A REPORT

ON

DESIGN AND SCHEMATIC IMPLEMENTATION OF DELAY LOCKED LOOPS (DLL)

BY

Megha Jain 2011A3PS081H
Sidharth Sahdev 2011AAPS098H
Mahesh Naidu 2011A3PS263H

In the course

EEE C443/F313 ANALOG AND DIGITAL VLSI DESIGN

AT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,PILANI
HYDERABAD CAMPUS

Academic year 2013-2014

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
HYDERABAD CAMPUS

Students:	**Megha Jain, B.E. Electrical & Electronics** **Sidharth Sahdev,B.E. Electronics & Communications** **Mahesh Naidu, B.E. Electrical & Electronics**
Expert:	**Prof. M.B. Srinivas**
Guiding Faculty:	**Mr. Sreehari**

Duration:	22nd August, 2013 to 21st November, 2013
Date of Submission:	23rd November, 2013
Title of the Project:	Design and Schematic implementation of DLL
Course	Analog and Digital VLSI Design
Key Words:	Delay Locked Loop, phase detector, NMOS, NAND gate, Charge pump, Jitter, Delay line
Project Area :	VLSI Design

Contents

ACKNOWLEDGEMENTS

Firstly, we are grateful to **Department of Electrical Engineering, BITS-Pilani Hyderabad Campus** for offering us Analog and Digital VLSI Design (ADVD) as a course in our curriculum.

We would like to express our sincere thanks to **Prof. M.B. Srinivas**, Department Of Electrical Engineering and Dean (Administration), BITS –Pilani Hyderabad Campus for giving us the opportunity to carry out a project in this esteemed organization.

We would also like to thank Mr**. V. Srihari**, Lecturer and our Project Guide for suggesting us the project and providing us valuable guidance and support throughout our work.

We would like to extend our gratitude to **Mr.Phaneendra,** PhD scholar and our friends who helped us out in every possible manner and supported us from time to time.

ABSTRACT

Generation and distribution of clocked signals inside the VLSI system is one of the most important problems in the design of VLSI systems. Because of process variations, clock signals delays vary for different paths. The clock signals should have zero clock skew, that is to say all the clock signals should arrive at the inputs of registers at the same time. Otherwise the components of the circuit (latches, flip-flops etc.) get clock signals at different time instances. In order to operate the circuit correctly, these differences should be eliminated, ideally to zero. One very important and upcoming way of eliminating these differences is Delay Locked Loop.

This report presents design and implementation of Delay locked Loop. It deals with various components of DLL like phase detector, charge pump, delay line etc. It begins with describing the general outlay of DLL and then went on to explain each component in detail. Special focus has been laid on designing the phase detector. A comparison between different phase detectors has been made and phase frequency detector has been implemented. Report also presents the schematic layout and implementation of other DLL components like charge loop and delay line. Simulations have been made for the phase detector using electric and results have been shown.

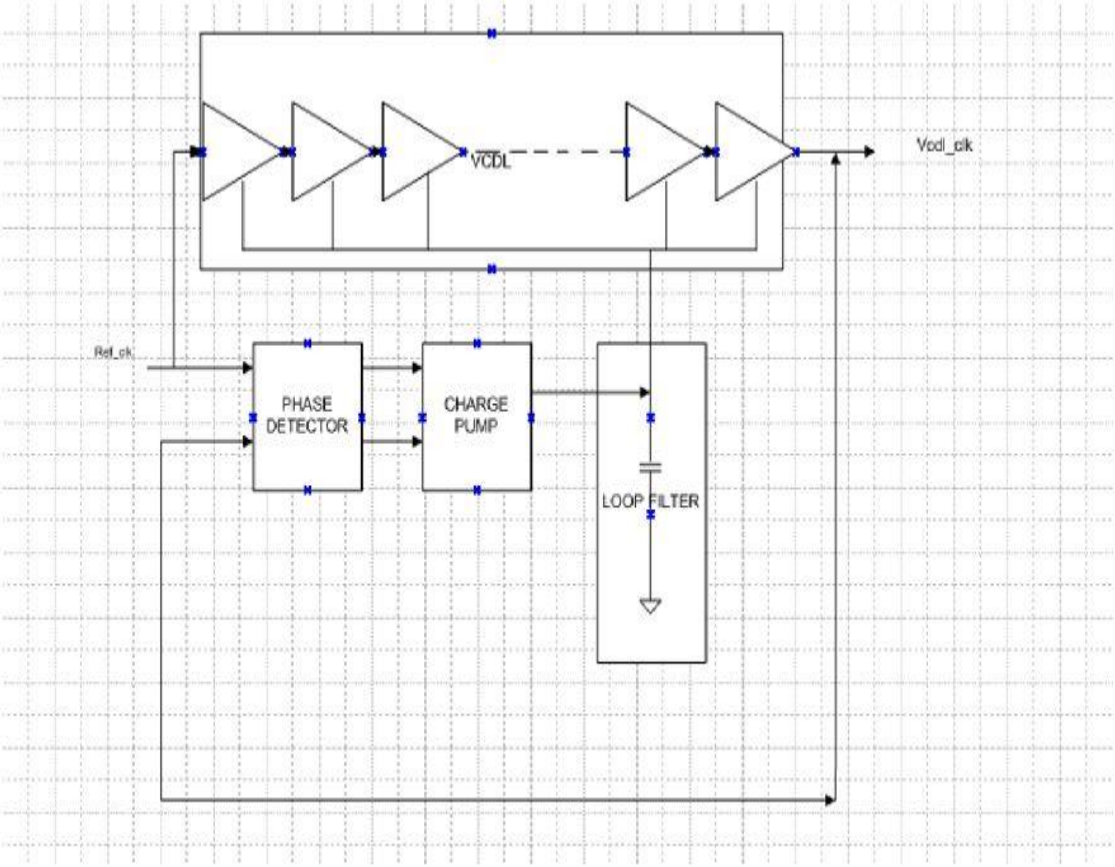
INTRODUCTION

The reduction of the clock skew is one of the important problems in VLSI design. Passive techniques such as clock network optimization cannot completely reduce the clock skew. Phase locked loops (PLLs) and delay locked loops (DLLs) are extensively used in VLSI circuits in order to decrease clock skew in the clocked networks. However, our main focus in this project will be DLL.

Broadly and practically speaking, a DLL is used in integrated circuits of any technological gadgets that use chips to make them run, like computers, or anything that uses a timed loop of circuitry to make it function efficiently and is automated. They are very important for the regulation of voltage coming in and out of the system.

DLL is a 'first order loop' that compares its input with a reference signal, than delay its output so that it can synchronize with the reference signal in a feedback fashion. It compares the phase of its last output with the input clock to generate an error signal which is then integrated and fed back as the control to all of the delay elements. The integration allows the error to go to zero while keeping the control signal, and thus the delays, where they need to be for phase lock. Since the control signal directly impacts the phase this is all that is required. The main components of DLL are as follows:

1. Phase detector
2. Charge pump
3. Filter
4. Voltage controlled delay line (VCDL)



Functions of the above components will be described in detail along with their schematic implementation during the course of this report.

A DLL controls a voltage-controlled delay line, which typically has many taps, in order to bring one of those taps into phase alignment with a reference signal. The input to the delay line is usually also the reference signal, so the various taps provide additional signals that are interpolated and/or extrapolated from the period of the reference signal.

For a DLL, locking time, lock range and jitter performance are the most important metrics. Locking time refers to the time interval a DLL takes to achieve a stable locking state from an initial state. Generally, locking time is related to the speed of the PD, the magnitude of the charging or discharging current in the CP, and the overall delay loop bandwidth. Lock range refers to the maximum and minimum delays of the VCDL, which set the range in which the delay of the VCDL can be varied. A DLL is able to achieve lock only in this range. Lock range directly affects the operating frequency range of a DLL. Phase noise, or time jitter, is the random variations of the period or phase of a clock signal. With the constantly rising data rate or clock frequency, the clock period becomes increasingly smaller, and so does the tolerance to the amount of time jitter. Therefore, a great deal of design effort should be aimed to improve

the jitter performance of a DLL.

DLLs are commonly used in high-speed communications among chips on a board (e.g., between a memory controller and its SDRAM chips) in order to "cancel out" things like input and output buffer delays as well as wiring delays, allowing very tight control over setup and hold times relative to the clock signal. This allows data rates to be much higher than would otherwise be possible.

DLL is very similar to PLL in its functioning with only a few differences. The biggest and most notable difference is that the voltage-controlled oscillator is not present like PLL, rather a delay line exists. DLL's advantage is that it can enhance the output timing of ICs or integrated circuits because it is self-regulating with its delay line. It gives periodic waveform consistently, and can be programmed or designed to become fully digital because it has the capacity to give constant delays or loops every time.

DLL and PLL can be used alternatively, but the PLLs are prone to frequency errors which gives an edge to the DLL system or circuit loop to rise above and be used by engineers more often. The factor involving no oscillator, as discussed earlier, makes DLL a favourite. Nonetheless, DLLs and PLLs functions for clock delays still do not change, and it's important to consider which one would work better for a circuitry project.

LITERATURE REVIEW

Delay locked loop is one of the most widely researched topic across world. Many new versions of DLL have been tried and implemented which offer their own advantages.

DLL overview

DLLs are widely employed in microprocessors, memory, and communication ICs in order to reduce on chip clock buffering delays and improve I/O timing margins. They can also be used to generate multiple clock signals on chip for applications such as for BIST circuits. The essential function of a DLL is to achieve phase alignment between the input clock and the output clock from the final stage of the VCDL. After the phase alignment is achieved, each VCDL delay stage is able to provide a stable clock signal which is phase shifted from the input clock. However, the rising clock speeds and integration levels of digital circuits have made the phase alignment task increasingly difficult . For example, power supply and substrate noise resulting from the switching of digital circuits affects the operation of DLL and leads to output clock jitter. Thus, a detailed study of prior work in DLL design and analysis is needed for us to achieve the design goals of wide lock range, low jitter, and fast locking.

DLL operation principle

A DLL is essentially a nonlinear negative feedback system. However, it is a common practice to characterize a DLL by linear analysis. Although linear analysis is not able to produce a very accurate result, it can still serve as a reasonable first-order approximation and can lead to some useful insights into a DLL is operation. In a DLL, the input clock signal propagates through the VCDL and develops phase shift (or time delay) at every delay stage of the VCDL. The phase shift of each delay stage is controlled by the voltage of a loop filter. The output is taken from one of the delay stages. The phase of the output signal is compared with the phase of the input clock in the PD. The phase error information generated by the PD (usually in the form of 8 a voltage or a current) is then transferred to the CP. The CP uses the phase error information to adjust the voltage of the loop filter and thus to change the delay of the VCDL. Owing to such a negative feedback mechanism, the phase error is gradually reduced until it finally becomes zero. At that time, the delay of the whole VCDL line becomes equal to one clock period, and the voltage of the loop filter is stabilized, which indicates that a locked state has been established.

Analysis of DLL jitter

Typically, jitter includes deterministic jitter as well as random jitter. Deterministic jitter is also referred to as systematic jitter and is generally caused by duty cycle distortion and device mismatch. In contrast, random jitter, which is also known as nonsystematic jitter, is mainly due to some random noise sources such as thermal noise, substrate noise, and power supply noise. The primary random noise sources in a DLL consist of the input clock noise and the noise originating from the VCDL buffer stages. The input clock noise is the noise associated with the input clock signal, while the VCDL noise refers to the noise originating from the MOSFET devices of each buffer stage as well as the substrate and power supply noise coupled to the VCDL. All these noise sources cause jitter in the DLL output signal. The precision and resolution requirements of the BIST application impose a strict limit on the allowable jitter in the output signal. As a result, minimizing random noise is one of the major considerations in VCDL design. The noise sources for each MOSFET device include channel thermal noise, channel flicker noise, and the noise caused by the gate leakage current.

DESIGN AND SIMULATION OF DIFFERENT DLL COMPONENTS

PHASE DETECTOR

Theory and concepts

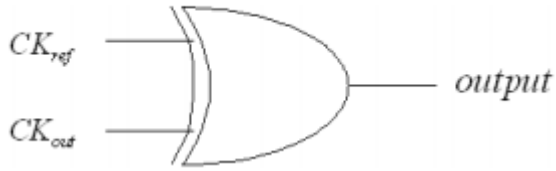
The Phase Detector is one of the most critical components within a DLL.The job of the PD is to detect the phase difference between the output signal from the VCDL and the input reference signal, and to pass the phase error information to the subsequent circuits

One of the structural differences between an analog and a digital DLL is that the inputs to the PD are different. Both analog and digital DLL takes the input reference clock as one input. For the other input to the PD, a digital DLL

usually uses the output of a digital tap whose phase is the closest to the input reference signal, and this tap is not necessarily the last tap of the VCDL. While for an analog DLL, since the delay of the entire VCDL must be one clock cycle in the locked state, the other input to the PD is directly taken from the end of the VCDL.

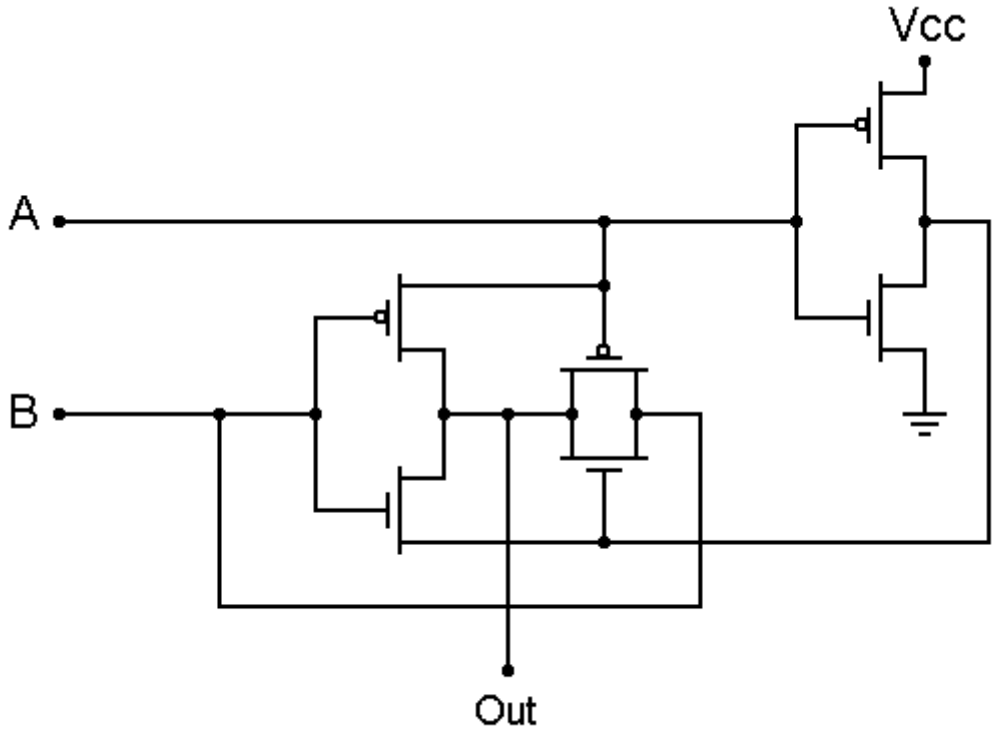
Different types of Phase detectors

1. **XOR PD (Exclusive OR Phase Detector)**



The output of an exclusive OR gate is only high when the first **OR** the other input two is high. If both inputs have the same logical state, either high (1) or low (0), the output of an XOR gate is low (0). This can be simplified in saying that the output of an exclusive OR gate is low (0) if the input states are the same. If the input states are different, the output will be high (1).

If one applies a rectangular signal with exact same parameters, such as frequency and amplitude, to an XOR gate, the output of the XOR gate will be a logical 0 if the phase difference between the two signals is 0°. This is because the two signals have the exact same logical state at any given point in time.



The opposite happens if the signals have a phase difference of 180°. Remember, 180° phase difference means that the second signal is a perfect mirror of the first signal. With 180° phase difference the second signal will be at a logic low (0) state if the first signal is high (1) and vice versa. Because the signals have exactly opposite logical states at any given point in time, the output of the XOR gate will constantly be high (1).

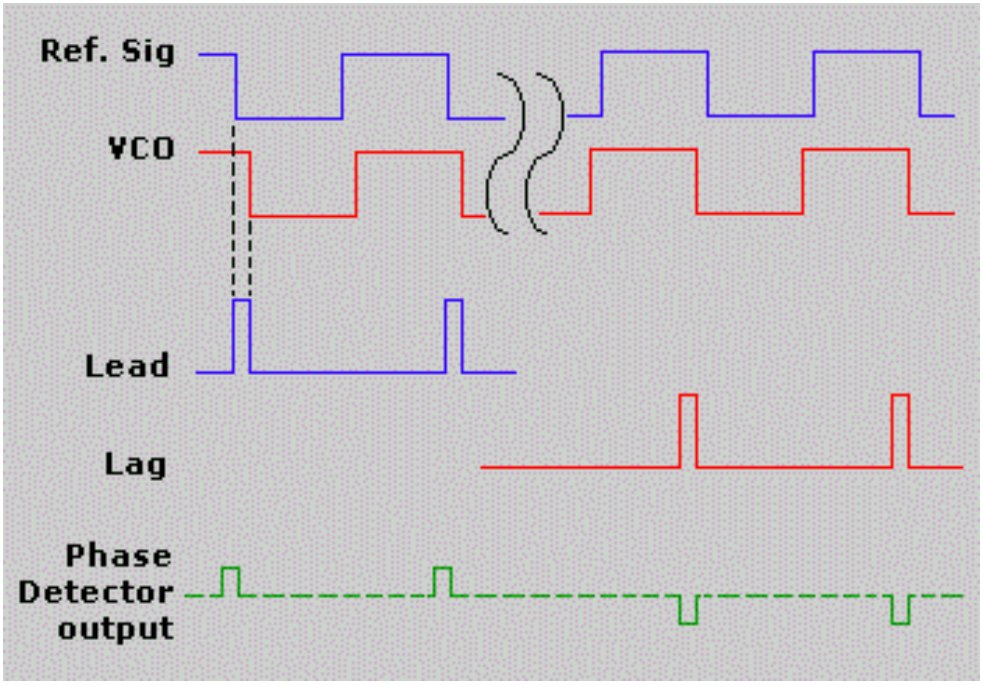
Disadvantages

The major disadvantage of XOR Phase Detector is that it can lock onto harmonics of the reference clock signal and most importantly it cannot detect a difference in frequency. To take care of these disadvantages, we have implemented the Phase Frequency Detector, which can detect a difference in phase and frequency between the reference clock signal and feedback signals. Due to this reason we decided to go for Phase Frequency Detector.

2. Dynamic Phase detector

A third type of PD, which is widely used recently in high-speed DLL designs, is a dynamic PD. The basic structure of a dynamic PD includes two blocks, which are used to generate the UP signal and the DOWN signal, respectively.

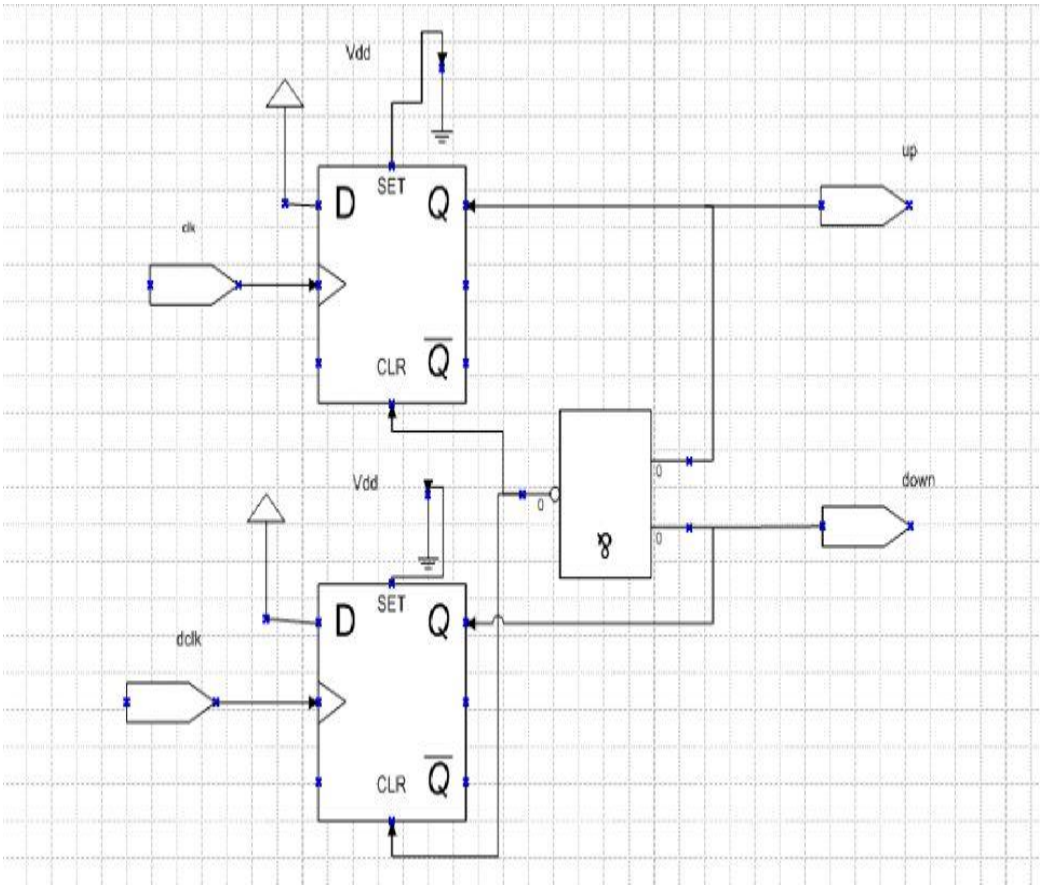
The two blocks have exactly the same design, except that the two input signals are switched in position. Each block consists of two cascaded stages with a pre charge PMOS in each stage. The Pre charge activity of the second stage is often controlled by the output of the first stage, as shown. The dynamic PD eliminates flip-flops and has advantages of simple structure and a fast transition time. However, dynamic PD needs to be carefully designed in order to minimize the dead zone.



3. Phase frequency detector

An improved PD is based on flip-flops. Since flip-flops offer edge detection, the duty cycle dependence problem with the XOR gate PD can be avoided. Figure 2-8 shows a commonly used flip-flop based linear PD, which consists of two flip-flops and one NAND gate called as Phase frequency detector.

The flip-flop based PD is capable of detecting both the phase and the frequency difference, which helps to increase the acquisition range and the locking speed. The disadvantage about the flip-flop based PD is that the reset time may limit the speed of the PD, which may further limit the operating frequency and the acquisition speed of the DLL.



These circuits have the advantage that whilst the phase difference is between ± 180° a voltage proportional to the phase difference is given. Beyond this the circuit limits at one of the extremes. In this way no AC component is produced when the loop is out of lock and the output from the phase detector can pass through the filter to bring the DLL into lock.

Moreover these Phase detectors have added advantage of detecting frequency difference. Thus, we chose this type of phase detector for our DLL Model as it works well with square wave pulses.

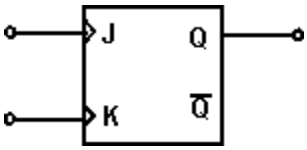
Types of Phase Frequency Detector

There are namely two types of **Phase Frequency Detectors.**

1. ***Dual D type phase comparator*** **-** This type of phase frequency detector uses two D type flip flops and an NAND gate, although there are a number of slightly different variants. This type of phase comparator is possibly the most widely used form of detector because of its performance and ease of design and use.

The circuit for the dual D-type comparator uses the two D-type flip flops with the reference and VCO signals being compared entering the clock inputs, one on each D-type. The NAND gate output is fed to the reset, R, inputs of both D-types. The inputs to the NAND gate are taken from the Q outputs and the output to the loop filter being taken from one of the Q outputs.

2. ***Edge triggered JK flip flop phase frequency detector -*** This form of phase comparator is used in some designs.



JK Flip Flop

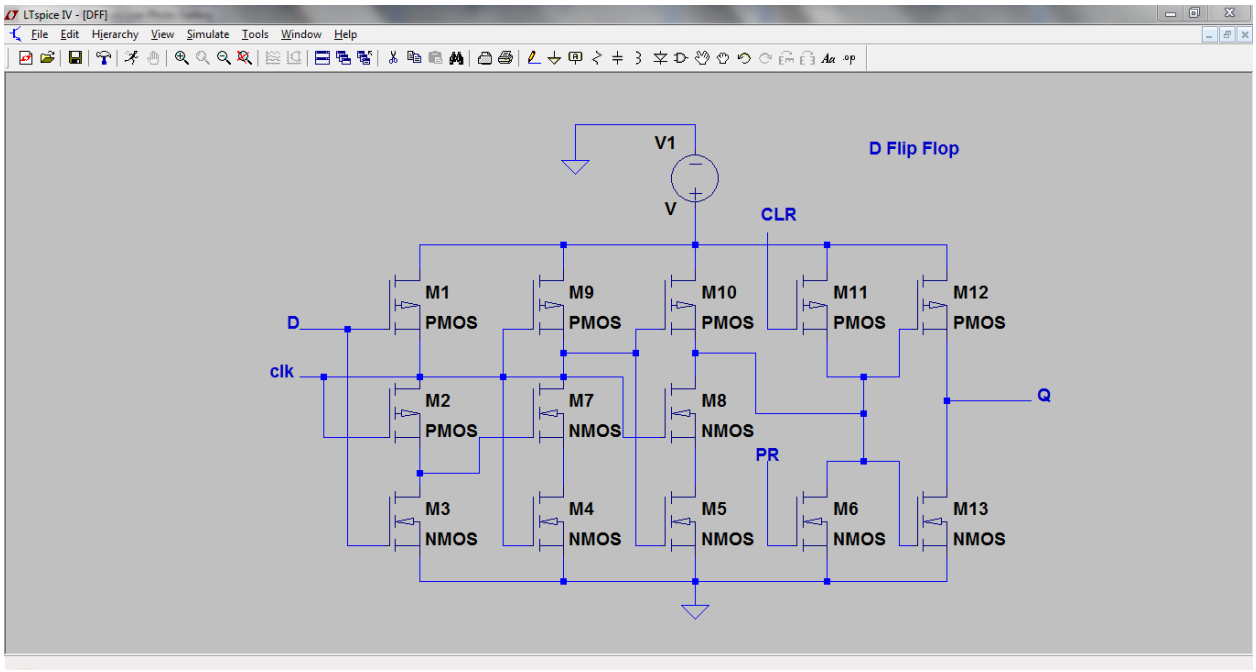
The idea behind the JK flip flop based comparator is that it is a sequentially based circuit and this can be used to provide two signals: one to charge, and one to discharge a capacitor.

Often when using this form of phase detector, an active charge pump is recommended.

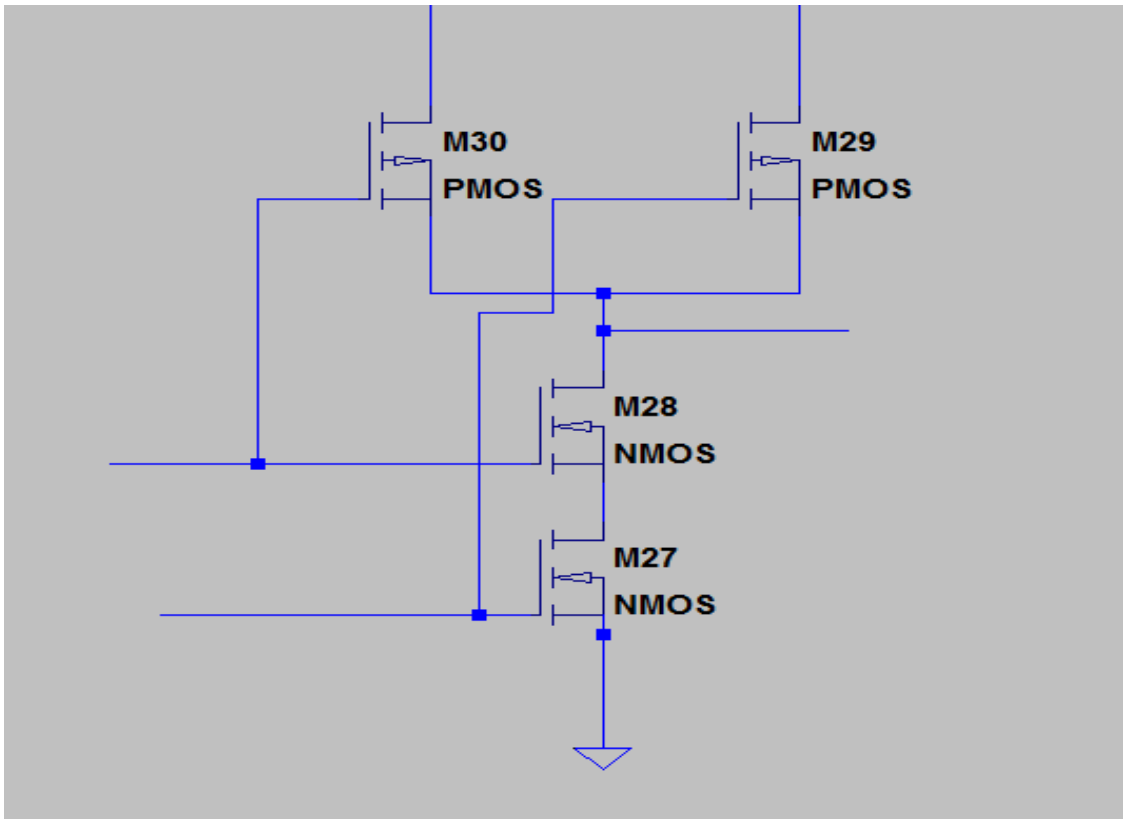
We have however implemented *Dual D type phase comparator*

Design Sequence of our Phase frequency Detector

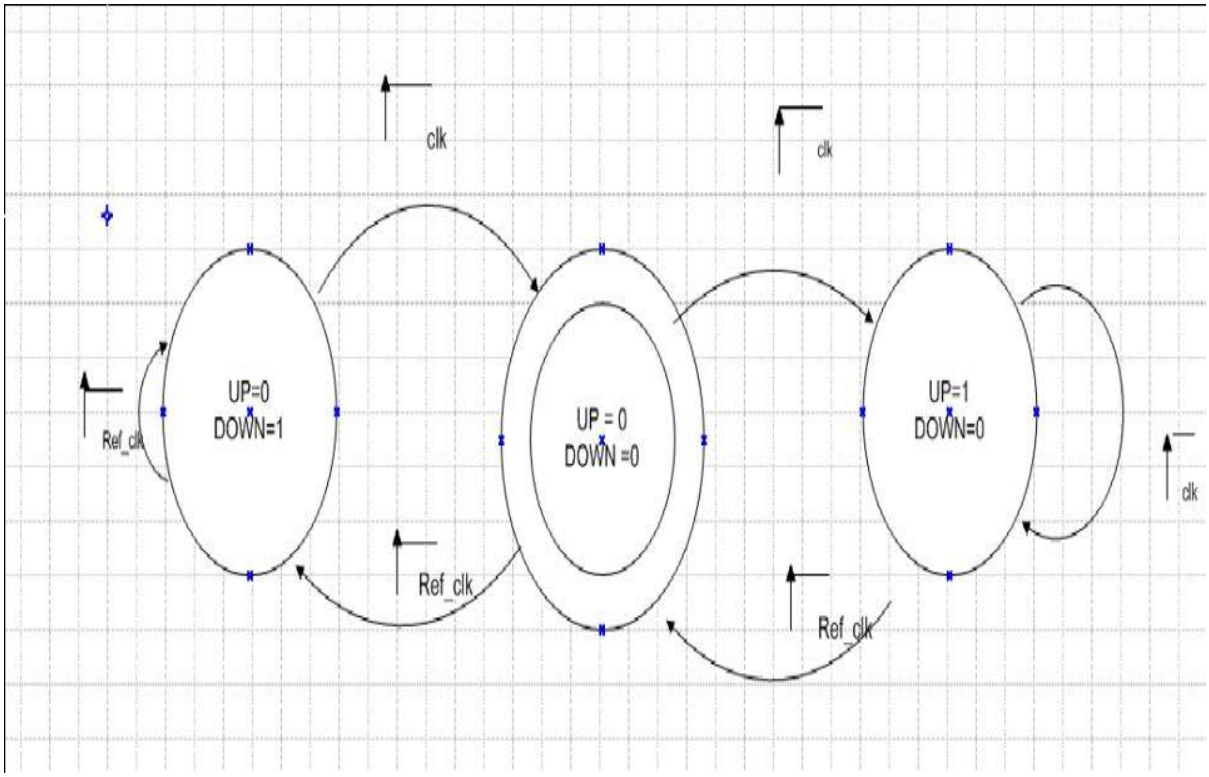
The basic building blocks of this type PFD are **D Flip flops** and **NAND Gate**.



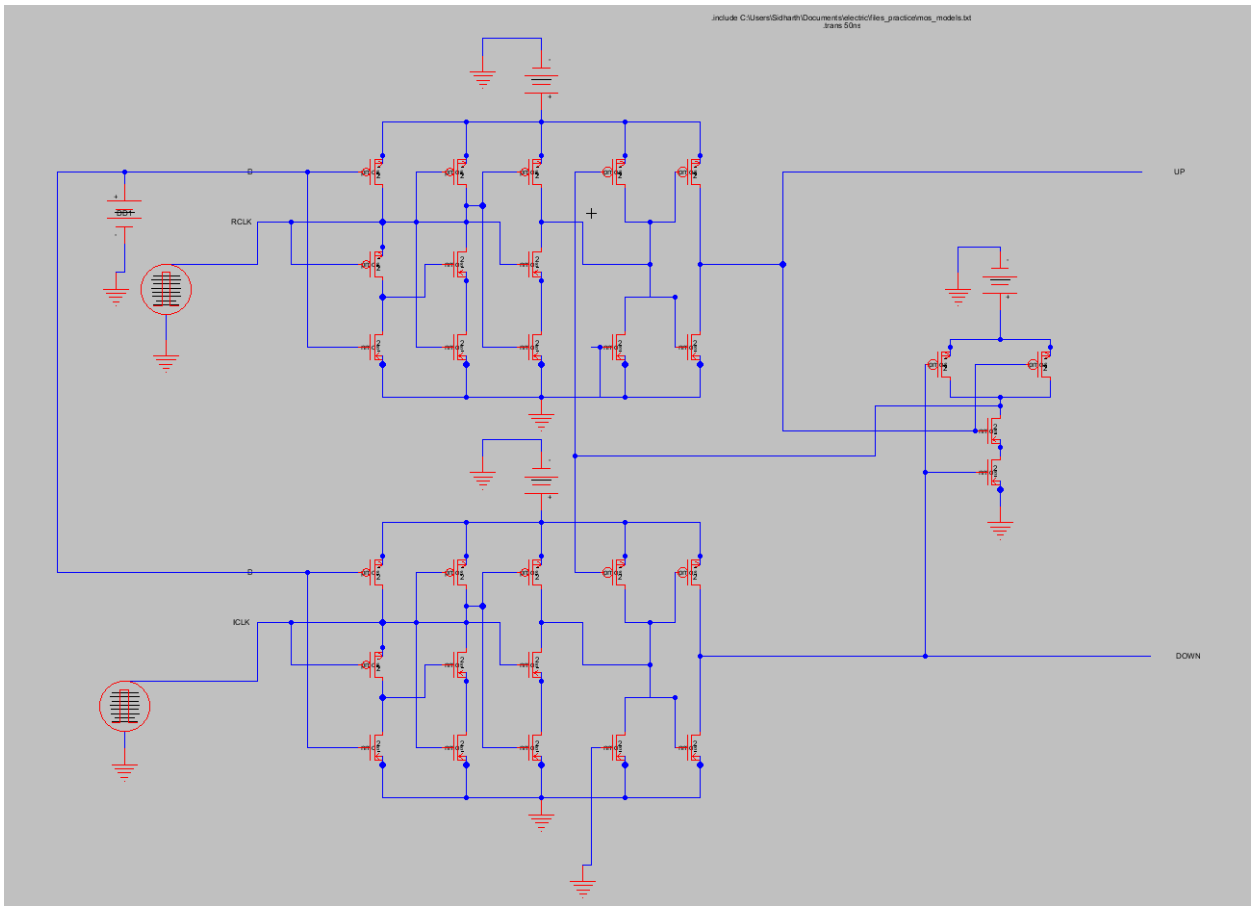
D Flip Flop



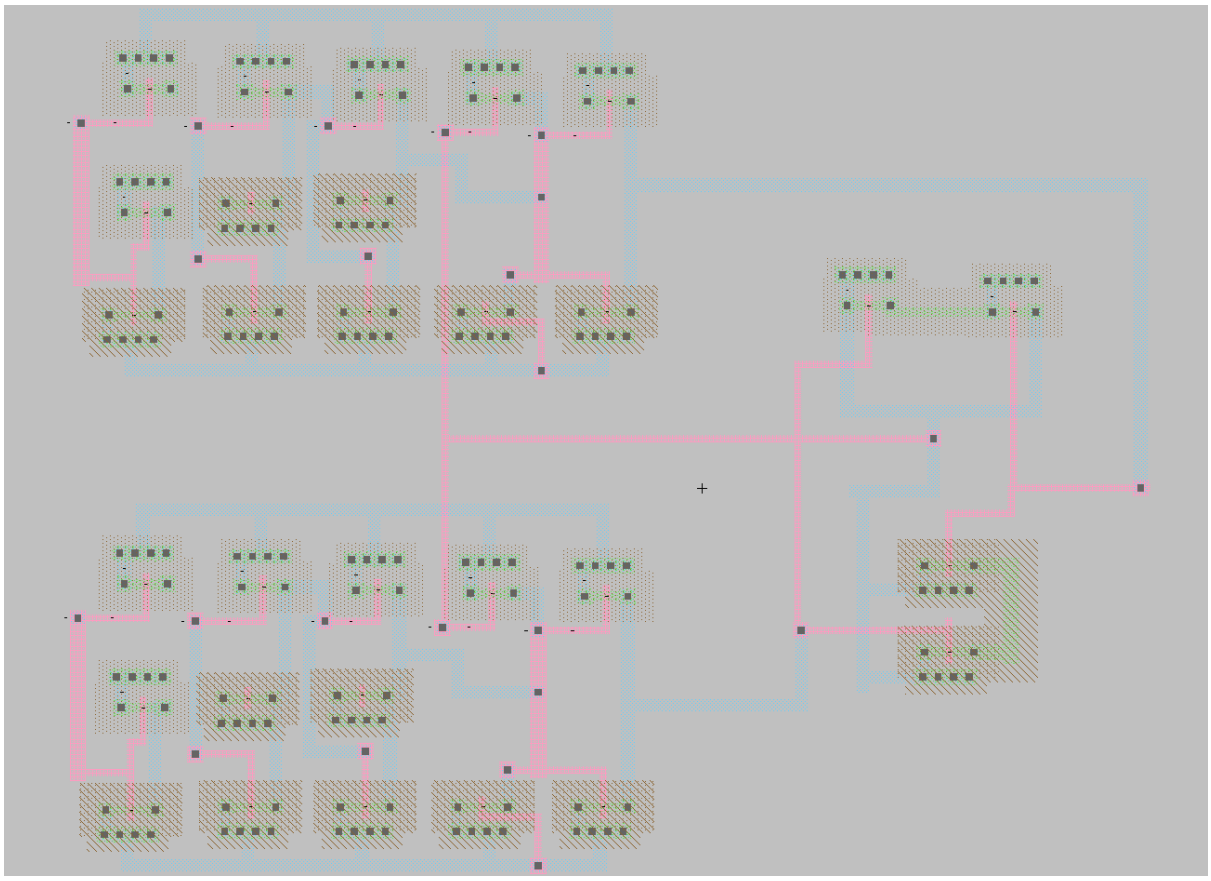
NAND Gate



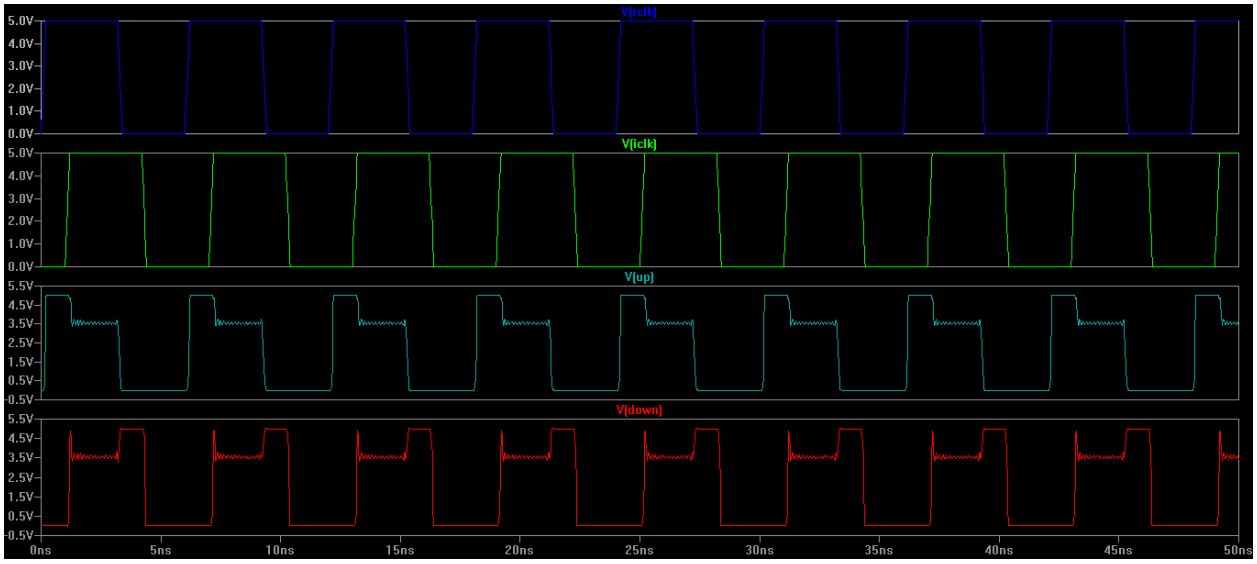
Schematic implementation of PFD on Electric



Layout design of PFD on electric -



Simulation Results



OBSERVATIONS

- Two clocks are given as input to the phase detector – rclk and iclk. The rclk is the original clock signal that is to be negatively delayed and the iclk is the output of the VCDL.
- We have assumed a phase difference between the two clocks in the above schematic for the time being.
- The output of the PFD are two pulse trains up and down. The Phase frequency detector detects the phase and frequency differences between the two input signals which can be seen in the plots V(up) and V(down). (output is proportional to the difference in phase or frequency).
- However, we are able to see some jitters at the outputs which are caused due to some internal design constraints. An error can also be spotted in each of the individual output pulses. This may be caused due to some internal offset voltage which is preventing the complete fall of the pulse.

CHARGE PUMP

Theory & concept

A charge pump is a kind of DC to DC converter that uses capacitors as energy storage elements to create either a higher or lower voltage power source. Charge pump circuits are capable of high efficiencies, sometimes as high as 90–95% while being electrically simple circuits. Charge is transferred from one capacitor to another under control of regulator and switching circuitry.

In a DLL the phase difference between the reference signal and the output signal is translated into two signals – UP and DOWN. The two signals control switches to steer current into or out of a capacitor, causing the voltage across the capacitor to increase or decrease. In each cycle, the time during which the switch is turned on is proportional to the phase difference, hence the charge delivered is dependent on the phase difference also. The voltage on the capacitor is used to tune a voltage-controlled delay line generating the desired output signal frequency.

A delay looked loop (DLL) having a charge pump gain independent of the operating frequency of the DLL. A charge pump is disclosed for providing a charge to a capacitive element on a voltage controlled delay line, wherein the charge is independent of a control voltage step cycle time of the DLL, the charge pump includes: a charge/dump signal generation stage that generates a charge signal and a dump signal during each period of a reference clock signal; a first switched capacitor stage that charges a first capacitor in response to the charge signal and dumps a positive charge from the first capacitor in response to the dump signal; a second switched capacitor stage that charges a second capacitor in response to the charge signal and dumps a negative charge from the second capacitor in response to the dump signal; and an output stage that selectively loads either the positive charge or the negative charge to the capacitive element on the voltage controlled delay line in response to an input signal from a phase detector.

In a DLL, the phase error between the input reference clock and the VCDL output clock is sensed by the PD and transferred to the CP in the form of voltage pulses or current pulses. The CP performs the function of adjusting the voltage of the loop filter and thereby altering the VCDL delay according to the phase error information from the PD. In principle, the CP simply consists of two controlled switches, one current source, and one current sink.

The two switches are controlled by the UP pulses and the DOWN pulses, respectively. Once the switch is closed, the current source or sink will start adding charge onto or removing charge from the loop filter (capacitor). This charging or discharging process will continue until lock is achieved. In the locked state, the voltage (charge) of the loop filter is kept constant. It is possible that equal charging and 22 discharging will still happen in the locked state. In fact, it is desired to have such activities to minimize jitter. However, the charging and discharging currents must be identical as well as very narrow so that the voltage of the loop filter will not be disturbed. In the actual implementation, the basic structure contains several non-ideal effects that may lead to time jitter. Firstly, the mismatch between the charging and discharging currents up I and down I contributes to time jitter. The mismatch is especially detrimental when both charging and discharging operations happen in the locked state. To suppress the effect of current mismatch, the turn-on time on Δt must be minimized. The second non-ideal effect of the CP is the timing mismatch caused by the PMOS and NMOS switches. Due to different characteristics of NMOS and PMOS switches, timing mismatch may occur

during charging and discharging operations. Similar to the current mismatch effect, the pulse width on Δt must be reduced to suppress timing mismatch. Both a single-ended topology and a differential topology CP exist in practical implementations. A single-ended topology has the advantages of smaller area and less power dissipation, but is more vulnerable to supply and substrate noise compared to a differential topology. Nevertheless, a single-ended topology is still popular in CP designs, because a single-ended CP does not need an additional loop filter and its power consumption is lower. There are three basic configurations for a single-ended CP: switching in the source, switching in the drain, and switching in the gate.

(a)Switching in drain (b) Switching in gate (c)Switching in source

Three single-ended CP configurations

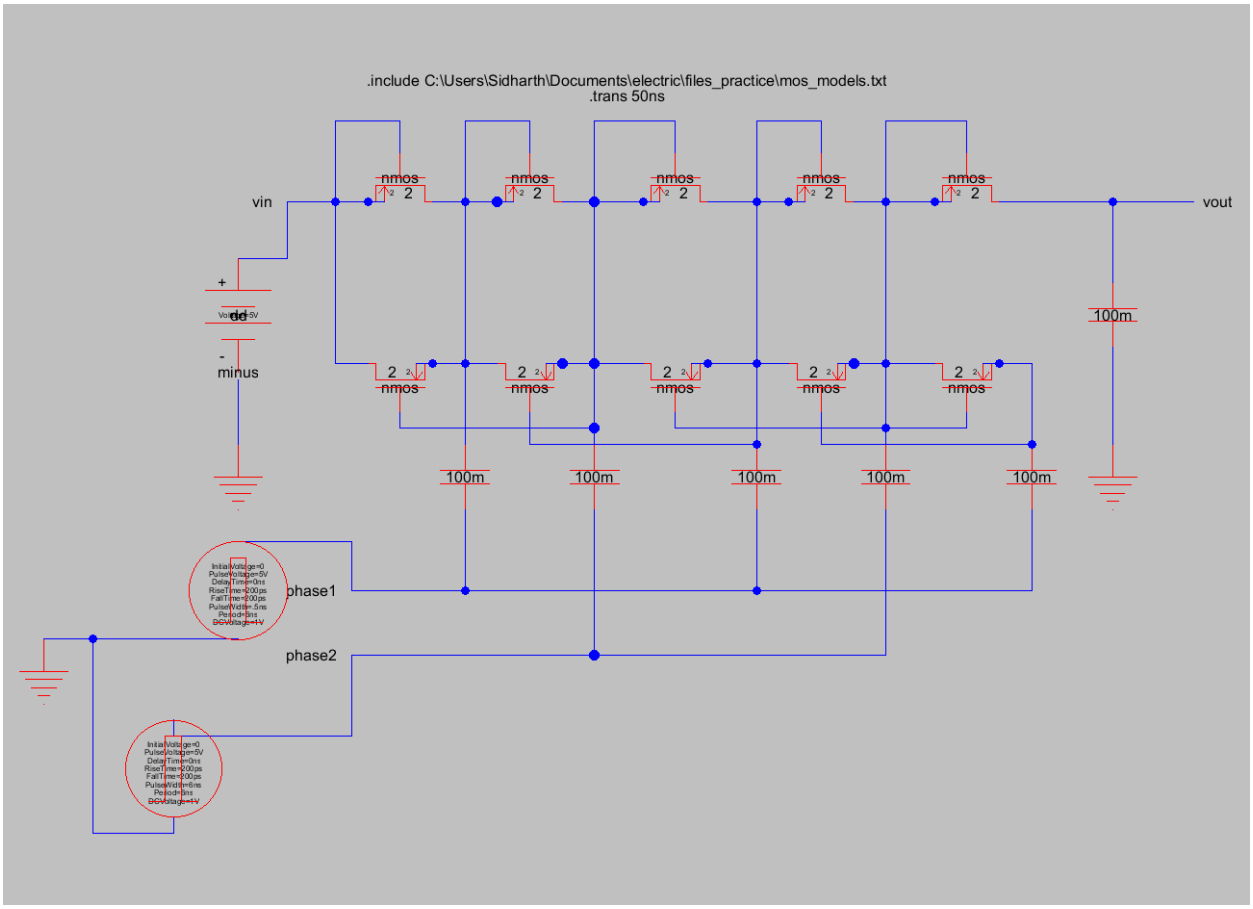
Among the three configurations, switching in source is preferred due to its simpler structure, lower power dissipation and faster switching time . Furthermore, studies show that in CMOS circuits, current switching provides a faster switching speed than voltage switching, if all the other conditions are the same . In addition, several new techniques are proposed to further improve the three basic CP structures. The configuration in the figure suffers from the charge sharing between the common drain of M1 and M2 and the loop filter when the switch is on (closed). A structure with an active unity-gain amplifier was proposed to solve this problem. Another enhancement to the basic CP is the adding of two additional current steering switches, which greatly improve the switching speed. The third proposed technique is to use only NMOS switches to avoid the mismatch between NMOS and PMOS.

A high-performance CP that combines several of the above techniques was proposed. In this structure, the mismatch problems are avoided by redesigning the NMOS and PMOS switches. Cascode current mirrors are used at the output to increase the output resistance so that the charging or discharging current are not be significantly disturbed.

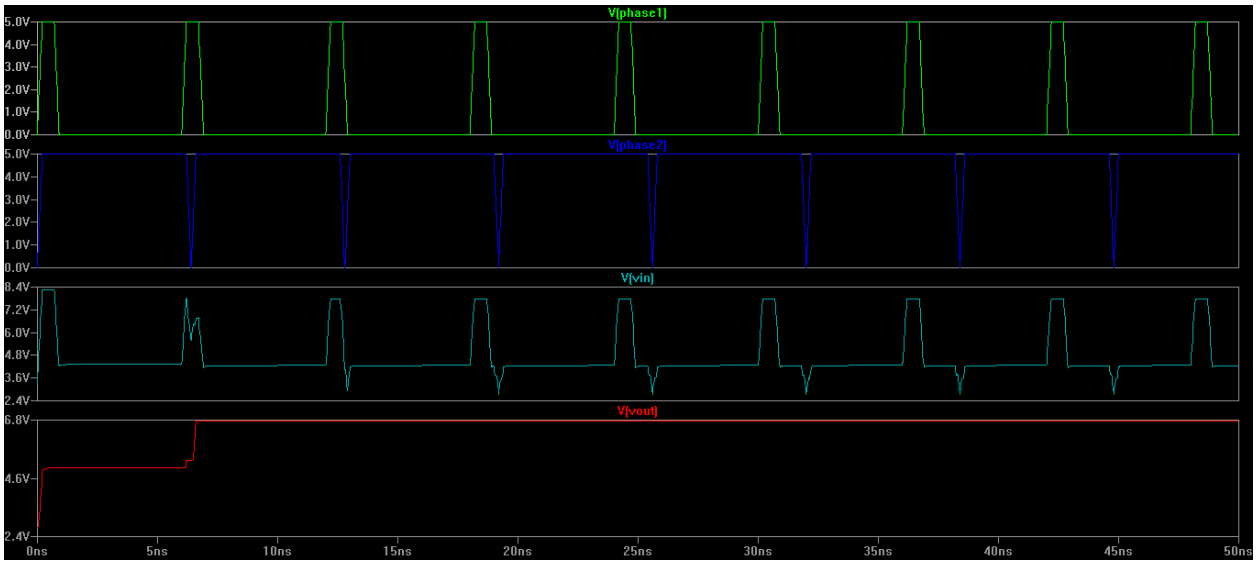
A high-performance single-ended CP

A fully differential CP consists of a set of NMOS switches, a set of PMOS switches, two loop filters, and some common-mode feedback circuitry. Although differential CPs are not as widely used as single-ended CPs, they do possess several unique advantages . Firstly, the fully differential structure offers better noise immunity to common-mode noise sources such as supply and substrate noise. Secondly, mismatch between the PMOS and NMOS switches in single-ended CPs does not exist in fully differential CPs. Lastly, the output voltage range can be doubled if the voltages of both loop filters are used. These advantages are achieved at the expense of double chip area and higher power dissipation. Generally, a single-ended CP can be converted to a fully differential CP by adding duplicate switches and duplicate loop filters.

Circuit diagram (next page)



Simulation Results



OBSERVATIONS

- The input to the charge pump are the two square wave pulse trains ‘ up and down’ that are nothing but the outputs of the PFD. (Characteristics of the output waveforms up and down have already been explained).
- In the above simulation, we have assumed two square wave pulses(similar to the outputs of the PFD) as input.
- The charge pump along with the loop filter sets a lock for the delay line (VCDL) to produce a negative delay effect.
- The output Vout settles at a constant value of 6.8V (approx).This constant voltage is the controlled input to the VCDL which locks it for a certain delay.

VOLTAGE CONTROL DELAY LINE

Theory & concept

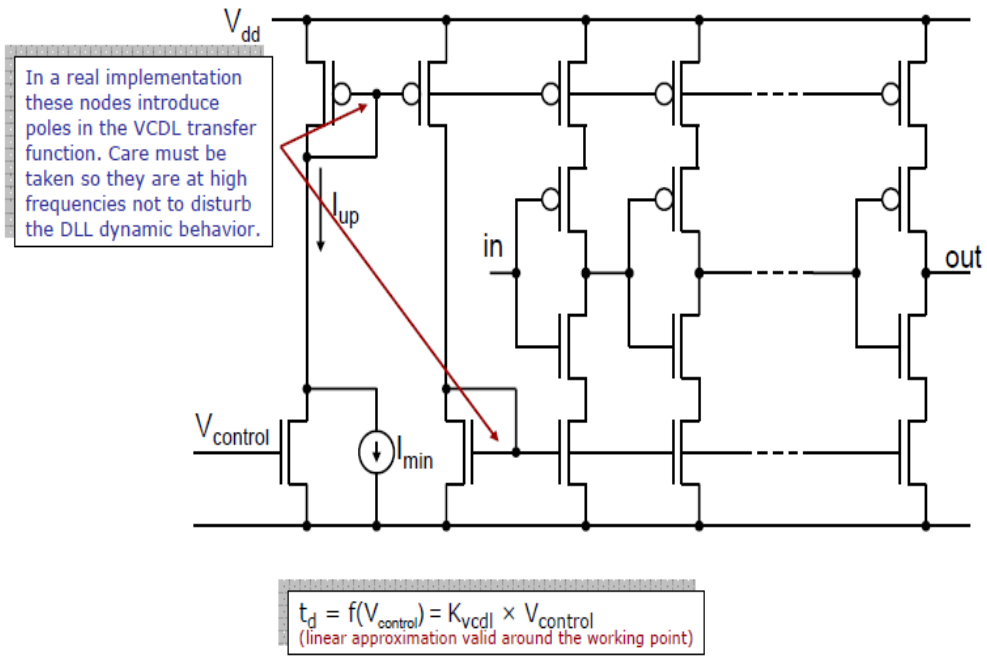
The main component of a DLL is a delay chain composed of many delay gates connected front-to-back. The input of the chain (and thus of the DLL) is connected to the clock that is to be negatively delayed.

A multiplexer is connected to each stage of the delay chain; the selector of this multiplexer is automatically updated by a control circuit to produce the negative delay effect.

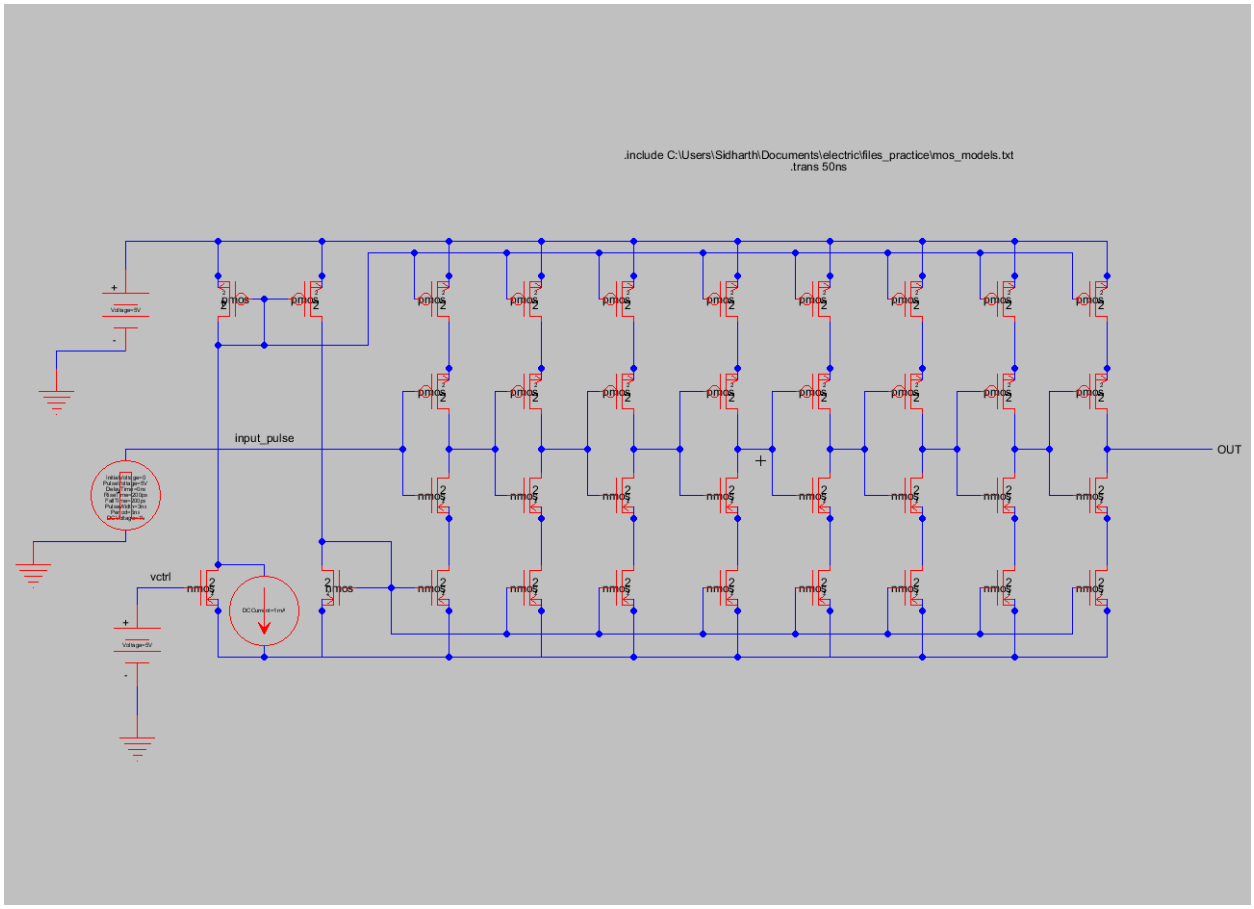
The voltage from the charge pump determines the power supply voltage across the gates in the voltage controlled delay line, so when it varies, the gates become incrementally slower or faster in response. The input to the VCDL is the DLL clock signal, whose delay we are trying to control.

The incoming clock signal is delayed such that when further processed by some circuit in the feedback loop, **the output is delayed by exactly one or more clock cycles**. Thus, the DLL employs an adjustable delay line element known as Voltage controlled delay line.

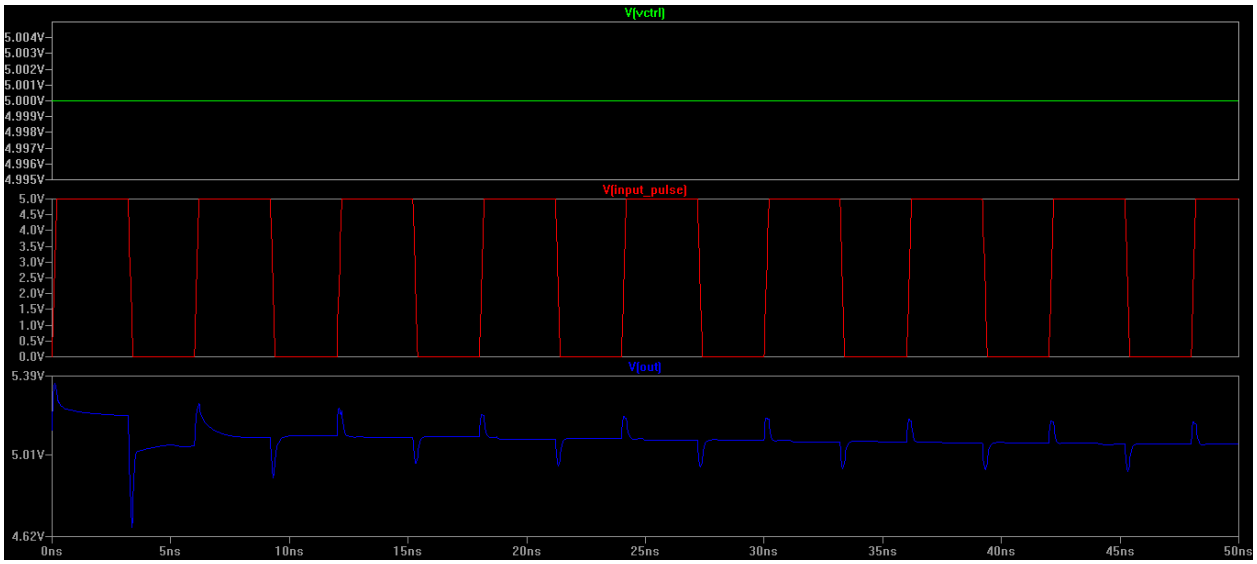
Circuit diagram



Schematic implementation on Electric



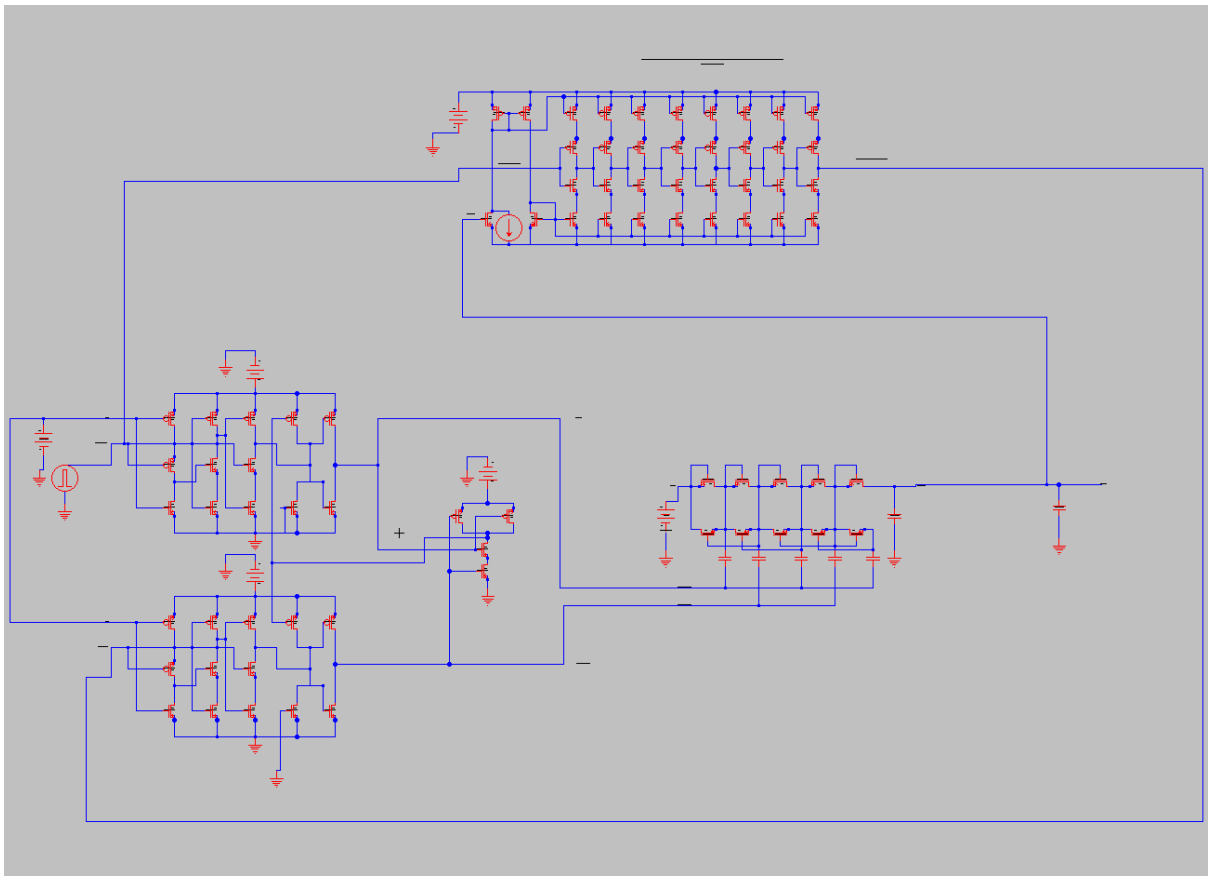
Simulation Results



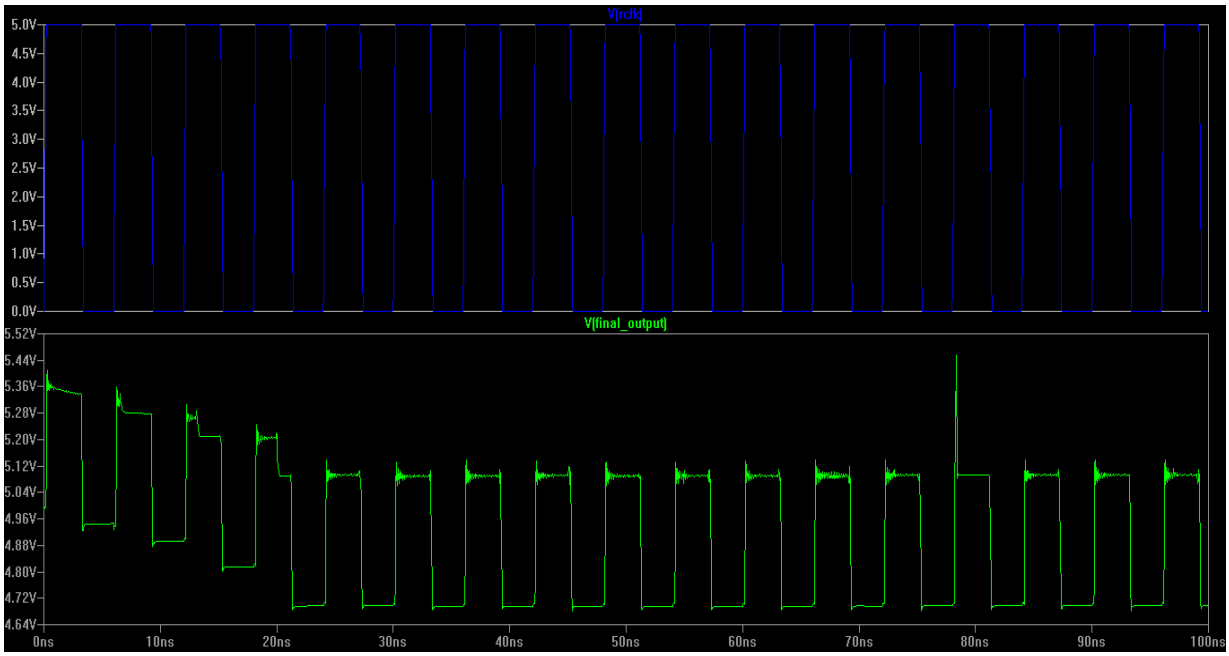
OBSERVATION

- The input to the VCDL is the voltage control (Vctrl) which is the output of the charge pump (has already been described). For testing of the VCDL, we have assumed a constant voltage of 5V as the Vctrl.
- The Vctrl sets the VCDL for a certain delay. The dealay depends on the number of delay elements as well as the voltage control.
- The output is as seen in the plot. This output is fed back as input pulse (iclk)to the PFD.

DESIGN AND IMPLEMENTATION OF COMBINED DLL CIRCUIT



Simulation Results



OBSERVATION

- We have drawn and tested/simulated all the individual components of the DLL.
- After testing, we have integrated all the components to make the complete DLL circuit.
- From the schematic , we observe that there is one input clock or reference clock and one output clock as seen in the above plot.
- It can be observed that both the pulses **are matched to a satisfactory level in terms of frequency and phase.**
- It can also be seen that the output takes certain time to match the input reference. This time interval is subject to design considerations.
- Also, one can notice some jitters which are inevitable in such large circuits.However, they can be rectified after proper testing if each individual (small or big) components.

- **Delay Compensation**

A DLL with a replica buffer chain in the feedback path can be used to mask the delay of a clock buffer tree

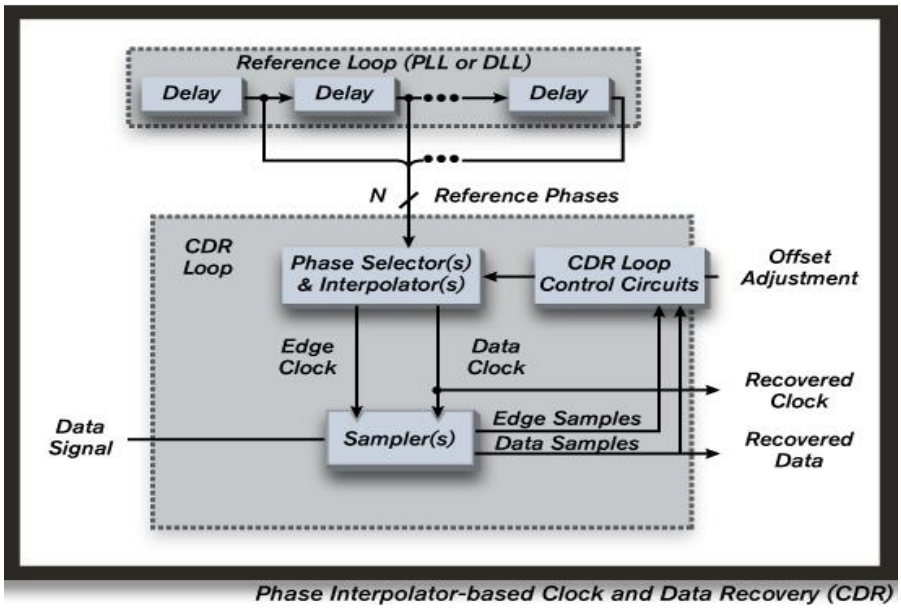
- **Multiphase Clock Generation**

A DLL can be used to generate multiple clock phases with precise phase spacing which is useful in CDRs and RF modulation and up/down-conversion

- **Clock and data recovery**

In many systems, data is transmitted or retrieved without any additional timing reference. For example, in optical communications, a stream of data flows over a single fiber with no accompanying clock, but the receiver is required to synchronously. The receiver generates a clock from an approximate frequency reference, and then phase-aligns to the transitions in the data stream with a DLL. This process is commonly known as **clock and data recovery** (CDR).

CDR: This type of CDR uses a PLL or DLL to implement a reference loop which accepts an input reference clock signal and produces a set of high speed clock signals, used as reference phases, spaced evenly across 360 degrees. These reference phases are then fed to a CDR loop which includes circuitry for selecting pairs of reference phases and interpolating between them to provide clocks for recovering the data from the data signal



Phase Interpolator-based Clock and Data Recovery (CDR)

INFERENCE/CONCLUSION

DLLs have been commonly used for generating precise phase delays of a signal and have been increasingly popular in clock generation and data recovery applications.Most importantly, because of the first loop characteristics that controls the phase directly, DLLs can be designed with high tracking bandwidths and do not exhibit the phase accumulation of an oscillator of an oscillator-based PLL.

The more simple loop characteristics belie many subtleties in DLL design.The delay-line input clock must have low-jitter and good duty-cycle. Furthermore, it must be carefully received and coupled to the input of the delay line to maintain good jitter performance.This source of jitter counter-balances the jitter performance.This source of jitter counterbalances the jitter accumulation of PLLs and the results in less jitter improvement. Additional circuitry is often needed to prevent false-locking. Since a delay line does not restore a clock's duty cycle, the output clock requires correction circuitry. To use DLLs in plesicronous systems, the delay line must have even more circuitry to achieve an unlimited delay range. In clock multiplication application, very careful matching in the DLL components is critical to eliminate reference tones. In many designs tht have addressed these subtleties, DLLs have demonstrated low-jitter clock outputs for a variety of clock generation and data recovery application.

Books and publications

- Digital Integrated Circuits by Rabaey Chandrakasan NikolicWeb
- Design of low power phase frequency detector for Delay Locked Loop(By Rajnandini,Himadari Singh, B.P. Singh(IJESE))
- Design of low glitch dynamic phase detector for Delay Locked Loop(Shirish Tripathi)
- Delay locked loop integrated circuit(By Robert W.Brocato)
- Delay Locked Loop (By Meera Puniya & Johnathan Chai)
- Delay locked loop –an Overview(By Chih-Kong Ken Yang) IEEE press 2005
- http://www.radio-electronics.com/info/rf-technology-design/
- https://smartech.gatech.edu/bitstream/handle/
- http://www.edaboard.com/thread108700.html