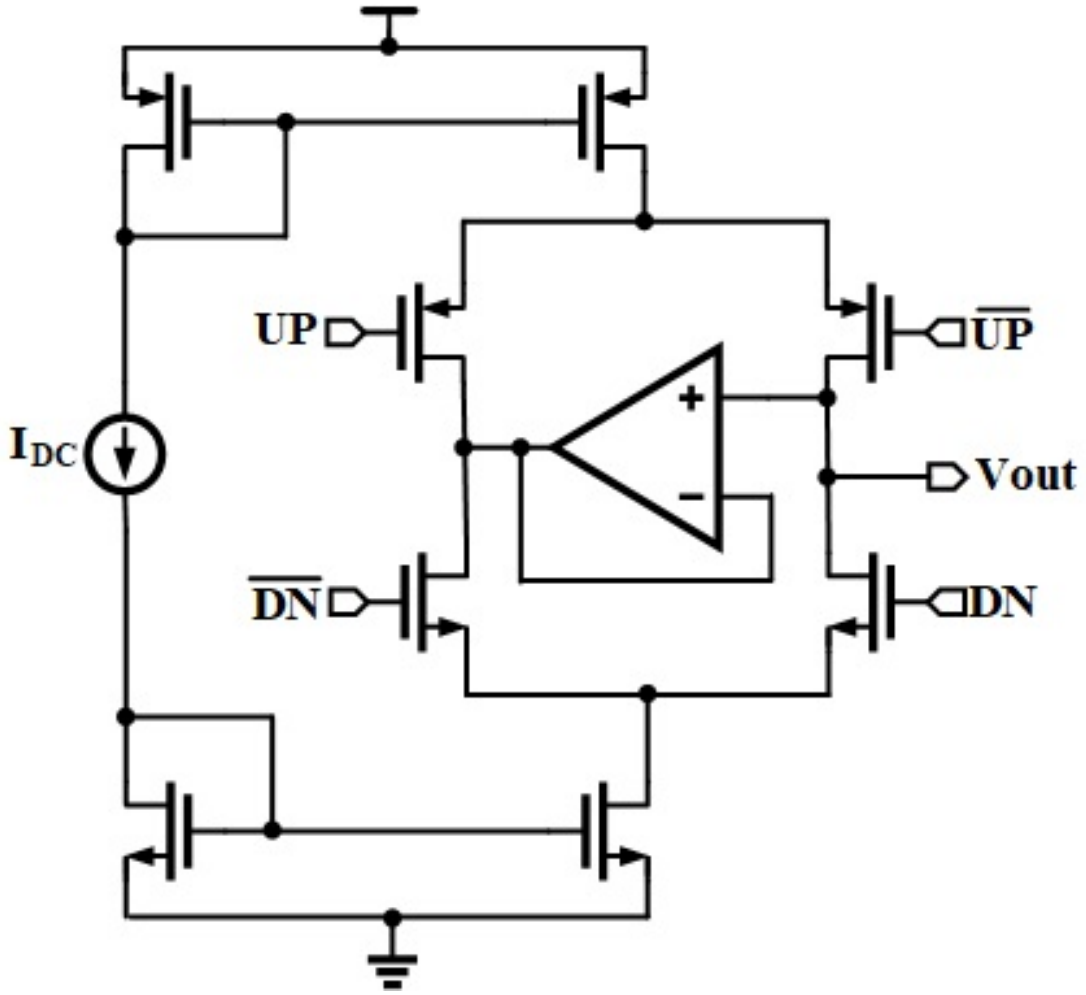

Figure 3.7: Charge Pump

The Bootstrapped charge pump makes use of a unity gain amplifier that ensures equal drain-to-source voltage V_{DS} for the UP and DN current sources, which allows for better current matching. For example, consider the case when there is no unity negative feedback opamp and DN=0. The voltage at source of DN transistor experiences a sudden peaking due to turning OFF of DN transistor. The voltage at the drain of DN transistor also sees the same peaking in order to maintaining zero current through DN transistor. During this transient time, spurious charge gets injected into the loop filter capacitor causing increased jitter. On the other hand, in the presence of unity negative feedback opamp, alternate path exists for I_{CP} current through $\bar{DN}$ transistor, which keeps the potential at source of DN transistor unaltered and hence no peaking occurs. This allows for reduction in jitter at the output. Another advantage of Bootstrapped charge pump is the ability to operate with low swing UP and DN signals. However, there is increased rail-to-rail power consumption than the convention Source switched/Drain switched charge pump because the parallel paths keep the current sources ON all the

time.

3.3 Voltage Controlled Delay Line

The voltage Controlled Delay Line is implemented as shown below.

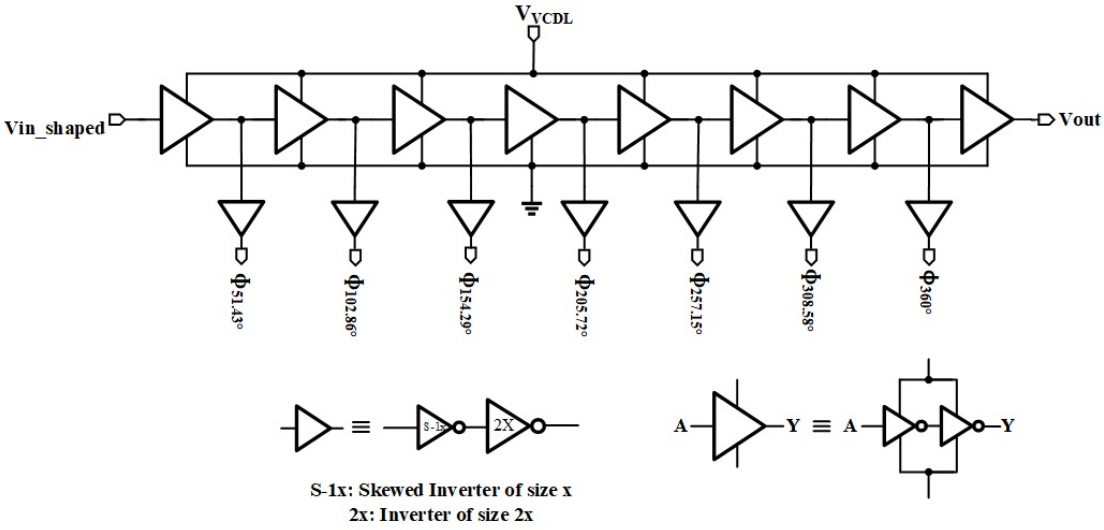

Figure 3.8: Delay Line

Each delay cell in the delay line has a delay of 1.905ns corresponding to $\frac{T_{ref}}{7}$ where T_{ref} is $\frac{1}{75MHz}$. The required delay tuning range from 33.33ns (= $\frac{1}{15MHz}$) to 13.33ns (= $\frac{1}{75MHz}$ is achieved by varying the supply voltage to the delay line V_{VCDL}. Each delay cell is composed of two inverters which are sized such that the delay contribution from each inverter is 952.38ps. Apart from these, skewed inverters are used to raise the swing limits of the phases from V_{VCDL} to 1.8V. The range of V_{VCDL} required to obtain the above tuning is [0.78V 1.54V] across all process corners and temperatures. The K_{VCDL} of the delay line is varying from 20.14ns/V to 98.63ns/V corresponding to the delay line operating frequency of 30MHz and 75MHz respectively. The wide tuning range of V_{VCDL} has necessitated to go for the design of a rail-to-rail opamp to be used in Bootstrapped charge pump and the regulator loop discussed later in this chapter.

3.4 Loop Filter

The loop filter is very important for lock acquisition and ripple suppression due to periodic charging/discharging of loop filter capacitor. The loop filter discussed in this thesis is a low pass filter.

The transfer function of loop filter is given by F(s) = $\frac{1}{sC}$.

The loop filter capacitance is evaluated as follows.

In order for the continuous time approximation to hold good for DLL, the Bandwidth is given by

BW(rad/s) = $\frac{I_{CP}K_{VCDL}}{T_{ref}C} \implies C = \frac{I_{CP}K_{VCDL}}{T_{ref}BW}$.

For maximum value of K_{VCDL}, C is maximum whose value is 785fF.

3.5 Rail-to-Rail opamp

In order to cater to the need of wide voltage range required by the delay line, rail-to-rail opamp with supply voltage as the common-mode voltage in open-loop was designed as shown below.

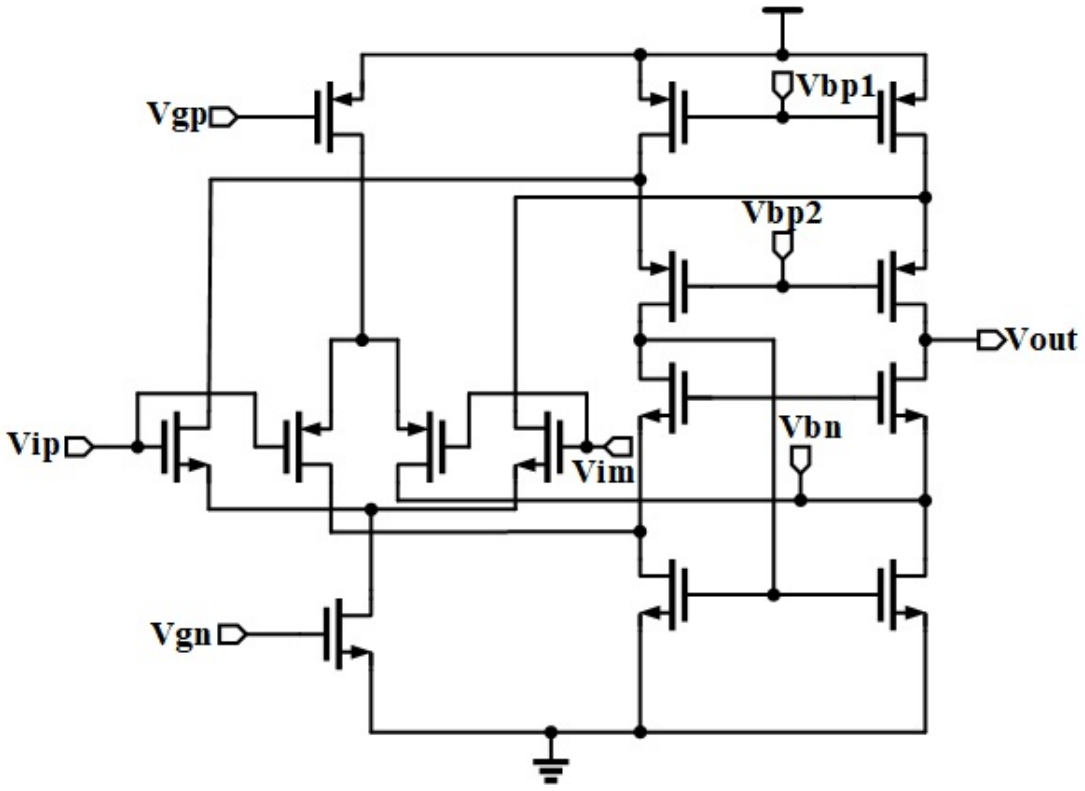

Figure 3.9: Charge Pump Opamp

This opamp consists of both nMOS and pMOS folded cascode stages. Assuming that the trans-conductances of nMOS and pMOS folded cascode stages are same i.e $gm_n = gm_p = g_m$, at low common-mode voltages, pMOS folded cascode stage gets activated and contributes a trans-conductance of g_m resulting in a gain of $g_m R_{out}$ where R_{out} is the output impedance of cascode stage. At common-mode voltages close to supply, nMOS folded cascode stage gets activated and contributes a trans-conductance of g_m resulting in a gain of $g_m R_{out}$. At common-mode voltages around mid supply, both the nMOS and pMOS folded cascode stages get activated and hence the overall trans-conductance of the combination is $g_m + g_m = 2g_m$ and the overall gain is $2g_m R_{out}$. Since, the purpose of charge pump opamp is to make V_{DS} same at both the nodes in the charge pump, i.e at DC, this fluctuation in the gain is acceptable. The loop gain magnitude and phase plots of charge pump opamp are as shown.

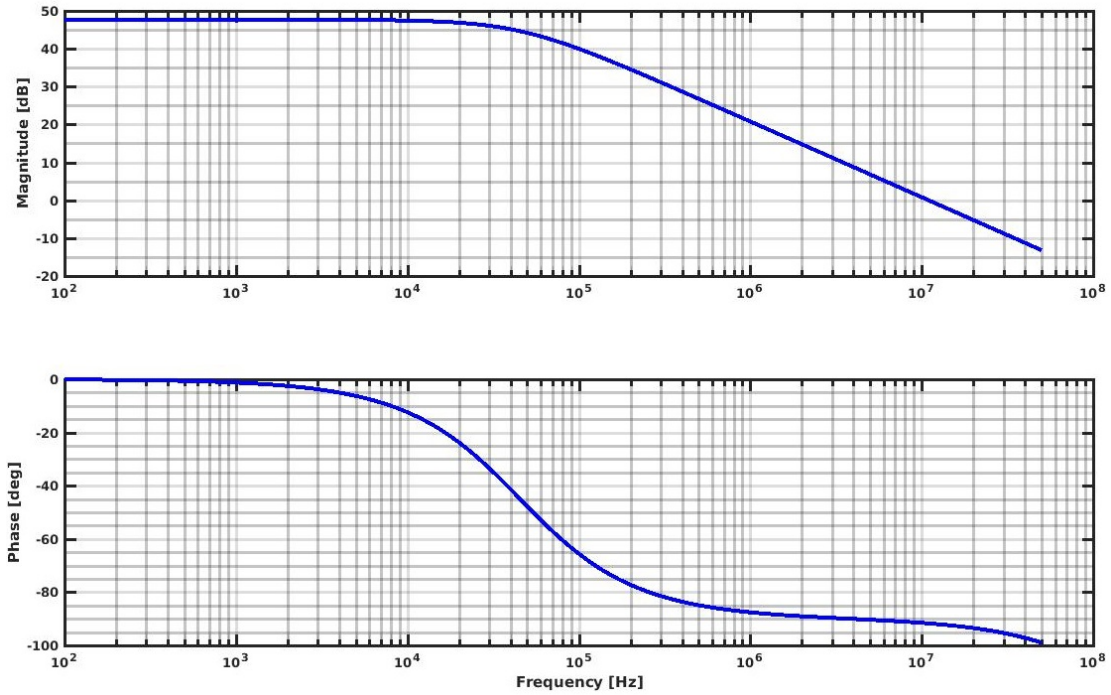

Figure 3.10: Loopgain Magnitude and Phase plots

This opamp has f_{ugb} of 10.96MHz with a Phase margin of 88.3°.

3.6 Regulator loop Opamp

This opamp is different from the charge pump opamp in that the gain of this opamp is uniform across the entire common-mode voltage range. The circuit diagram of this

opamp is as shown.

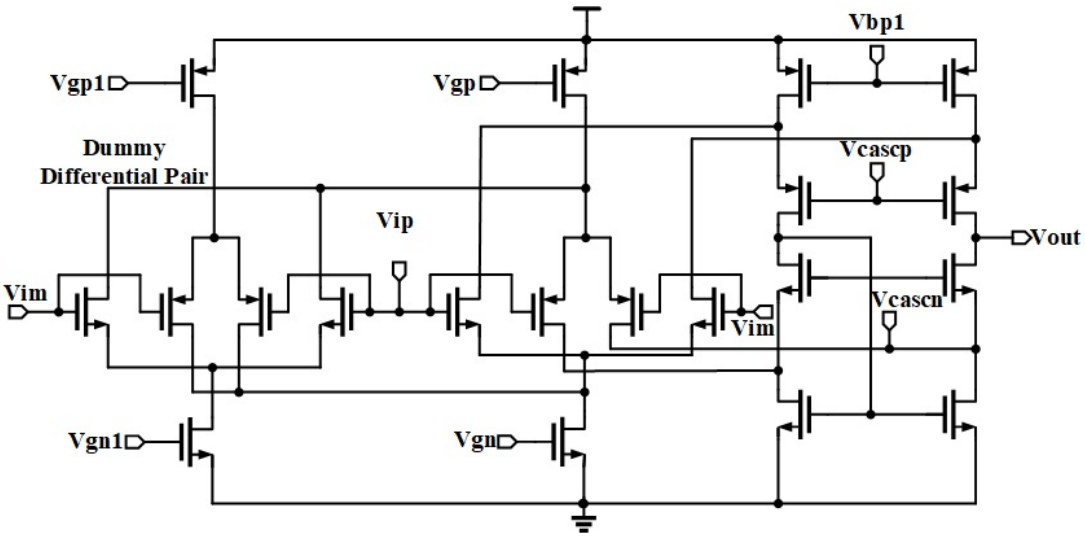

Figure 3.11: Regulator loop opamp

There is a complementary nMOS and pMOS input pair as well as dummy nMOS and pMOS pair. The dummy nMOS pair is connected to the tail current source of the pMOS input pair and the dummy pMOS pair connects to the tail current source of nMOS input pair.The dummy nMOS and pMOS pair serves the purpose of reducing the bias currents through the input pairs when the common-mode voltage is in the mid-supply. The dummy pair will essentially have no effect when the common-mode is near the supply rails. The results of this opamp suggests that the variation in gm proves to be better than that in charge pump opamp. The loop gain magnitude and phase plots of charge pump opamp are as shown.

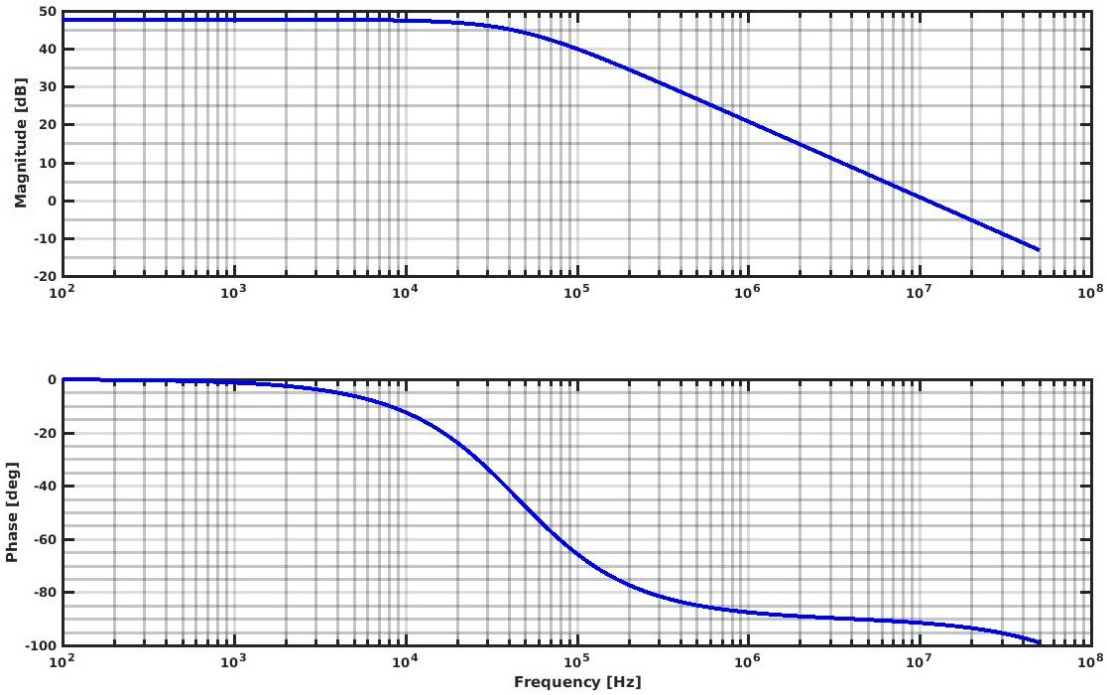

Figure 3.12: Loopgain Magnitude and Phase plots

This opamp has f_{ugb} of 10.96MHz with a Phase margin of 88.3°.

3.7 Regulator loop

The regulator loop suppresses any ripple on the supply from reaching V_{VCDL} there by ensuring less jitter at each of the output phases. The circuit diagram of regulator loop is as shown below.

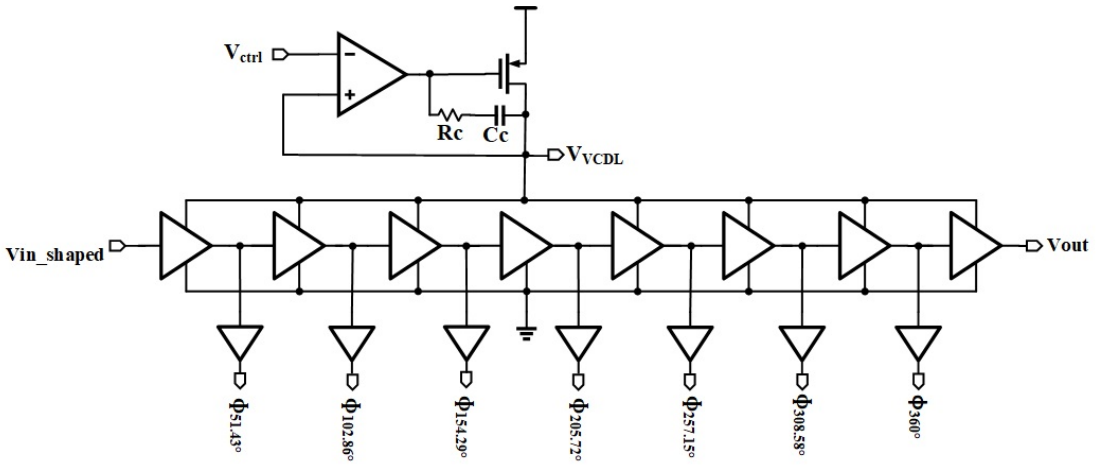

Figure 3.13: Regulator loop

The delay line acts a load to the regulator loop. The resistor R_c and capacitor C_c are

used for loop compensation.In addition to the task of power supply ripple rejection, this loop also acts as a low pass filter to eliminate the ripple caused at loop filter capacitor node V_{ctrl} node. This is because the periodic voltage waveform caused due to charging and discharging events of loop filter capacitor can be decomposed into a number of sinusoid and cosinusoid signals with fundamental frequency and its harmonics as per Fourier series. Subsequently, the higher order harmonics get filtered off by the low pass system, and the magnitude of fundamental components gets scaled down by the low pass system magnitude response with a corresponding shift in phase. This is shown as below.

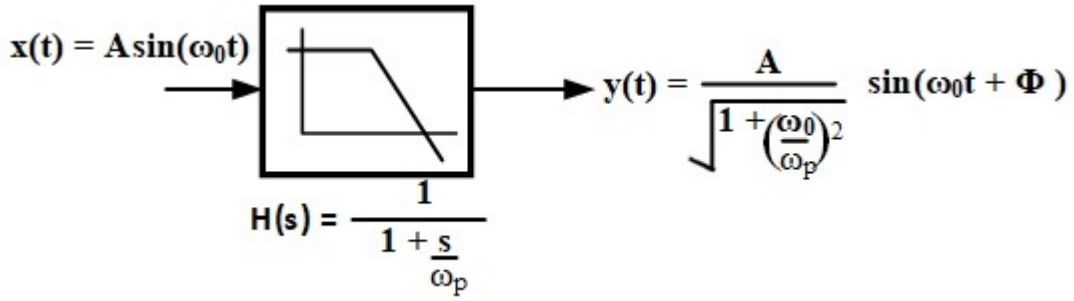

Figure 3.14: Low pass system response to periodic input

where $\phi = -tan^{-1}\frac{\omega_0}{\omega_p}$. Thus, the ripple gets attenuated at V_{VCDL}. The V_{ctrl} values required for the delay line to work across all process corners and temperatures is tabulated as shown below. The loop-gain magnitude and phase plot and PSNR of regulator

Table 3.1: V_{ctrl} values across all process corners and temperatures

Sl.No	Chain Delay (ns)	TT@$27°C(Volts)$	FF@$-40°C(Volts)$	SS@$125°C(Volts)$
1	33.33	0.83	0.78	0.89
2	25	0.93	0.85	1.03
3	20	1.04	0.93	1.17
4	16.67	1.14	1.00	1.31
5	13.33	1.30	1.12	1.54

loop are as shown below.

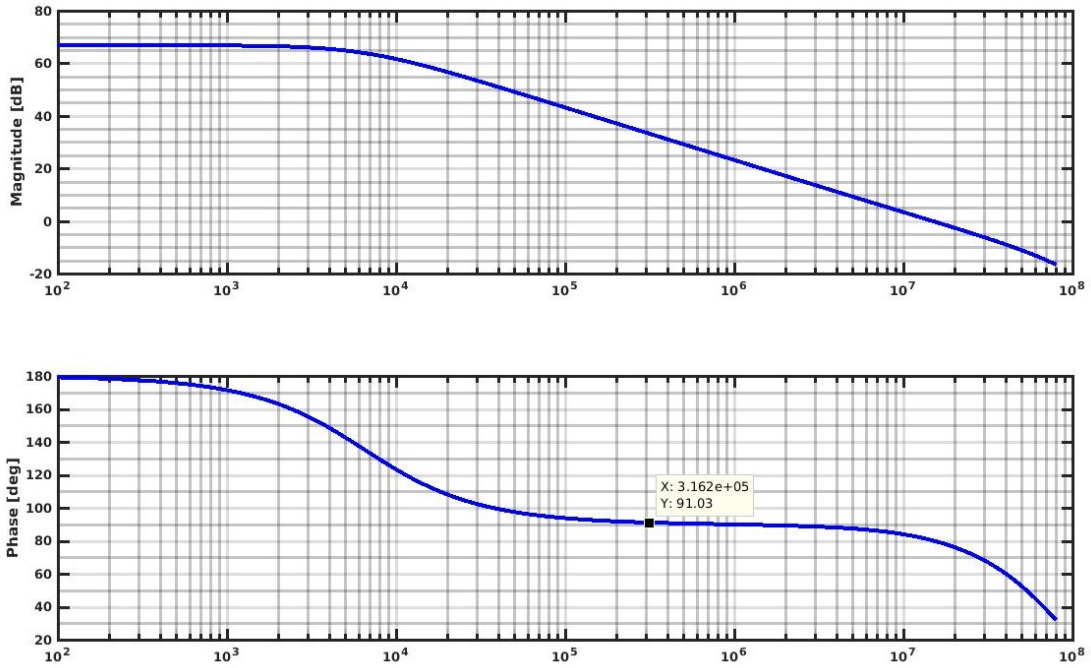

Figure 3.15: Regulator open loopgain response

The regulator loop has f_{ugb} of 14.79MHz with a Phase margin of 80.2°.

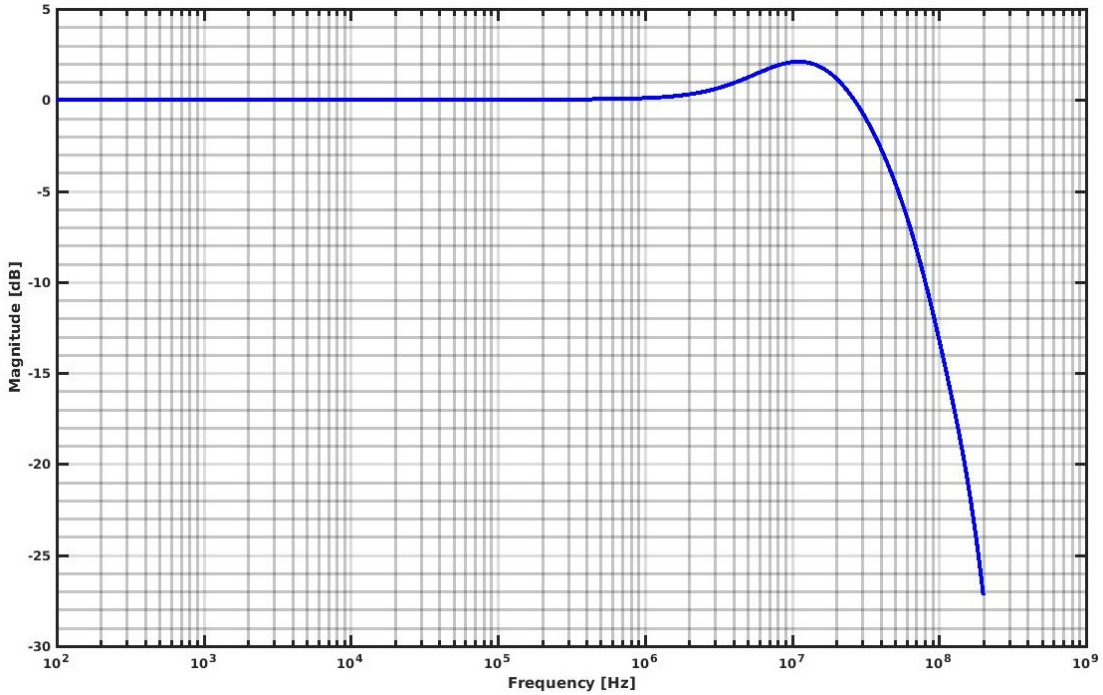

Figure 3.16: Regulator closed loopgain response

From the above figure, the -3dB bandwidth of regulator f_{-3dB} is 42MHz.

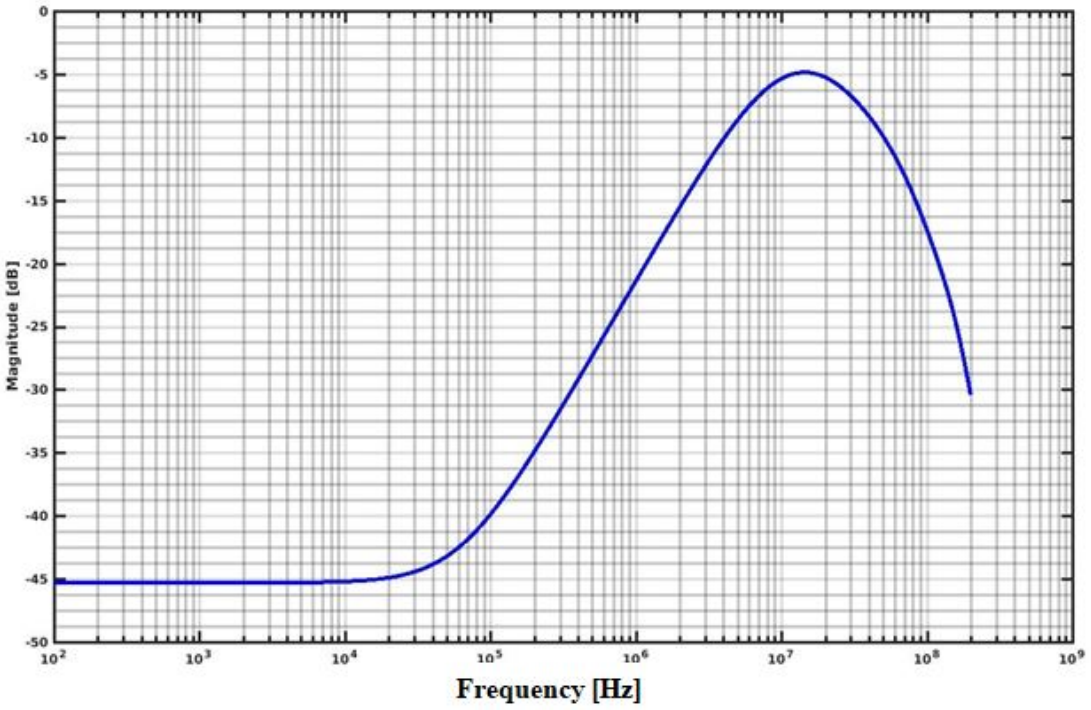

Figure 3.17: Regulator closed loopgain response

3.8 Divider

The Divider block is implemented as shown

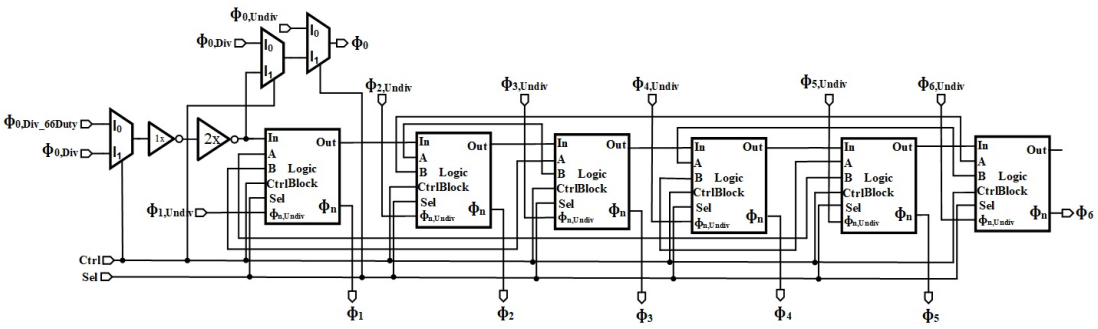

Figure 3.18: Frequency Divider block

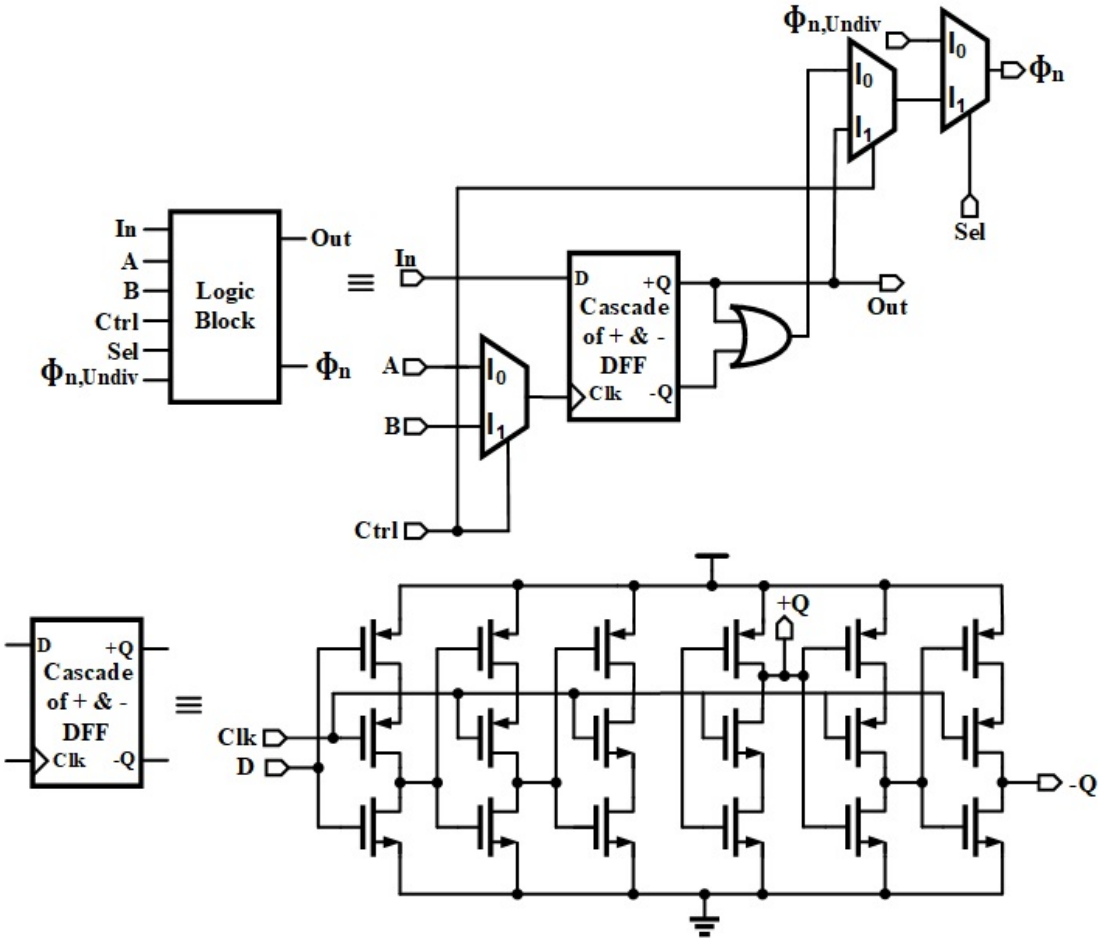

Figure 3.19: Cascade of +ve and -ve edge triggered D Flipflops

Based on the select signal 'Sel', the divider block either divides the clock phases by 2/3 or pass undivided phases to the ouputs. If control signal 'Ctrl' is '0', the divider block divides the clock phases by 3 and performs duty cycle correction to restore the duty cycle of each clock phase back to 50 % and pass them to outputs. On the other hand, if 'Ctrl' is '1', the divider block divides the clock phases by 2 and pass them to the outputs. In this case, no duty cycle correction is required.

The 2/3 divider circuits is as shown below.

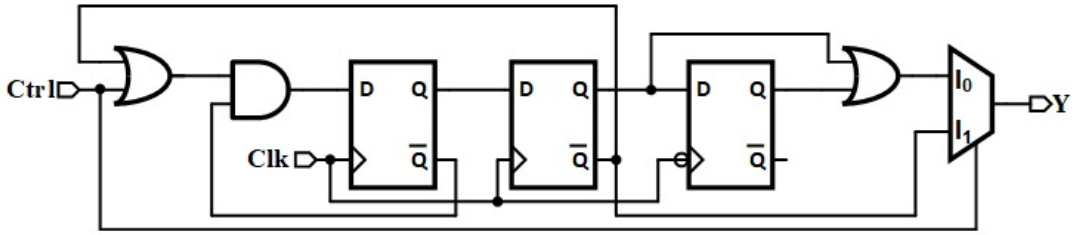

Figure 3.20: 2/3 Divider

3.9 Buffer Block

Keeping in view the routing capacitance between the Regulator loop and Divider block, Buffer block is placed so that the phases from regulator loop have sufficient drive strength.

CHAPTER 4

Implementation of Serializer

The serializer converts data from multiple channels into a single data stream at a higher data rate. This conversion is achieved by multiplexing the data bits from multiple channels with in the serialized bit period.

The ciruit diagram of the Serializer implemented in this thesis is as shown below.

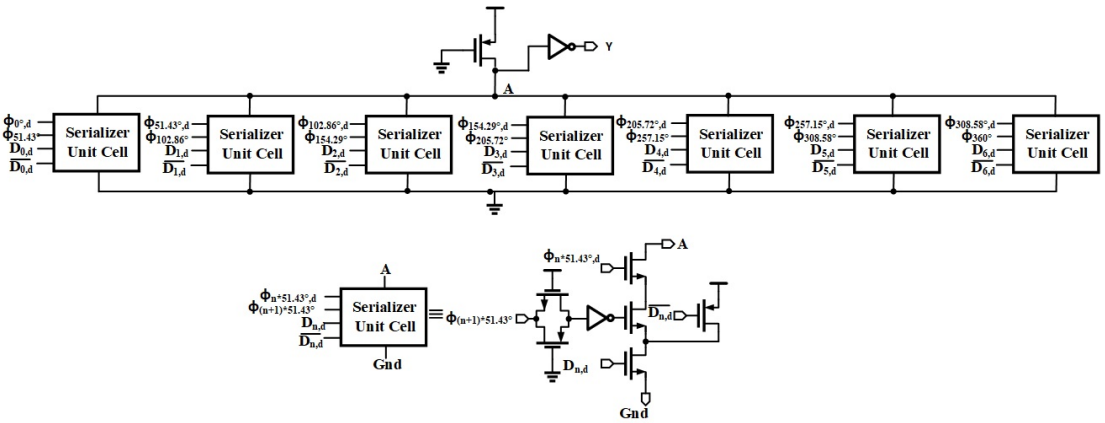

Figure 4.1: Serializer

The inputs to the serializer are clock phases generated by the DLL. Time-division parallel to serial data conversion is achieved by gating each transmission path sequentially. Since, only a single path gets activated at a time, there is no interference from other paths in the circuit. In the phase difference between $\phi_{n*51.43^\circ}$ and $\phi_{()n+1)*51.43^\circ}$, data bit D_n is sandwiched and the node potential at A is either pulled to ground or pushed to supply depending on the data bit D_n. A pMOS precharge transistor is added in each path to precharge the internal node to a high level before a '1' is sent out after transmitting a long sequence of '0's. This technique alleviates the charge sharing effects, which significantly suppress systematic data jitter. Since, the serializer is implemented using CMOS logic which is an inversion logic, an inverter is placed at the output to negate the effect of inversion caused by CMOS logic and to obtain the correct data. The RC extracted eye diagram of the serializer when simulated at the highest data rate with a Pseudo Random Binary Sequence (PRBS) is as shown below.

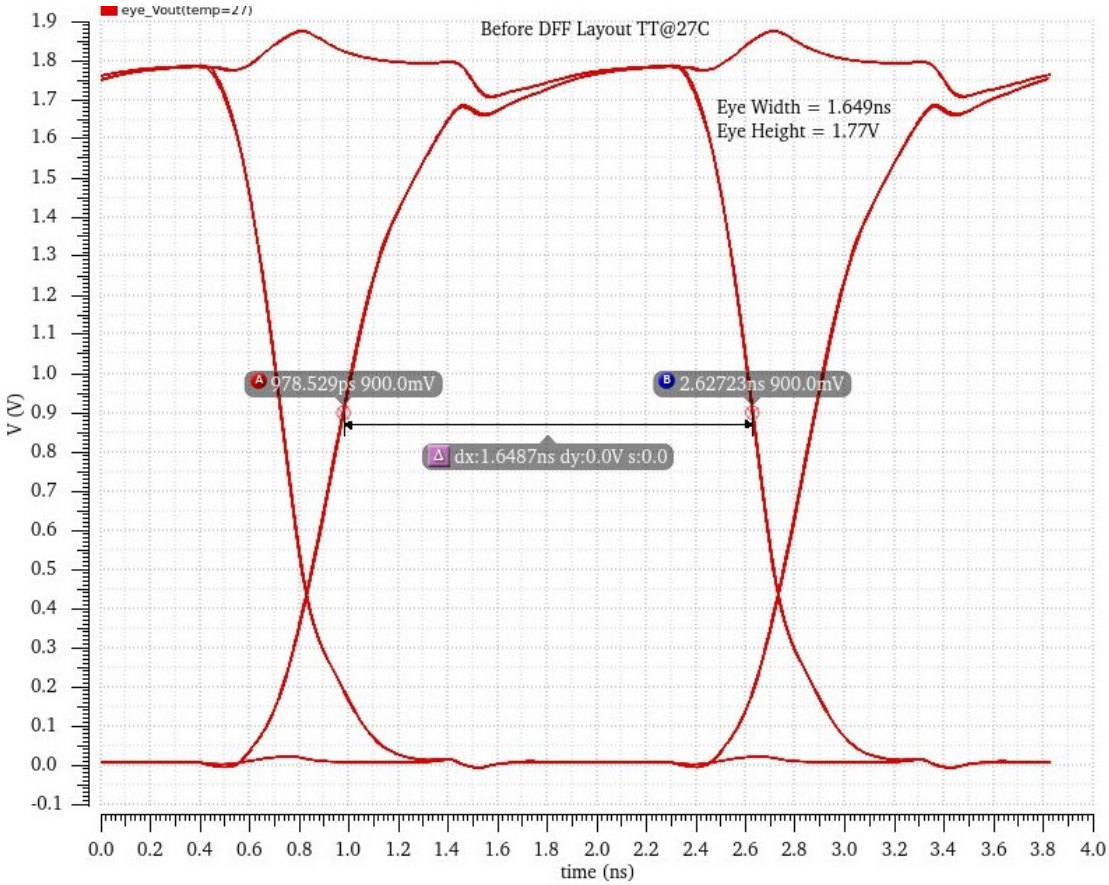

Figure 4.2: Serializer Eye Diagram

4.1 Synchronizer

The synchronizer is implemented as a Sense amplifier based D Flipflop. Assuming, that a clean clock is generated by a PLL at the same rate as serialized data stream, the Serialized data is given as input to synchronizer which is clocked by PLL output,a serialized data stream with the same jitter as that of PLL output is obtained.

The synchronizer remain active only during the positive half cycle of clock signal and in the negative half cycle, both the outputs R and S regenerate to the supply voltage. Hence, an RS latch is used to latch the decision made by the sense amplifier exactly at the end of positive half cycle.

The implementation of synchronizer is as shown below.

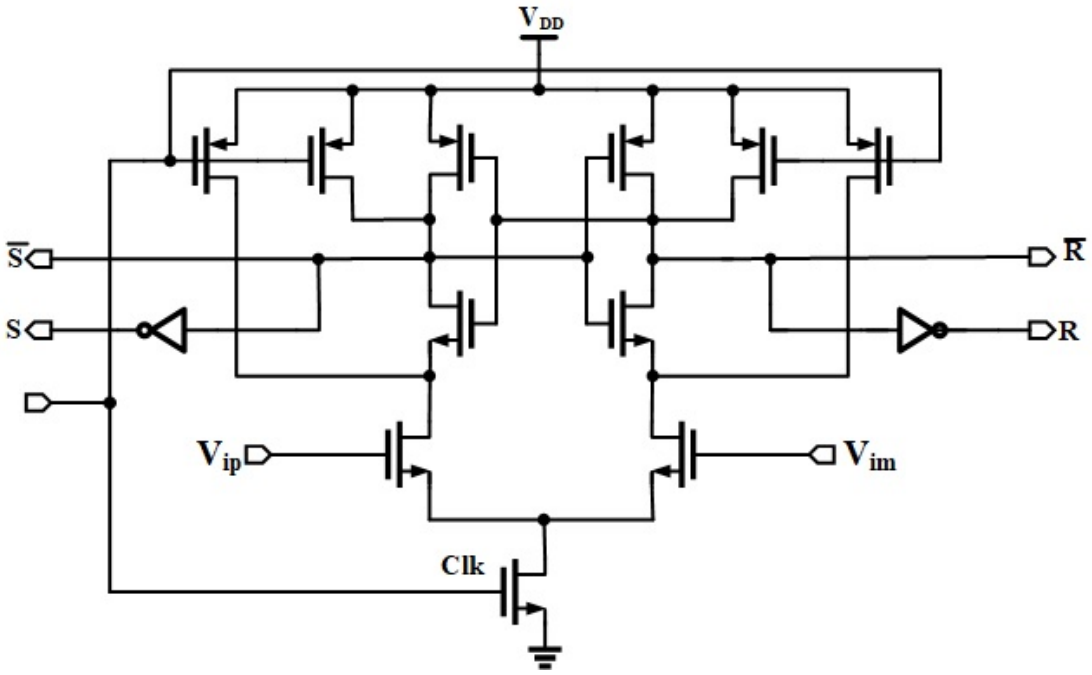

Figure 4.3: Sense amplifier

When CLK = '0', all the intermediate node potentials are pulled to supply, as all the top pMOS transitors turn ON. When Clk = '1', all the top pMOS transistors get turned OFF and the supply potential at the intermediate nodes from the previous half cycle gets stored in the parasitic capacitances at the respective nodes. Assuming $V_{ip} > V_{in}, M_0$ turns ON and the intermediate node potential (supply) from the previous cycle causes nMOS transistors M_2 and M_3 to turn ON and pMOS transistors M_4 and M_5. As a result, the intermediate node potentials at RBAR and SBAR begins to fall. Because of the positive feedback due to back-to-back connected inverters, at a particular instant, either side of the pMOS transistor in inverter turns ON and nMOS transistor on the other side turns OFF causing $\bar{R}$ to go high and $\bar{S}$ to go low. Thus, the difference in $V_{ip} - V_{in}$ is resolved completely. In order to latch this decision during the negative half of clock cycle,, RS flipflop is used as shown in the schematic below.

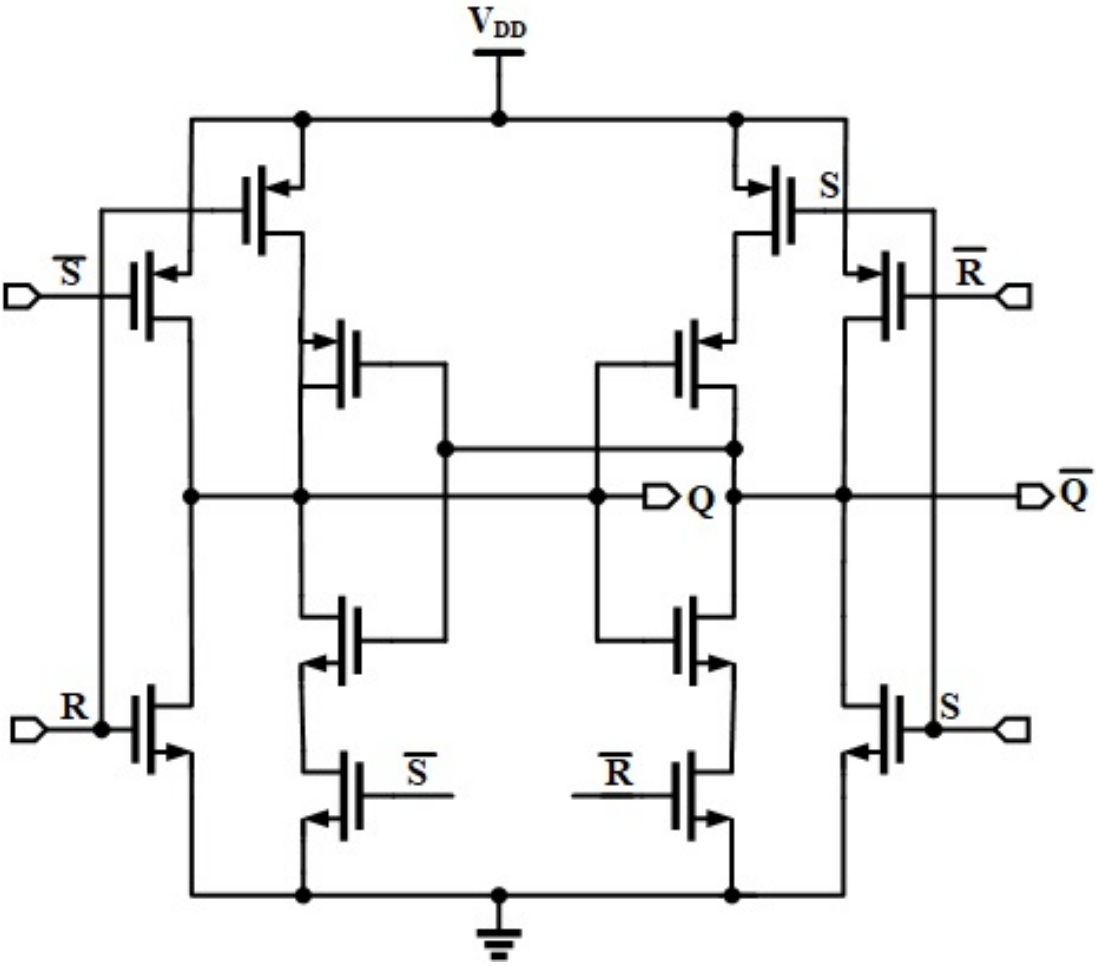

Figure 4.4: RS Latch

The advantage of this RS latch is that Q and $\bar{Q}$ signals are generated less than an inverter delay.

Sense amplifier + RS latch = Synchronizer

This is further explained by the following diagram which is self-explanatory.

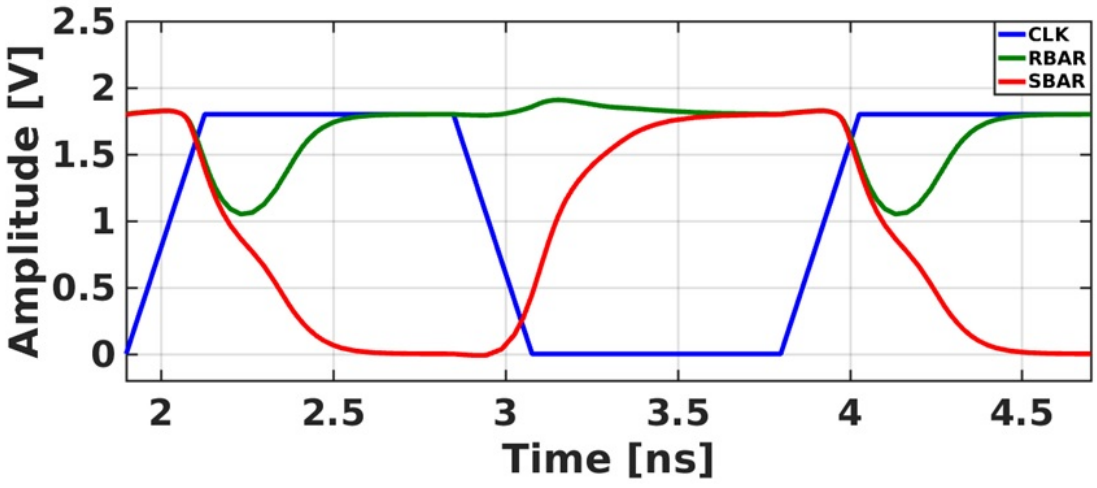

Figure 4.5: Response of Synchronizer for $V_{ip} > V_{in}$

The integration of synchronizer with the serializer is as shown below.

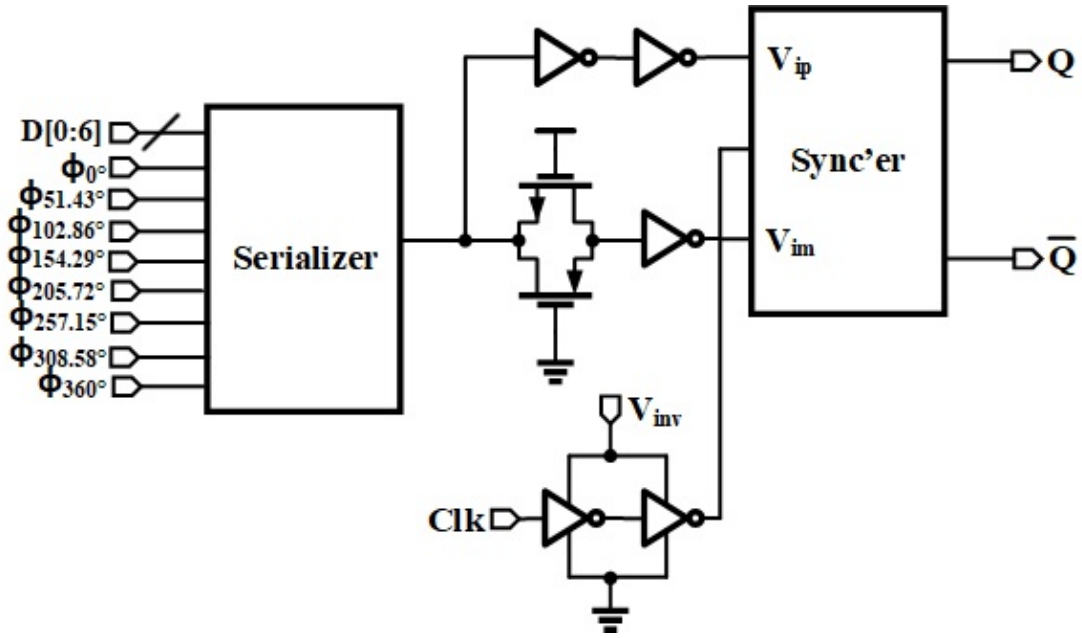

Figure 4.6: Serializer with Synchronizer

The resulting eye diagram at the synchronizer output Q when the combined circuit is simulated at the highest data rate using a PRBS sequence is as shown below.

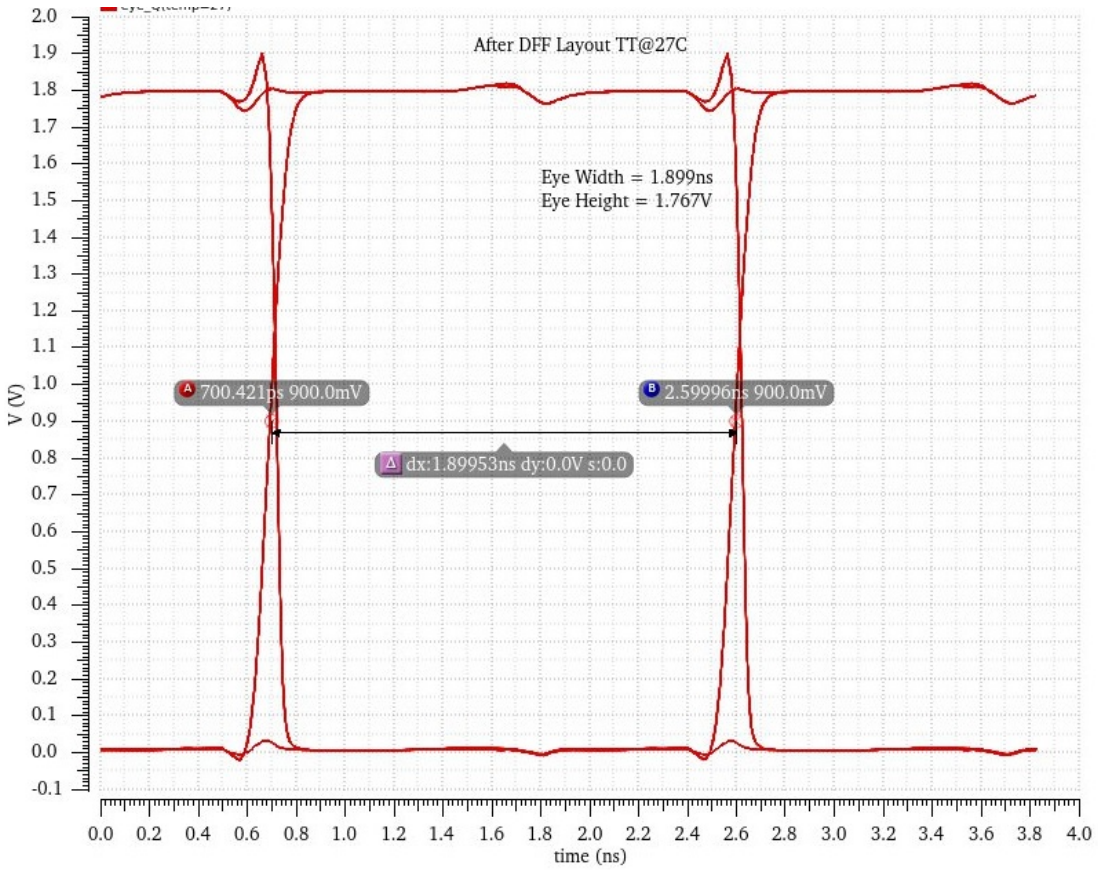

Figure 4.7: Serializer with Synchronizer Eye Diagram

clearly, the synchronizer output has less relative peak to peak jitter compared with serializer output.

CHAPTER 5

Implementation of PLL

The PLL is impleneted as shown below.

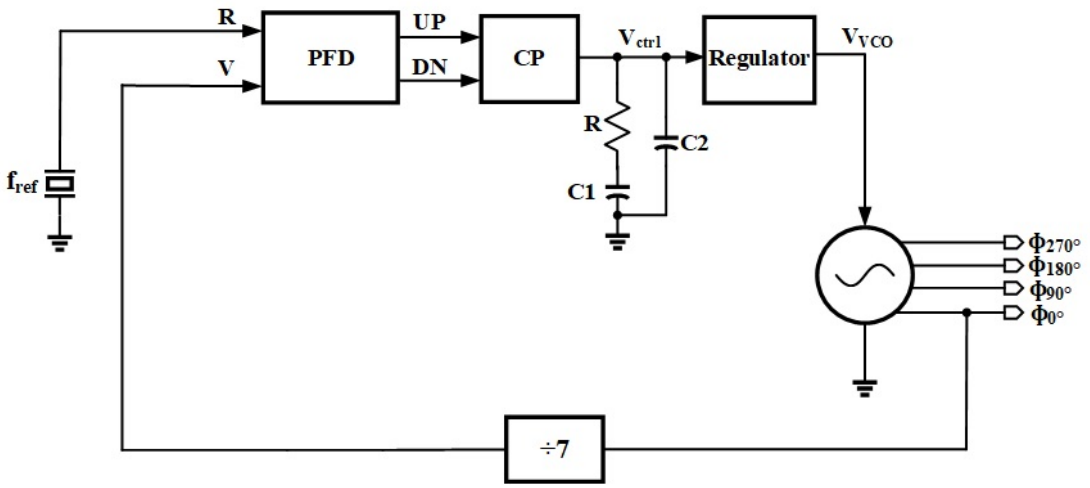

Figure 5.1: Block diagram of PLL

5.1 PFD

It is the primary block of a phase locked loop, which compares phase and frequency of divided signal with the reference signal and generates UP and DN pulses based on the difference between the two signals.

A conventional 3-state PFD has a problem called dead-zone,where the PFD can not identify small phase differences at the input owing to finite reset time limited by the technology.

Its other counterpart, NAND Based PFD shown in the figure can prevent the problem of dead-zone by having more reset delay so that UP and DN pulses are properly generated and this configuration detects small phase differences and hence improves the frequency acquisition range. Reset delay of this configuration is given by $\mathrm{T_{RST}} = 2\mathrm{T_{NAND2}} + \mathrm{T_{NAND4}}$.

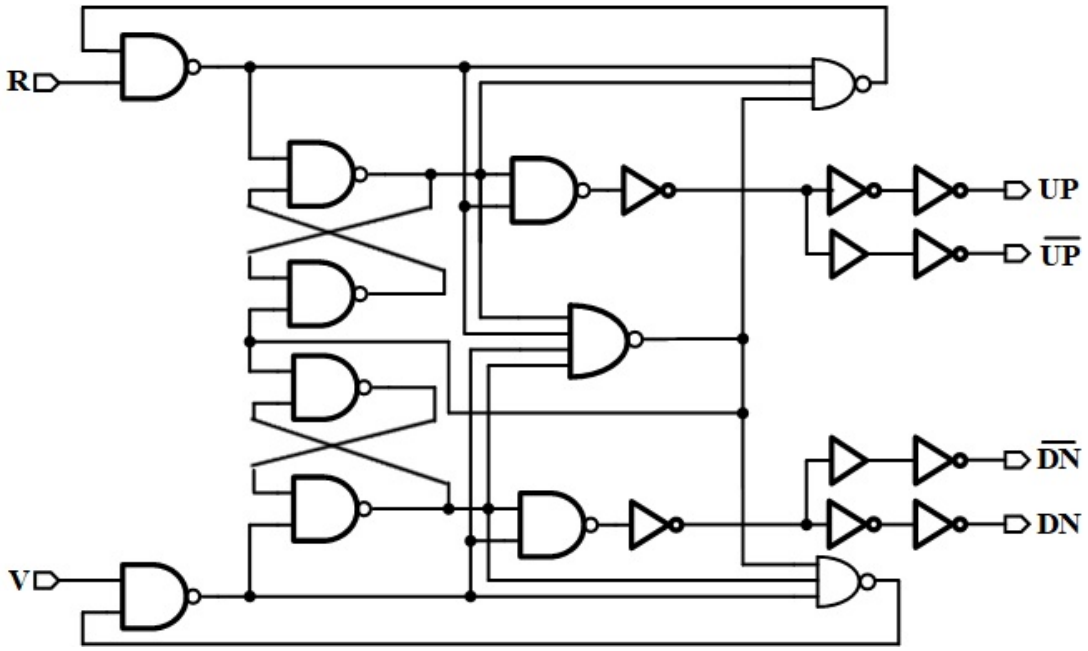

Figure 5.2: NAND based PFD

5.2 Charge Pump (CP)

The charge pump converts the UP and DN input pulses coming from PFD into current pulses that charge/discharge the loop filter capacitor to give the desired control voltahe V_{ctrl}. Due to a phase difference between R and V signals, PD produces UP and DN pulses causing I_{CP} current to flow into/out of the loop filter capacitor. The UP current causes charge to be dumped on the loop filter capacitor and hence V_{ctrl} increases. On the other hand, the DN current pulse causes charge to be removed from loop filter capacitor and hence V_{ctrl} decreases. In order to maintain a good match between the UP and DN currents, boot strapped charge pump with current steering DAC topology was used in this thesis.

In this above topology, four bits are used to control the current flowing into loop filter to have tunability over loop bandwidth so that the PLL works across wide operating frequency.

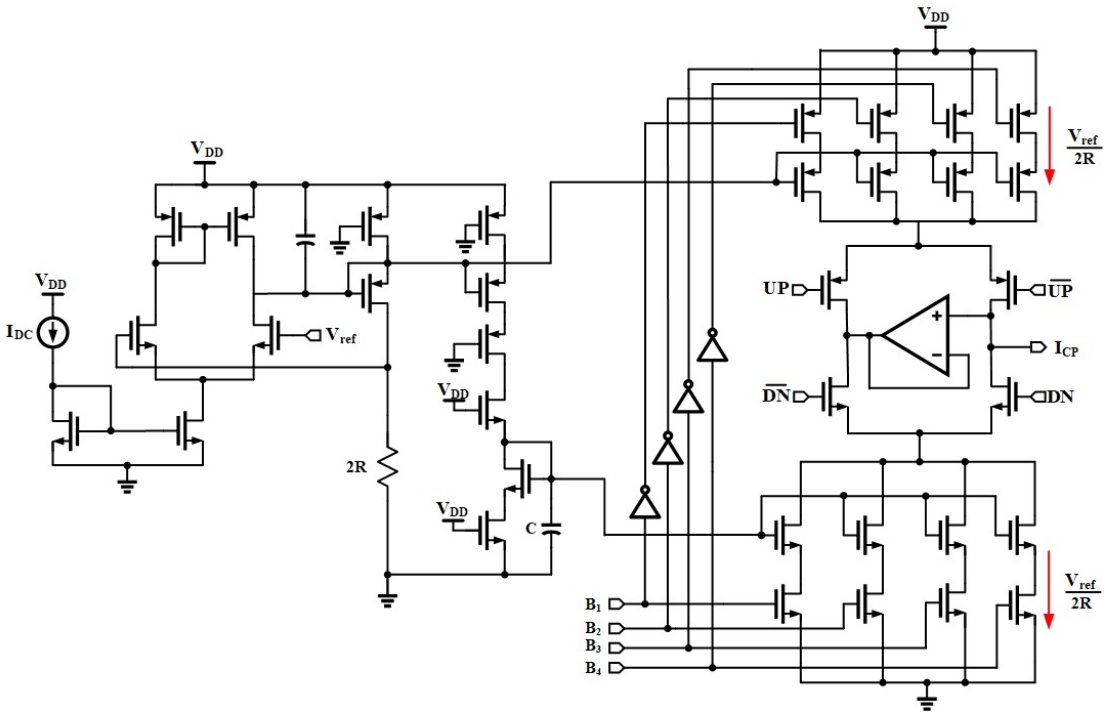

Figure 5.3: Current steering DAC Charge Pump

5.3 Loop Filter (LF)

The transfer function of the loop filter is given by F(s) = $\frac{1+\frac{s}{\omega_z}}{s^2(1+\frac{s}{\omega_{p3}})}$, $\omega_z = \frac{1}{RC_1}$ and $\omega_{p3} = \frac{1}{R\frac{C_1C_2}{C_1+C_2}}$

Pole ω_{p3} is introduced to reduce ripple in control voltage and to suppress the reference spur at the output. Note that $I_{CP}, R, C_1 and C_2$ values are chosen based on phase margin (ϕ_m) and loop bandwidth.

5.4 Regulator loop

The regulator loop suppresses any ripple on the supply from reaching V_{VCDL} there by ensuring less jitter at each of the output phases. The circuit diagram of regulator loop is as shown below.

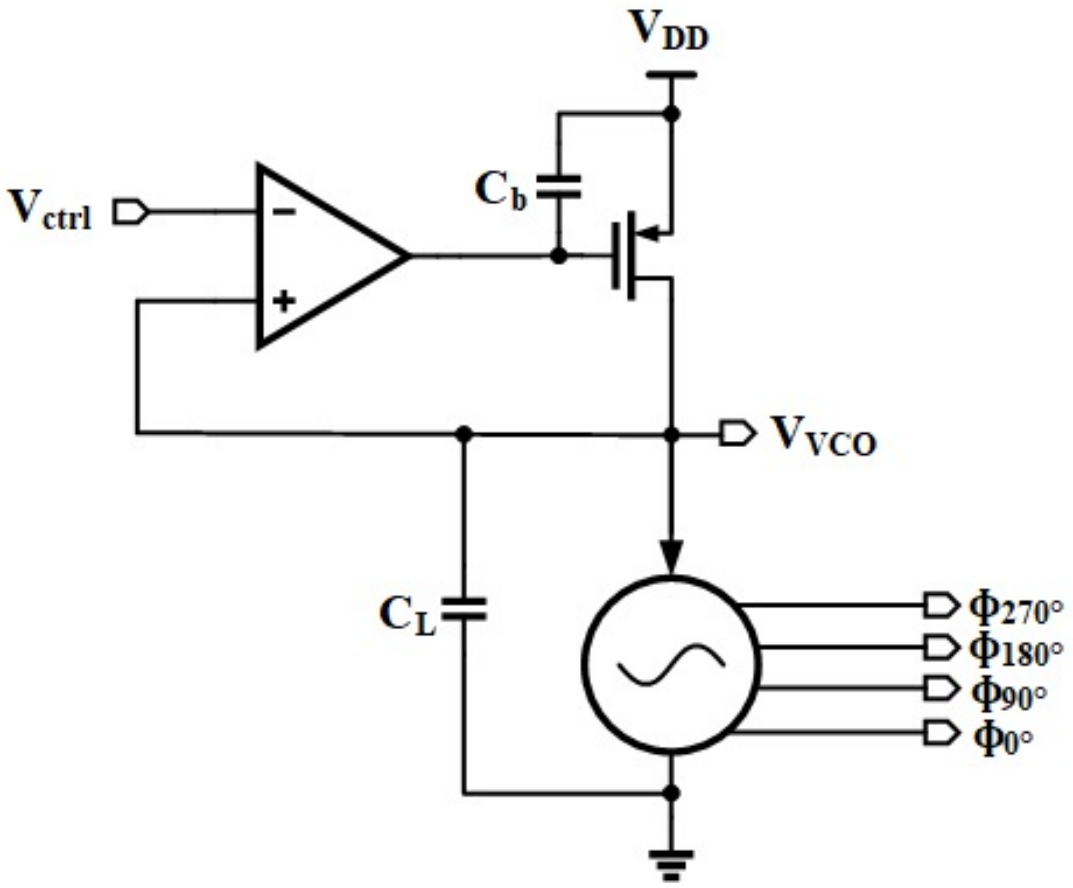

Figure 5.4: Regulator loop

Both the capacitor values C_b and C_L are chosen such that better ϕ_m and PSRR are achieved.

5.5 Voltage Controlled Oscillator (VCO)

This generates periodic waveforms whose frequency varies with respect to V_{ctrl}. If V_{ctrl} increases, output frequency increases and vice versa. Eight inverter based Pseudo differential ring oscillator with cross coupled inverters is designed as shown in figure such that it's output frequency varies from [105MHz - 525MHz] with V_{ctrl} range being [0.68V - 1.48V] across PVT.

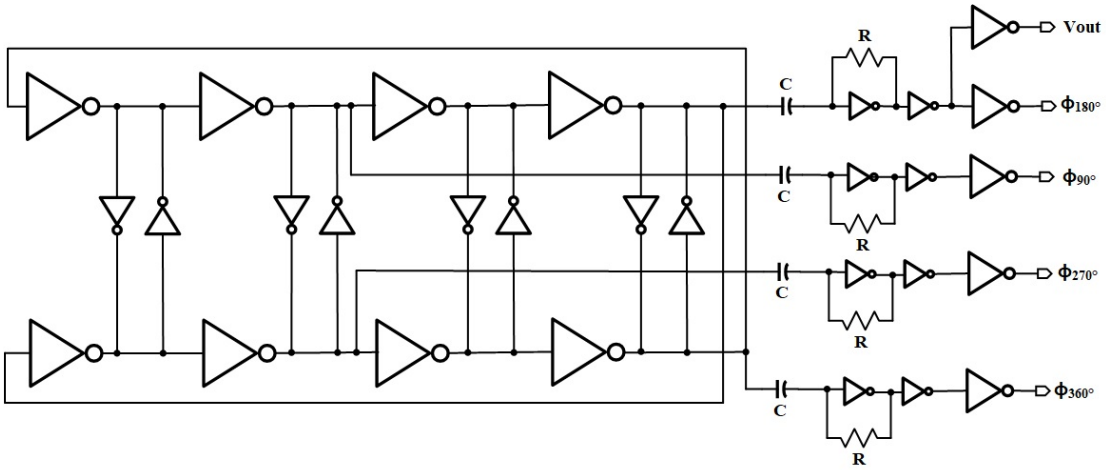

Figure 5.5: Voltage Controlled Oscillator

5.6 Frequency Divider

This divides higher output frequency to lower frequency to match it with reference frequency to have feedback action.Since f_{ref} varies from 15MHz to 75MHz and f_{out} is 105MHz to 525MHz , divide by 7 counter is used. Synchronous divide by 7 counter as shown in figure is designed to have 50 percent duty cycle.

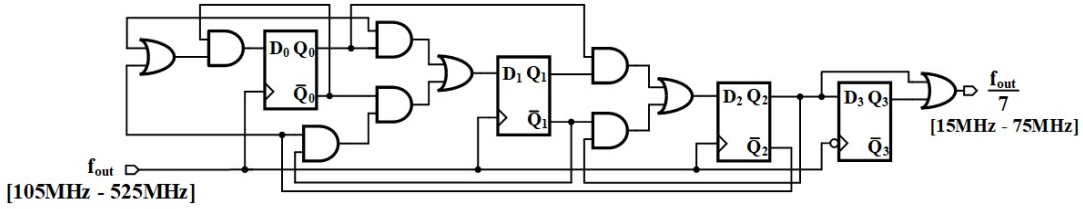

Figure 5.6: Divide by 7 counter

$$D_0 = \bar{Q}_2 Q_1 + Q_1 Q_0;$$
$$D_1 = \bar{Q}_1 Q_0 + \bar{Q}_2 Q_1 \bar{Q}_0;$$
$$D_0 = (\bar{Q}_1 + \bar{Q}_2) Q_0;$$
$$D_0 = Q_2 + Q_3;$$

CHAPTER 6

LVDS Driver

The LVDS driver launches the data bit on to the LVDS channel. For a bit '1' to be transmitted, a differential voltage of 350mV over a common mode voltage of 1.2V is incident on the LVDS channel. For a bit '0' to be transmitted, a differential voltage of -350mV over a common mode voltage of 1.2V is incident on the LVDS channel. The schematic of LVDS driver is as shown below.

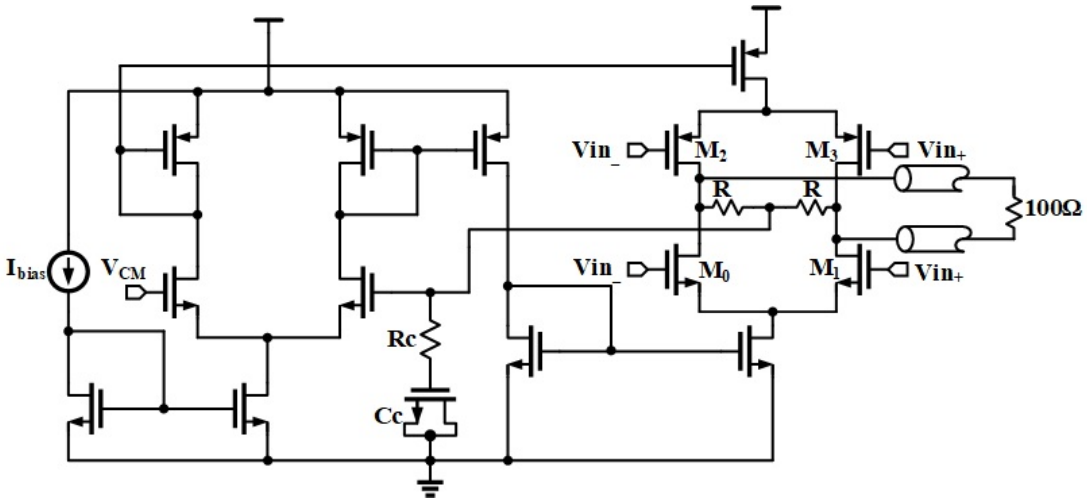

Figure 6.1: LVDS Driver

Since, the LVDS driver sees a differential load impedance of 100Ω, the bias current required to flow through the load for generating a differential voltage of 350mV as per specification is 3.5mA. The left half of the circuit is a Common Mode Feedback Circuit (CMFB) that pins the output DC voltage at 1.2V. The resistor R_c and capacitor C_c are used for CMFB loop compensation.

When bit '1' is sent, switches M_1 and M_2 turn ON and a current of +3.5mA flows through the load generating a differential voltage of +350mV. On the other hand, when bit '0' is sent, switches M_0 and M_3 turn ON and a current of -3.5mA flows through the load generating a differential voltage of -350mV.

CHAPTER 7

REFERENCES

REFERENCES:

[1] **Meng-Tzer Wong and Wei-Zen Chen**, "A 2.5Gbps CMOS Data Serializer", Department of Electrical Engineering, National Central University, Taiwan, 2002.

[2] **Wenjing. Y, Rajesh. I, Amr Elshazly, Brian. Y and Pavan Kumar Hanumolu**, " A 0.7 to 3.5GHz 0.6 to 2.8mW Highly Digital Phase-Locked Loop With Bandwidth Tracking", IEEE Journal of Solid-State Circuits Vol 46, No 8, August 2011.

[3] **Jiu Xiong, Jin Liu, Yuchend Dai and Hlaing Minn**, "An Open-Loop 28GHz 16-Phase Clock Generator in 28nm CMOS", Department of Electrical and Computer Engineering, University of Texas, Dallas.

[4] **Behzad Razavi**, "The Delay-Locked Loop", A Circuit for all Seasons.

[5] **Yan Lu and Ruo He Yao**, "Low-voltage constant-gm rail-to-rail CMOS operational amplifier input stage", Solid State Electronics 52 (2008) 957-961.

[6] **Rien Lerone Beal**, "A Low-voltage rail-to-rail operational amplifier with constant operation and improved process robustness", Iowa State University, 2009.

[7] **Ashwarya Rajwardan**, "Design and simulation of a charge pump phase-locked loop in 65nm CMOS technology", Senior Thesis in Electrical Engineering, University of Illinois at Urbana-Champaign.

[8] **Willy Sansen**, "Analog Design Essentials", The Springer International Series in Engineering and Computer Science, 2006.