Delayed Locked Loops
For High Speed Serial Interface

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Ramani Ashish V.
(14MECV25)

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2016

Delayed Locked Loops
For High Speed Serial Interface

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Ramani Ashish V.
(14MECV25)

Under the guidance of

External Project Guide:

Mr. Venkatesh Rao
Circuit Design Manager,
Intel India Technology Pvt. Ltd.,
Bangalore.

Internal Project Guide:

Dr. Amisha Naik
VLSI Design,
Institute of Technology,
Nirma University, Ahmedabad.

तमसो मा ज्योतिर्गमय
NIRMA
UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
Dec 2015

NIRMA
UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

Certificate

This is to certify that the Major Project entitled **"Delay locked loop for high speed serial interface"** submitted by **Ramani Ashish V. (14MECV25)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Internal Guide

Dr. Amisha Naik
(VLSI Design)

Program Co-ordinator

Dr. N. M. Devashrayee
(Professor,EC)

Director

Prof. P. N. Tekwani
(Head of EE Dept.)
(Director, IT-NU)

Intel Technology India Pvt. Ltd.

Certificate

This is to certify that the Major Project entitled **"Delay locked loop for highspeed serial interface"** submitted by **Ramani Ashish V. (14MECV25)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Bangalore

Mr. Giridhar Bajpe
IPG-MIG India
Intel Technology India Pvt. Ltd.,
Bangalore.

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.

2. Due acknowledgment has been made in the text to all other material used.

- Ramani Ashish V.
14MECV25

Acknowledgements

First and foremost, sincere gratitude to my managers Mr. Venkatesh Rao and Mr. Shivraj Thakare. Also I want to thank Intel Technology India Private Limited, Bangalore for assigning me such project and guide me through.

I would like to express my gratitude and sincere thanks to my mentor Mrs. Bheemisetti Chakravarti at Intel Technology India Private Limited, Bangalore for his valuable guidance throughout this period. he has given me valuable advices and support for my project work which I am very lucky to benefit from.

I would like to express my gratitude and sincere thanks to our Director Dr. P. N. Tekwani, Head of Electrical Engineering Department for allowing me to undertake this thesis work and for his guidelines during the review process.

I would like to thank my Program Coordinator, Dr. N. M. Devashrayee, Professor, EC (VLSI Design), Institute of Technology, Nirma University, Ahmedabad for giving valuable support and motivation throughout the academic period.

I would also thank to my Project Guide, Dr. Amisha Naik, Professor, VLSI Design, Institute of Technology, Nirma University, Ahmedabad for being a source of inspiration, giving valuable support and timely guidance for the project work.

I take immense pleasure to thank my team member Mr. Mallikarjuna Rao, Bheemisetti Chakravarti and other colleagues in the Intel, special thanks for helping me on this path and for making project at Intel more enjoyable.

I wish to thank my classmates for their delightful company which kept me in good humor throughout the journey.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete the degree program successfully.

- Ramani Ashish V.
14MECV25

Abstract

The performance of high-speed wire-line data links depend crucially on the quality and precision of their clocking infrastructure. with aggressive scaling of device susceptibility of signal to process, voltage, and temperature variation increase tremendously. because of high speed and low power design timing and voltage margin reduce respectively. For future applications, such as microprocessor systems that require terabytes/s of aggregate bandwidth, signaling system designers will have to become even more aware of detailed clock design trade offs in order to jointly optimize I/O power, bandwidth, reliability, silicon area and testability. Digital communications technology as well as integrated circuit scaling trends has enabled the industry to dramatically scale the bandwidth of high-loss networks such as DSL and Ethernet. Many of these networks are channel bandwidth limited and have had to leverage sophisticated equalization techniques to push well beyond the uncompensated channel bandwidth. Proper clocking architecture dramatically reduce burden on sophisticated equalization technique for generating a high bandwidth on and off chip link .so trade off between power, performance and area (complexity) can solve by proper handling of clock through distribution. Delay locked looped provide easy means of phase synchronization of sampling clock and data. Due to optimum sampling point in time-domain decision threshold get extra tolerance against noise, inter symbol interferences, jitter, and skew. All digital delay locked is presented here with wide frequency range, small locking time, less jitter and stable phase of delayed clock. two types of delay line coarse delay unit and fine delay unit together provide a wide delay regulation range and finer delay resolution. bang-bang phase detector with delay less than clock period enable us to update delay value at each clock cycle hence time to lock improved.

Contents

List of Figures

Abbreviations

DLL	Delayed locked loop
ADDLL	All digital delayed locked loop
CDU	Coarse delay unit
FDU	Fine delay unit
PD	phase detector
PVT	Process voltage temperature
DC	Duty cycle
DRAM	Dynamic random access memory
DDR-DRAM	Double data rate dynamic random access memory
UI	Unit interval
CP	charge pump
CSTRV	Current starved
CDN	Clock
DQS	Data strobe
FSM	Finite state machine
VCDL	Voltage controlled delay line
DCC	Duty cycle correction circuit
LS	Level shifter
MVD	Multi voltage domain
CTLE	Continuous time linear equalization
CML	Current mode logic
MGT	Multi-Gigabit Transceiver
CRC	Clock recovery circuit
CTS	Clock Tree Synthesis

Chapter 1

Introduction

Demand for multi Gb/s serial I/O has placed a great challenge to clock and data recovery architecture and circuit design. Due to great improvement in semiconductor technology along with device scaling has created high speed interface capability in microprocessor and memory. All high speed and performance benefits are enjoyable in high bandwidth communication link but scenario in low bandwidth communication is slight different. Various additional factor came into picture when we switch from on chip to off chip communication. Some of them are link bandwidth, jitter, noise and skew. This factor are limiting factor for off chip serial link's power, performance and complexity. Its demand extra functionality and novel design for clock and data recovery circuit to mitigate an additional challenges specifically imposed by off chip communication application. Various additional effort made in section of channel equalization, channel coding, proper link termination has improved off chip communication capacity very far from link bandwidth. But by doing so the tradeoffs triangle has not optimize. Calculated approach toward high speed I/O should include each and every possible optimization to improve a tradeoff triangle which is power, performance (speed) and area (complexity). So novel clock distribution is critical for further capacity improvements. Improved clocking methods have played a central role in the widespread adoption of multi-gigabit data links.

1.1 Challenges for high speed clock and data recovery circuit

In high speed transceiver system clock and data recovery (CDR) is a critical function. Optical communication, backplane routing, and chip-to-chip interconnects are the main application area of CDR. Due to high speed operation allowed by asynchronous communication, clock must be recovered from noisy data to allow a synchronous operation to reduce a bit error rate in communication. Data retiming is necessary to remove jitter which is accumulated during transmission to allow a high timing margin. Clock get extracted from noisy data by clock recovery circuit and data retiming operation is performed by data recovery circuit.

Communication standard impose a stringent specification on CDR which it must satisfy. To design a CDR within communication standard boundary create a difficult challenges to circuit designer and system designer. Figure show clock recovery circuit, clock is require to perform a synchronous operation such as data retiming and de-multiplexing of random data, receiver must generate a clock. Clock recovery circuit generate a periodic clock from noisy data. This recovered periodic clock then drive a D-flip-flops to retime a data. D-flip-flop here act as a sampler so we can called it a decision circuit.

Clock generated by this circuit must satisfy below three condition for proper operation of CDR.

1. Frequency of extracted clock must be equal to the incoming data rate. For example, clock frequency of 10 GHz (with a period of 100 ps) must be extracted from data rate of 10 GB/s (each bit 100 ps wide).

2. Received data possess a significant amount of noise because of transmission loss and noise coupling. So clock recovered from this noisy data should replicate all noise component of data onto itself. If it happens then clock always bear phase relationship with data, and it can achieve an optimum sampling point in decision circuit. If sampling occur at midpoint of data valid window then we can achieve a maximum timing and voltage merging for jitter and other uncertainty in time domain as well as in voltage domain.

3. It can add a jitter to extracted clock, this jitter directly reflect on re-timed data jitter because very high speed operation small amount of jitter can introduce a high degradation in timing and voltage margin. So we can define it as a primary jitter source within a receiver circuit. Jitter can introduced because of power supply variation or modulation occurred because of inter-symbol interference (low channel bandwidth. jitter may be random or deterministic. To counter jitter we should design an optimum channel coding technique, equalization and very precise supply references. This compensation technique limits random jitter within acceptable range.clock is extracted from data itself so all high frequency jitter get tracked by clock it self.so data recovery circuit bandwidth to track a high frequency jitter is actually no more required. so it can operate at much less bandwidth. so it will reduce jitter which is introduced by clock recovery circuit.

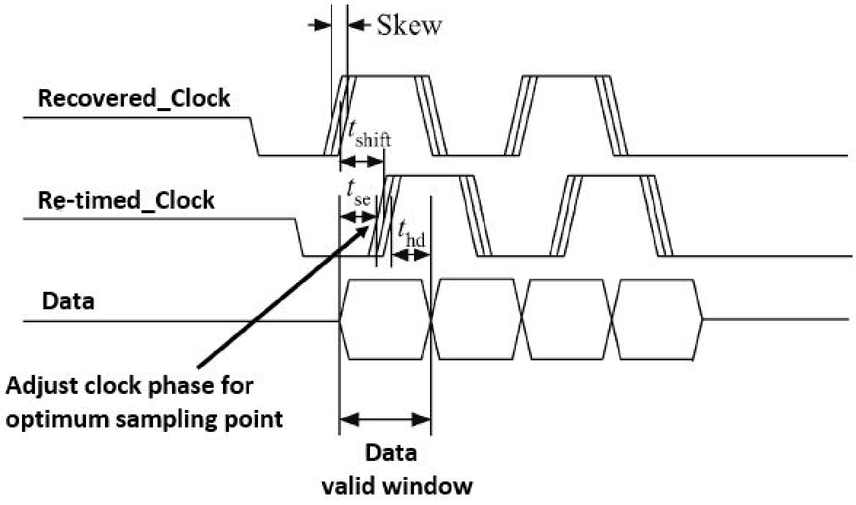


Figure 1.1: Timing budget of CDR

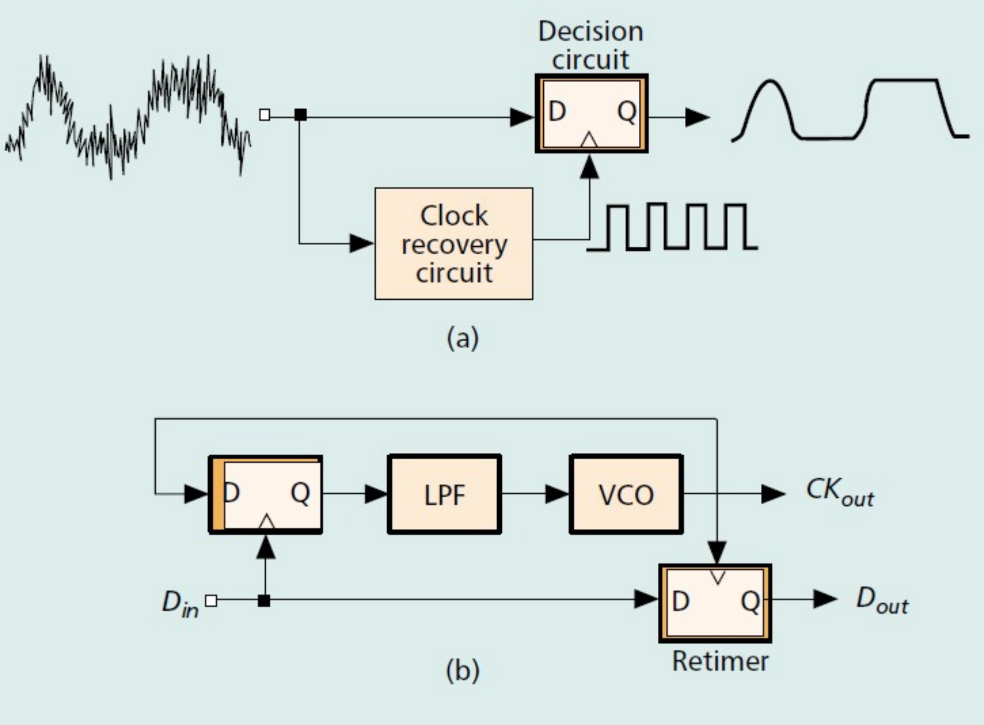


Figure 1.2: a) The role of a CDR circuit in re-timing data; b) an example of CDR implementation

1.2 Jitter

Jitter is a major disturbance parameter in any clock path circuit including DLL and PLL.Jitter is random in nature. Maximum value of jitter determines a final timing and voltage budget margin and eventually define the speed and performance of circuit. Jitter generated from different types of sources land themselves as different type jitter as random jitter, deterministic jitter, data dependent jitter etc. so nature of jitter completely depend on nature of its source. We will analyses jitter in analog delay locked loop and digital delay locked loop separately.

Advanced equalization and precise clocking are integral part of well-designed multi GB/s data link architecture. Proper combination of above two factor can optimize the design triangle performance, power efficiency and cost. Data link design which are over emphasis on advanced equalization may meet the performance goal but with increased complexity and resources power dissipation and cost become worst. Which is not acceptable. So low jitter clock path design should be major design goal while design multi GB/s serial I/O to optimize the design triangle.

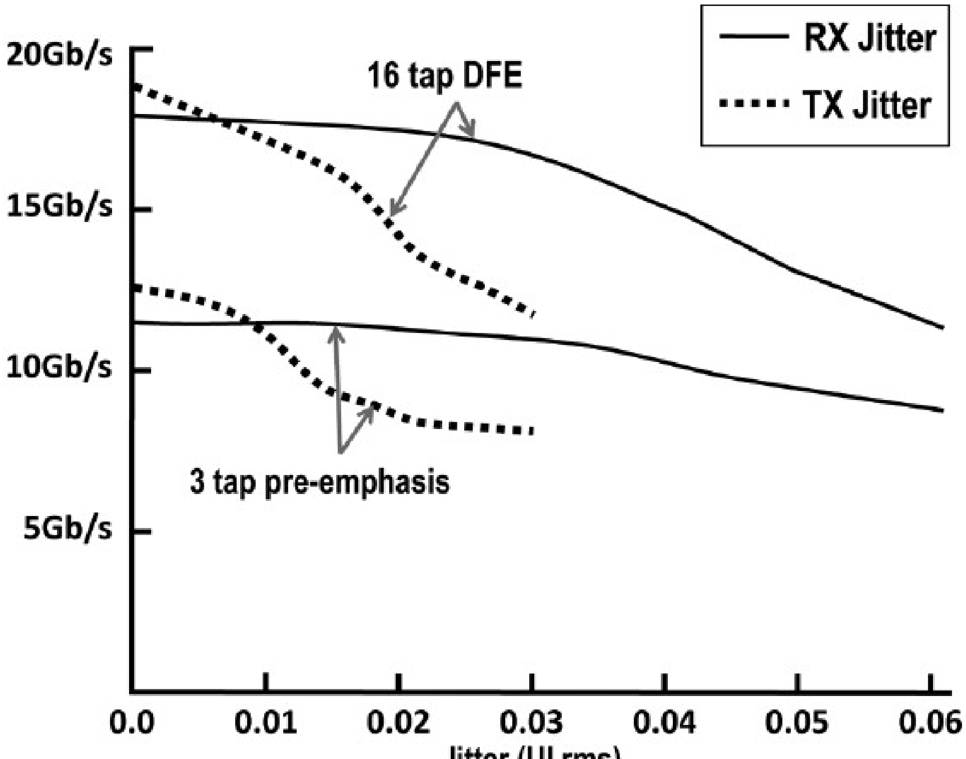


Figure 1.3: Maximum data rate as a function of TX jitter magnitude for a data link with varying amounts of equalization complexity. Baseline parameters for link simulation include: Channel loss 15 dB@5 GHz, BER= 1e-12. 16 tap DFE link includes 3 tap TX pre-emphasis. Rx input-referred noise 1 mV rms. TX jitter 1/2 ps rms, RX jitter 1ps rms unless otherwise specified. Jitter is normally distributed

In data communication link we broadly categorized jitter in two category based on their source as channel induced jitter and circuit induced jitter. Inter-symbol interference (ISI) and co-channel interferences are two major souses for channel induced jitter. Inter-symbol interference (ISI) comes from electromagnetic coupling within channel (low pass nature) and co-channel interference (CCI) comes from an electromagnetic coupling with another channel. Channel induced jitter is deterministic clock and data recovery for high speed interface is affected by it. Due to deterministic it can be model effectively as far as channel characteristics are known. Various equalization technique can be adopted to mitigate channel induced jitter. Circuit induced jitter are much diverse and modeling of it is very complex as compared to channel induced jitter. Various sources for circuit induced jitter are thermal and flicker noise which can deviate a normal circuit operation and generate voltage induced timing noise. Power supply noise and substrate noise are another sources of jitter which can inject a timing noise and create a delay variation in clock circuit. Circuit design defects such as layout mismatch, locked loop update glitch, coupling effect also inject a timing noise in clock path.
As shown in figure with rise in data rate to achieve a low jitter in communication link with equalization required a large resources compare to slow communication link. As we switch from 10GHz to 15 GHz for same level of jitter in both speed link the equalization resources approximately rising 5 times in high speed link. So we can compensate TX and RX jitter by designing a low jitter clock path to maintain a timing budget. Which eventually lower load on equalization and reduce equalization complexity and resources optimizing design triangle.

1.2.1 Jitter peaking by channel

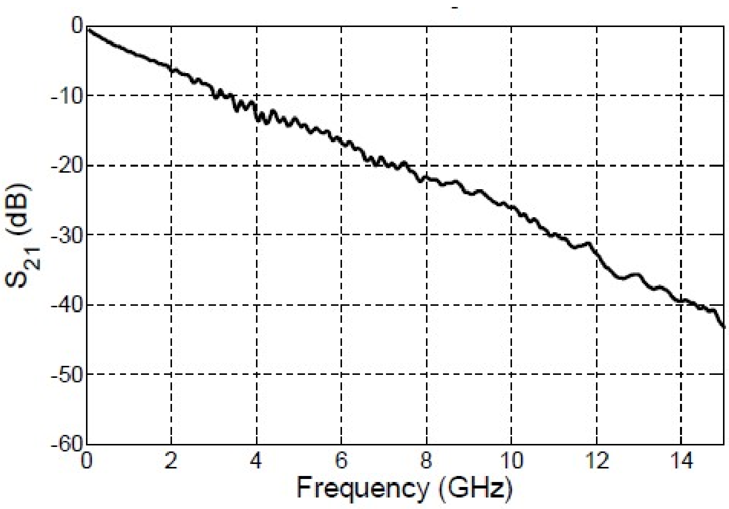


Figure 1.4: Channel Response[3]

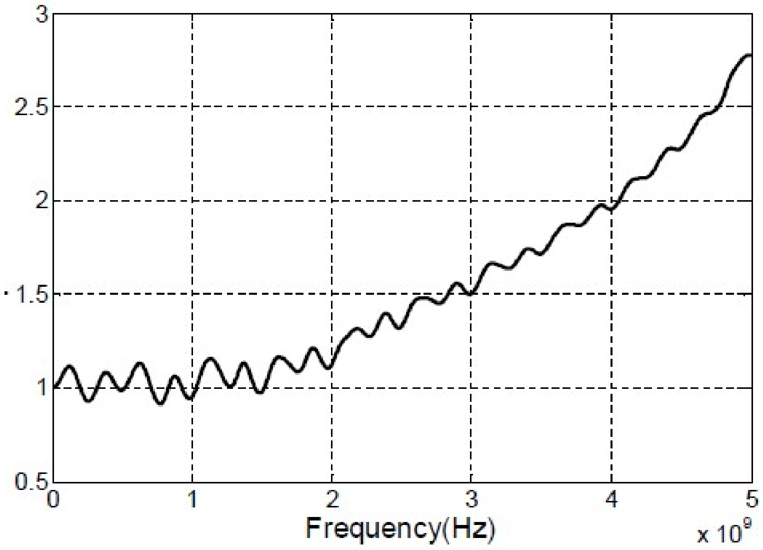
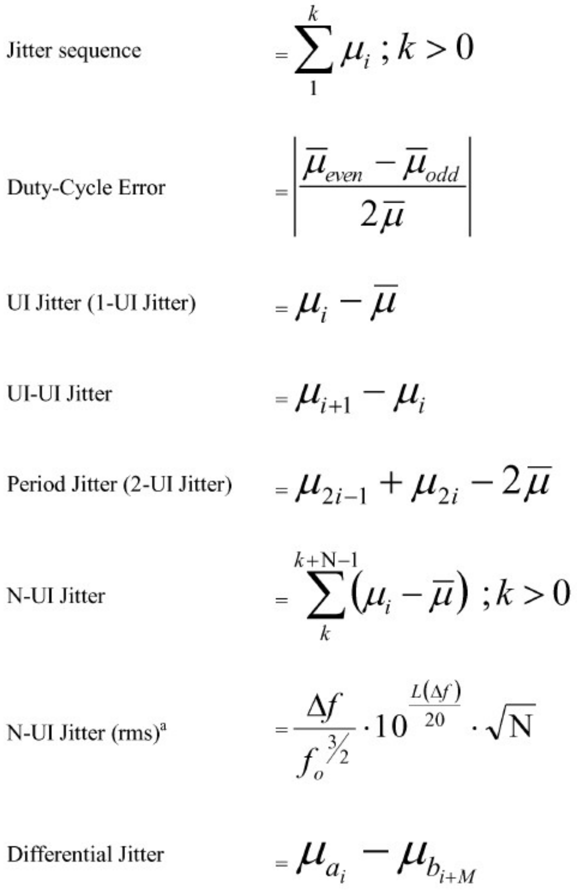


Figure 1.5: Jitter Transfer/Amplification[3]

Low-pass frequency response (buffer, distribution interconnect) is similar to a high-pass jitter filter.
Finite bandwidth of DLL can amplify clock jitter.



Jitter sequence	$= \sum_{1}^{k} \mu_i \; ; k > 0$
Duty-Cycle Error	$= \left\| \dfrac{\overline{\mu}_{even} - \overline{\mu}_{odd}}{2\overline{\mu}} \right\|$
UI Jitter (1-UI Jitter)	$= \mu_i - \overline{\mu}$
UI-UI Jitter	$= \mu_{i+1} - \mu_i$
Period Jitter (2-UI Jitter)	$= \mu_{2i-1} + \mu_{2i} - 2\overline{\mu}$
N-UI Jitter	$= \sum_{k}^{k+\mathrm{N}-1} \left(\mu_i - \overline{\mu}\right) \; ; k > 0$
N-UI Jitter (rms)[a]	$= \dfrac{\Delta f}{f_o^{3/2}} \cdot 10^{\frac{L(\Delta f)}{20}} \cdot \sqrt{\mathrm{N}}$
Differential Jitter	$= \mu_{a_i} - \mu_{b_{i+M}}$

Figure 1.6: Jitter representation

1.2.2 Jitter analysis in analog delay locked loop

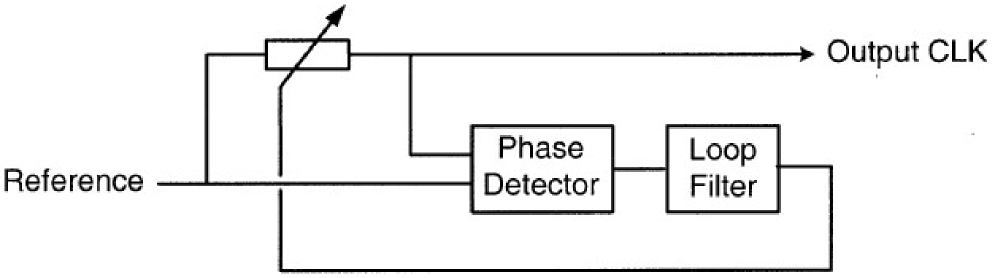


Figure 1.7: simplified type-1 analog DLL

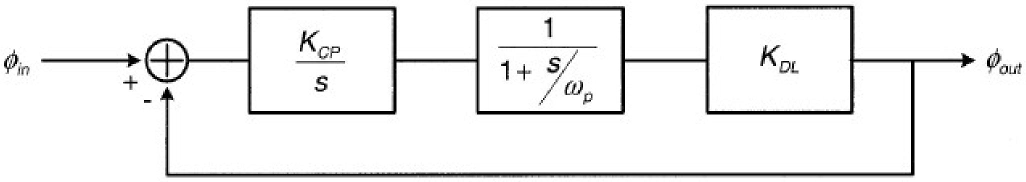


Figure 1.8: S-domain model for Analog DLL

- Kcp–gain of charge pump and loop filter.
- Wp-loop filter pole.
- Kdl–delay line gain.

Generally loop filter is integrator but to account second order effects we set pole at Wp.

$$Kcp = Icp/2Pi * Wp$$

where Icp is charge pump current.

$$\frac{\phi_{\text{out}}}{\phi_{\text{in}}} = \frac{1}{\frac{s^2}{\omega_p K_{\text{DL}} K_{\text{CP}}} + \frac{s}{K_{\text{DL}} K_{\text{CP}}} + 1}.$$

Figure 1.9: jitter transfer function for Analog DLL

If we assume a loop filter as an ideal integrator then Wp=0 and jitter transfer function left with single pole at Kdl*Kcp. There would not be jitter peaking. Jitter transfer function is second order system so jitter peaking is possible. With making it over damped we can achieve jitter attenuation.

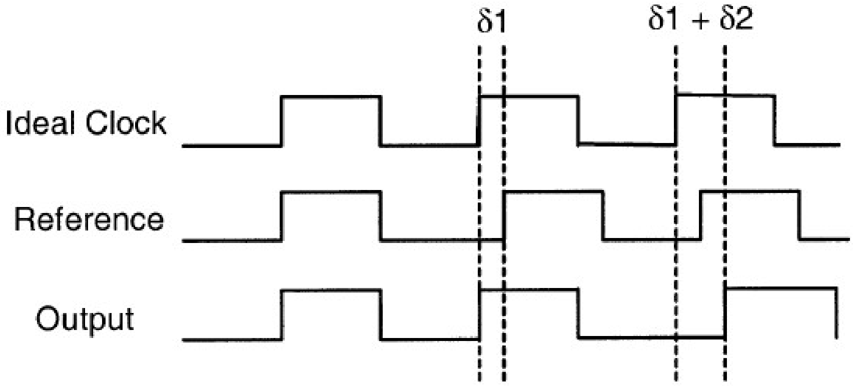


Figure 1.10: Timing diagram illustration of jitter peaking

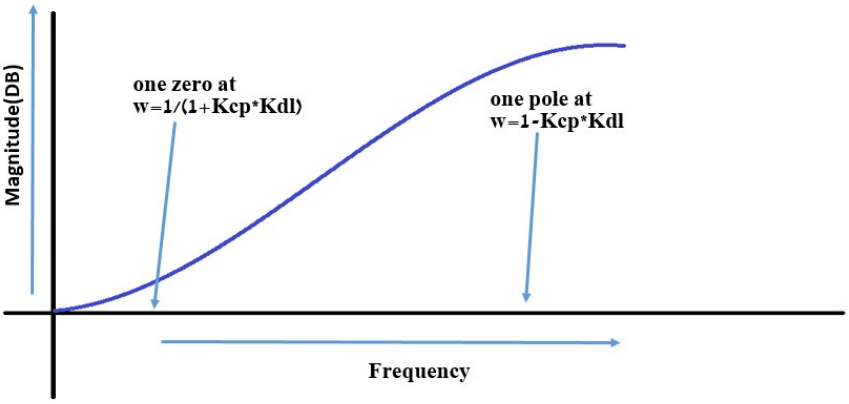


Figure 1.11: Bode plot for jitter transfer characteristics of first order analog DLL

1.2.3 Jitter analysis in digital delay locked loop

Power supply noise and substrate bias noise can alter the delay of delay elements. We can adopt a various circuit design technique to mitigate effect of noise of power distribution networks on delay amount of delay line. Supply voltage from constant reference to delay elements to avoid power supply noise. Use local current mirror to mitigate locally generated and distributed power distribution network noise. Current starve delay elements and differential delay elements are less prone to power supply and power distribution noise.

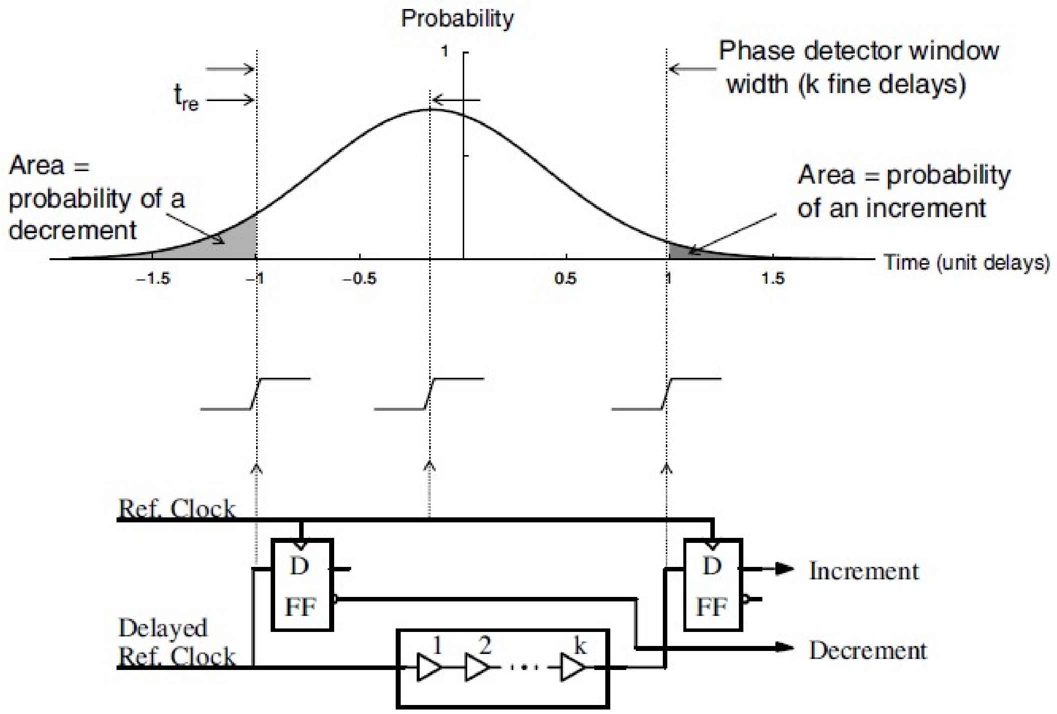

Probability
1
Phase detector window width (k fine delays)
t_{re}
Area = probability of a decrement
Area = probability of an increment
−1.5
−1
−0.5
0.5
1
1.5
Time (unit delays)
Ref. Clock
D
FF
D
FF
Increment
Decrement
Delayed Ref. Clock
1
2
k

Figure 1.12: phase detector and probability distribution function of reference clock

Generally digital DLL exhibit all pass jitter transfer characteristics.so in forwarded clock serial I/O architecture DLL is very efficient.

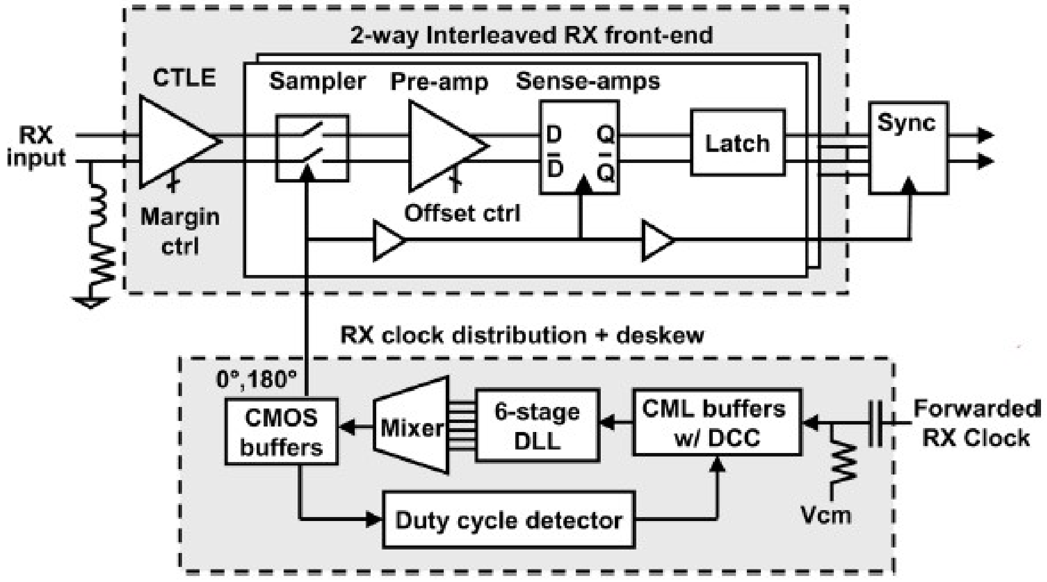


Figure 1.13: DLL based forwarded clock architecture

some advantage of DLL over PLL in clock and data recovery circuit are mentioned below.

- DLL displays an all-pass jitter transfer characteristics.
- All-pass jitter transfer characteristics is Desired for correlated jitter.
- It is not desired for uncorrelated jitter.
- No jitter accumulation.
- Inherently stable.
- Simpler and less area than PLL.
- Finite bandwidth of DLL delay line can result in jitter amplification

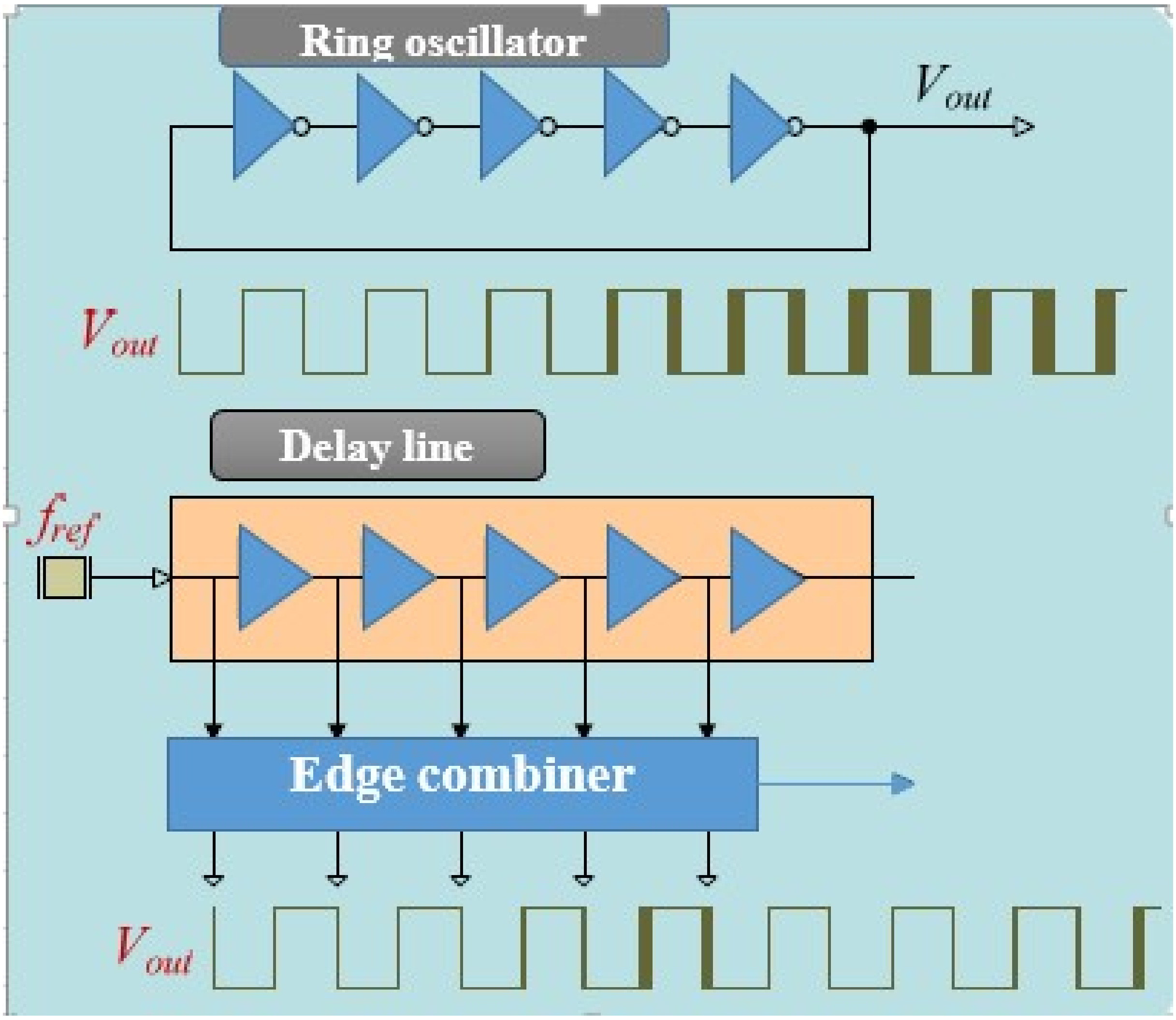


Figure 1.14: Jitter in PLL and DLL

1.3 Application of delay locked loop

delay locked loop has wide range of application.

- clock tree synthesis in multi-voltage domain.
- frequency multiplier.
- DDR-DRAM
- Delay compensation and multiphase clock generation

1.3.1 clock tree synthesis in multi-voltage domain

Today in very high density VLSI chip operating at GHz frequency must have a low power design. Multi voltage domain (MVD) design is one of the most common technique for low power design in today's integrated circuit design. Due to frequent transition of clocks from one domain to another domain via level shifter create a relative phase difference between the clocks in clock tree reducing the data valid window and eventually reducing the speed.By using simplified version of DLL in each domain we can achieve a controlled (adaptive) skew in clock path by which we can achieve a phase synchronization between clocks from different voltage domain.

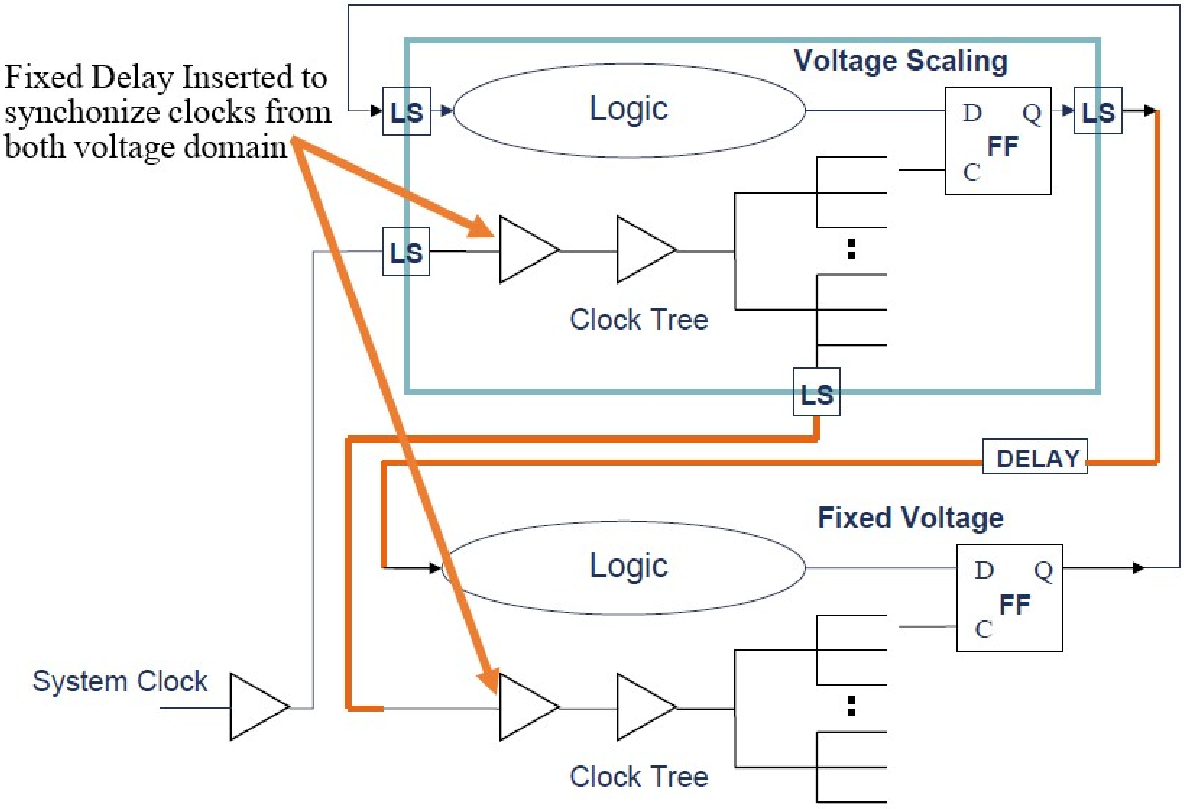


Figure 1.15: pseudo synchronization at voltage scaling domain boundary

Here a fixed amount constant delays are inserted in clock tree. Delay is independent of actual skew between clocks and change with variation in process, voltage, and temperature. So sufficient margin must require in timing budget to compensate an uncertainty create by PVT variation. So achievable speed down comparatively.
To design a clock trees in multi voltage domain(MVD) with optimum timing budget and get maximum throughput possible in given technology, skew between clocks must be adaptable to the PVT variation.to full fill this requirements we can use simple version of DLL in clock tree synthesis of multy voltage domain (MVD) .

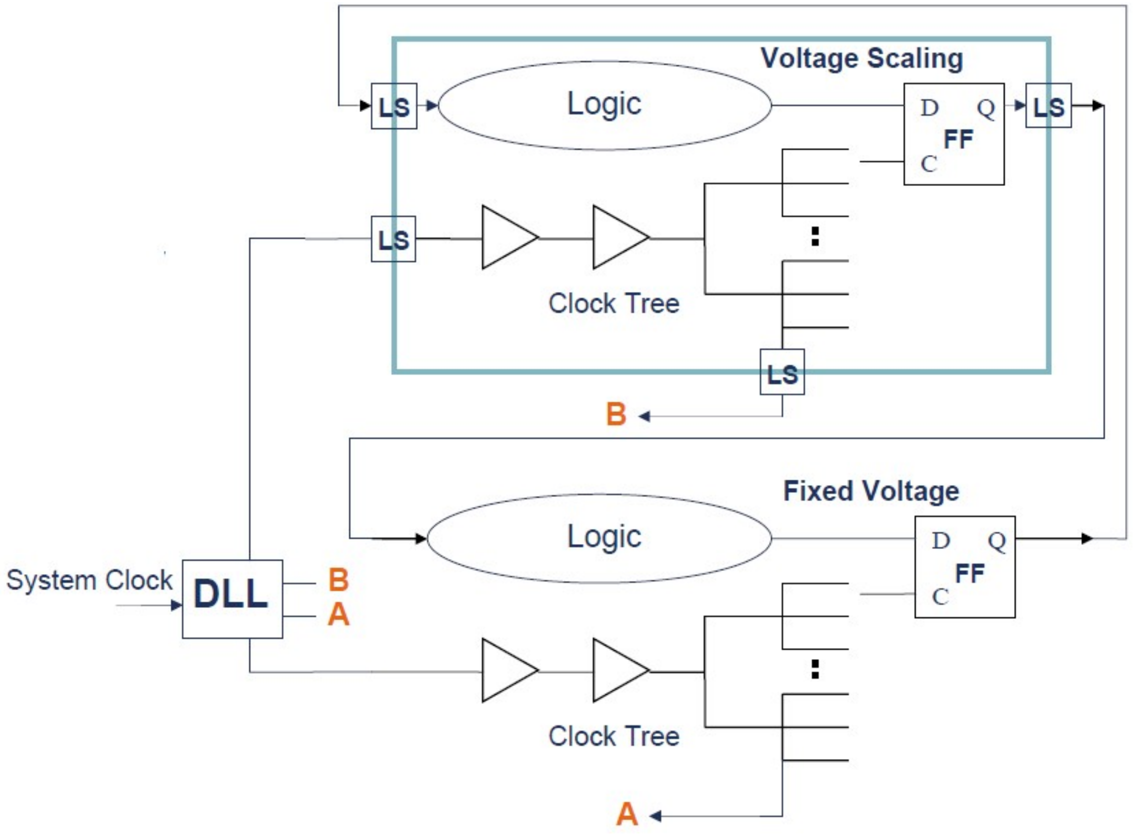


Figure 1.16: DLL synchronization at voltage scaling domain boundary

here once delay locked loop enters in locked state all clocks in all domain get synchronize.so clock edge at all sampler (D-flops) fall somewhere in data valid window and no shrink in timing and voltage margin. that ensure expected bit error rate within multi-voltage inter-domain communication.it ensure a highest possible performance of system but it add an extra design complexity and increase area and power requirements. compare to PLL and other for clock synchronization across multi-voltage domain DLL is still a very good solution for this particular application.

1.3.2 Frequency multiplication

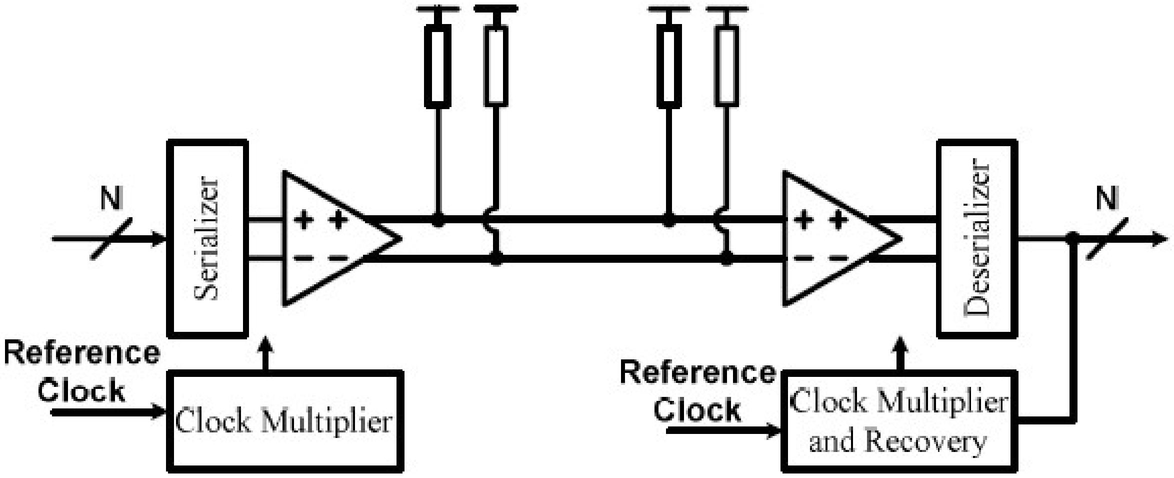


Figure 1.17: Basic block diagram of high speed serial I/O

As shown in figure for high speed serial I/O high precision clock multiplier required for clock generation. Precision timing circuits and clock generator are designed using phase-locked loops (PLLs) or delay locked loop (DLL). Due to higher order system PLL based clock multiplier are very hard to design. There are so many advantage of using DLL based clock multiplier over PLL based clock multiplier. Voltage controlled oscillator design in PLL is difficult to achieve a low jitter level compare to a delay line and edge combiner design in DLL.

Typical clock multiplier circuit design with DLL is shown in figure.

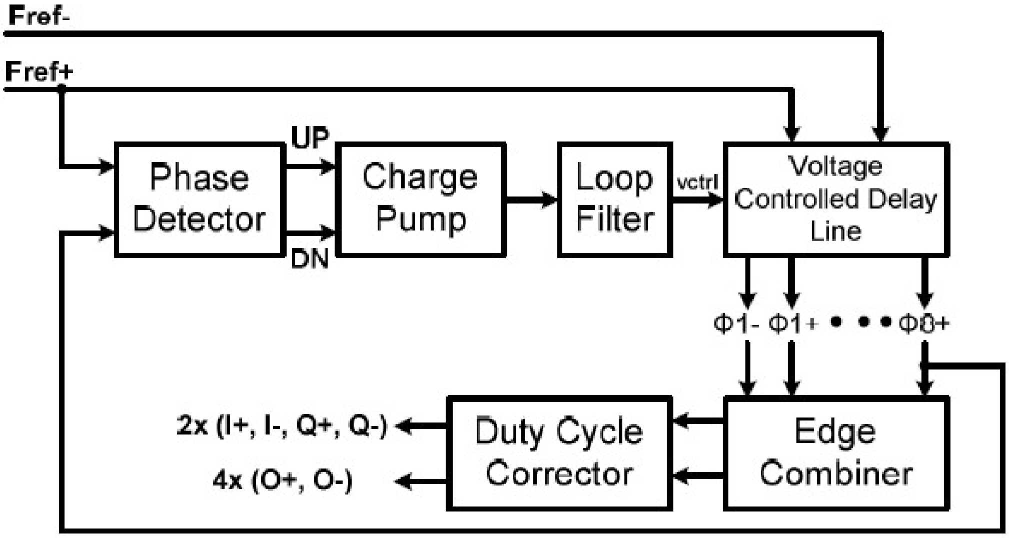


Figure 1.18: basic DLL based clock multiplier

In high speed serial interface major noise contributor is frequency multiplier. Generally frequency multiplier takes a low frequency clean clock as a reference clocks and then it multiplies reference clocks by PLL and DLL to get a high frequency clocks. PLL or DLL contain VCO (PLL), VCD (DLL), charge pumps and phase detector. All this modules of clock multiplier are very noisy and create a maximum amount of jitter in high speed serial I/O application.

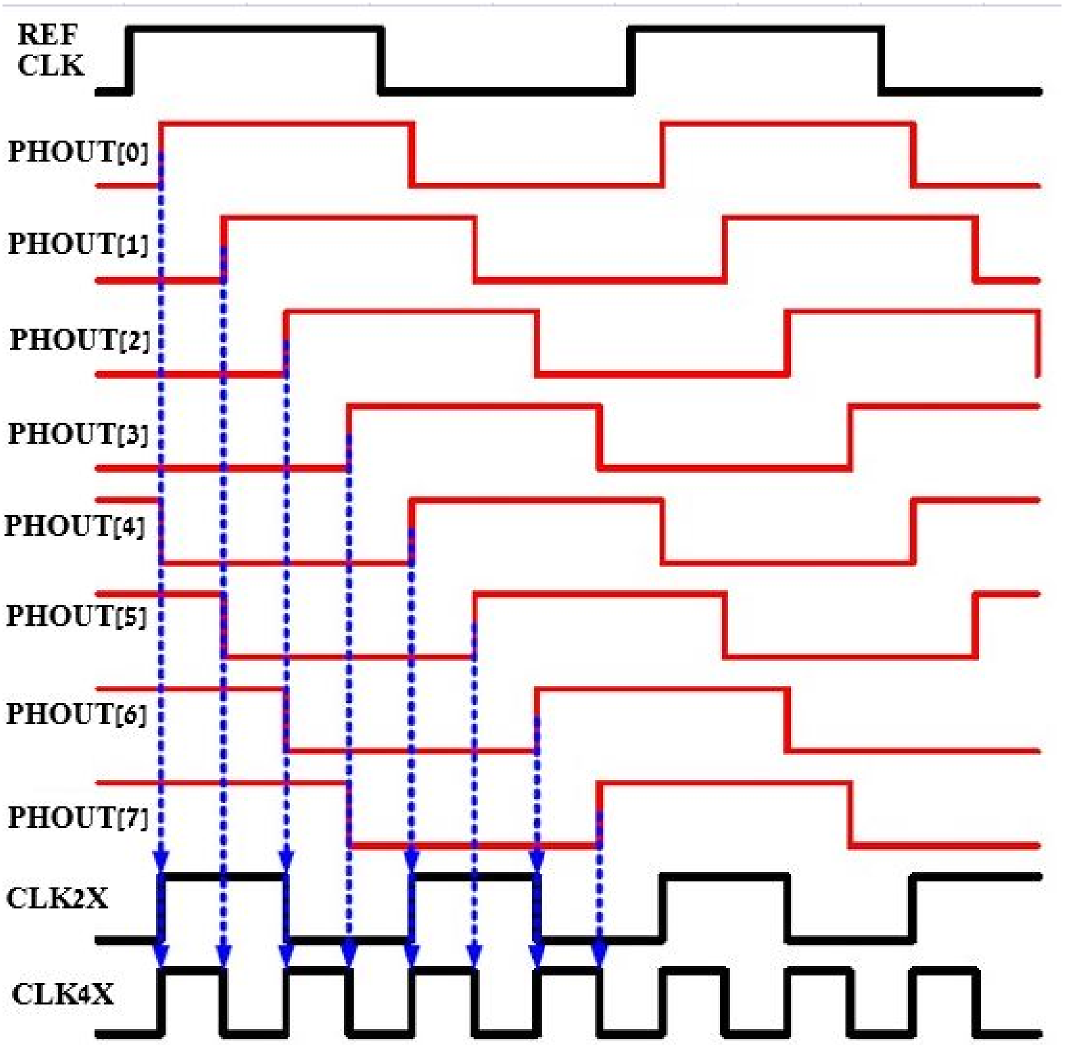


Figure 1.19: frequency multiplication by processing the output phases of DLL

1.3.3 DDR-DRAM

In double data rate (DDR) DRAM controller design bidirectional data strobe (DQS) signal control the data transfer. Bidirectional data strobe signal transmitted with output data signal (DQ) .fig giving an approximation of timing budget while read operation. Phase of DQS and DQ should be ideally aligned. But pin-o-pin mismatch between the DQS and DQ and skew induced due to PCB phase difference exist between DQS and DQ when they reach at controller. Which left data valid window very narrow reducing the timing margin. If we delay DQS with 90 degree phase shift to center of the data window we get a wide a wide data valid window and eventually an enough timing margin. To achieve a 90 degree phase shift we can use a simple buffer chain for delay. But this open loop arrangement have disadvantage of variation in skew with variation in process, voltage and temperature. Digital delay locked loops are preferred for this application because of its various characteristics which best suit for this application such as it has a fast locking time and ease of migration over different processes with interface of high speed DDR memory.

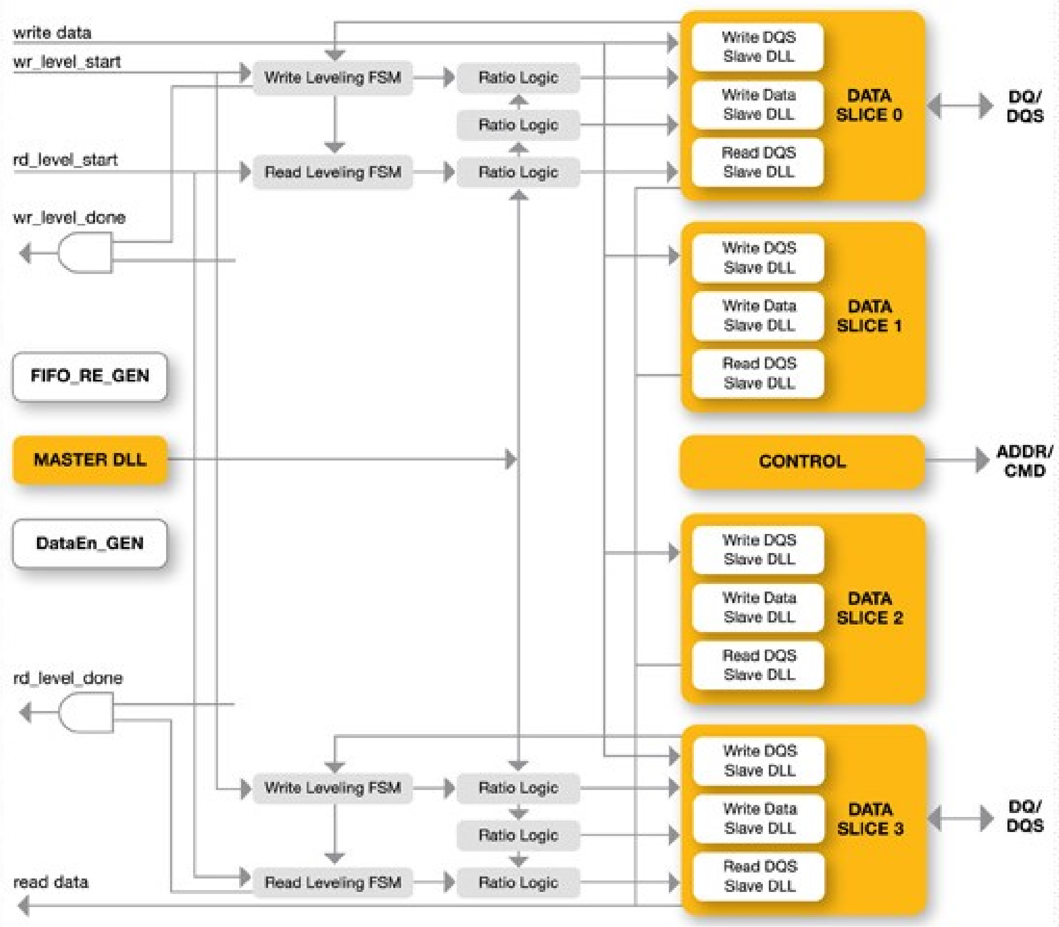


Figure 1.20: All digital high performance DDR PHY/DLL implementation

At low signaling speeds, the data valid window (the time over which data can be sampled reliably by the receiver) can be large. Even in the presence of a substantial shift in the data valid window across operational extremes, the resulting data valid window can still be large enough to transmit and receive data reliably. This is the case for DRAM technologies such as SDRAM. However, for higher-speed DRAM technologies such as RDRAM and DDR, variations in process, voltage, and temperature can result in the loss of the data valid window.

1.3.4 Delay compensation and multiphase clock generation

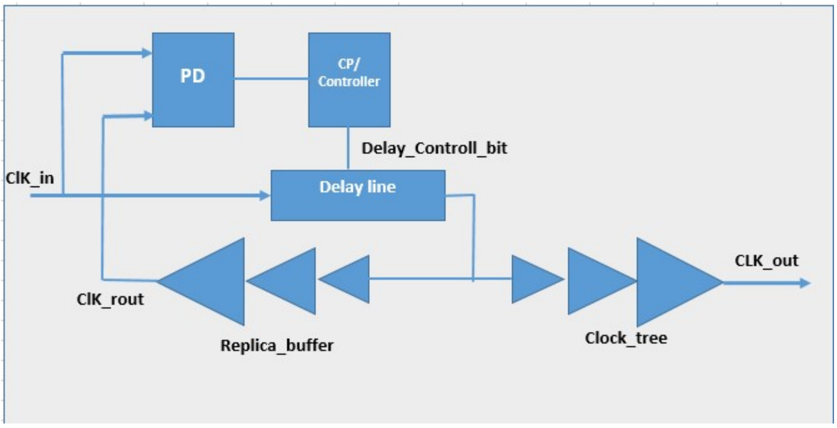


Figure 1.21: delay compensation by DLL

Design of clock distribution network must me very aware to PVT variation effect on that and mitigation of that. As shown in figure if common data line is sampled by clk_in and clk_out then phase of both clocks must be match. Due to inherent delay of clock tree and it variation due to pvt variation needs a continuous calibration and same is achieve by DLL as sown in figure.

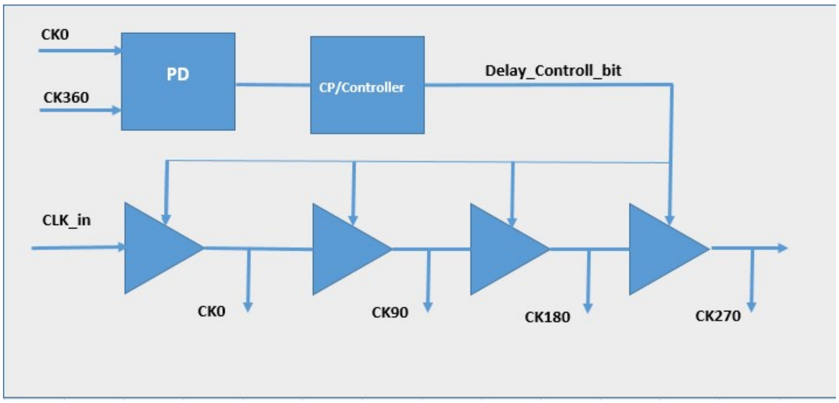


Figure 1.22: Multi phase clock generation

Multiphase clocks are useful in many application belongs to various domain. In high speed serial interface multi-phase clocks process data streams with a speed much higher than internal clock frequency.multiphase clocks are combined to produce the desire output frequency for the synthesizer in clock multiplier. clock constraint can be ease in per-charge logic in microprocessor. multi phase clock can be use to find better sampling point for analog-to-digital converter in wireless LAN base band design to improve overall system performance.

Chapter 2

Literature Review

2.1 Types of DLL

Mainly we can define a two types of delay locked loop based on nature of operation

- digital delay locked loop.
- analog delay locked loop.

Both digital delay locked loop and analog delay locked loop have their own merits and demerits. What kind of dll to be used is decided by application and performance expectation from dll. We can compare both type of dll by their various performance parameter. Those performance parameters are calibration range, delay regulation nature, power consumption, jitter level, I/O characteristics nature.
Combination of one or two above parameter decide what kind of operation nature require for given application. In some cases we can use function of both types dll to meet an application requirements and that form a hybrid configuration of dll. For better control and precision we should use either digital or analog configuration entirely.

2.1.1 digital delay locked loop-performance parameters

1-regulation range:

regulation range for all digital delay locked loop is very high. Factor behind the very high regulation range is digital nature of delay line. Generally digital delay line composed of coarse and fine delay elements. Coarse delay elements generally provide a large time step in initial phase of adaptation and then fine delay elements do fine tuning of phase with comparatively small time step. So with coarse time step we can achieve wide time range under a reach of calibration and eventually we end up with a wide regulation range. So this is an advantage of using a digital delay line over analog delay line where a high uncertainty of relative phase exist. With wide range of regulation dll never goes in inoperable region.

2- Delay regulation characteristic:

delay regulation characteristic for digital delay locked is very linear. Because of discrete digital delay elements control by monotonic code gives a uniform delay step size which eventually leads characteristics into linear in nature. For linear delay characteristics across all operating condition (process, voltage, and temperature) individual delay elements must be immune enough to PVT variation.

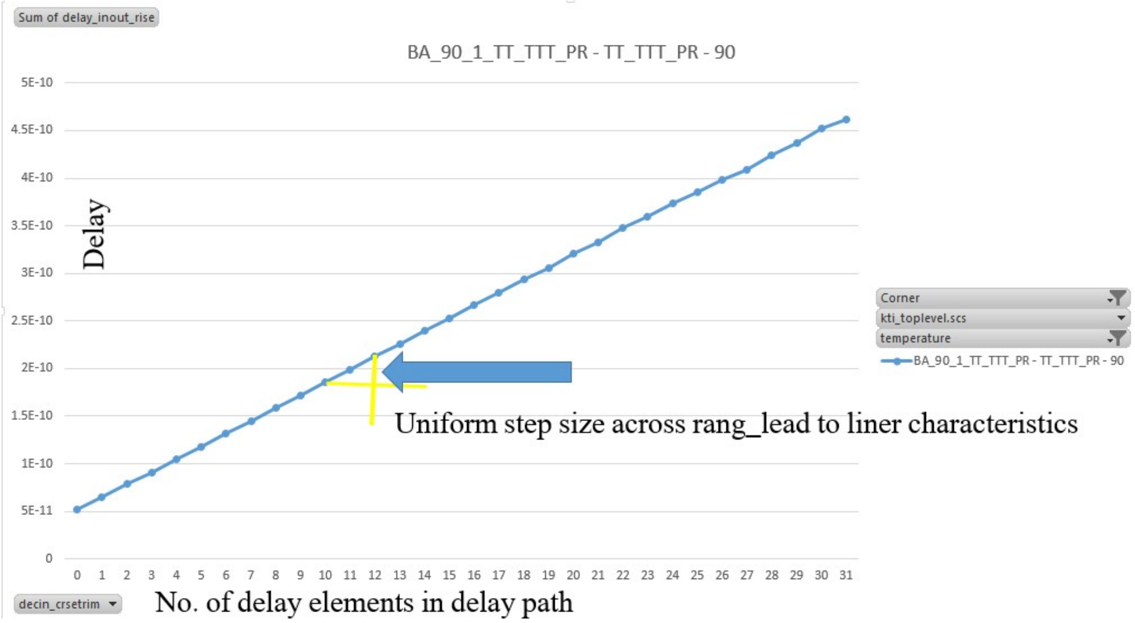


Figure 2.1: Linear delay characteristics

3-power consumption:

digital delay locked loop generally consume low power relative to analog delay locked loop. In digital delay locked loop operation of delay elements are majority c-mos. So it provide a very low statics power dissipation. Number of caps and is lumped value is very low in digital delay locked loop leading to very low switching power dissipation.

4-jitter:

- jitter level in digital delay locked loop is comparatively high due to coarse regulation. There is a high correlation coefficient between power supply and delay of delay elements. So power supply noise create a high jitter in coarse regulation unit.

5-phase resolution:

due to discrete (digital) phase control phase resolution of digital delay locked loop is very poor. Phase resolution depends on fine delay line delay step size.

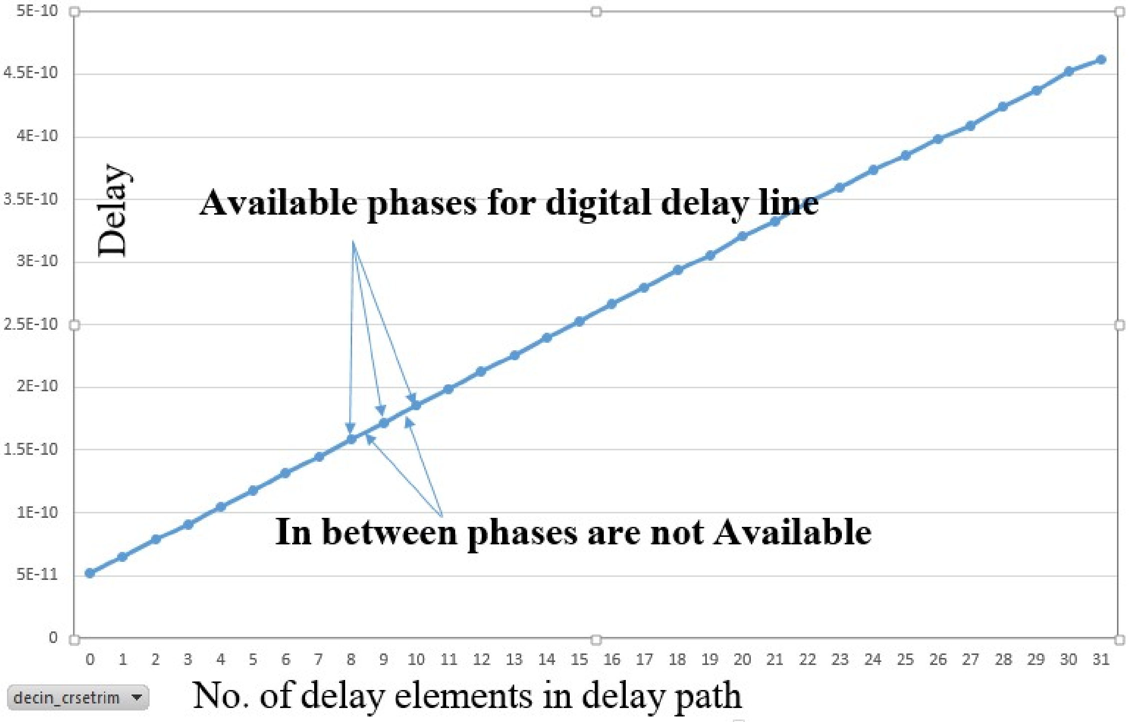


Figure 2.2: delay resolution for digital delay line

2.1.2 analog delay locked loop-performance parameters

1-regulation range:

for analog delay locked loop delay regulation is small compare to digital delay locked loop. In analog delay locked loop delay lines are voltage control delay line. Where delay is control by modulating a load cap value. Generally they use MOS-cap for load cap in delay elements. And hence Range of load cap value is limited by power and area limit of design. For stringent power requirements load cap range reduce to very narrow range which leads to a very small regulation range. Because of very narrow regulation range it cannot control an operation with highly uncertain relative phase difference. Analog delay line has very narrow clock-decked capability compare to digital delay locked loop. So application where a large difference between clock paths exist there we should use digital delay locked loop instead of analog delay locked loop.

2- Delay regulation characteristic:

delay regulation characteristics for analog delay locked loop is highly nonlinear. We can reduce an analog voltage control delay line by following model.

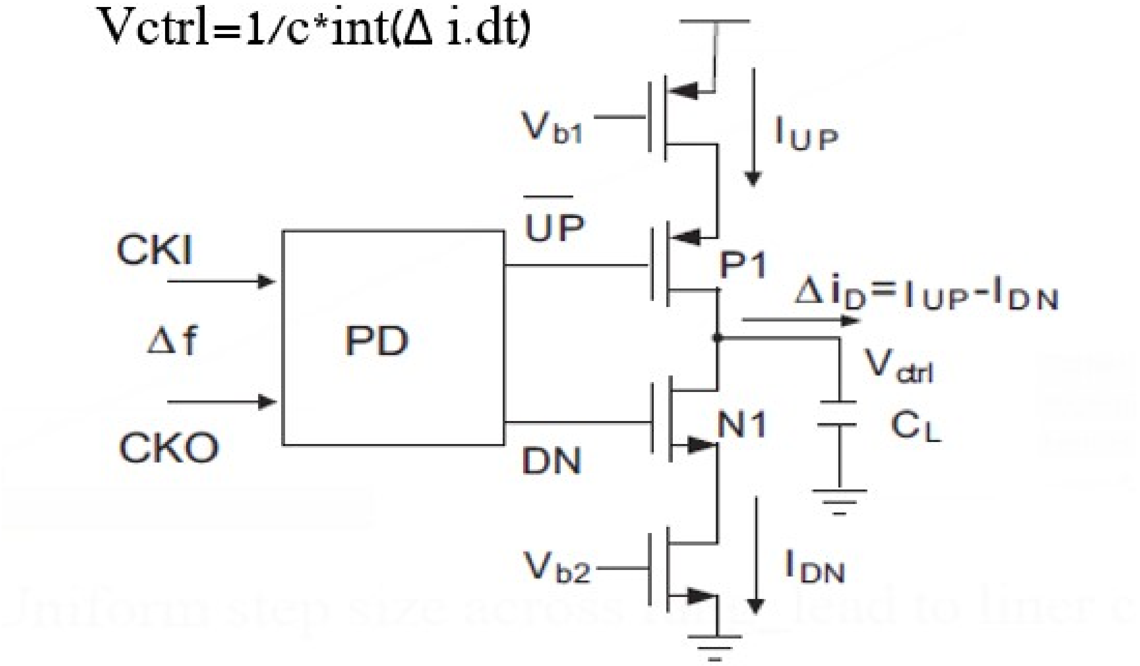


Figure 2.3: Vctrl generation in analog delay locked loop

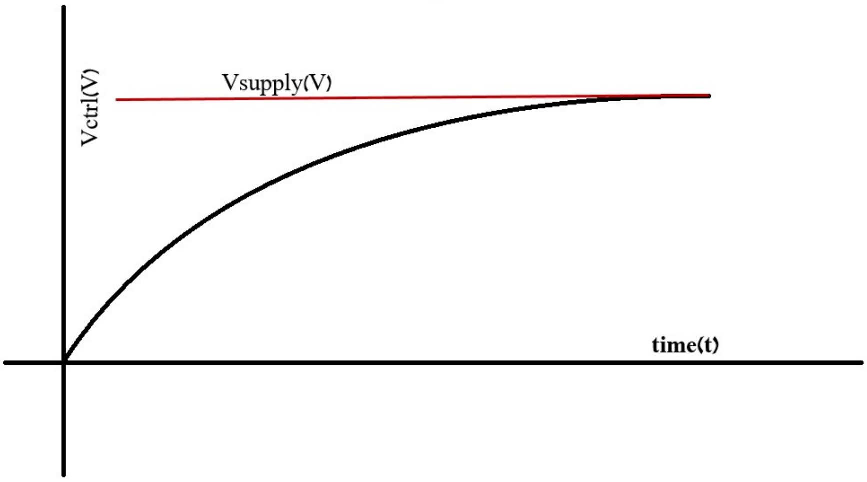


Figure 2.4: nonlinear characteristics of Vctrl VS relative phase difference between two clocks

3-power consumption :

power consumption for analog delay locked loop is generally very high compared to digital delay locked loop. Generally voltage control delay line in analog delay locked loop adopt a cstrv delay elements. Large value of load capacitance create a large switching power dissipation.

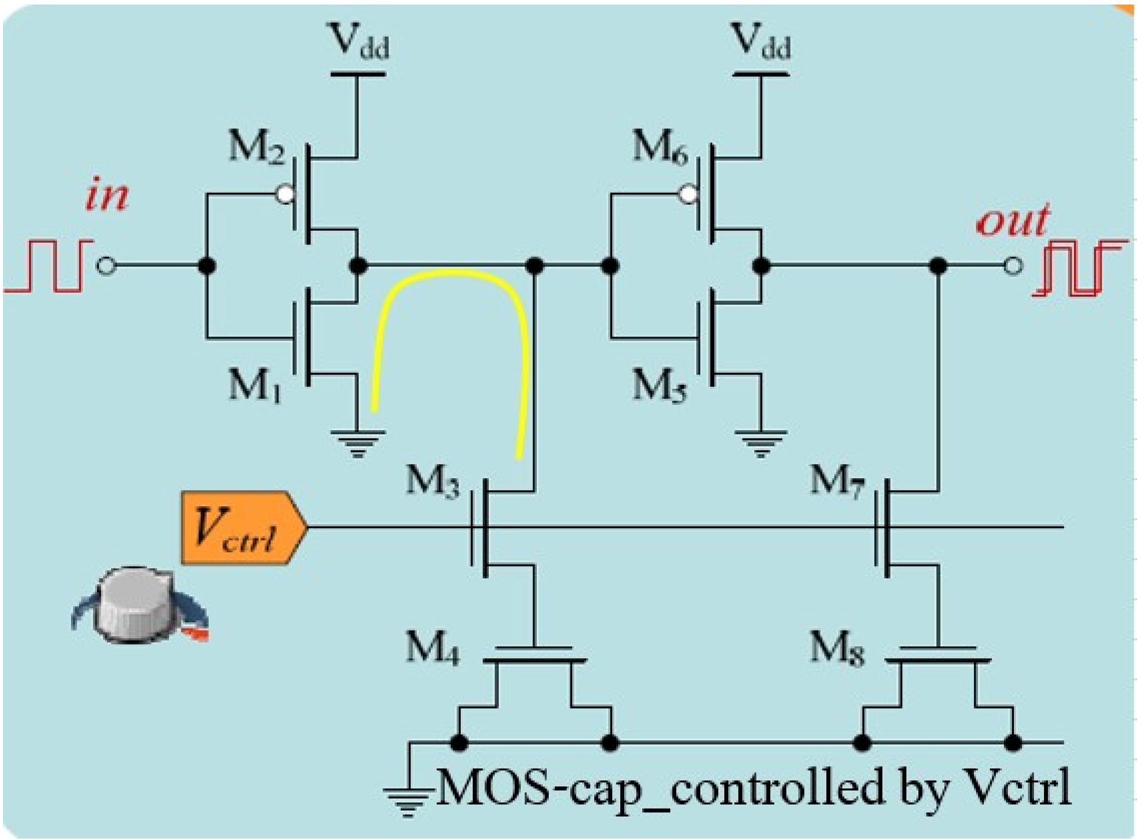


Figure 2.5: power dissipation in voltage control delay in analog delay locked loop.

4-jitter :

jitter level in analog delay locked loop is comparatively low compared to coarse controlled delay locked loop. Delay control in voltage control delay lines are very smooth and continues random switching in signal path does not exist in case of analog delay line which leads to a low jitter accumulation in analog delay locked loop.

5-delay resolution :

delay resolution for analog delay line is very good. All possible phase can be generate by voltage controlled delay line.

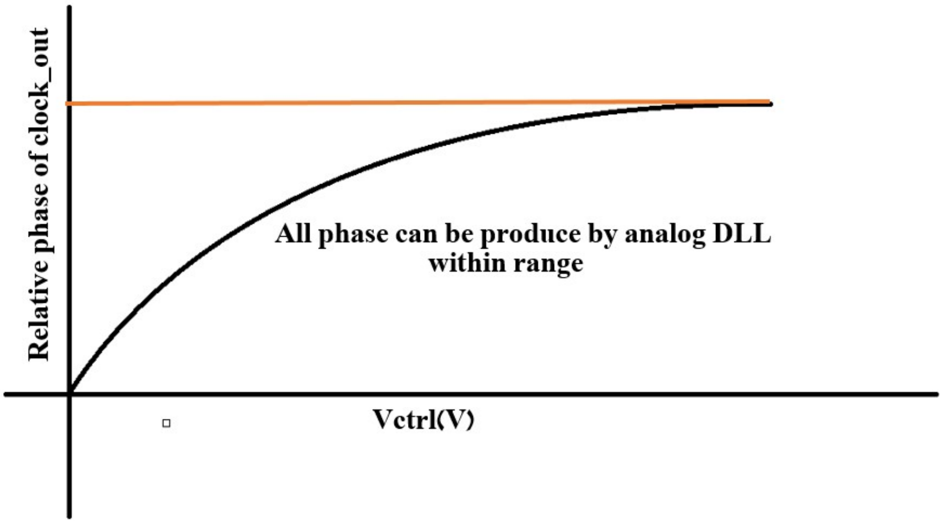
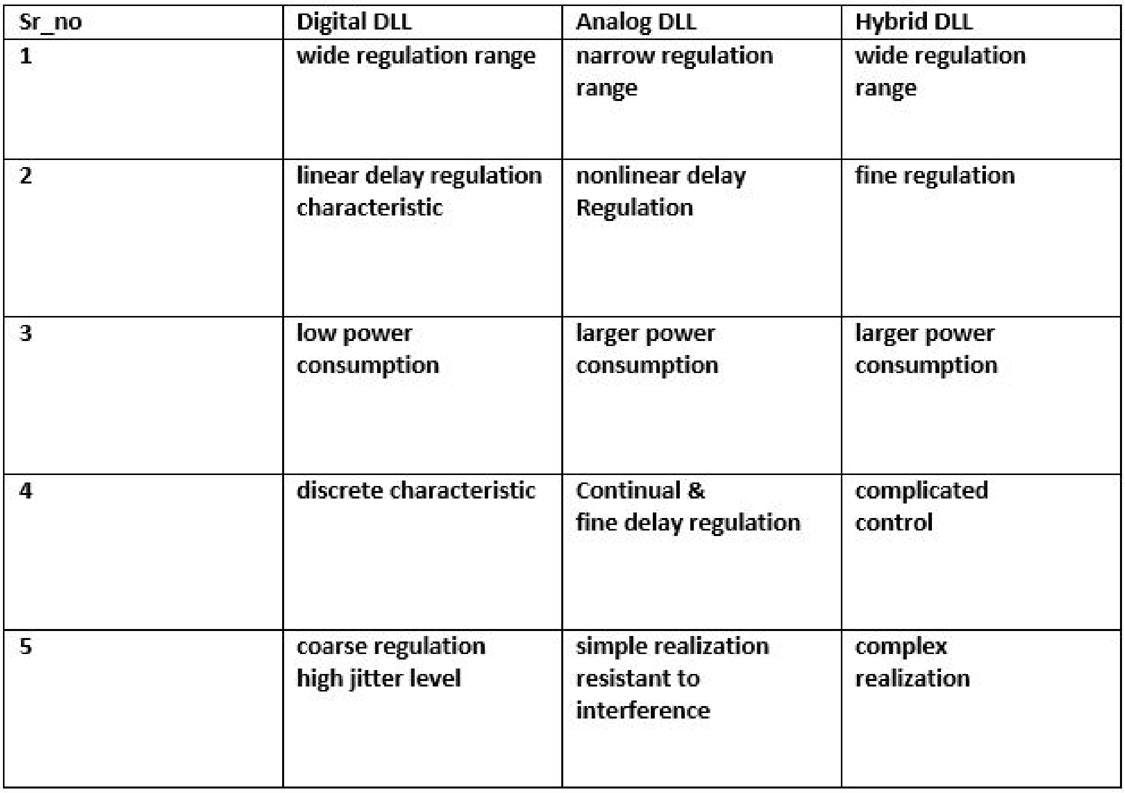


Figure 2.6: Delay resolution for analog delay locked loop

2.1.3 performance summary



Sr_no	Digital DLL	Analog DLL	Hybrid DLL
1	wide regulation range	narrow regulation range	wide regulation range
2	linear delay regulation characteristic	nonlinear delay Regulation	fine regulation
3	low power consumption	larger power consumption	larger power consumption
4	discrete characteristic	Continual & fine delay regulation	complicated control
5	coarse regulation high jitter level	simple realization resistant to interference	complex realization

Figure 2.7: SUMMARY OF PERFORMANCE PARAMETER FOR DIFFERENT DLL

2.2 Analog delay locked loop

The key component in designing an analog delay locked loop are mentioned below

- voltage control delay line.
- charge pump
- phase detector.

We will discuss each component and its parameter affecting the performance of DLL individually.

2.2.1 Voltage control delay line

There are different types of configuration available for voltage controlled delay line. We will go through all relevant configuration to explore their advantage and disadvantage.

current starved delay elements

We can abstract operation of current starve delay elements as Vctrl controlling reference current of current mirror to control the speed of subsequent inverter. Due to mirror controlled delay power supply variation induced jitter is very less. So supply independent operation is great advantage for current starved delay elements. Delay control is independent of load cap value so switching power dissipation is less. It required two more transistor for each delay elements.

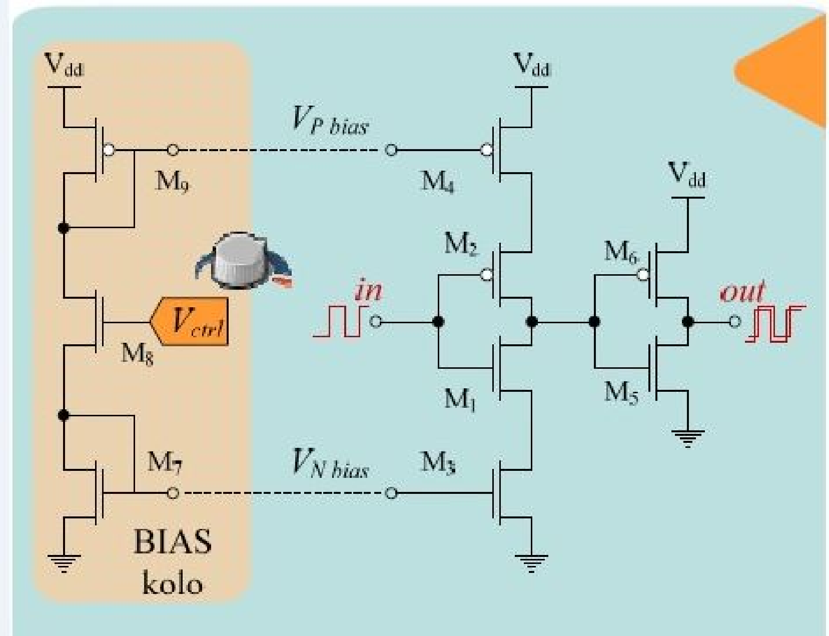


Figure 2.8: current starve delay elements

Capacitive load Delay elements

Due to large load capacitance switching power dissipation of capacitive load delay elements is very high. Highly nonlinear delay characteristics. Delay is directly proportional to supply voltage so supply noise create a high jitter.

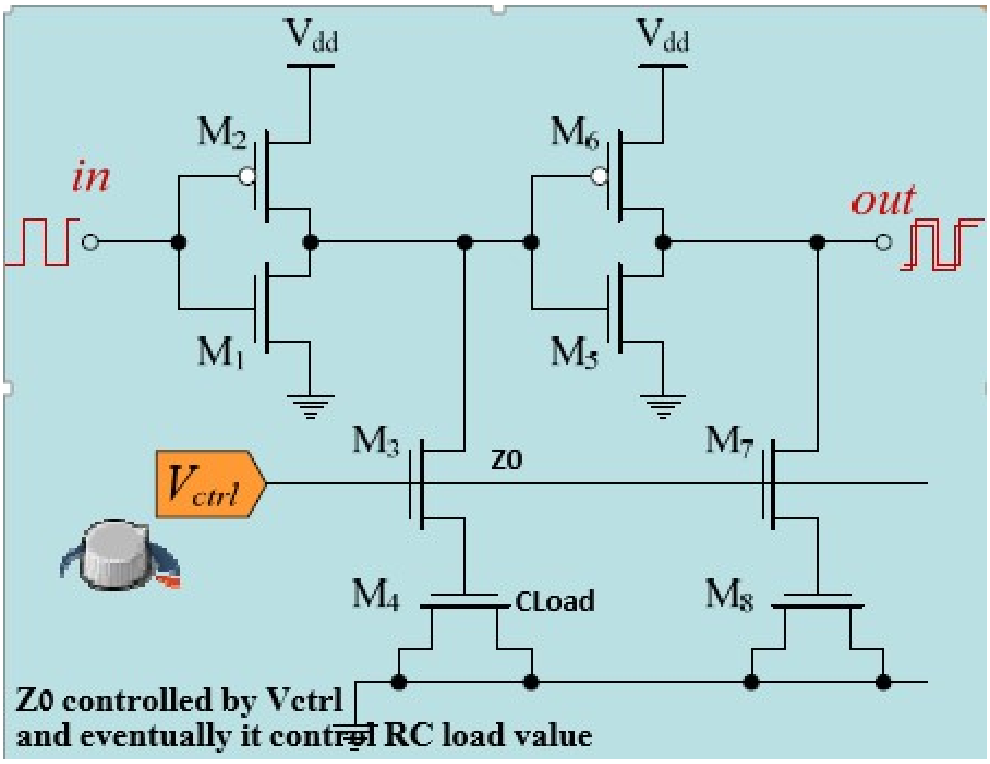


Figure 2.9: capacitive load delay elements

delay elements with differential Input/output

Good resistance on interference and compliance of rising and falling edge delay Differential delay elements are design such that it use limited charge/discharge current and parasite output capacitance to obtain delay. Which leads us to a very small amount of switching power dissipation.

delay resolution of differential delay elements is very good.power supply jitter of differential delay elements is relatively low.better common mode noise rejection and less duty cycle distortion.it required level transition . It remove any skew between differential signal . It has least dependency on PVT variation on delay among all analog delay elements.

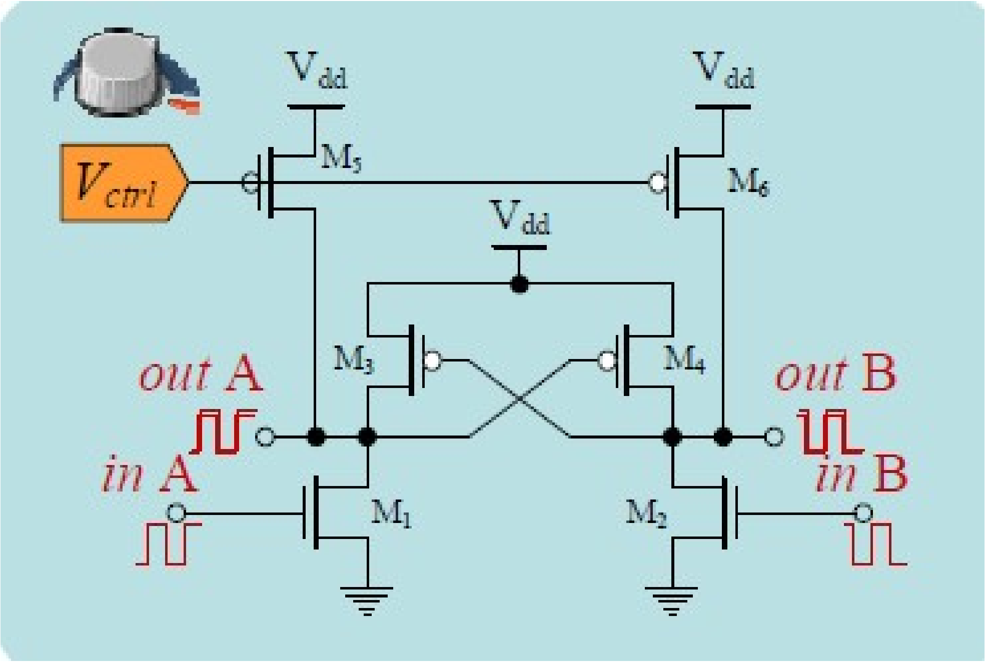


Figure 2.10: delay elements with differential Input/output

Relative delay transfer characteristics

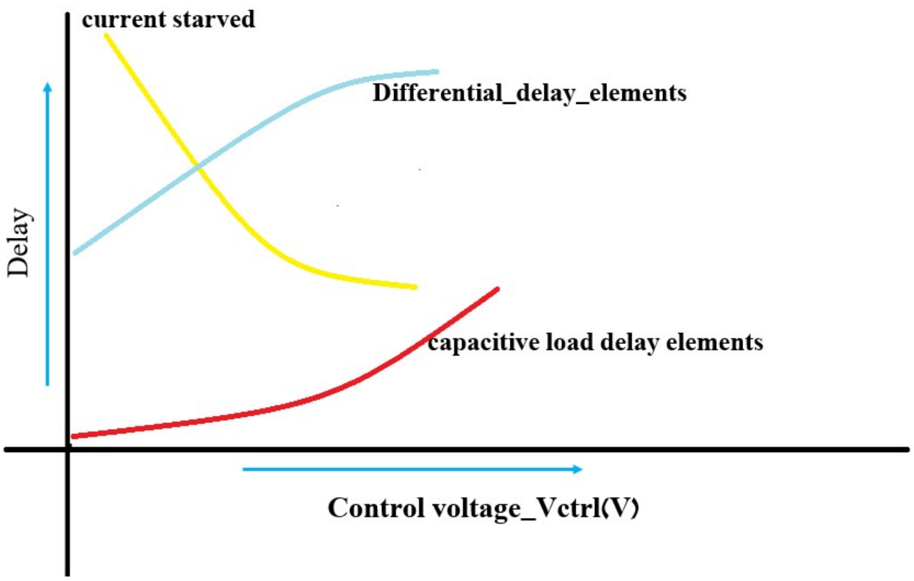


Figure 2.11: relative delay transfer characteristics

As we can see from Figure differential delay elements has very finer delay transfer characteristics among all delay unit. current starve delay elements has better control one delay. capacitive load delay elements are very non-linear and very susceptible to noise.

2.2.2 charge pump

The conventional charge pump has a problem of current mismatch as shown in fig current mismatch generates due to variation in drain and source voltage (VDS) of transistor P1 and N1. To eliminate this current mismatch charge pump output voltage should be exactly half of supply voltage which is not in case of conventional charge pumps. Process variation is another factor which can generate current mismatch. Phase error of get generated due to current mismatch at output of DLL. Charge injection into charge pump capacitor at each clock cycle under mismatch condition is

Q = ICP*.

Where Icp is the charge pump current IUP or IDN, is phase error of phase detector CKI and CKO. The amount of charge in capacitor when DLL is locked is given by

Q = ton*Id

Ton is turn on time of phase detector and Id is current mismatch between IUP and IDN. So phase detector uncertainty and charge pumps current mismatch induce a phase error at output of delay elements. Static phase error generated due to current mismatch at output of DLL between CKI and CKO is given by

= Id*ton /ICP

Current mismatch in charge pumps can create a static phase error at output of delay locked loop so we have to include some technique to mitigate this problems.

Chapter 3

All digital delay locked loop

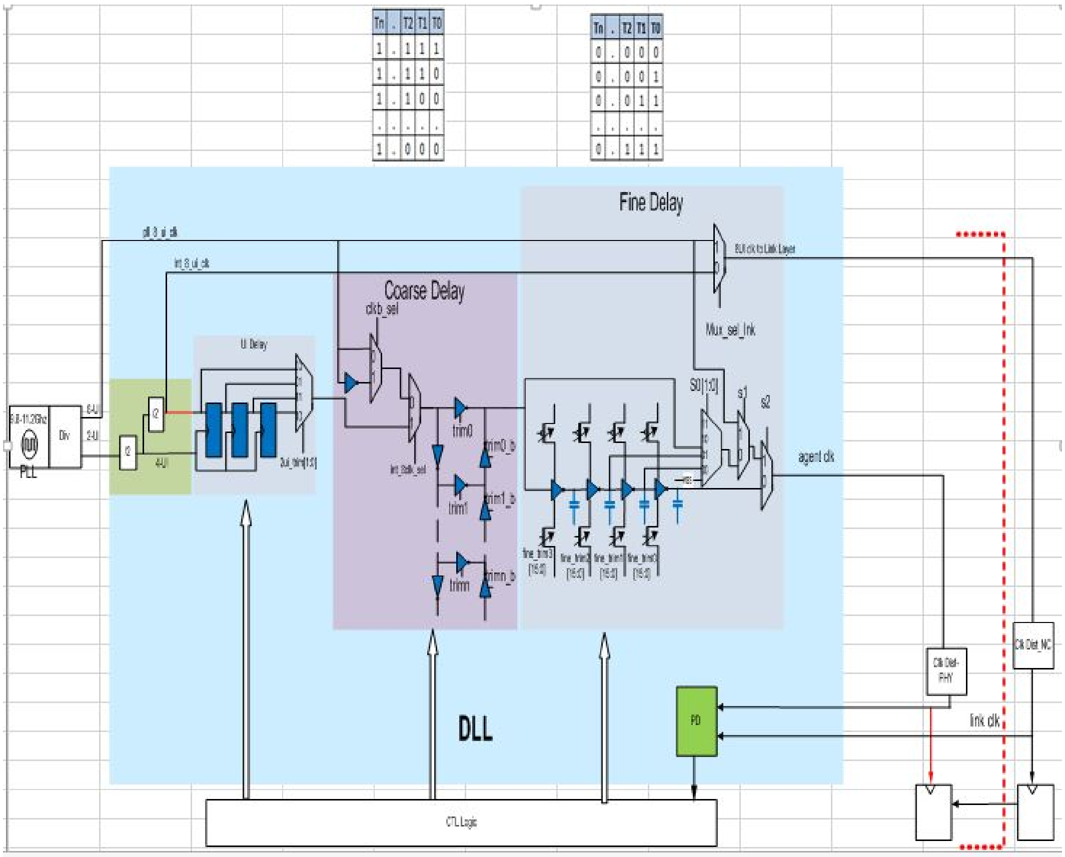
As we have seen in chapter 2 there are two major category of delay locked loop and we have also seen merits and demerits of each type of DLL implementation. I this section we will explore architecture of proposed digital delay locked loop.

3.1 Architecture of all digital delay locked loop

Major component of proposed delay locked loop are

1. Frequency divider
2. Coarse delay line
3. Fine delay line
4. Phase detector
5. Control logic
6. Phy Clock distribution network
7. Node controller clock distribution network

Above block diagram giving us a glance of top level functional and signal flow information. We will look into individual component in detail in subsequent section. Here coarse delay line and fine delay line operation is controlled by a thermal code generated by digital control logic blocks to control the delay of it. No analog controlled voltage used for delay control that's why above configuration is an all-digital delay locked loop configuration. Because of digital nature of operation no charge pump and loop filter to generate an analog control voltage.so little clock path jitter generated due to DLL operation.



Tn	.	T2	T1	T0
1	.	1	1	1
1	.	1	1	0
1	.	1	0	0
.	.	.	.	.
1	.	0	0	0

Tn	.	T2	T1	T0
0	.	0	0	0
0	.	0	0	1
0	.	0	1	1
.	.	.	.	.
0	.	1	1	1

Figure 3.1: Block diagram of proposed digital delay locked loop

3.1.1 frequency divider

We used to scale down frequency of source for which we required a phase synchronization by factor of 4. For this purpose frequency divider of d-flip-flops and delay buffer and inverter are shown in fig.

Delay value of buffer and inverter is decided by value of setup time of d-flip-flops and clock frequency. Basic idea here is that when inverted value of Q valued reached to D then it should be at its valid logic level not at an intermediate value between two valid logic level. If we connect an inverted Q net directly to the D net then operation becomes oscillatory with high frequency of oscillation around the mid-logic level and structure stuck in metastability state. So some amount of delay between Q and D is required to allow a D-value to be settle at valid logic level and hence a Q value. What is the minimum amount of delay required to avoid metastability is depend on setup time of d-flip-flops, and maximum limit of delay is limited by clock frequency and hold time of d-flops.

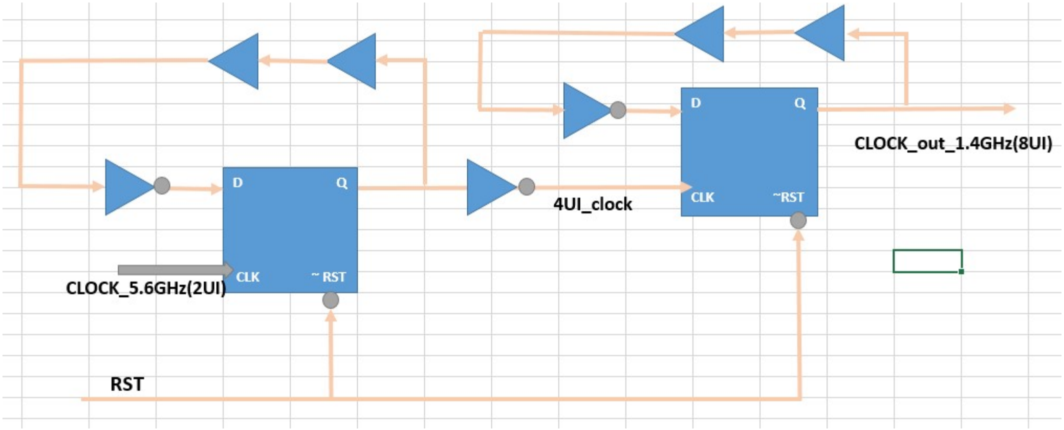


Figure 3.2: basic block diagram of frequency divider

First consider a minimum required delay value evaluation. If d-flop has setup time=Tse_up. Then Tdeay¿Tset_up.

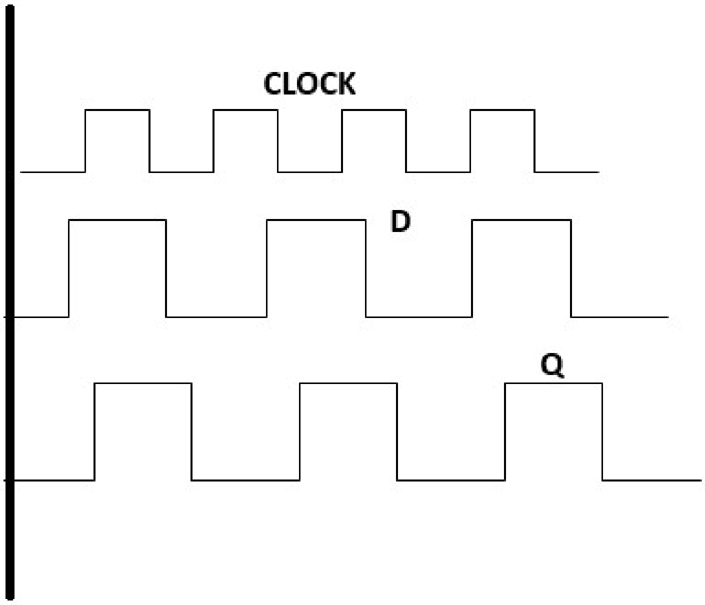


Figure 3.3: delay adjustment in frequency divider

Maximum allowable delay Tdelay (max) =T(clock_period)-Thold. As clock frequency increase max limit of delay decrease so we left with a very small valid delay range and margin also decrease. To count a valid operation across PVT variation of frequency divider we should have an enough valid delay range. Decrease setup and hold time to increase a valid delay range of frequency divider.

Flop 1 has a stringent spec regarding setup and hold than flop 2 because operating frequency for flop 1 is 2X higher than flop 2. All timings and simulation results are presented in chapter-4.

3.1.2 Coarse delay line

As we have seen in chapter 2 that DLL resolution depends on minimum delay value of single delay elements of delay line, adaptation speed and delay range depends on size of delay step and maximum achievable delay by delay line respectively. Coarse delay line take care of later two performance parameter adaptation speed and delay range. Generally delay step size in coarse delay line is very high comparatively fine delay line step size. Delay step size means amount of increments in delay by increasing one code value. By increasing one code value we actually adding one more delay elements in delay line. Actually delay step size is equal to delay of single delay elements. Because of large delay step size in coarse delay line by increasing small number of code we get a sufficient amount delay and hence phase shift. Each code increments takes some time and next code value can be applied after some time decided by DLL closed loop bandwidth. Actually after updating a code value delay line take some time get settle down and reflect its effect as a phase shift at input of phase detector. Phase detector output value get sampled by speed of closed loop bandwidth of DLL so value taken at optimum timing threshold is finally applied to the control of delay line. So as number of code update increase it increase time to lock and adaptation speed decrease. With large step size of coarse delay line with small number of code update we get a significance amount of delay to synchronize the phase of clocks. Also with large delay by small number of delay elements we can get a large delay range which can handle a large amount of uncertainty.

In proposed all digital delay locked loop coarse delay line is CMOS inverter based delay line as shown figure. for less duty cycle distortion delay elements should provide an equal amount of delay for both rising edge and falling edge.In CMOS inverter rising edge delay is controlled by PMOS transistor and falling edge delay is controlled by NMOS transistor individually. for equal rising edge and falling edge delay proper sizing of NMOS and PMOS required. Also this two should maintain equality across all possible PVT variation.

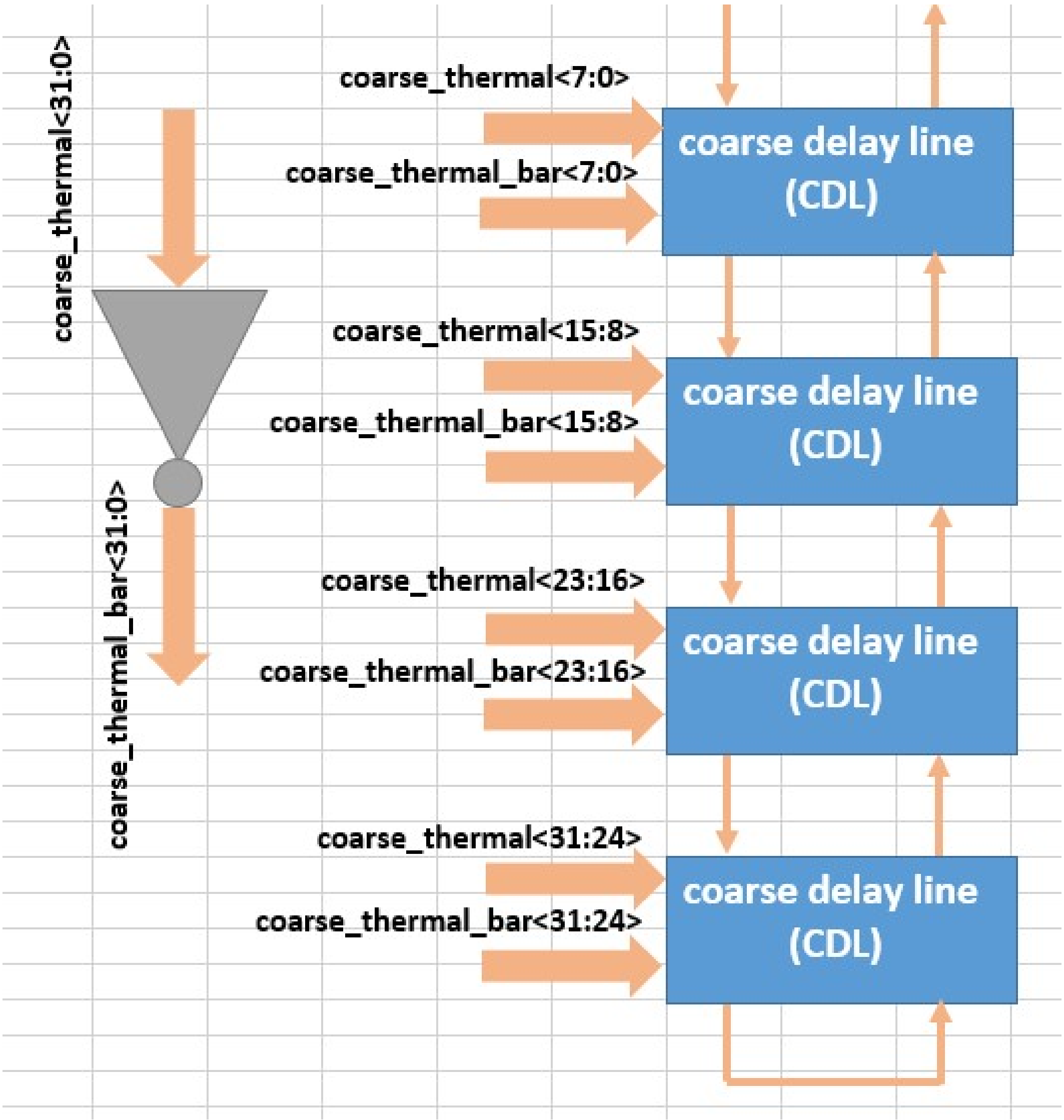


Figure 3.4: top level block diagram of coarse delay line

Tunable range of coarse delay line can increase by cascading more number of coarse delay unit as shown shown in figure. In proposed DLL four basic delay unit each contain 8 intrinsic delay elements are cascaded.As we have shown earlier that operating frequency of DLL is limited by a maximum intrinsic delay value of individual Daley elements. So while cascading more number of delay elements care must be taken that intrinsic delay of basic elements should not increase. so maximum operating frequency put a restriction on maximum cascading.

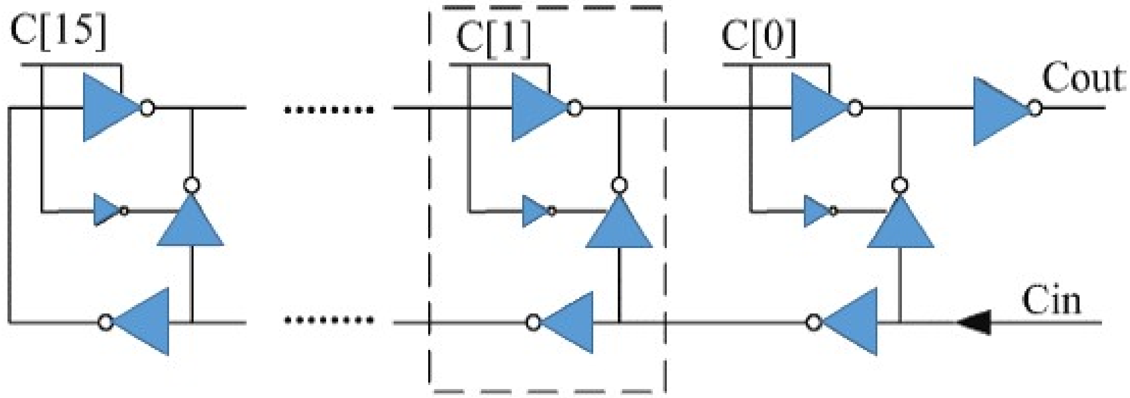


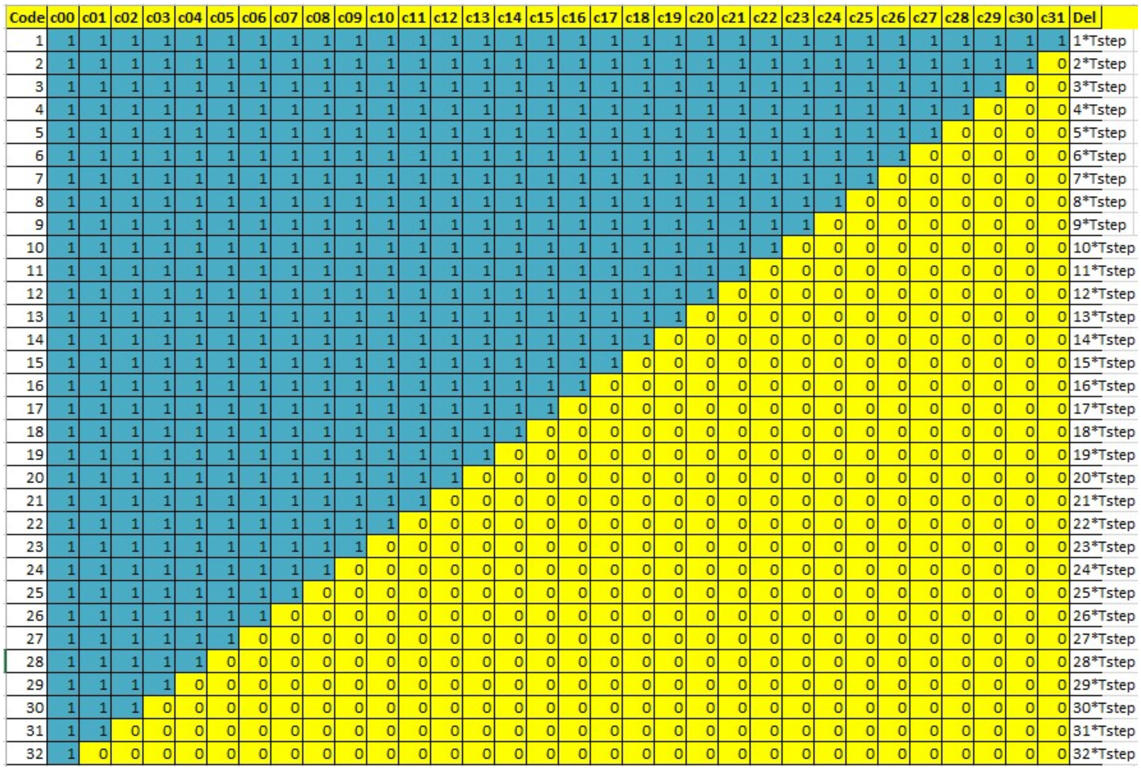
Figure 3.5: coarse delay line

As shown in figure coarse delay line control by thermal code so digital nature of operation.Cin is an input clock and Cout is a delay version of Cin. Consider a case when below code is given as a control of delay line.

C0	C1	C2	C3	C4	....
1	0	0	0	0	....
C_bar0	C_bar1	C_bar2	C_bar3	C_bar4	....
0	1	1	1	1	....

For this thermal code condition clock get a smallest possible path and get delay equal to delay of one delay elements.

Digital controlled delay line is a vital component of digital delay locked loop because ultimately it only defines frequency range, linearity of delay, and delay resolution. For extend the tunable range as we discussed earlier delay of each coarse delay elements should high enough. But to track PVT variation and fine tuning of phase we should have a very granular delay line. We have used 32 block of delay for coarse delay line. Each block count delay of 3-inverter.Timing data and simulation results for coarse delay line are present in chapter-4.



Code	c00	c01	c02	c03	c04	c05	c06	c07	c08	c09	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26	c27	c28	c29	c30	c31	Del
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1*Tstep
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	2*Tstep
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	3*Tstep
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	4*Tstep
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	5*Tstep
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	6*Tstep
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	7*Tstep
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	8*Tstep
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	9*Tstep
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	10*Tstep
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	11*Tstep
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	12*Tstep
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	13*Tstep
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	14*Tstep
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15*Tstep
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16*Tstep
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17*Tstep
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18*Tstep
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19*Tstep
20	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20*Tstep
21	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21*Tstep
22	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22*Tstep
23	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23*Tstep
24	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24*Tstep
25	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25*Tstep
26	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26*Tstep
27	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27*Tstep
28	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28*Tstep
29	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29*Tstep
30	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30*Tstep
31	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31*Tstep
32	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32*Tstep

Figure 3.6: mapping of thermal code in coarse delay line

3.1.3 Fine delay line

As we have seen in chapter-2 that delay resolution of DLL is very important performance parameter in application where we need a very precise phase synchronization and tracking across PVT variation. Delay resolution can be define by minimum delay provide by delay line. Delay fine delay element is comparatively very low compare to coarse delay line. High delay resolution is provided by fine delay line. In calibration process first coarse calibration enable us a fast phase shifting. Once phase difference becomes small than coarse delay resolution disable the coarse adaptation and enable fine delay line for further fine tuning. Simplified block diagram for fine delay is shown in figure.
We use four such fine delay cells to construct a fine delay line. Total of 64 code and 16 for each block thermal code required for fine delay control. Due to large fine delay line range any worst case PVT variation cannot switch operation from fine tuning to coarse tuning and hence jitter component eventually get vanished.

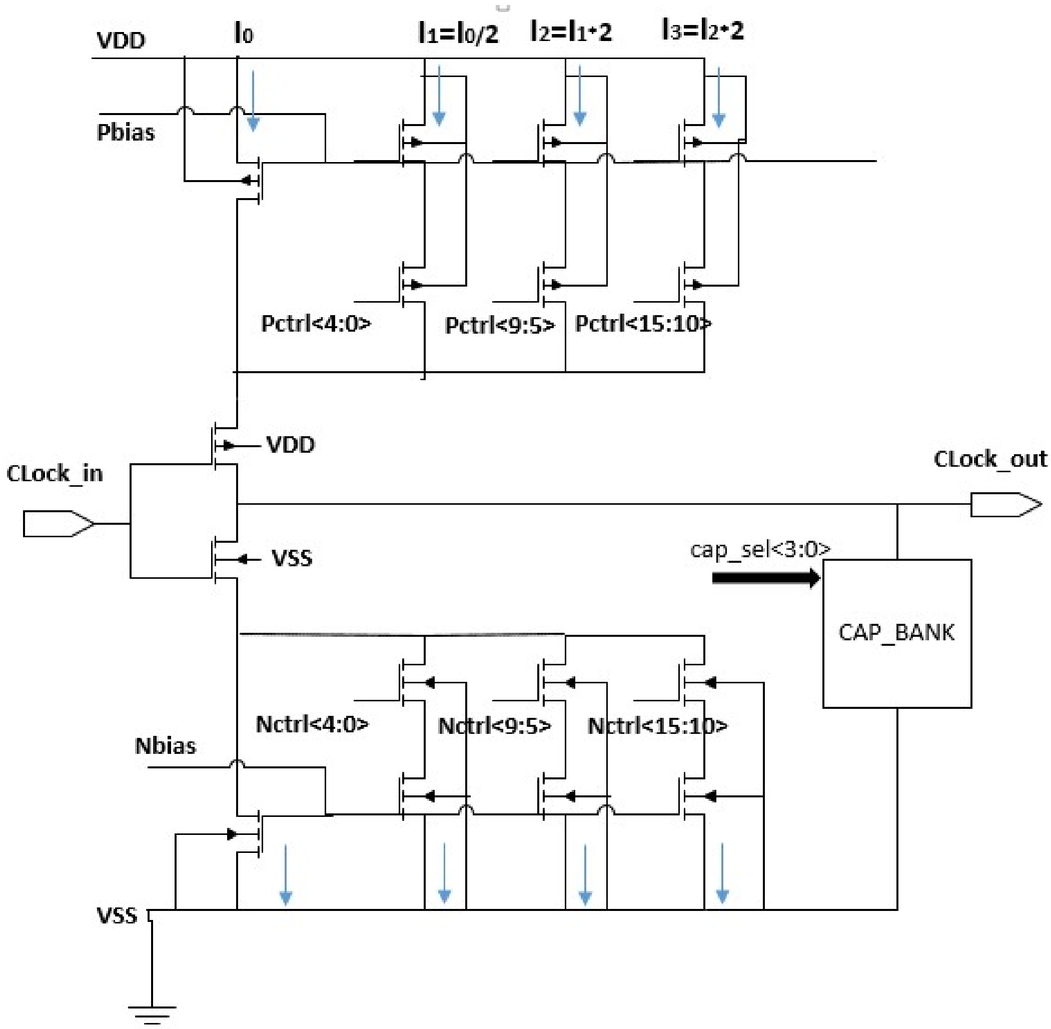


Figure 3.7: - basic block diagram of fine delay line

Maintain a delay line uniform delay granularity is main concern while designing the delay line. So sizing tuning of the transistor in fine delay unit (FDU) need to be tuned carefully. We can analyze fine delay line delay as follow. Transistor biased with a Pbias and Nbias are the biasing transistor which supply a predefine amount of current to inverter. Transistor which gates are controlled by Nctrl and Pctrl are switching transistor which control the amount current to pass to inverter from bias transistor and hence the speed of inverter and delay through inverter.

$$\left[\left(\frac{W}{L}\right)_{k+1}^{\text{eq}}\right]^{-1} - \left[\left(\frac{W}{L}\right)_{k}^{\text{eq}}\right]^{-1} = \left[\left(\frac{W}{L}\right)_{k}^{\text{eq}}\right]^{-1} - \left[\left(\frac{W}{L}\right)_{k-1}^{\text{eq}}\right]^{-1}$$

Figure 3.8: Binary weighted delay relation

Conventional relation between two consecutive (W//L) differences is given by above equation but with large number of biasing transistor to maintain a linear delay by above relationship is not possible. We have a three separate group which are individually following above relation. But as we are moving for higher code we are injecting more and more current into inverter to make it faster.

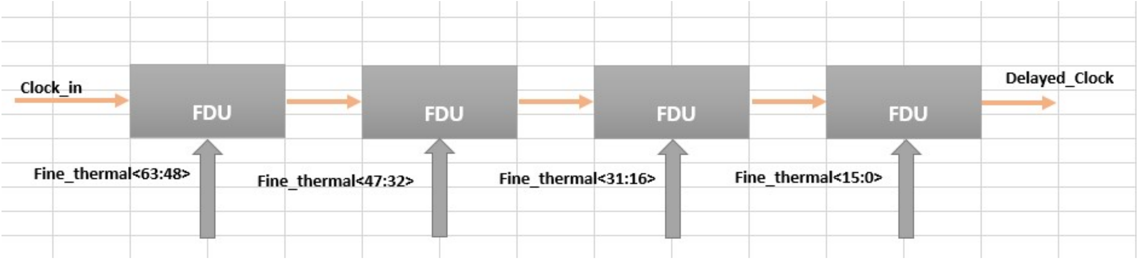


Figure 3.9: Top level block diagram of fine delay line

As shown in each fine delay unit (FDU) contain a four lags fine delay elements structure. When all switched disable then we get a minimum current in inverter and maximum amount of delay from in to out. We configured basic fine delay unit such that for one group of 20 code only one branch of switch can alter the delay. Consider a one example that we are operating in code range [19:0]. When this code applied to delay line in incremental order then for code [4:0] code first branch of switch from first FDU get controlled, for [9:5] first branch of switch from second FDU get controlled and so on for remaining code. We have three individual switch groups with having an identical size of switch within the group. First and second group has 5 switch for each FDU and at top level that is 20, and third group has 6 switch for each FDU and at top level that is 24. So we have 64 possible discrete delay value. Which is a very large number and give us a very high delay resolution.
In sub-micron technology intrinsic delay of inverter is order of tens of micro second but major contributor delay is interconnect lumped RC parasitic. estimation of delay caused by parasitic RC component is very hard and less accurate. Here in proposed fine delay unit have used inverter based delay model without considering the parasitic. Reason of validity of proposed delay model is that it is taking a relative delay count instead of actually delay.so every time incremental delay will add up in parasitic generated delay plus intrinsic delay of inverter to give FDU a incremental delay. so fixed offset of parasitic RC generated delay will add up for each update.

$$\tau_{PHL} = \frac{C_{load} \cdot \Delta V_{HL}}{I_{avg,HL}} = \frac{C_{load} \cdot \left(V_{OH} - V_{50\%}\right)}{I_{avg,HL}}$$

$$\tau_{PLH} = \frac{C_{load} \cdot \Delta V_{LH}}{I_{avg,LH}} = \frac{C_{load} \cdot \left(V_{50\%} - V_{OL}\right)}{I_{avg,LH}}$$

$$I_{avg,HL} = \frac{1}{2}\left[i_C\left(V_{in} = V_{OH}, V_{out} = V_{OH}\right) + i_C\left(V_{in} = V_{OH}, V_{out} = V_{50\%}\right)\right]$$

$$I_{avg,LH} = \frac{1}{2}\left[i_C\left(V_{in} = V_{OL}, V_{out} = V_{50\%}\right) + i_C\left(V_{in} = V_{OL}, V_{out} = V_{OL}\right)\right]$$

$$\tau_P = \frac{\tau_{PHL} + \tau_{PLH}}{2}$$

Figure 3.10: Propagation delay model of cmos inverter

As time taken by signal to switch state from low to high and high to low are controlled by PMOS and NMOS separately. switch sizing for equal delay is a major concern hare. duty cycle of clock propagating through this delay elements are very sensitive to delay difference between high to low and low to high state switch. it will add a cycle to cycle jitter to clock and eventually reduce timing margin.so proper sizing of PMOS and NMOS for every switching condition is major design challenge.

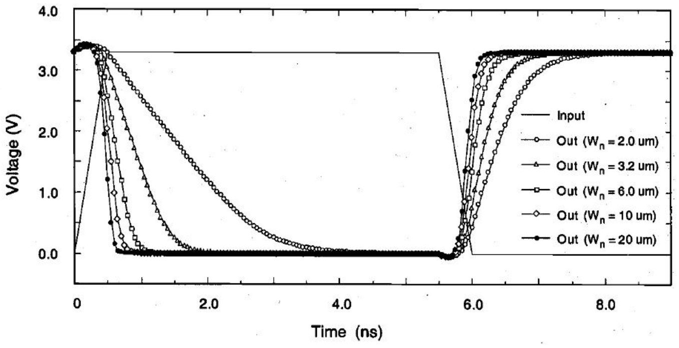


Figure 3.11: W/L vs Propagation delay[10]

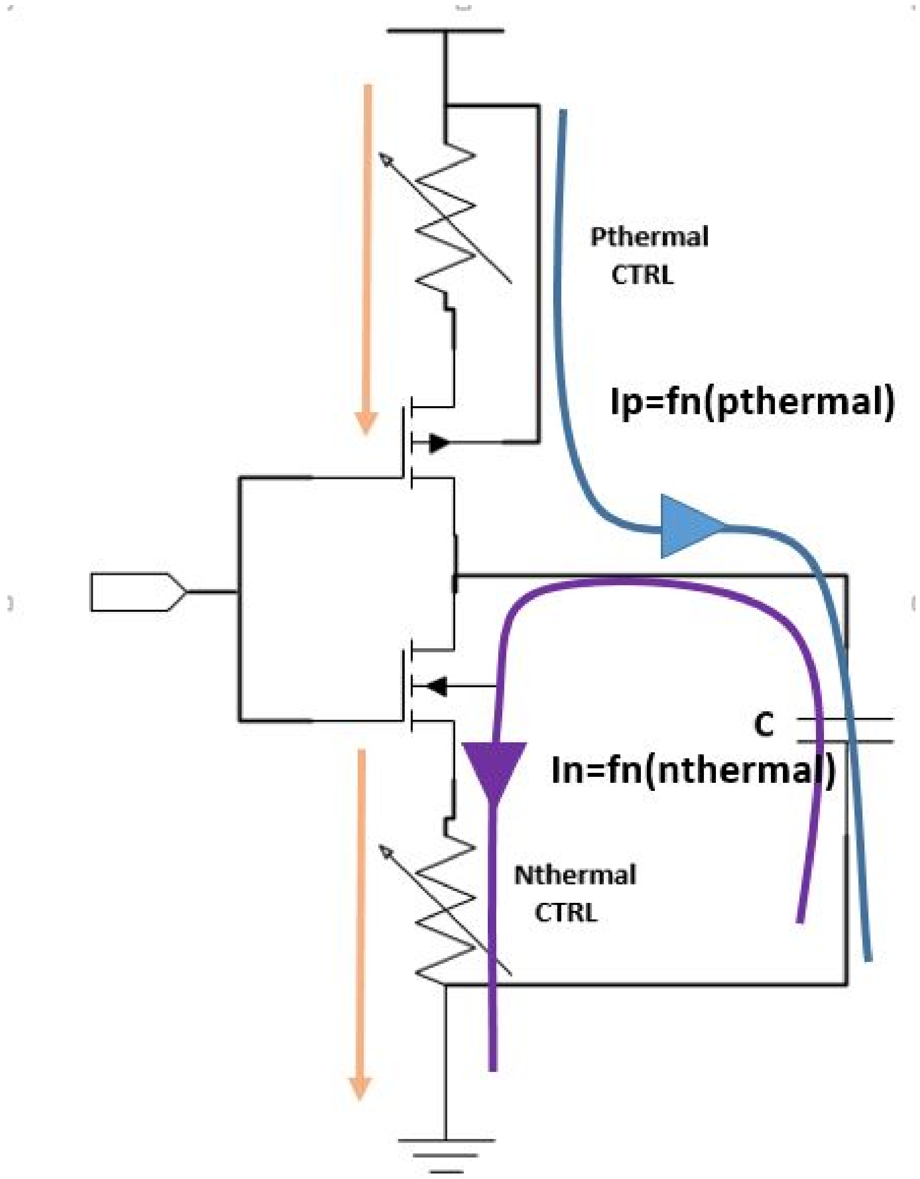


Figure 3.12: inverter delay model

3.1.4 Phase Detector

Fig show a three state bang-bang phase detector with lock state window. Up and down states are used for either increments or decrements of delay of delay line. When DLL get locked means both clk_ref and clk_out phases get completely synchronize then neither up nor down signal should affect the delay. But up and down store a previous state and act according it. Continuous up and down control for delay move phase of clk_out back and forth across clk_ref phase and cycle to cycle jitter of value equal to delay resolution of fine delay line getting added into a clk_out. So when phase of clk_out comes in close vicinity of clk_ref we should disable the up and down delay control. In this condition state of delay line is called a locked state and no delay update happen in this state. When DLL operate in locked state phase of clk_out remain constant and it changes with PVT variation. When PVT variation sweep clk_out phase enough that it goes out of locked window then low locked state signal asserted and DLL operate in normal condition to trap clk_out phase in locked window again.

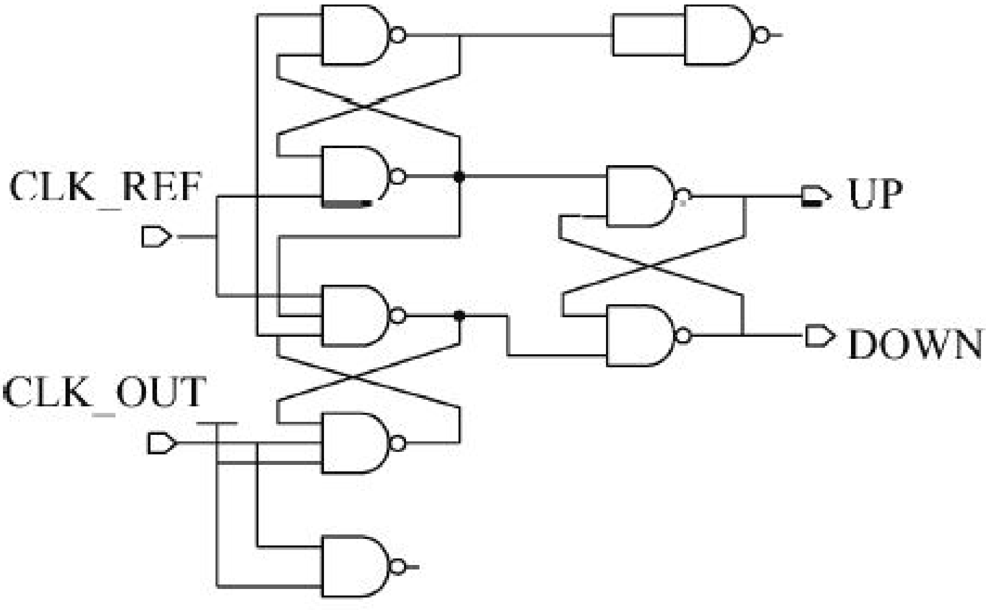


Figure 3.13: block diagram of two state bang-bang phase detector

When phase of CLK_REF and CLK_OUT comes in close vicinity than output of lower side latch may go in metastability state. which make value of reset pin of output flop uncertain. this uncertainty cause a toggling of output state UP and DOWN and operation of phase detector becomes unstable.To avoid metastability for any combination of clock phase proper NAND gate design is required to give optimum delay for signal.upper most flip flop used to store a previous state from down side latches. when phase of CLK_REF and CLK_OUT comes in close vicinity to each other. momentary glitch occur at output of middle latch.If this glitch duration will be more than setup time of upper most latch than it will store it as a previous state value and reflects it as a valid output. this create a momentary transition on UP and DOWN state that can alter operation of DLL. In some case momentary glitch put DLL in never lock condition or long time to lock.

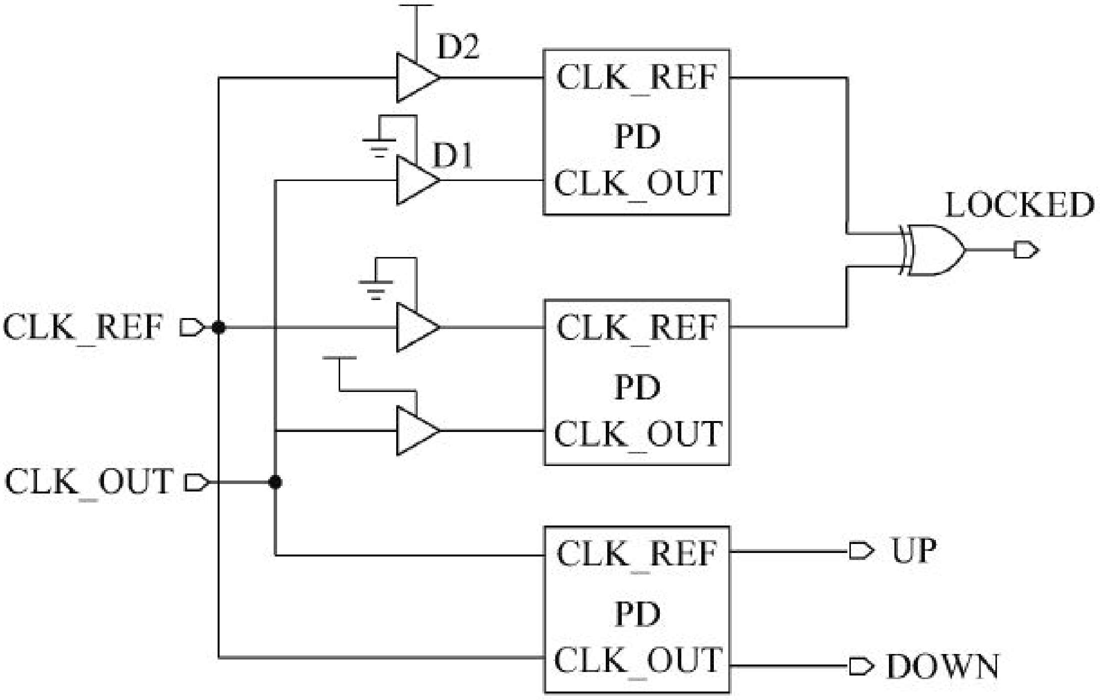


Figure 3.14: block diagram of three state bang-bang phase detector

Locked state window width is equal to 2t, where t is a delay difference between delay elements D1 and D2. Locked window width is depend on delay resolution of delay line, locked window width must be higher than delay resolution of DLL so it can accommodate minimum one step sweep variation in locked range. For delay locked window width less than delay resolution of DLL, DLL never goes in locked state. Phase detector shown in fig is two state bang-bang phase detector based on RS latch. It change UP/DOWN signal by detecting phase difference between clk_ref and clk_out. Second figure shows a three state bang-bang phase detector. First two state phase detector compare the phase of clk_ref and clk_out and second two state phase detector compare delay version clk_ref and clk_out to generate a lock_state signal. As shown in figure width of locked window is decided by delay difference between D1 and D2.main challenge here is two maintain a constant width of lock window across the process, voltage and temperature variation. Because with PVT variation delay value of D1 and D2 varies significantly changing t and hence locked window width. Here we have designed delay elements which delay value is relatively constant across PVT variation. Various problem coming from high frequency operation of above configuration is discussed in subsequent section.

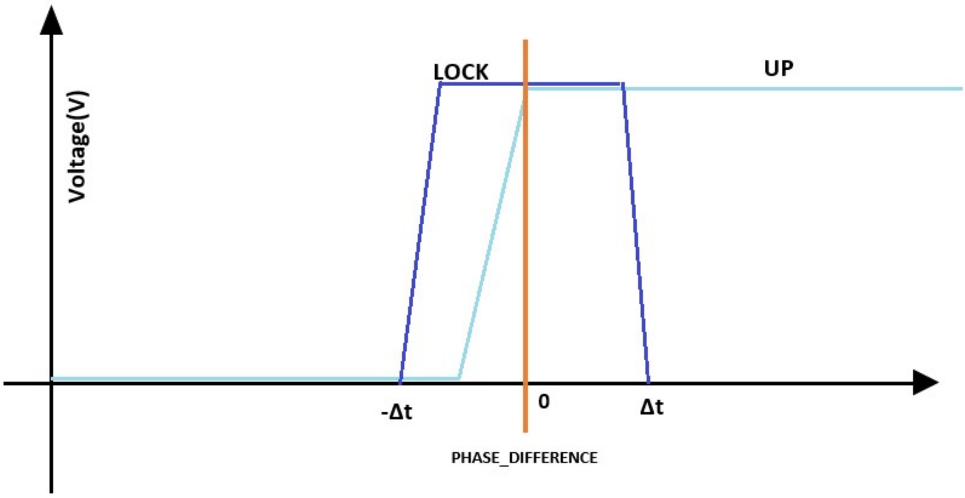


Figure 3.15: locked window of phase detector

Because of separate delay given to both two state phase detector for lock state generation sometimes we get a glitch on lock signal because of up and down signal are not synchronize. Which can false locking of DLL at half clock period to avoid this condition we required a rising edge synchronize.

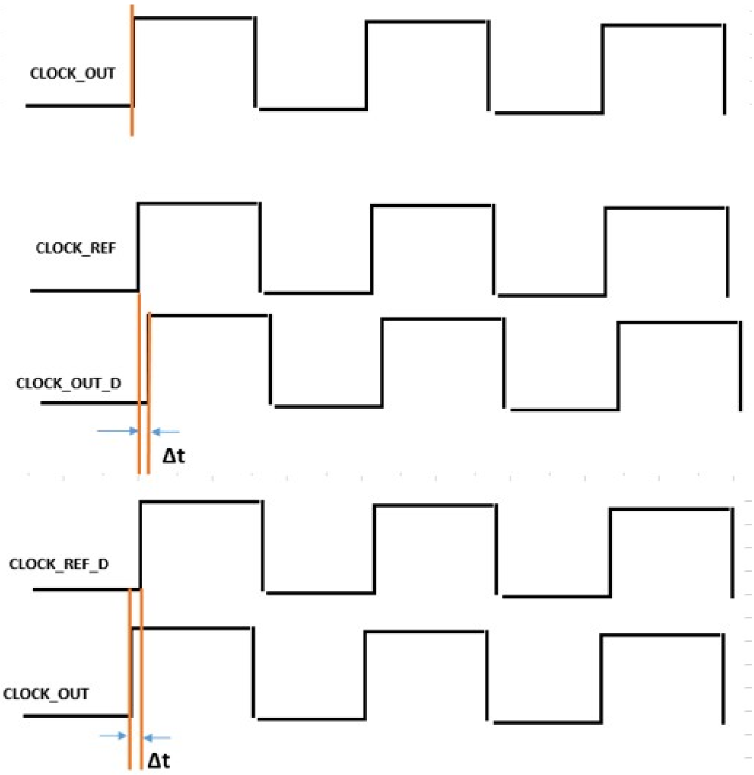


Figure 3.16: pre lock signal of phase detector

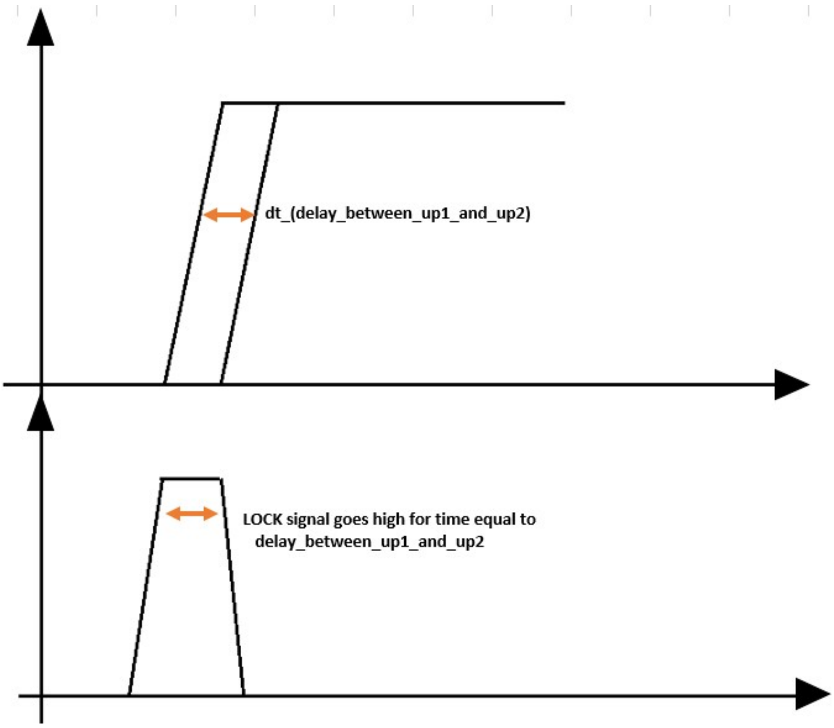

dt_(delay_between_up1_and_up2)
LOCK signal goes high for time equal to
delay_between_up1_and_up2

Figure 3.17: Glitch on LOCK signal

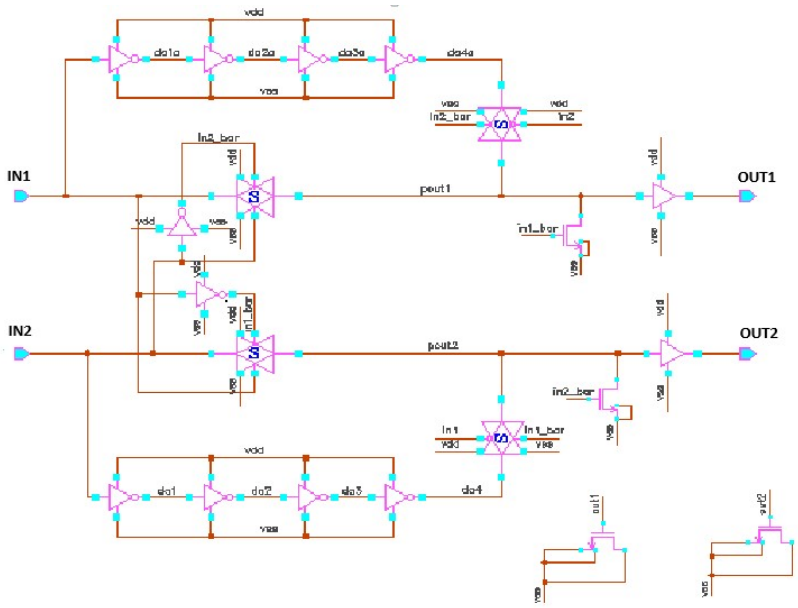


Figure 3.18: Rising edge synchronizer to avoid glitches on LOCK signal

3.1.5 Digital control logic (FSM)

As we have discussed as earlier that closed loop bandwidth of DLL is importance performance parameter. From flow chart we can say that this is a best possible method to increase a closed loop bandwidth. On reset FSM first check for LOCK signal if LOCK=1 then it preserve the state of coarse delay line and fine delay line code value and hence delay value. If LOCK=0 then it look for half lock signal state. Half lock signal state goes high when two clocks are out of phase in range of half lock window. Half lock window width is equal to lock window width. When it found high half lock signal then it delay clock by 180 degree to make both clock in phase this action get complete in one calibration cycle. So clock loop bandwidth increased significantly in this particular case. After accessing half lock state if it found half lock state low then it move to initialize the coarse delay unit. Coarse delay unit get initialize at mid code value so adding an inherent delay equal to half of its delay range. If up is asserted high then coarse code is increment by 1 otherwise it is decrement by 1. Coarse code update accrued only if coarse state is low otherwise it move to fine code update bypassing coarse code update. Coarse state is suggesting that initial adaptation occurred and DLL got locked state once. But due to PVT variation phase get changed frequently and DLL moved away from locked state. By in scenario the relative phase difference is very small and it can be handle by only fine delay unit. We use fine delay unit for phase synchronization after a one time locked because due to small step size we can get a higher delay resolution. Initial coarse code update provide us a fast adaptation and closed loop bandwidth.

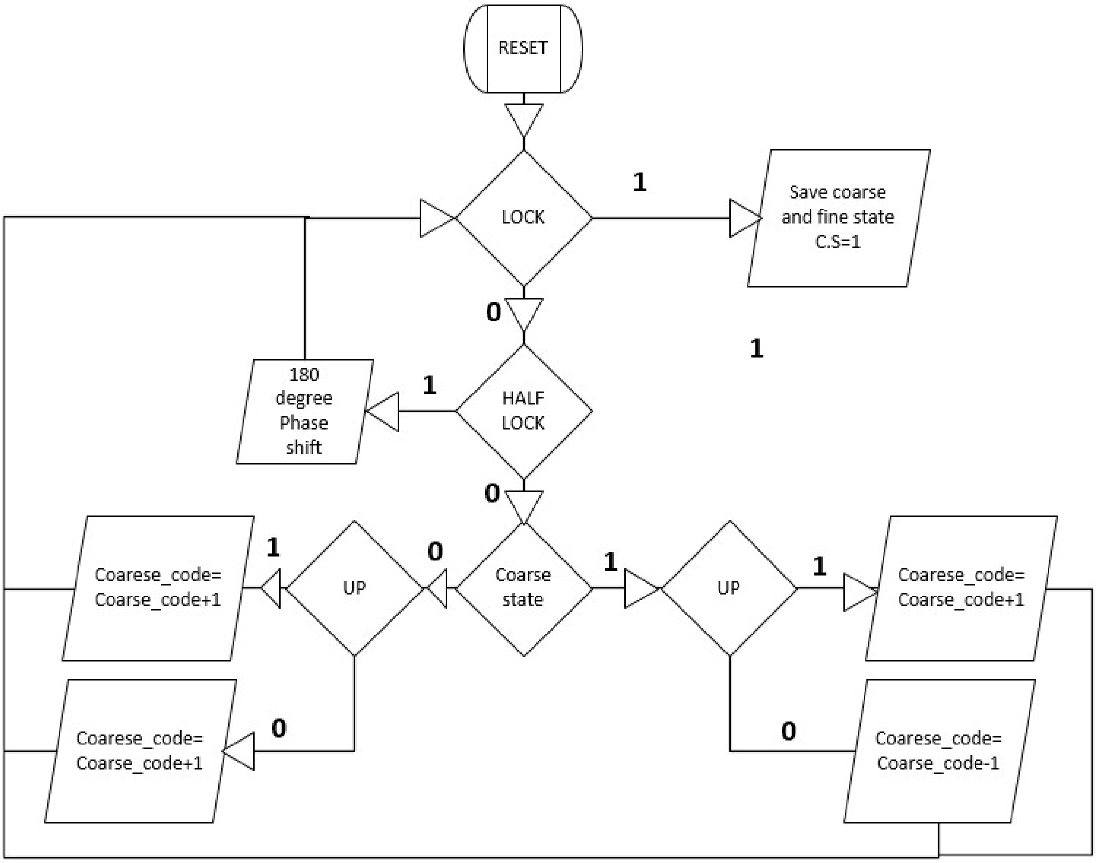

RESET
LOCK
1
Save coarse and fine state C.S=1
0
1
180 degree Phase shift
1
HALF LOCK
0
Coarese_code= Coarse_code+1
1
UP
0
Coarse state
1
UP
1
Coarese_code= Coarse_code+1
Coarese_code= Coarse_code+1
0
0
Coarese_code= Coarse_code-1

Figure 3.19: Digital control flow chart of DLL

Chapter 4

Results and Observations

simulation results for proposed all digital delay locked loop are presented here. All simulation results are derived from pre-layout netlist.operating frequency range of DLL is 9.6 GHz to 11.2 GHz.closed loop bandwidth of All digital delay locked loop is 1.4GHz.

4.1 Delay line

delay line composed of two types of delay elements.

- Coarse Delay Line
- Fine Delay Line

Important parameters needs to consider here are

- delay range
- delay resolution
- linearity

4.1.1 Coarse delay line

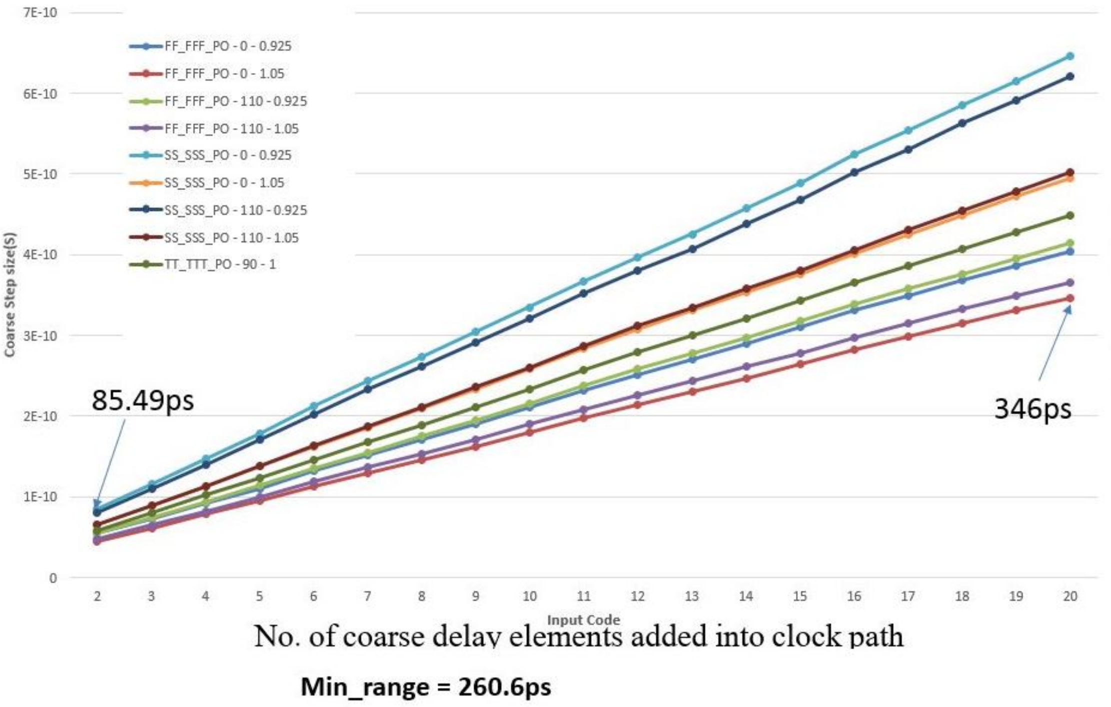


Figure 4.1: Coarse delay line range across PVT

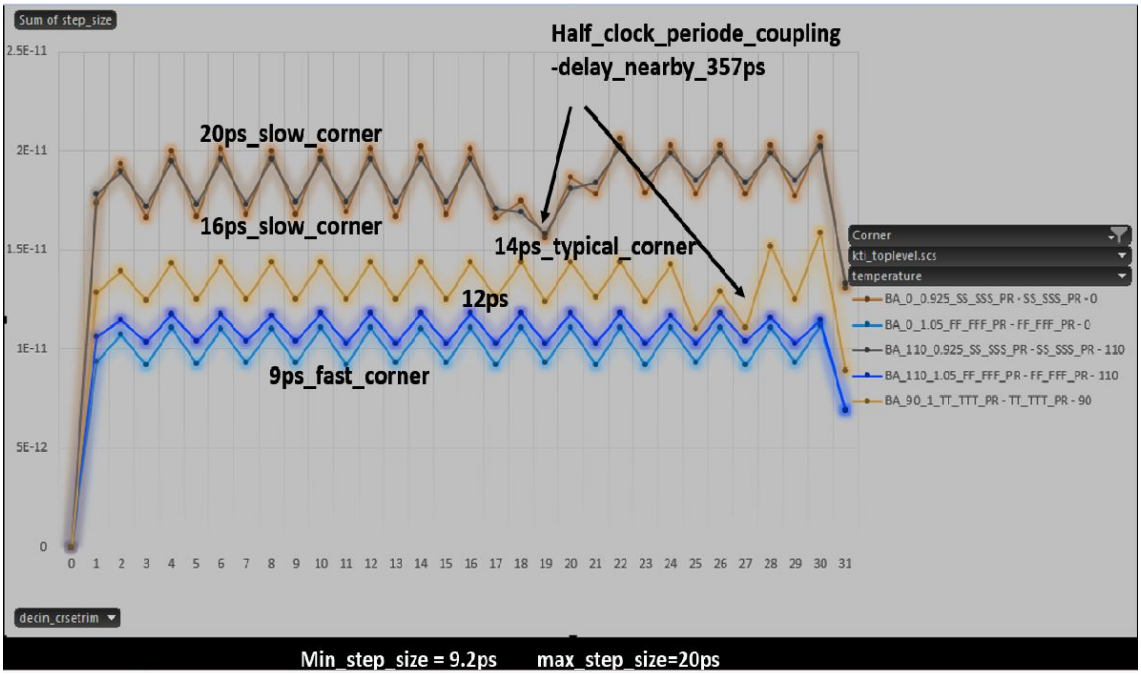


Figure 4.2: Coarse delay step size

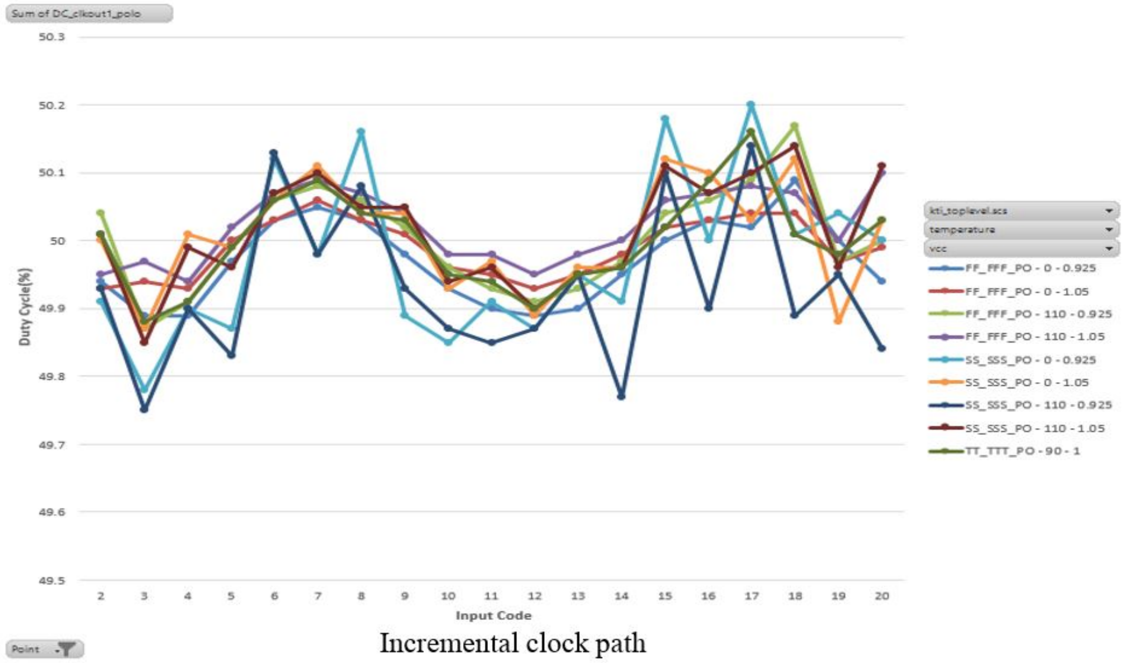


Figure 4.3: Coarse delay line output duty cycle

Matching rising edge and falling edge delay ensure less duty cycle degradation.

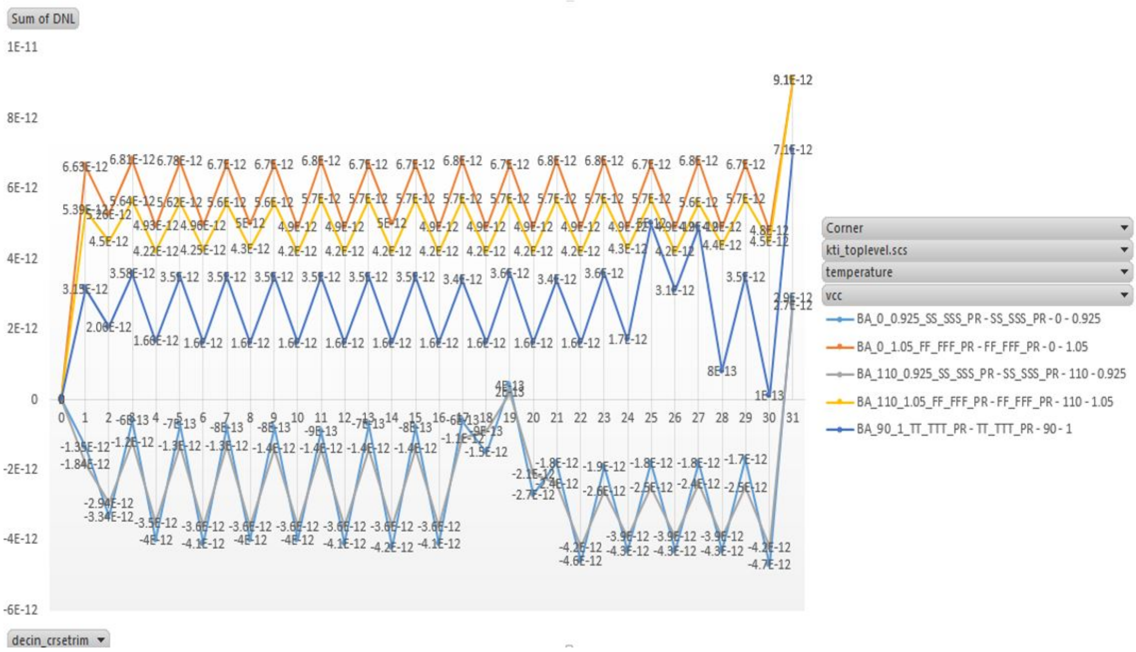
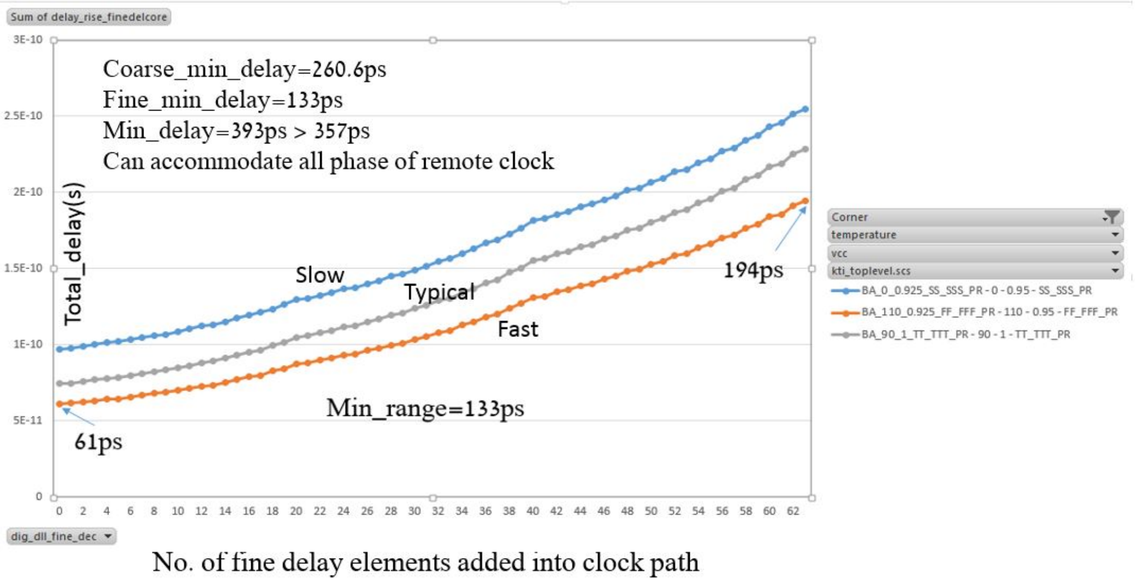


Figure 4.4: Differential non-linearity of coarse delay line

Differencial non-linearity is a measure for linearity of delay transfer characteristics. As we can see here maximum DNL observed here is 6.2ps in fast process corner and minimum value is 1.6ps in typical corner. so relative small DNL proves linearity of delay transfer characteristics.

4.1.2 Fine delay line



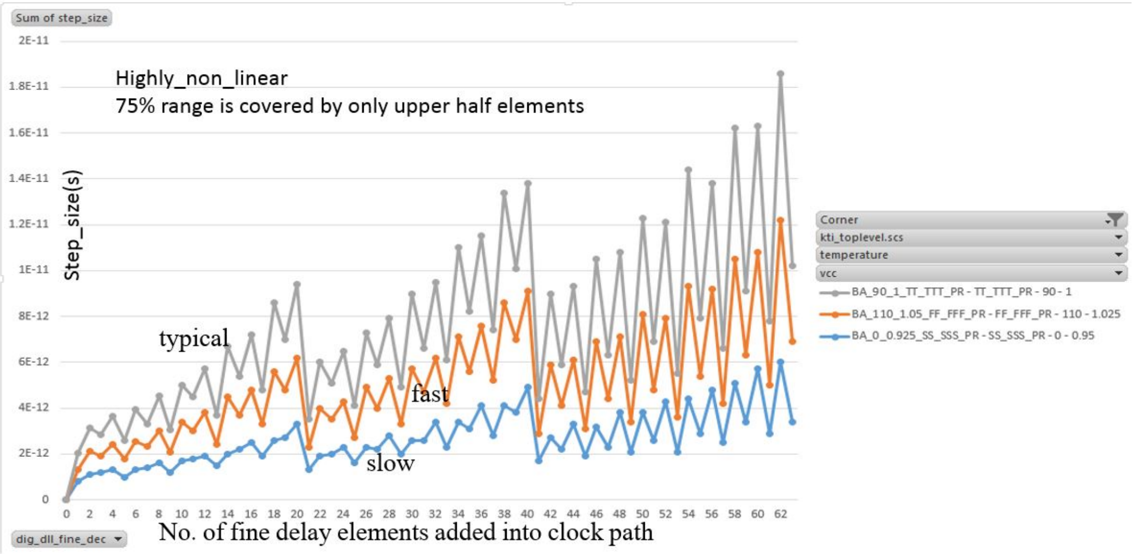
Coarse_min_delay=260.6ps
Fine_min_delay=133ps
Min_delay=393ps > 357ps
Can accommodate all phase of remote clock

Min_range=133ps

Min_range = 133ps max_range=164ps

Figure 4.5: Fine delay line range across PVT

Figure 4.5 shows a delay range of fine delay unit as we can see it is relatively non-linear compare to coarse delay line especially for lower code value. By combining delay of both coarse and fine delay unit it can accommodate any phase variation.



Highly_non_linear
75% range is covered by only upper half elements

Min_step_size = 0.65ps max_step_size=18.8ps

Figure 4.6: Fine delay line step size across PVT

step size is higher in higher code range compare to lower code range.

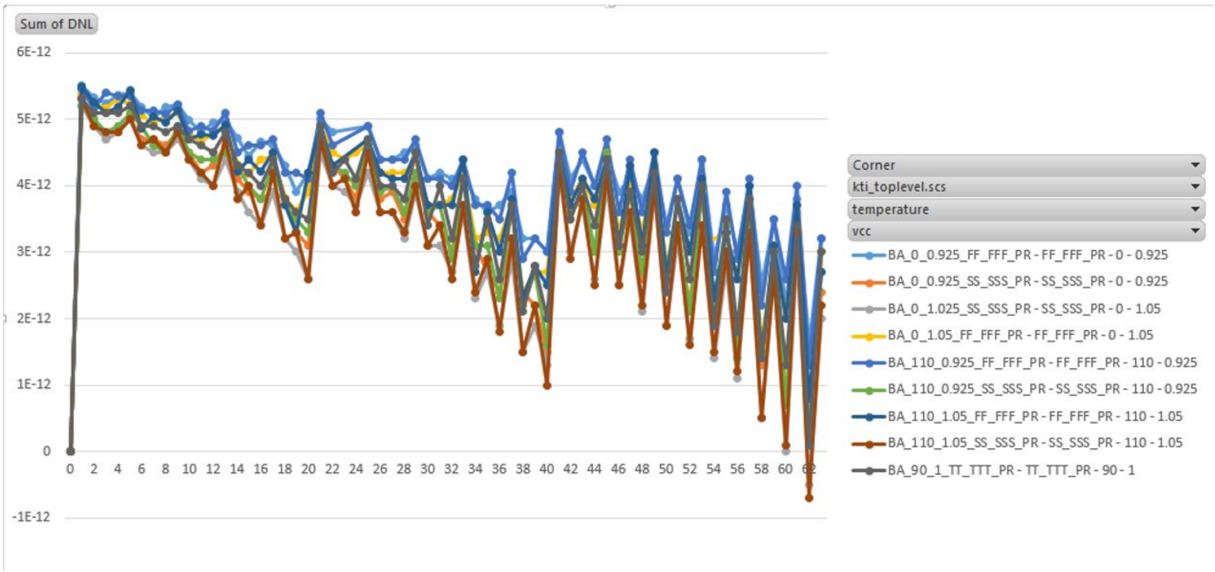


Figure 4.7: Differential non-linearity of fine delay line

As we can see from figure for small delay renge DNL is very good for FDU. But as we move for higher delay value DLL become worst. so it is mendatory to operate DLL at low delay value of FDU.

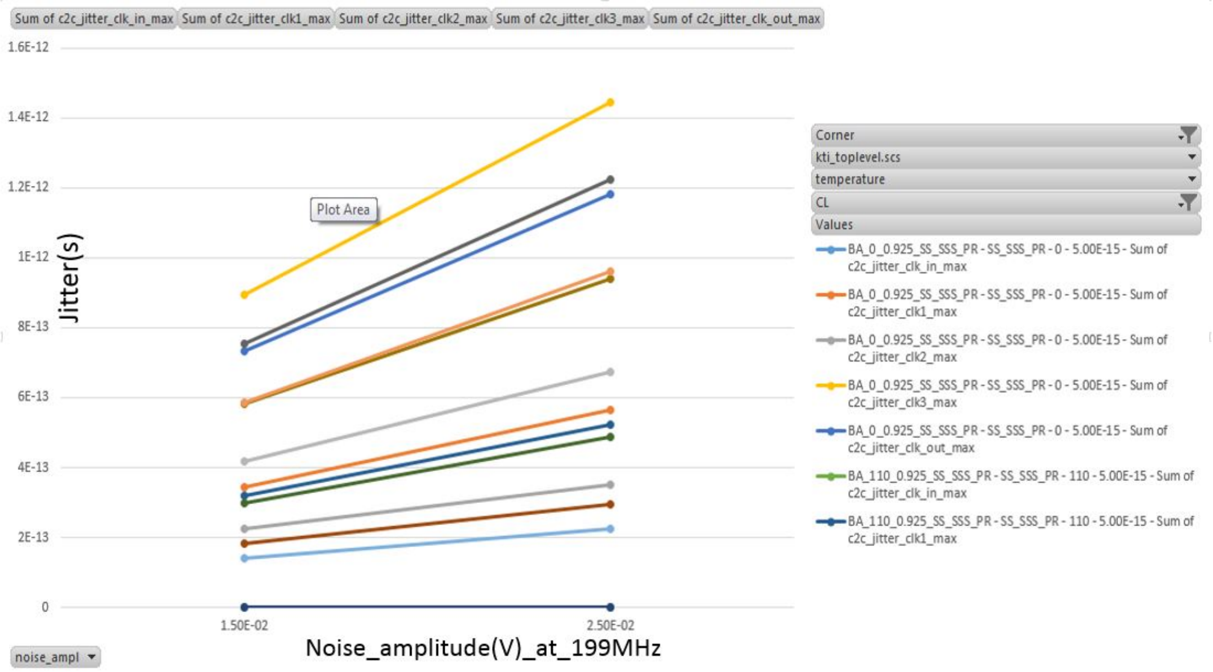
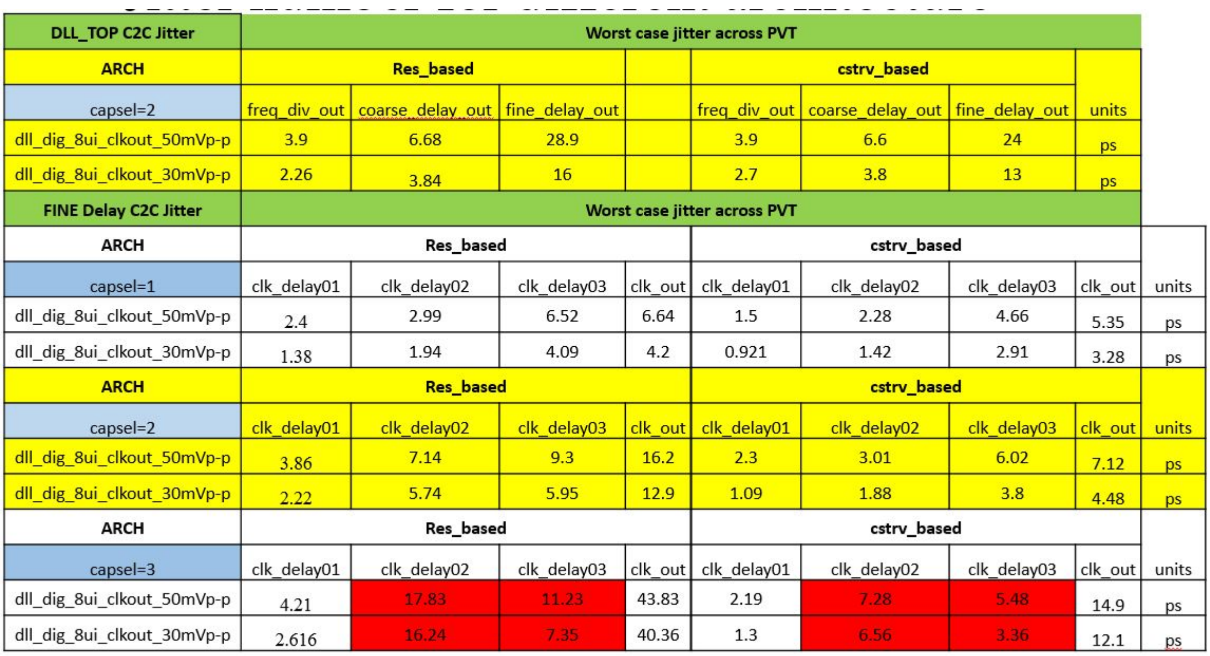


Figure 4.8: Jitter contribution by individual fine delay elements

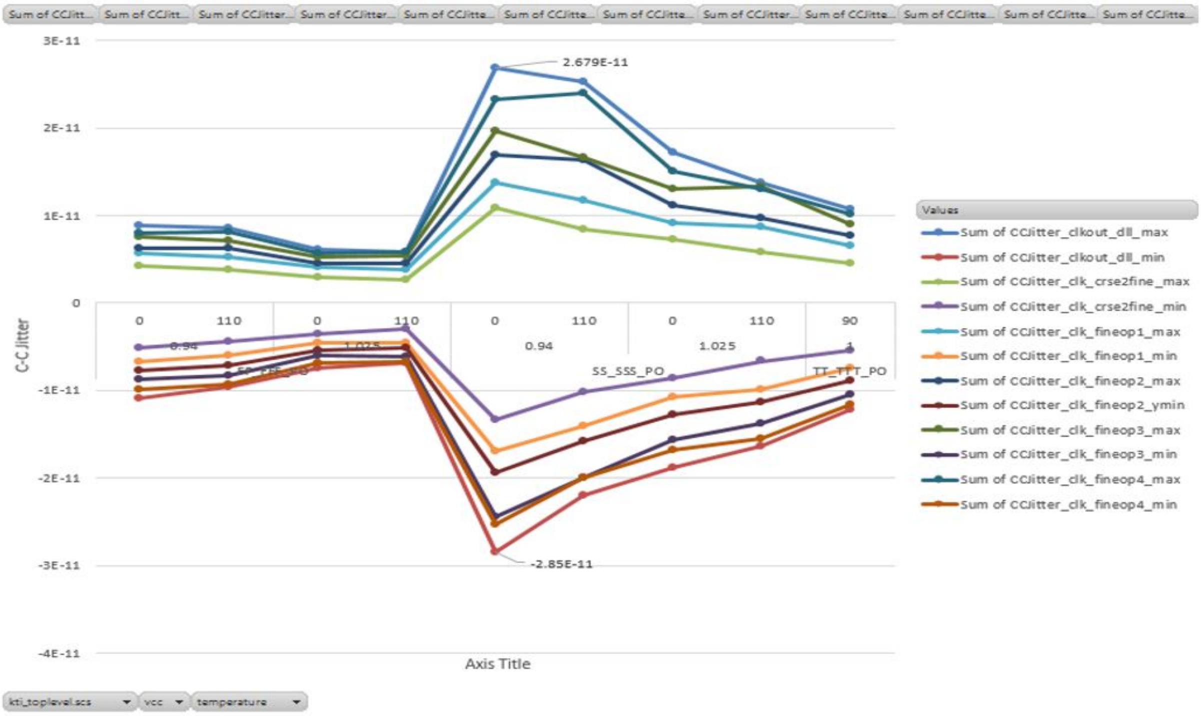
Jitter measured at each stage of FDU at 199MHz and noise level of 15mv and 25mv. Nose amplitude is not critical for each individual stage in terms of jitter. But combining all stage jitter difference for different noise amplitude is significant.

4.2 Jitter



DLL_TOP C2C Jitter	Worst case jitter across PVT								
ARCH	Res_based				cstrv_based				
capsel=2	freq_div_out	coarse_delay_out	fine_delay_out		freq_div_out	coarse_delay_out	fine_delay_out	units	
dll_dig_8ui_clkout_50mVp-p	3.9	6.68	28.9		3.9	6.6	24	ps	
dll_dig_8ui_clkout_30mVp-p	2.26	3.84	16		2.7	3.8	13	ps	
FINE Delay C2C Jitter	Worst case jitter across PVT								
ARCH	Res_based				cstrv_based				
capsel=1	clk_delay01	clk_delay02	clk_delay03	clk_out	clk_delay01	clk_delay02	clk_delay03	clk_out	units
dll_dig_8ui_clkout_50mVp-p	2.4	2.99	6.52	6.64	1.5	2.28	4.66	5.35	ps
dll_dig_8ui_clkout_30mVp-p	1.38	1.94	4.09	4.2	0.921	1.42	2.91	3.28	ps
ARCH	Res_based				cstrv_based				
capsel=2	clk_delay01	clk_delay02	clk_delay03	clk_out	clk_delay01	clk_delay02	clk_delay03	clk_out	units
dll_dig_8ui_clkout_50mVp-p	3.86	7.14	9.3	16.2	2.3	3.01	6.02	7.12	ps
dll_dig_8ui_clkout_30mVp-p	2.22	5.74	5.95	12.9	1.09	1.88	3.8	4.48	ps
ARCH	Res_based				cstrv_based				
capsel=3	clk_delay01	clk_delay02	clk_delay03	clk_out	clk_delay01	clk_delay02	clk_delay03	clk_out	units
dll_dig_8ui_clkout_50mVp-p	4.21	17.83	11.23	43.83	2.19	7.28	5.48	14.9	ps
dll_dig_8ui_clkout_30mVp-p	2.616	16.24	7.35	40.36	1.3	6.56	3.36	12.1	ps

Figure 4.9: Jitter number for different architecture



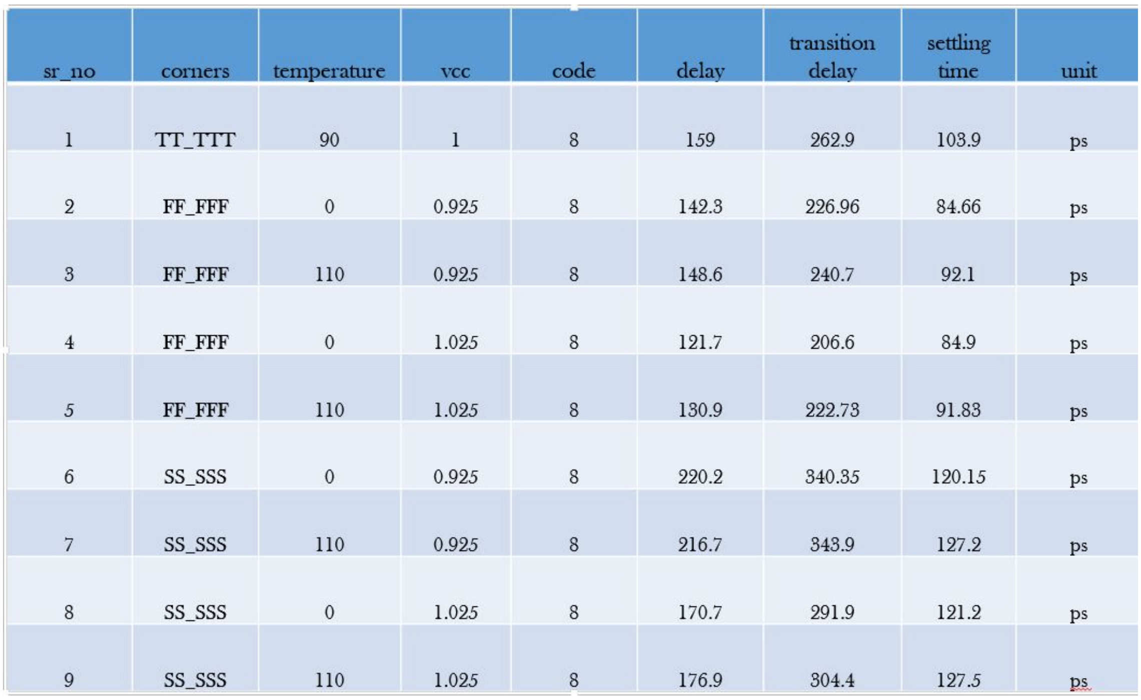
Noise on PS is 50mV p-p, 199MHz, Coarse code-5, Fine code-Max value

Figure 4.10: DLL TOP stage wise C-C Jitter across PVT

4.3 Settling time

Setup condition to measure settling time.

- dig_dll_fine_dec - 32
- Clk_freq - 1.4G
- decin_crsetrim - transition from 7 to 8
- Settling time is measured as a delay difference
- Delay at decin_crsetrim=8 and delay at decin_crsetrim transition from 7 to 8.



sr_no	corners	temperature	vcc	code	delay	transition delay	settling time	unit
1	TT_TTT	90	1	8	159	262.9	103.9	ps
2	FF_FFF	0	0.925	8	142.3	226.96	84.66	ps
3	FF_FFF	110	0.925	8	148.6	240.7	92.1	ps
4	FF_FFF	0	1.025	8	121.7	206.6	84.9	ps
5	FF_FFF	110	1.025	8	130.9	222.73	91.83	ps
6	SS_SSS	0	0.925	8	220.2	340.35	120.15	ps
7	SS_SSS	110	0.925	8	216.7	343.9	127.2	ps
8	SS_SSS	0	1.025	8	170.7	291.9	121.2	ps
9	SS_SSS	110	1.025	8	176.9	304.4	127.5	ps

Figure 4.11: Settling time summary

settling time define here as time taken by delay elements to settle at valid logic level from time of switching it state. i.e delay line operating with 5 delay elements in clock path and if we insert an extra delay elements in clock path then definitely that will add up in delay value but that delay value will be much higher than intrinsic delay value. time difference between total increments in clock path delay to intrinsic delay of delay elements we can define it as an settling time in case of closed loop control delay line. It will add a switching jitter in clock path. some time high settling time move phase of delay clock back and forth across the reference clock and DLL never goes locked. this create a cycle to cycle jitter in clock path.

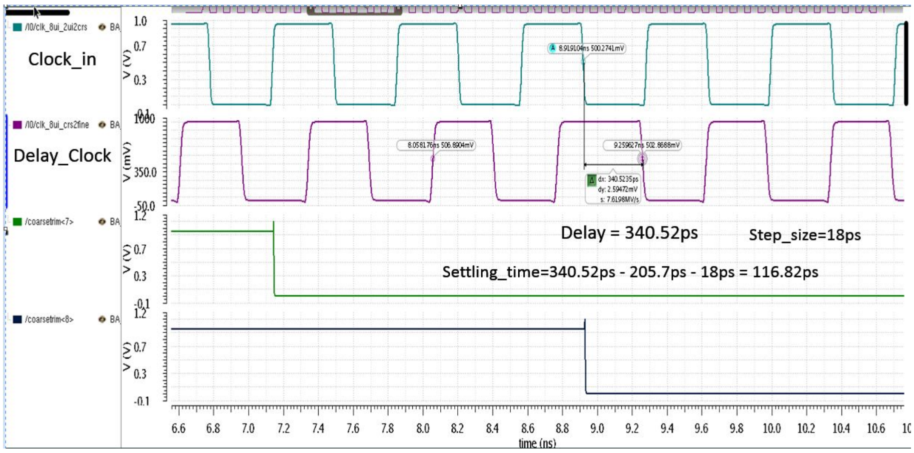


Figure 4.12: Delay decin crsetrim transition 7to8

Figure shows us how duty cycle distortion occurring while switching delay line configuration. so clock must not be used when coarse adaptation happening to sample the data. once lock acquire, only fine delay control the delay with very small settling time. If settling time exceed the clock period false locking can occur in DLL.

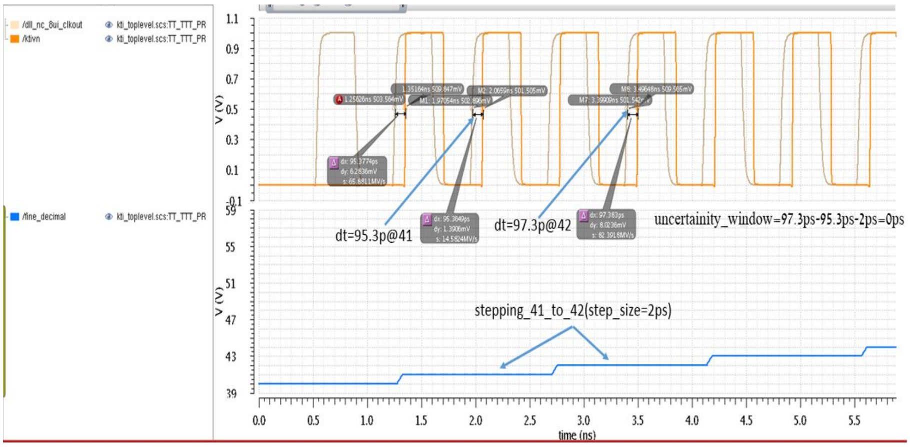


Figure 4.13: Settling time fine delay line

Delay increasing from 95.36ps to 97.38 while inserting one extra delay elements in fine delay line. Delay step size for that range of code is 2ps.No settling time issue in FDU so it can trace PVT variation induced phase shift at every clock cycle. basic delay elements of FDU is current starved delay elements connecting in parallel. when switch from lower delay to higher delay value, basically we are disabling the one leg of FDU. when disabling of FDU leg and rising edge of input clock occur simultaneously then only settling time comes into picture.

4.4 Phase detector

Phase of delayed clock is swipe across phase of reference clock to measure locked window width.
1 step on x-axis = 2ps.

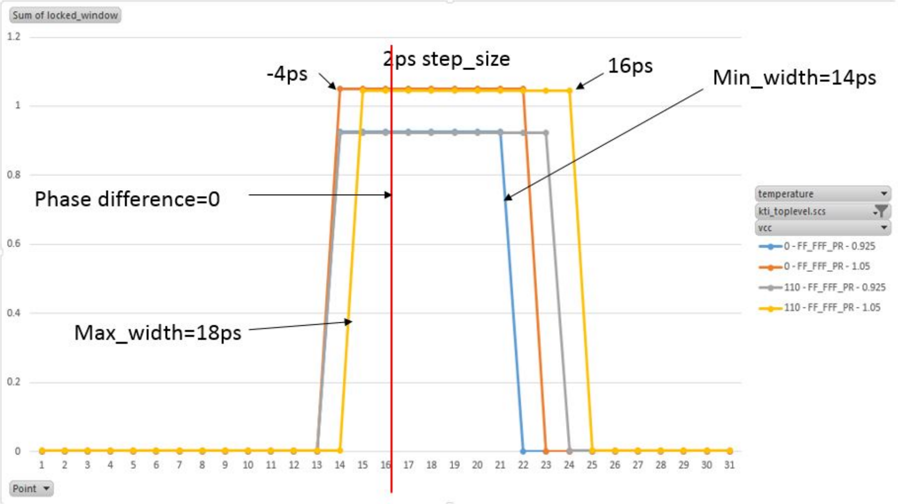


Figure 4.14: LOCK window across fast corner

Figure shows relative clock difference between delayed clock and reference clock for which DLL remain in locked state. In locked state DLL freezes both FDU and CDU, clock get constant delay, no switching, less duty cycle distortion. Maximum width occurring in slow corner of 20ps. Relative phase difference of 20ps will not be corrected by DLL in this specific corner and DLL remain in locked state. In fast corner minimum value of locked window is 14ps. Maximum locked window difference is 6ps which is hardly three time than FDU resolution in middle code range.

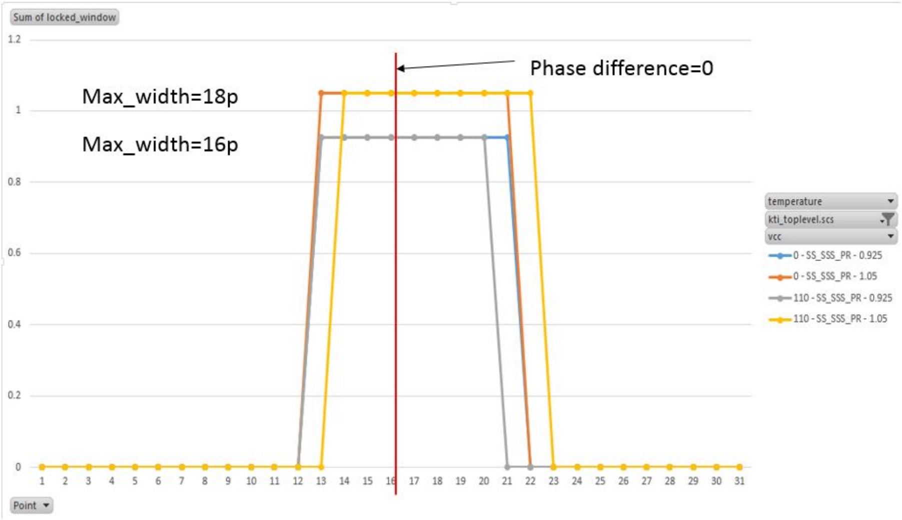


Figure 4.15: LOCK window across slow corner

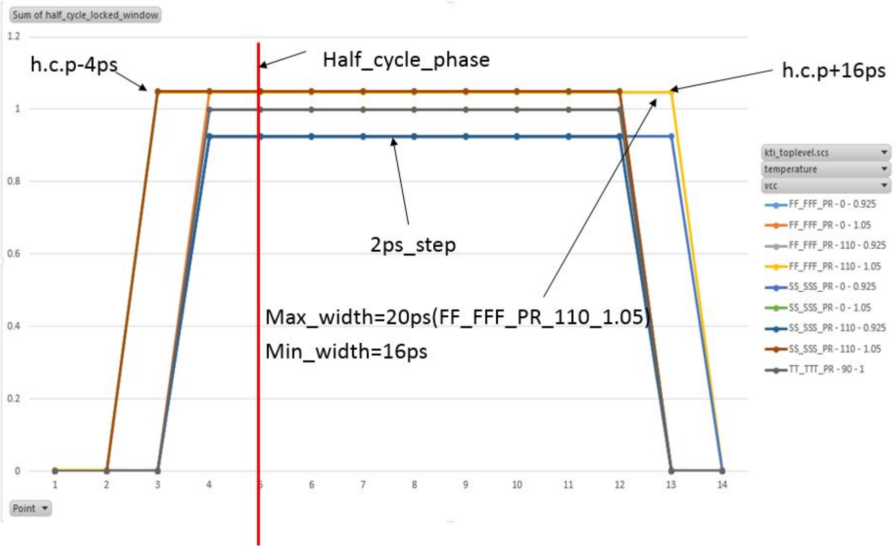


Figure 4.16: Half cycle locked window across pvt

Half cycle locked window suggest that both clocks are out of phase by 180 degree. The benefit of detecting clocks for 180 degree phase shift is that we can easily delay clock by 1 UI by sampling the 2UI clock with 1UI clock. This operation occur in only one clock cycle improving the lock time of DLL. To compensate the delay of 180 degree which is the maximum possible delay that can occur in delay line by CDU and FDU it may take 64 to 96 clock cycle. Half cycle locked window is not symmetric around half cycle phase i.e 180 degree phase. It cover more range of 16ps while delayed clock lead reference clock and cover less range of 4ps while delayed clock lag reference clock. Width variation of half cycle locked window across PVT is 4ps. ideally half cycle locked window width should be one forth of clock period. If this happen then actual delay tracking range reduced to half.

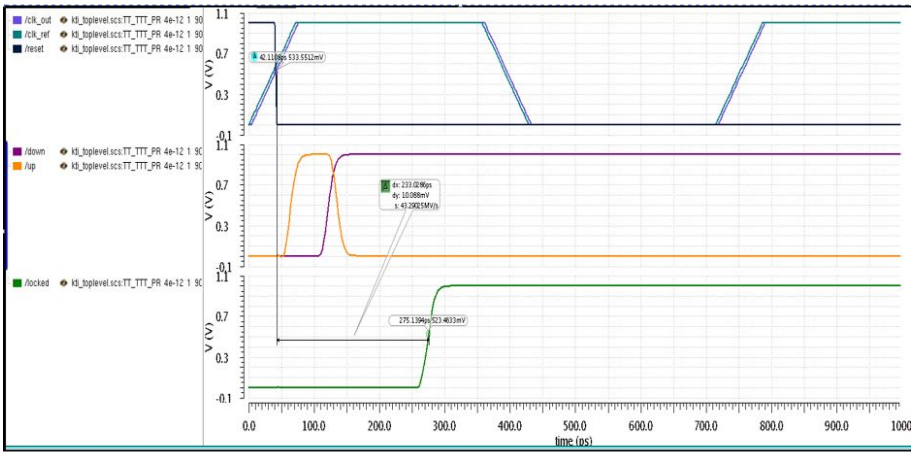

/clk_out kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
/clk_ref kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
/reset kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
/down kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
/up kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
/locked kti_toplevel.scs:TT_TTT_PR 4e-12 1 90
V (V)
1.1
0.7
0.3
-0.1
42.1108ps 533.5512mV
dx: 233.0286ps
dy: 10.088mV
s: 43.29025MV/s
275.1394ps 523.4633mV
0.0
100.0
200.0
300.0
400.0
500.0
600.0
700.0
800.0
900.0
1000
time (ps)

Figure 4.17: Clock to locked delay typical

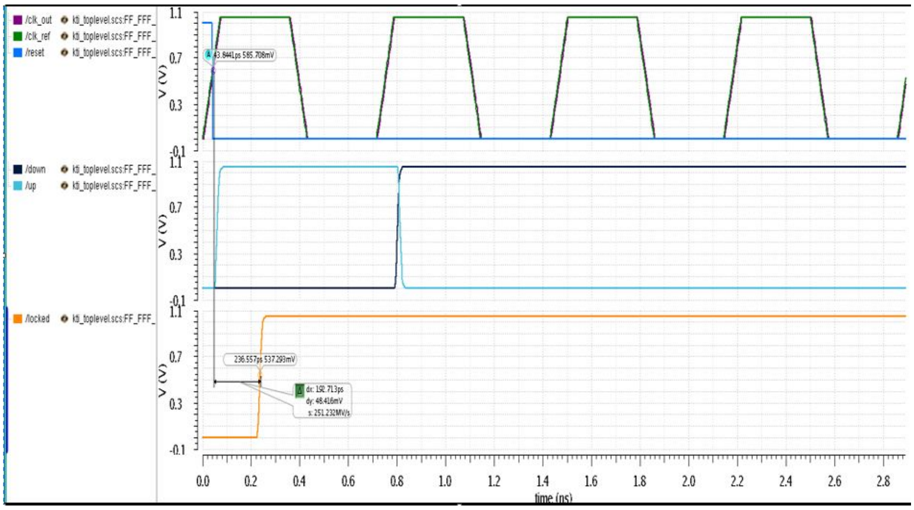


Figure 4.18: Clock to locked delay fast

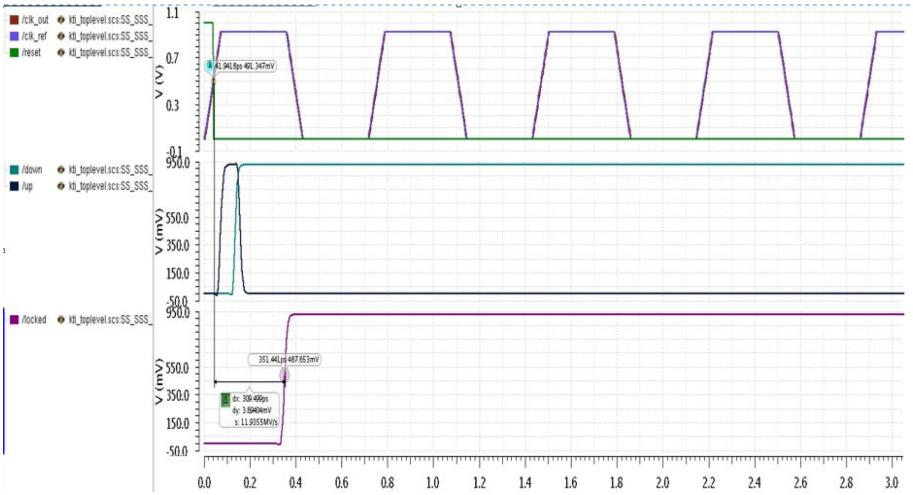


Figure 4.19: Clock to locked delay slow

figure-4.20 shows a clock to lock delay. this is a delay between actual phase comparison happening at phase detector to locked signal goes high. dominantly it is a phase comparator intrinsic delay. To generate locked window and half cycle locked window we have added two discrete delay on complimentary phase skew. it will access close vicinity of reference phase to identify its locked state. As we have seen that this processes add maximum delay of 20ps. When phase of both CLK_REF and CLK_DELAYED into setup uncertainty phase detector flops tends to move in metastability state. To avoid metastability flops gates are designed with optimum delay of 18ps to each. Which eventually add up in total clock to locked delay. When CLK_REF and CLK_DELAY becomes out of phase by 180 degree along with half lock detect signal, locked detect signal also get glitch to overcome this glitch added circuitry add more 48ps delay to clock to locked delay timing budget. Maximum clock to locked delay is 309ps in slow corner.Which is far less than 714ps clock period so while operating DLL in closed loop configuration we can update delay value at end of each clock cycle.

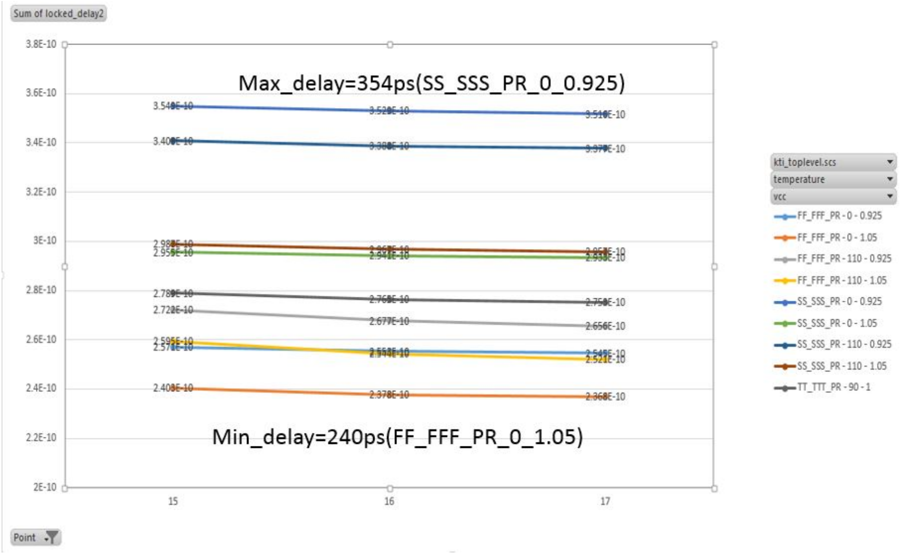


Figure 4.20: Clock to locked delay

clock to lock delay across PVT variation. it is varying from max value 354ps in slow corner to minimum value of 240ps in fast corner. As clock period is 714ps this number are under the range and we can update code at every cycle.

4.5 Closed loop calibration

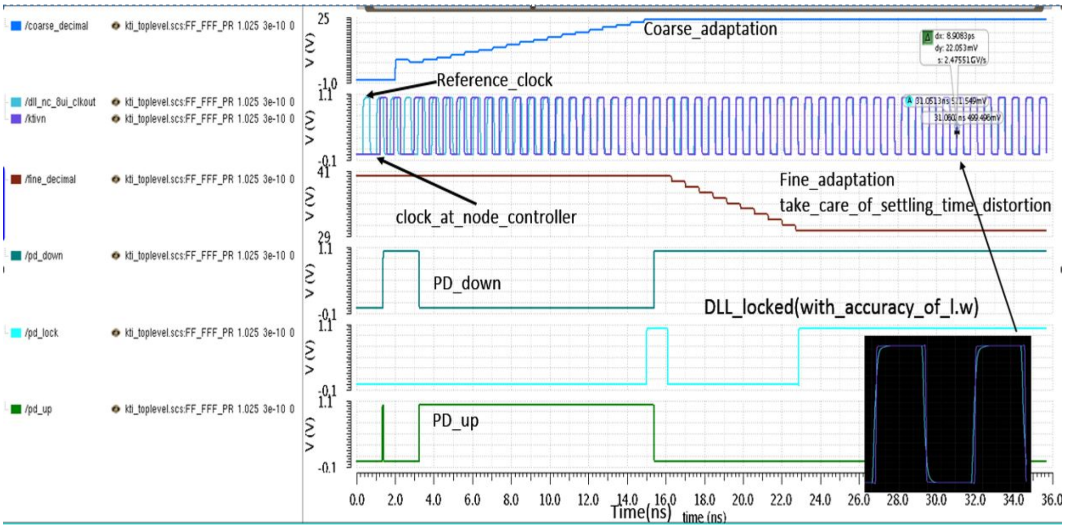


Figure 4.21: Closed loop calibration

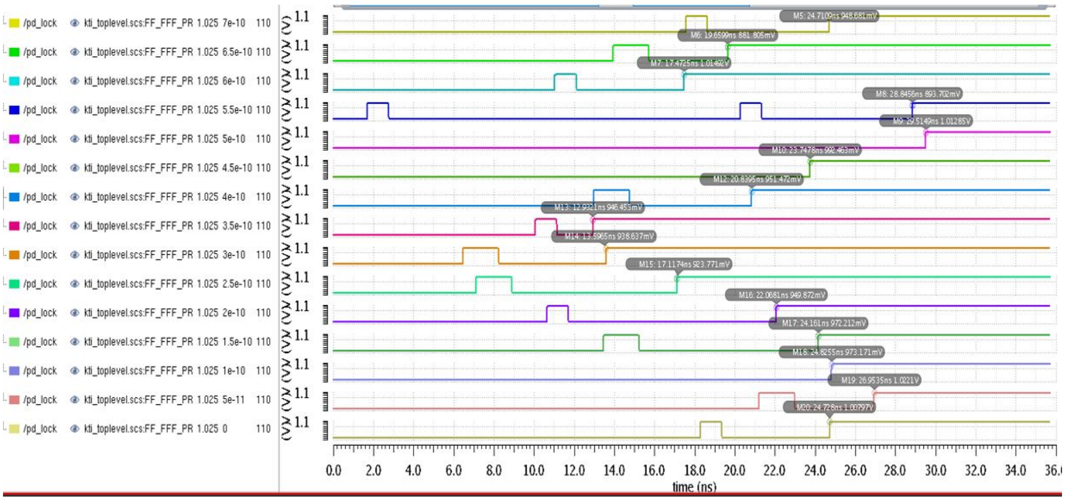


Figure 4.22: Lock time for different phase skew

figure shows a time taken by DLL to lock phase of clock with that of reference clock for different initial relative phase difference for fast process corner.in-term of lock time fast process corner is worst because of its relative small delay step size it take more time to get locked.

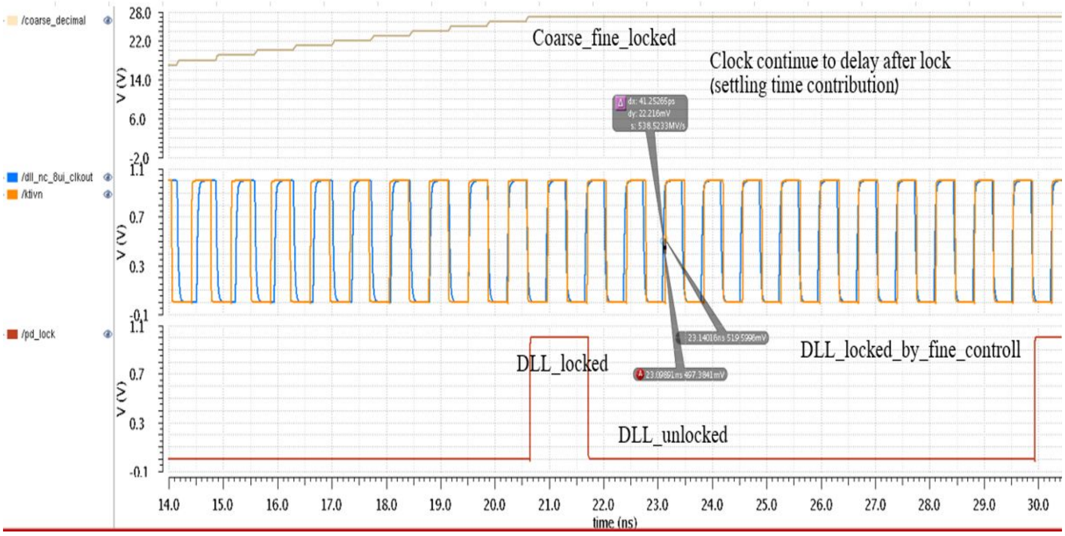


Figure 4.23: Settling time compensation by fine tuning

coarse code is updating at every clock cycle(4UI). clock delay elements has settling time which is more than a locked window of dll. after getting locked DLL frequently move to unlocked state after coarse calibration. phase usually deviate from sync with reference clock phase. fine tuning of phase is again carried out by fine delay elements.due to small settling time of fine delay elements every clock cycle update of fine code can not unlock DLL.

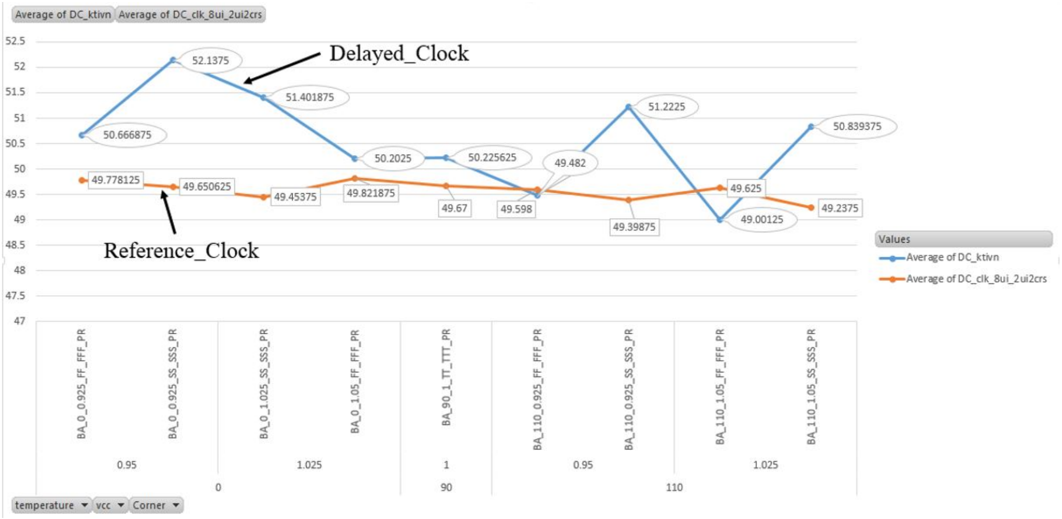


Figure 4.24: Duty cycle variation across PVT

Chapter 5

Conclusion

DLL operating on clocks with frequency range of 1.4GHz. Closed loop operating frequency of DLL is 1.4GHz. As we have seen in chapter 4 that because of coarse delay proposed DLL getting large delay range very less time to locked which provide a capability of fast tracking of PVT variation to DLL. Also with wide delay regulation range it can track any worst case PVT variation. Fine delay unit is responsible for delay resolution because of its finer delay transfer characteristics it can produce large number of phase. once the relative phase difference comes under coarse delay step size, coarse delay unit get shutdown, other wise clock phase will move back and forth to the reference clock. Locked signal generated from phase detector generate stable phase. Without DLL locked delayed clock get extra 4ps cycle to cycle jitter which is comparatively high. fine delay unit is current starved based delay elements, power supply variation doesn't make much difference on delay value. jitter generated due to power supply variation in FDU is 7.12ps in worst case while it was 16.2ps in resistive-capacitive based delay elements. some times load capacitor value set at higher value can degrade jitter number. Worst case jitter of overall all digital delay locked loop with clock distribution network is 26ps. Major component of jitter comes from clock distribution network which is nearly 11ps in worst case. Jitter introduced by coarse delay line is 6ps in worst case. other components of DLL as frequency divider and UI delay unit add remaining portion of delay. Due to each cycle delay update worst case locked time is 96 clock cycle of 8UI clocks.

References

[1] Chen Zhujia, Yang Haigang, Liu Fei, Wang Yu, *"A fast-locking all-digital delay-locked loop for phase/delay generation in an FPGA"*, Journal of Semiconductors, October 2011.

[2] Chia-Yu Yao ,Yung-Hsiang Ho, Yi-Yao Chiu, and Rong-Jyi Yang *"Designing a SAR-Based All-Digital Delay-Locked Loop With Constant Acquisition Cycles Using a Resettable Delay Line"*, IEEE Conference Publications, ISBN 0-7803-7659-5.

[3] Bryan Casper, Frank O'Mahony, *"Clocking Analysis, Implementation and Measurement Techniques for High-Speed Data Links"*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 56, NO. 1, JANUARY 2009.

[4] Behzad Razavi, *Challenges in the Design of High-Speed Clock and Data Recovery Circuits*

[5] Shao-ku Kao, *"A delay-locked loop with self-calibration circuit for reducing phase error"*, Microelectronics Journal44(2013)663–669.

[6] M.-J. Edward Lee, Member, IEEE, William J. Dally, Fellow, IEEE, Trey Greer, Hiok-Tiaq Ng, Member, IEEE, Ramin Farjad-Rad, Member, IEEE, John Poulton, Senior Member, IEEE, and Ramesh Senthinathan, *"Jitter Transfer Characteristics of Delay-Locked Loops—Theories and Design Techniques"* IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 4, APRIL 2003.

[7] James R. Burnham, Gerard K. Yeh, Enoch Sun, Chih-Kong Ken Yang, *"Design and Analysis of a Jitter-Tolerant Digital Delay-Locked-Loop Based Fraction-Of-Clock Delay Line"* ISSCC 2004.

[8] Hsiang-Hui Chang, Student Member, IEEE, and Shen-Iuan Liu, Senior Member, IEEE, *"A Wide-Range and Fast-Locking All-Digital Cycle-Controlled Delay-Locked Loop"* IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 40, NO. 3, MARCH 2005.

[9] Jinn-Shyan Wang, Member, IEEE, and Chun-Yuan Cheng, Member, IEEE, *”An All-Digital Delay-Locked Loop Using an In-Time Phase Maintenance Scheme for Low-Jitter Gigahertz Operations”* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 62, NO. 2,FEBRUARY 2015.

[10] SUNG-MO (STEVE) KANG, YUSUF LEBLEBIGI, *”CMOS DIGITAL INTEGRATED CIRCUITS Analysis and Design”*