

**DESIGN AND SIMULATION OF CMOS BASED DELAY LOCKED
LOOP FOR MULTIPHASE CLOCK GENERATION AND ITS
APPLICATION AS A PROGRAMMABLE CLOCK MULTIPLIER**

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

MASTER OF TECHNOLOGY

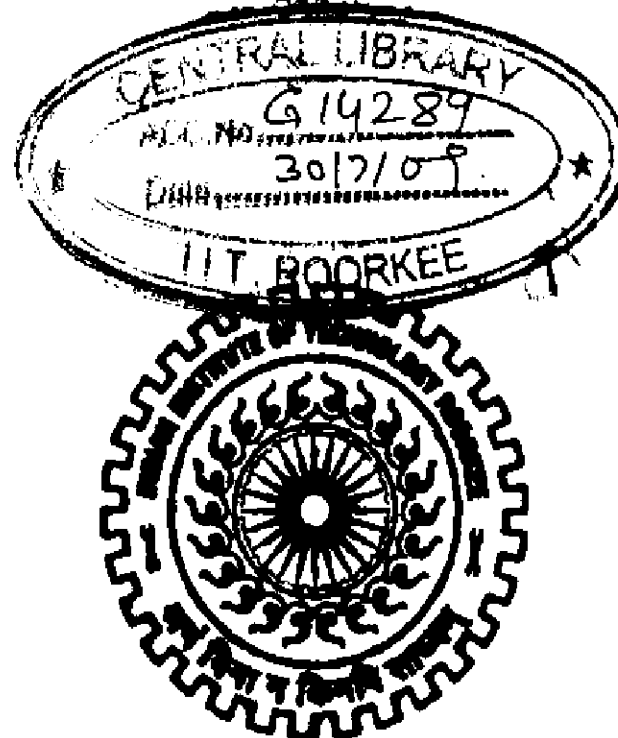
in

ELECTRONICS AND COMPUTER ENGINEERING

(With Specialization in Semiconductor Devices & VLSI Technology)

By

ABHAY GUPTA



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2008**

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being reported in this dissertation report, entitled “**Design and Simulation of CMOS based Delay Locked Loop for Multiphase Clock Generation and its Application as a Programmable Clock Multiplier**”, is being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Semiconductor Devices and VLSI Technology**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work, carried out from June 2007 to June 2008, under guidance and supervision of **Dr. A.K.Saxena**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date : **JUNE, 2008**

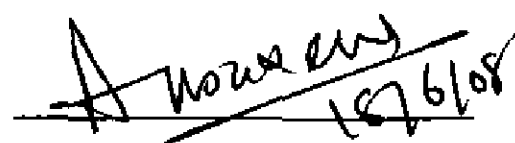
Place : Roorkee



Abhay Gupta

CERTIFICATE

This is to certify that the statement made by the candidate is correct to best of my knowledge and belief.



Dr. A.K Saxena

Professor

ACKNOWLEDGEMENT

At the outset, I express my heartfelt gratitude to Dr. A.K Saxena, Professor, Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for his valuable guidance, support, encouragement and immense help. I consider myself extremely fortunate for getting the opportunity to learn and work under his able supervision. I have deep sense of admiration for his innate goodness and inexhaustible enthusiasm. It helped me to work in right direction to attain desired objectives. Working under his guidance will always remain a cherished experience in my memory and I will adore it throughout my life.

My sincere thanks are also due to rest of the faculty in the Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for the technical knowhow and analytical abilities they have imbibed in us which have helped me in dealing with the problems I encountered during the project. I also extend my sincere thanks to all the technical and non-technical staff of my VLSI Design Lab for providing me various tools and encouraging me through out my work.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample morale support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Abhay Gupta
M. Tech. (SDVT)

ABSTRACT

Delay Locked Loops (DLLs) have been widely used in various applications like clock distribution networks (to remove clock skew), frequency multipliers (to generate high frequency clock) etc , where conventionally Phase Locked Loops (PLLs) were used. This is because DLLs offer several advantages over PLLs, in terms or their simplicity, stability, easy to design and integrate on chip, low power consumption and better jitter performance. In nowadays, more and more applications such as local oscillators in communication systems, on chip clock generators in high speed microprocessors and clock distribution networks in synchronous circuits employ DLL. So, the DLL will be more significant in future.

In this thesis, in the first part the design of Delay Locked Loop for multiphase clock generation is presented at 180 nm technology node with low power and fast locking considerations. The previously proposed circuits for various units of DLL were modified for better performance and characteristics at desired technology node. The operating frequency range of the proposed DLL is 170 MHz- 252 MHz . The simulations were carried in T-SPICE with 1.8 V power supply and 200 MHz reference frequency. The lock time of proposed DLL is less than 300 ns and average power consumption is 6.46 mw.

In the later part of the thesis, the work is extended and the multiphase clocks obtained from the DLL, designed, are utilized for the purpose of frequency multiplication of the reference clock signal. A DLL based programmable frequency/clock multiplier with multiplication factor of 1X, 2X and 4X is proposed. The output frequencies of DLL based frequency multiplier are 170 MHz – 252 MHz (multiply by 1), 340 MHz – 504 MHz (multiply by 2) and 680 MHz – 1.004 GHz (multiply by 4). We used the same reference frequency of 200 MHz and showed (through T-SPICE simulations) the output with 200 MHz, 400 MHz and 800 MHz, with exactly 50 % duty cycle, for multiplication factor of 1X, 2X and 4X respectively. The average power consumption in all the three cases is less than 10 mw.

CONTENTS

Candidate's declaration and certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	vii
List of Tables	x
List of Abbreviations	xi
Chapter 1: Introduction	1
1.1 Background	1
1.2 Thesis Contribution	2
1.3 Thesis Organization	3
Chapter 2: Overview of DLL	4
2.1 Architecture of DLL	4
2.2 Components of DLL	5
2.2.1 Phase Detector (PD)	5
2.2.2 Charge Pump (CP)	9
2.2.3 Voltage Controlled Delay Line (VCDL)	11
2.3 Dead Zone in DLL	12
2.4 Locking Range of DLL	14
2.5 Stability Analysis of DLL	16
2.6 Critical Issues of Delay Locked Loop	17
2.7 Applications of DLL	20
2.7.1 Application in Clock Distribution Networks in Synchronous Circuits.	20
2.7.2 Application in Digital Testing (Built in Self Test (BIST))	22

2.7.3	Application in frequency/clock multiplication	22
Chapter 3:	DLL Based Frequency/Clock Multiplier	23
3.1	Basic of DLL based frequency multiplier	24
3.2	Comparative study of PLL/DLL based frequency multipliers	26
3.2.1	Overview of PLL based frequency multiplier	26
3.2.2	Comparison of PLL and DLL based frequency multiplier	29
3.3	DLL based Frequency/Clock Multiplier in Recent Past	32
Chapter 4:	Design of Delay Locked Loop for multiphase clock generation	35
4.1	Design of Phase Detector	35
4.1.1	Conventional Phase Detector	36
4.1.2	Phase Detector Incorporated in Proposed DLL	38
4.1.3	Simulation Output And Results of Incorporated Phase Detector	40
4.2	Design of Charge Pump	44
4.2.1	Conventional Charge Pump	45
4.2.2	Charge Pump Incorporated in Proposed DLL	48
4.2.3	Simulation Output And Results of Incorporated Charge Pump	52
4.3	Design of Voltage Controlled Delay Line	57
4.3.1	VCDL Incorporated in Proposed DLL	58
4.3.2	Simulation Output and Results of Incorporated VCDL	59
4.4	Loop Filter Capacitor	61
4.5	Complete Circuit	62
4.5.1	Simulation result and output of complete circuit	62
Chapter 5:	Design of DLL based Programmable Frequency/Clock Multiplier	67
5.1	Principle of Frequency Multiplication	67
5.2	Architecture of Proposed Programmable Frequency/Clock Multiplier	68

5.2.1	Glitch generator	70
5.2.2	Phase Selector and Edge Combiner	72
5.3	Simulation Output and Results of Programmable Frequency Multiplier	74
Chapter 6:	Conclusions	78
References		80

List of Figures

Fig. No.	Title of Fig.	Page No.
Fig. (2.1)	Block Diagram of DLL	4
Fig. (2.2)	Tristate Phase Detector	5
Fig. (2.3)	Waveforms for F_{ref} and F_{out} in phase	6
Fig. (2.4)	Waveforms for F_{ref} leading F_{out} in phase	6
Fig. (2.5)	Waveforms for F_{ref} lagging F_{out} in phase	7
Fig. (2.6)	State diagram of tri-state PD	7
Fig. (2.7)	Phase characteristics of tri-state PD (a) ideal (b) practical	8
Fig. (2.8)	Conceptual diagram of loop filter	9
Fig. (2.9)	Rise of output waveform	9
Fig. (2.10)	Conceptual diagram of VCDL	11
Fig. (2.11)	Ideal waveforms of PD (a) in locked state (b) with small phase difference	12
Fig. (2.12)	Charge Pump current in presence of dead zone of $\pm\Phi_0$ rad	12
Fig.(2.13)	Circuit to eliminate dead zone	13
Fig (2.14)	Response of actual PD (a) in locked state (b) for small phase difference	13
Fig. (2.15)	Multiphase DLL waveforms (a) correctly locked (b) falsely locked	14
Fig. (2.16)	Conditions to avoid false locking	15
Fig. (2.17)	Linear mathematical model of DLL	16
Fig. (2.18)	Timing jitter	18
Fig. (2.19)	Linear model of DLL with noisy source	18
Fig. (2.20)	Bode diag. of equ. (2.10)	19
Fig. (2.21)	Linear model of DLL for noisy VCDL	19
Fig. (2.22)	Bode diag. of equ. (2.11)	20
Fig. (2.23)	Conceptual clock distribution network	21
Fig. (3.1)	Conceptual block diagram of DLL based frequency multiplier with waveforms	24

Fig. (3.2)	(a) DLL as frequency multiplier (b) output waveforms	25
Fig. (3.3)	Block diagram of PLL based frequency multiplier	26
Fig. (3.4)	Linear model of PLL	27
Fig. (3.5)	Loop filter of PLL	28
Fig. (3.6)	Ring oscillator and jitter accumulation in its output waveform	29
Fig. (3.7)	Jitters in output waveform of edge combiner of DLL	30
Fig. (3.8)	Clock multiplier in [8]	32
Fig. (3.9)	Frequency multiplier in [17]	33
Fig. (3.10)	Block diagram of clock generator in [22]	33
Fig. (4.1)	Conventional phase detector using NAND gates	36
Fig. (4.2)	Pre-charge type phase detector	37
Fig. (4.3)	Phase detector using TSPC type D flip flop	37
Fig. (4.4)	Phase detector in proposed DLL	38
Fig. (4.5)	Simulated waveforms for F_{ref} leading F_{out}	41
Fig. (4.6)	Simulated waveforms for F_{out} leading F_{ref}	41
Fig. (4.7)	Simulated waveforms for F_{ref} in phase with F_{out}	42
Fig. (4.8)	Phase characteristics of designed phase detector	43
Fig. (4.9)	Conceptual circuit of charge pump	45
Fig. (4.10)	Conventional Charge pump	46
Fig. (4.11)	Block diagram of charge pump incorporated in proposed DLL	48
Fig. (4.12)	Pump up circuit	49
Fig. (4.13)	Complete charge pump incorporated in proposed DLL	50
Fig. (4.14)	Schematic diagram of low voltage wide swing cascode current mirror	51
Fig. (4.15)	Simulated waveforms of low voltage wide swing cascode current mirror	52
Fig. (4.16)	Simulated plot for voltage at node N	53
Fig. (4.17)	Simulated plot of pump up current	54
Fig. (4.18)	Simulated plot of pump up current	54
Fig. (4.19)	Simulated plot of pump up operation	55
Fig. (4.20)	Simulated plot of pump down operation	56
Fig. (4.21)	Conceptual diagram of VCDL	57
Fig. (4.22)	Bias stage and delay stage in VCDL of proposed DLL	58

Fig. (4.23)	Output of first two delay cells (a) without buffer (b) with buffer	59
Fig. (4.24)	Simulated transfer characteristics of designed VCDL	60
Fig. (4.25)	Simulated output of VCDL for $V_{ctrl}= 0.6$ V	60
Fig. (4.26)	Simulated output of VCDL for $V_{ctrl}= 2.32$ V	61
Fig. (4.27)	Locking curve of proposed DLL along with F_{ref} and F_{out}	64
Fig. (4.28)	Ripples on control voltage line	64
Fig. (4.29)	Multiphase clock output of DLL in locked state	65
Fig. (5.1)	Basic circuit of a frequency multiplier with 4 stage VCDL and its output Waveforms	67
Fig. (5.2)	Architecture of proposed programmable frequency/clock multiplier	68
Fig. (5.3)	Different cases of frequency multiplication	70
Fig. (5.4)	Glitch generator	70
Fig. (5.5)	Simulated waveform of glitch generator	71
Fig. (5.6)	Basic unit of phase selector	72
Fig. (5.7)	Edge Combiner	73
Fig. (5.8)	Simulated output for $(p1, p0) = (1, 0)$, output frequency = 800 MHz	75
Fig. (5.9)	Simulated output for $(p1, p0) = (0, 1)$, output frequency = 400 MHz	75
Fig. (5.8)	Simulated output for $(p1, p0) = (0, 0)$, output frequency = 200 MHz	76

List of Tables

Table No.	Title of table	Page No.
Table (4.1)	Summary of designed phase detector	43
Table (4.2)	Summary of designed charge pump	56
Table (4.3)	Summary of proposed DLL	66
Table (4.4)	Performance comparison of proposed DLL with prior similar designs	66
Table (5.1)	A,B and C inputs for various cases	72
Table (5.2)	Summary of proposed programmable frequency/clock multiplier	76
Table (5.3)	Performance comparison with prior clock multiplier	77

List of Abbreviations

Abbreviation	Meaning
DLL	Delay Locked Loop
PLL	Phase Locked Loop
PD	Phase Detector
PFD	Phase Frequency Detector
CP	Charge Pump
VCDL	Voltage Controlled Delay Line
VCO	Voltage Controlled Oscillator
FD	Frequency Divider
LF	Loop Filter
TSPC	Truly Single Phase Clock
PT	Pre-charge Type
TPL	Toggle Pulsed Latch
PCS	Personal Communication System
LO	Local Oscillator
BIST	Built In Self Test

Chapter 1

Introduction

1.1 Background

With the evolution of CMOS process technology and as the technology node advances in nanometer regime, the need for high performance VLSI system continues to grow. Clocks are used in high performance digital systems to sequence operations and provide synchronization between various functional units. Requirement of higher data rates and high performance have forced technology scaling, due to which we are moving towards higher clock frequency.

Clock skew and jitter poses a major frequency limitation and hampers the performance of synchronous circuits. As the clock frequency increases, clock skew and jitter minimization in clock distribution network becomes more and more significant. Also, due to interconnection problems, there is a need for on chip frequency multiplier/clock generator in high performance microprocessors.

In the field of communication the growing demand of wireless communication systems, like cordless, cellular phones etc. for voice and data have led to the increase in the level of integration of RF transceivers. It led to the implementation of all the RF functions in CMOS technology because of its low cost and high level of integration. All the applications consists of a RF local oscillator (LO) block to down convert entire RF band to an intermediate frequency(IF). For an integrated transceiver the phase noise and spurious tone of this LO block is very critical and should be kept low.

Conventionally Phase Locked Loops (PLLs) are used in all the above mentioned applications namely: in clock distribution networks for skew minimization, as a clock generator in high speed systems and also in local oscillators of communication system. Delay Locked Loops (DLLs) offer sever advantages over PLLs in all the above mentioned applications. First of all it (DLL) is a simple 1st order system which is inherently stable and requires only single capacitor in the loop filter as compared to PLL which require a minimum of 2nd order loop filter and thus making PLL a third order system. Due to this DLL is easy to integrate on chip. Another major advantage of DLL is its better jitter performance as compared to PLL. DLL does not have the problem of jitter

accumulation over many cycles, which is present in PLL and, thus results in low phase noise and stable output signal. This provides an excellent long-term jitter performance, or, equivalently a low close-in phase noise. Better jitter performance makes DLL very advantageous in clock distribution networks where jitter poses a frequency limitation, and also in frequency multipliers in local oscillators, wherein phase noise and spurious tones generated are to be kept below a certain level.

In most of the applications of DLL, the multiphase output's of Voltage Controlled Delay Line (VCDL), which is the final unit of DLL, are used to complete circuit function. Therefore in the initial part of the thesis we present the design of CMOS based Delay Locked Loop for multiphase clock generation at 180 nm technology node, for low power consideration. In the later part we present one of the major application of DLL which is DLL as a frequency multiplier.

1.2 Thesis Contribution

The key contributions of this thesis are:

1. Previously proposed circuits of various units of DLL are modified for better performance and characteristics at designed technology node.
 - a) The proposed DLL incorporates a high performance phase detector, which improves the previously proposed phase detector based on TSPC (Truly Single Phase Clock) type D flip flop, by removing its additional reset circuitry, but retaining its advantage of better linearity range and negligible dead zone.
 - b) The charge pump unit incorporated in the proposed DLL improves the previous similar circuit as it solves the problem of current mismatch (due to Channel Length Modulation (CLM) effects) which was present before, by using a cascode current mirror load.
 - c) In the voltage controlled delay line, an additional buffer circuit is added in order to reduce rise time and fall time of output waveforms, which is essential when DLL is used as a frequency multiplier.

2. A new programmable frequency multiplier, based on DLL, comprising of a phase selector, glitch generator and an edge combiner is proposed, with a multiplication factor of 1X, 2X or 4X controlled by an external digital signal.

1.3 Thesis Organization

The thesis organization is as follows:

Chapter 2 deals with an overview of DLL. All the units of DLL, namely phase detector, charge pump, loop filter and voltage controlled delay line, are discussed. Various issues such as dead zone, locking range and stability analysis of DLLs are also discussed. Finally there is a brief mention of various application of DLL.

Chapter 3 deals with an overview of DLL as a frequency multiplier/clock generator. A comparative study of DLL and PLL based frequency multiplier is done and various advantages of DLL based frequency multiplier over conventional PLL based frequency multiplier is deduced. In the end various DLL based frequency multipliers/clock generators developed in recent past have been presented.

In chapter 4 we present the design and simulation of DLL for multiphase clock generation. The simulation output and results of various units and finally of the complete circuit is presented.

In chapter 5 a programmable frequency multiplier/clock generator based on DLL designed in chap.4 is presented.

Chapter 6 concludes the thesis

Chapter 7 gives the reference details

Chapter 2

Overview of DLL

The basic idea of delay locked loop as a variant to phase locked loop, for better performance, was introduced by Mark and Edwin [1]. They proposed a variable delay line PLL for CPU-coprocessor synchronization, wherein they abandoned traditionally used voltage controlled oscillator(VCO) and used a voltage controlled delay line(VCDL) to improve noise immunity, ease loop stabilization, and permit dynamically adjustable clock periods. From then on DLL became a major unit in various applications like skew minimization in clock distribution networks, as a clock generator in high speed circuits and also as frequency multipliers in local oscillators etc.

Design and implementation of DLL continue to be challenging as design requirements of DLL such as low power consumption, fast locking, minimum static phase error, minimum chip area, minimum jitter etc become more and more stringent. In order to understand the challenges in DLL design , this chapter introduces basic operation of DLL along with certain design issues and some applications.

2.1 Architecture of DLL

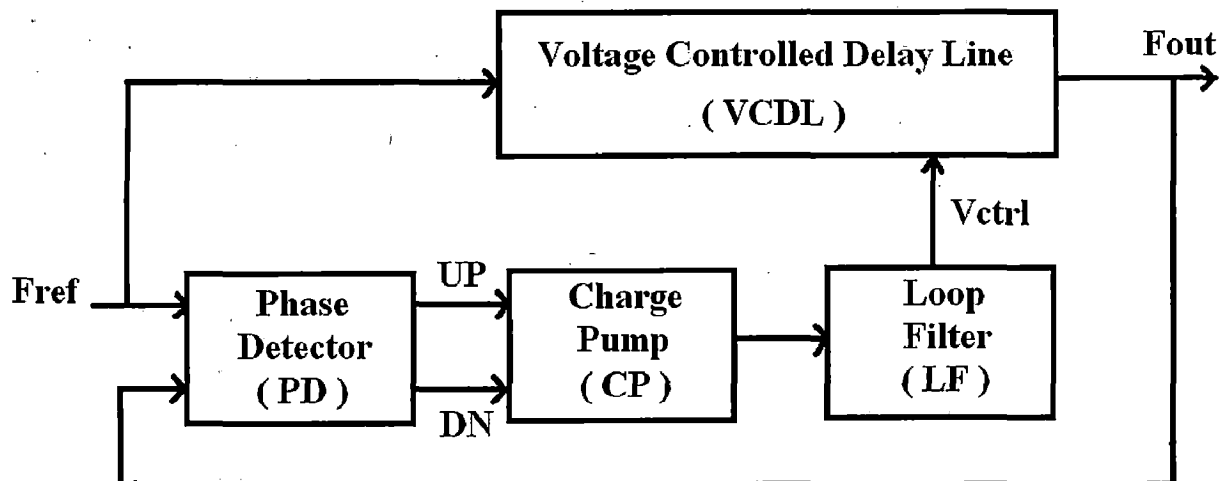


Fig. (2.1) Block Diagram of DLL

The block diagram of DLL is shown in fig. (2.1). It is a negative feedback control system that compares the output signal phase with input signal phase. It keeps the output signal in synchronism with the input signal. It also tracks the changes in the input signal , within its operating frequency range.

It consists of a phase detector (PD) , a charge pump (CP), a loop filter (LF) and a voltage controlled delay line(VCDL). As seen from the figure, the two inputs to the DLL and hence to PD are the external reference signal F_{ref} and the output signal F_{out} from the VCDL. The purpose of PD is to detect the phase difference between these two signals which are applied to its input. The PD generates UP or DN signal, depending on the phase difference. The CP unit, which follows the PD, utilizes these UP and DN signal. Depending on whether it receives an UP or a DN signal from the PD, it pumps in or pumps out charge and hence voltage from the loop filter. The output of loop filter is the control voltage V_{ctrl} which is given to VCDL. The input reference signal drives the VCDL which comprises a number of cascaded variable delay buffers. The loop's negative feedback drives the control voltage to a value that forces a zero phase error between the output clock and the reference clock. When the DLL is locked V_{ctrl} is constant and F_{ref} and F_{out} are exactly in phase.

2.2 Components of DLL

The various components of DLL are:

2.2.1 Phase Detector (PD)

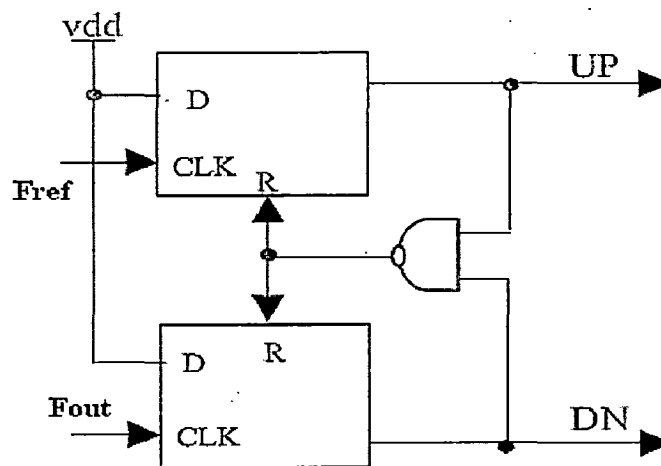


Fig. (2.2) Tristate Phase Detector

The purpose of the phase detector is to detect the phase difference of the two signals applied at its input. Conventionally an EX-OR gate or a J-K flip flop was used as a phase detector [2]. They have a poor linearity range. An EX-OR gate can resolve phase differences in $\pm \pi/2$ range and JK flip flop in $\pm \pi$ range. Also, in EX-OR gate the output is duty cycle dependent. The most widely used phase detector is a tri-state phase detector as shown in fig. (2.2). It is independent of duty cycle of the inputs. It consists of two D flip flops and a NAND gate in its reset path. The reset path must provide a sufficient width reset pulse in order to avoid dead zone (explained later). The leading edge of $F_{ref}(F_{out})$ sets UP(DN) pulse which is reset by lagging edge of $F_{out}(F_{ref})$. Fig. (2.3-2.5) shows the output waveforms of the PD for all the three possible cases.

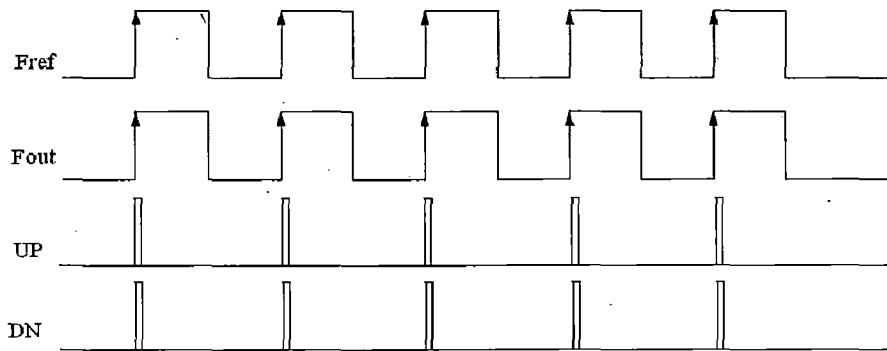


Fig. (2.3) Waveforms for F_{ref} and F_{out} in phase

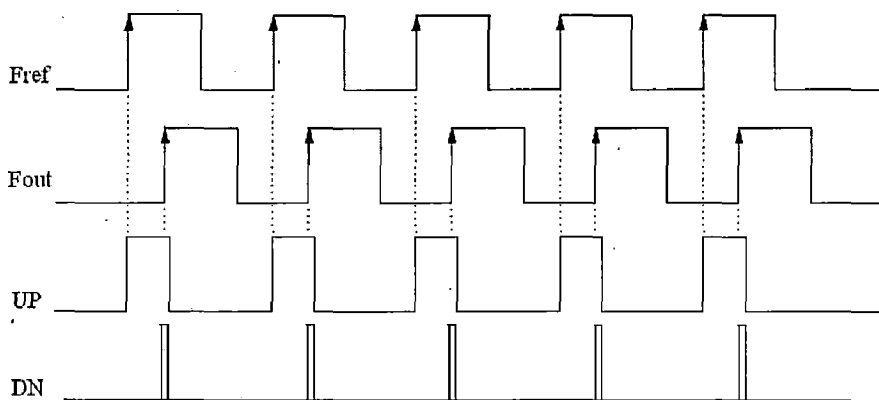


Fig. (2.4) Waveforms for F_{ref} leading F_{out} in phase

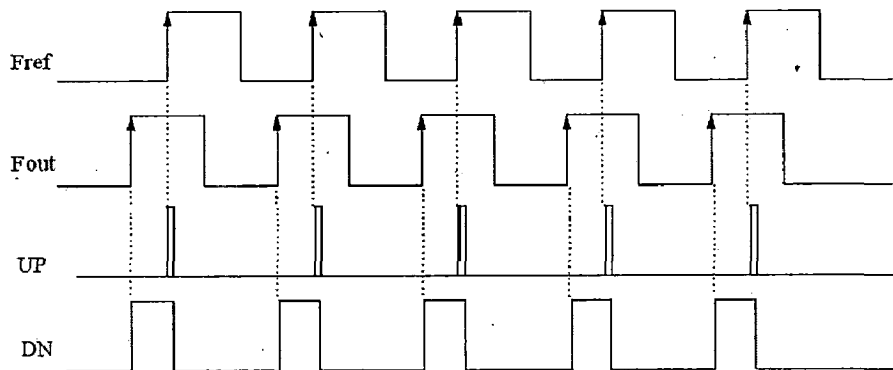


Fig. (2.5) Waveforms for F_{ref} lagging F_{out} in phase

Depending on the operation described above the PFD can be in one of the three states:

- $UP = 0, DN = 0$ ----- state-0
- $UP = 0, DN = 1$ ----- state-2
- $UP = 1, DN = 0$ ----- state-1

The state $UP=1, DN=1$ is prevented due the presence NAND gate that forces the PD to state 0 when such condition arises. The state diagram of the tri-state PD can be drawn as shown in Fig.(2.6).

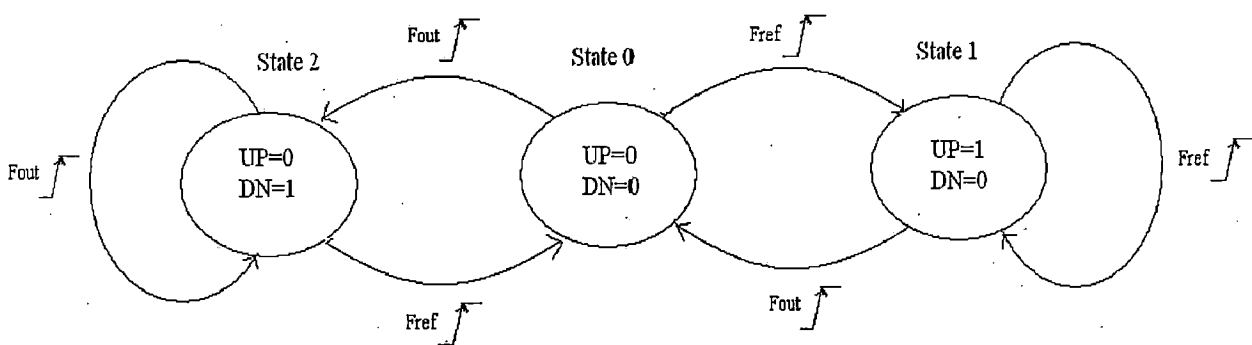


Fig. (2.6) State diagram of tri-state PD

If the PD is in state 0, then a transition on F_{ref} will take the circuit to state 1, where the state values are $UP = 1$ and $DN=0$. The circuit remains in this state until a positive transition occurs at

the F_{out} and the PD returns to state 0. The transition from state 0 to state 2 is same as transition from 0 to 1 state. The only difference is that a positive transition at F_{out} occurs instead at F_{ref} .

Phase characteristics of tri-state PD

The tri-state PD has a linearity range of $\pm 2\pi$. Fig. (2.7) shows the ideal and practical characteristics of tri-state PD [3].

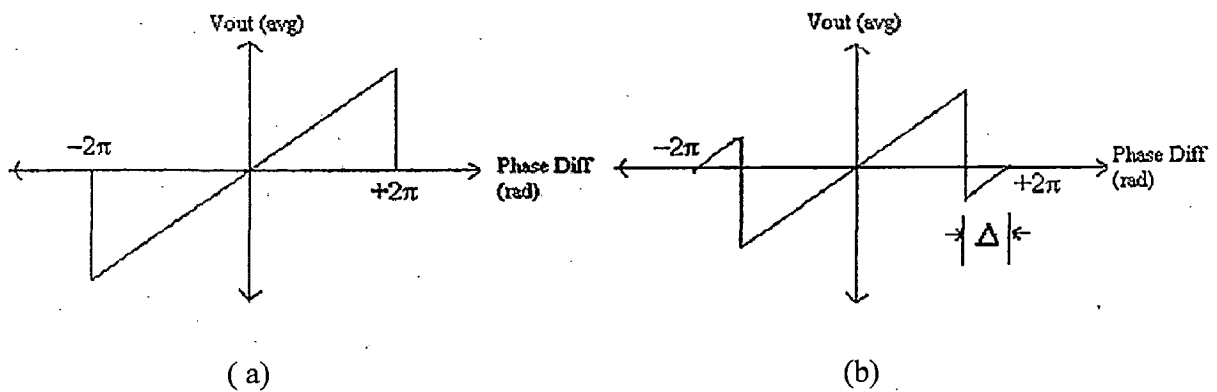


Fig. (2.7) Phase characteristics of tri-state PD (a) ideal (b) practical

As seen from Fig. (2.7) the PD's phase characteristic is ideally linear for the entire range of input phase differences from -2π to 2π . However due to delay of the reset path, the linear range is less than 4π . This is because; when the phase difference is nearing 2π (F_{out} lagging w.r.t F_{ref}) the next leading edge of F_{ref} arrives before the flip flops are reset due to finite reset delay. The reset overrides the new F_{ref} edge and does not activate UP signal. The subsequent F_{out} edge causes a DN signal. This effect appears as a negative output for phase difference higher than $(2\pi - \Delta)$ where $\Delta = 2\pi \cdot T_p / T_{ref}$.

T_p - reset pulse width

T_{ref} - period of the reference signal (F_{ref})

Maximum frequency of operation is given by [4].

$$F_{max} = 1 / 2 T_p \quad (2.1)$$

2.2.2 Charge Pump and Loop Filter

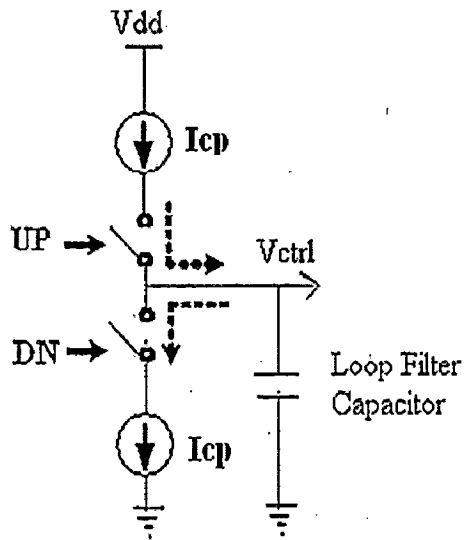


Fig. (2.8) Conceptual diagram of loop filter

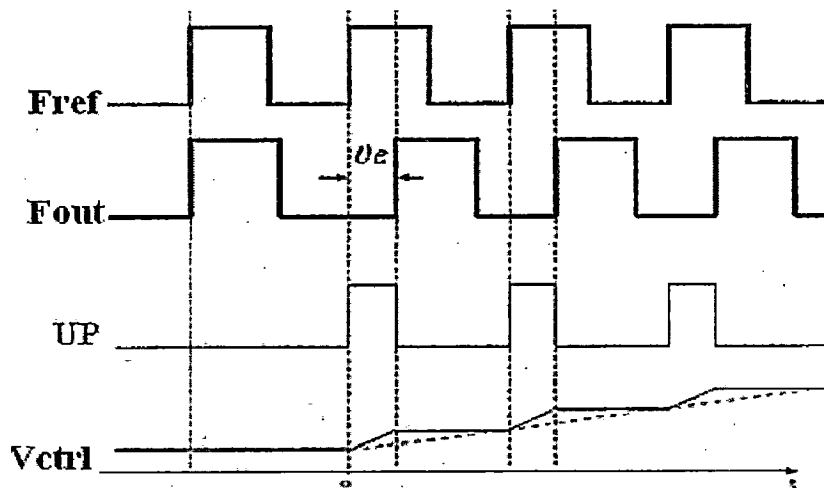


Fig. (2.9) rise of output waveform

The charge pump is the second unit of a DLL, following PD. It converts the digital signal at the output of PD into a continuous analog signal. The conceptual diagram of a charge pump along with loop filter is shown in fig.(2.8). The loop filter in DLL is a simple capacitor, which is

sufficient enough to ensure stability of DLL. This is unlike PLL where a second order loop filter is required for stability concerns (discussed later).

The charge pump consists of two switched current sources that pump charges into or out of the loop filter according to the PD outputs. VCDL control voltage, V_{ctrl} , rises when UP is active and the amount of voltage change is dependent on the duty cycle of the UP signal, which can be seen in fig. (2.9). Similarly, V_{ctrl} decreases when DN is active. As shown in the fig. (2.9), the case when the signal F_{ref} leads the signal F_{out} , UP signal goes high for the duration $t_p = \theta e \cdot T_{ref} / 2\pi$. This causes the current source to pump charge to the loop filter capacitor for the same duration, resulting in a staircase type waveform at the output V_{ctrl} (solid line). This generates a VCDL control voltage that alters the delay provided by VCDL in proper direction. Similar is the case when waveform at F_{out} leads to that at F_{ref} , where in DN signal goes high and the lower current source sinks the charge from the loop filter capacitor.

Improper designing of charge pump can lead to several non-idealities [5]. The various non-idealities and their affects are:

- **Leakage current**

Small currents flow even when the charge pump switch is off. This is due to sub threshold conduction. As the technology node decreases the leakage currents become more and more severe. They tend to produce a small static phase offset between the two input signals (F_{ref} and F_{out}).

- **Current mismatch of charge pump**

The two current sources used are never exactly equal. In the locked state, PD generates pulses such that net charge injected to the loop filter is zero. Due to unequal current sources, the PD generates pulses such that lower valued charge pump current has larger pulse width and higher valued charge pump current has lesser pulse width, such that net charge injected is zero. Thus DLL creates a static phase offset between the incoming signals in order to generate unequal pulse width signals. These unequal pulses in locked state also give rise to ripples on control voltage.

- **Timing mismatch**

During locked state, the PD generates UP and DN signal of equal widths, thus resulting in both UP and DN current pulse for equal duration. The timing mismatch between these two currents should be as small as possible, otherwise would result in large ripple on voltage control line which would result in spurious tones at the output.

- **Charge sharing, Channel charge injection and Clock feed through**

The parasitic capacitance of the current sources and of the switches causes the problem of charge sharing , channel charge injection and clock feed through, resulting in sudden jumps and ripples .

2.2.3 Voltage Controlled Delay Line

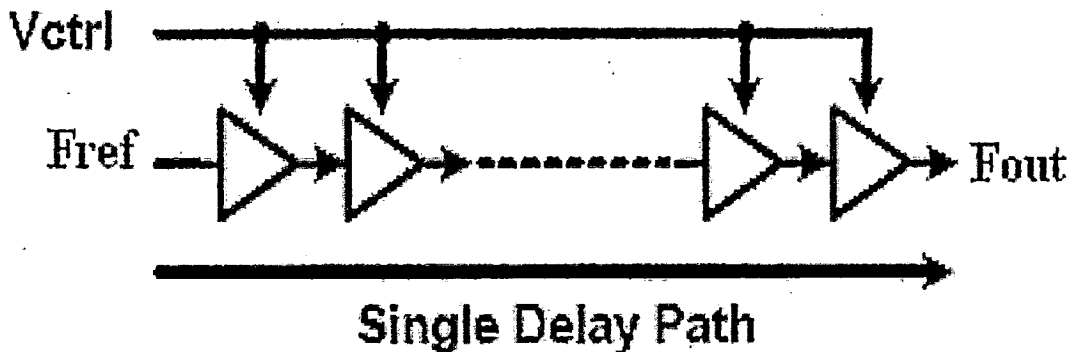


Fig. (2.10) Conceptual diagram of VCDL

The voltage Controlled Delay Line (VCDL) is the final unit of DLL. It provides necessary delay to the input reference signal, depending on the control voltage (V_{ctrl}) applied to it. Fig. (2.10) shows the conceptual diagram of a VCDL. It consists of several stages of buffers (delay elements) connected in cascade. The delay of each buffer is controlled by V_{ctrl} applied to it. The VCDL is an open loop configuration by itself, so it does not oscillate and thereby is different from the voltage-controlled oscillator (VCO) in a PLL. The delay range of VCDL depends on the number of delay elements. When the DLL is locked the total delay of VCDL is equal to one period of the reference signal. If N is the number of delay stages in VCDL, then each delay stage will provide a delay of T_{ref}/N , where T_{ref} is the period of reference signal. The operating frequency range of DLL is mainly dependent to the minimum and maximum delay provided by VCDL.

2.3 Dead Zone in DLL [6]

Dead zone is one of the important parameters of DLL. It is defined as the undetectable phase difference between its input (F_{ref} and F_{out}). It can result in static phase error between the two signal in locked state. The dead zone occurs due to finite time required by charge pump switches (in order to supply current corresponding to phase difference) to turn on owing to finite rise time of UP and DN signals.

Consider an ideal phase detector with zero delay in the reset path. In the locked state there will not be any reset pulse in UP and DN signal as shown in fig. 2.11(a).

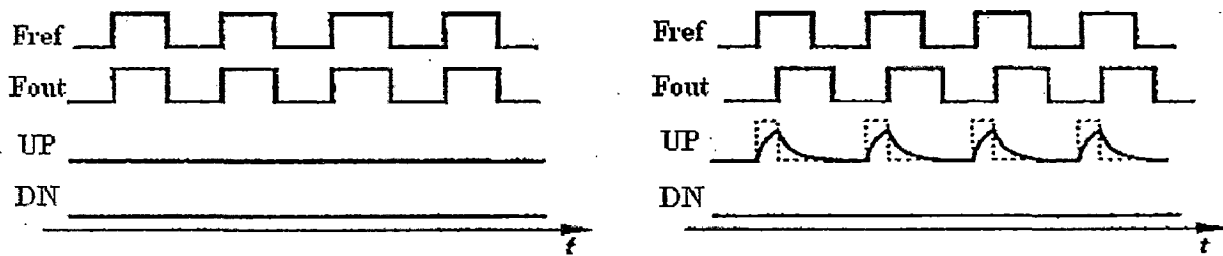


Fig. (2.11) Ideal waveforms of PD (a) in locked state (b) with small phase difference

Now if there is a small phase difference between the inputs (fig. 2.11(b)) the PD generate narrow pulses on UP (for F_{ref} leading w.r.t F_{out}) or on DN (for F_{ref} lagging w.r.t F_{out}). Since these pulses have finite rise time and fall time, it might be possible that for a very small phase difference they might not reach to a level to switch on the current sources, thus letting the phase difference $|\theta_e| < \Phi_0$ go undetected, as shown in fig. (2.12).

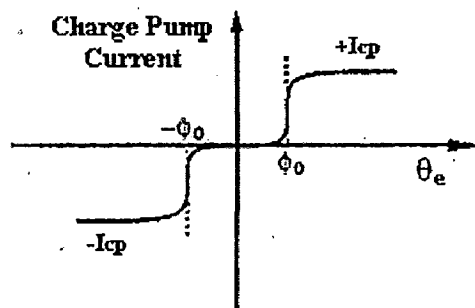


Fig. (2.12) Charge Pump current in presence of dead zone of $\pm\Phi_0$ rad

This undetectable range $\pm\Phi_0$ is called dead zone of DLL. This dead zone is undesirable as it causes static phase error between the input signals. Also the timing jitters present in the output feed back signal in the range $\pm\Phi_0$ go undetected .

To eliminate dead zone, a delay is introduced in the reset path of phase detector as shown in fig.(2.13).

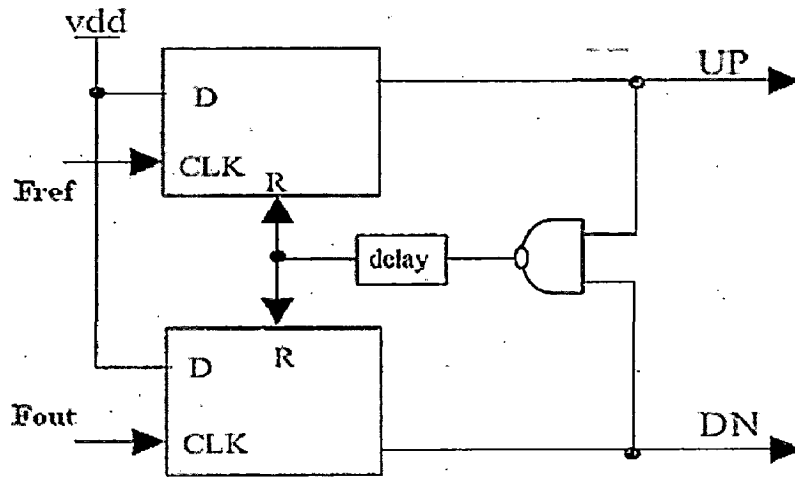


Fig.(2.13) Circuit to eliminate dead zone

The dead zone will vanish if the delay width is long enough to allow UP or DN to reach a valid logical level and turn on the switches in the charge pump. This delay might comprise of NAND gate delay , flip flop delay and other gates incorporated to remove dead zone.

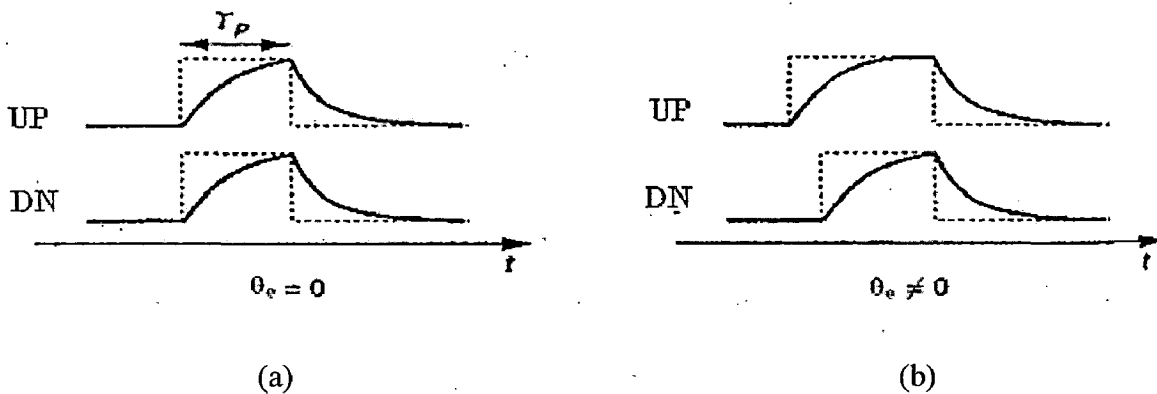


Fig (2.14) response of actual PD (a) in locked state (b) for small phase difference

As shown in fig.(2.14), T_p is the total delay of the reset pulse, which is just sufficient to switch on the current sources. Therefore, even a small amount of phase difference causes respective current source to switch on and initiating a corrective action.

2.4 Locking Range of DLL

The DLL has a poor locking range, which corresponds to a very small operating frequency range of the input reference frequency. The DLL may suffer from harmonic locking or false locking over wide operating range. Fig (2.15) shows the multiphase output waveform of eight stage VCDL with correct and falsely locked DLL.

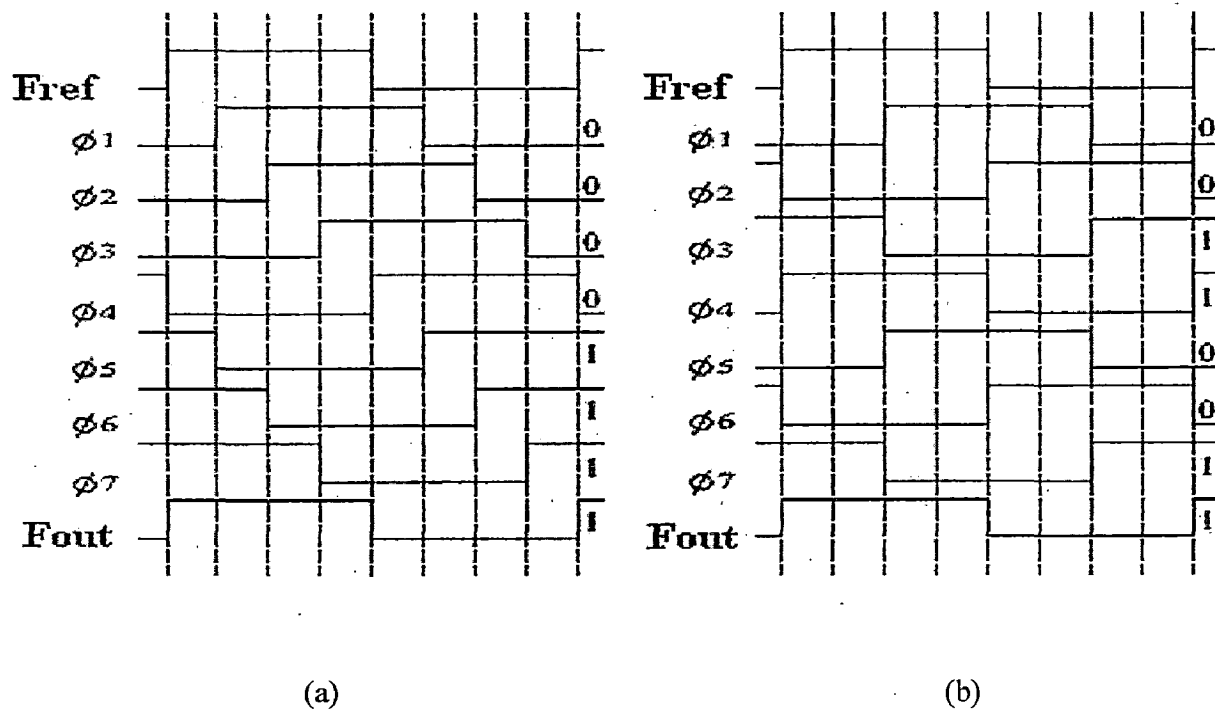


Fig. (2.15) Multiphase DLL waveforms (a) correctly locked (b) falsely locked

As seen from fig. 2.15(b) the DLL is falsely locked to two period of the input reference signal. In order to overcome incorrect locking problem, the maximum delay and the minimum delay, of VCDL has got an upper and a lower boundary [7]. The minimum delay $T_{VCDL_{MIN}}$ of the VCDL should be located between $0.5 \times T_{ref}$ and T_{ref} , and the maximum delay $T_{VCDL_{MAX}}$, should be located between T_{ref} and $1.5 \times T_{ref}$. On the other hand the initial delay of VCDL needs to be located between $0.5 \times T_{ref}$ and $1.5 \times T_{ref}$. The conditions are summarized as follows

$$0.5 T_{ref} < T_{initial} < 1.5 T_{ref} \quad (2.2)$$

$$T_{VCDL_{MIN}} < T_{ref} < T_{VCDL_{MAX}} \quad (2.3)$$

$$0.5 T_{ref} < T_{VCDL_{MIN}} < T_{ref} \quad (2.4)$$

$$T_{ref} < T_{VCDL_{MAX}} < 1.5 T_{ref} \quad (2.5)$$

where T_{ref} is the period of input reference signal. Fig. (2.16) describes the above relation

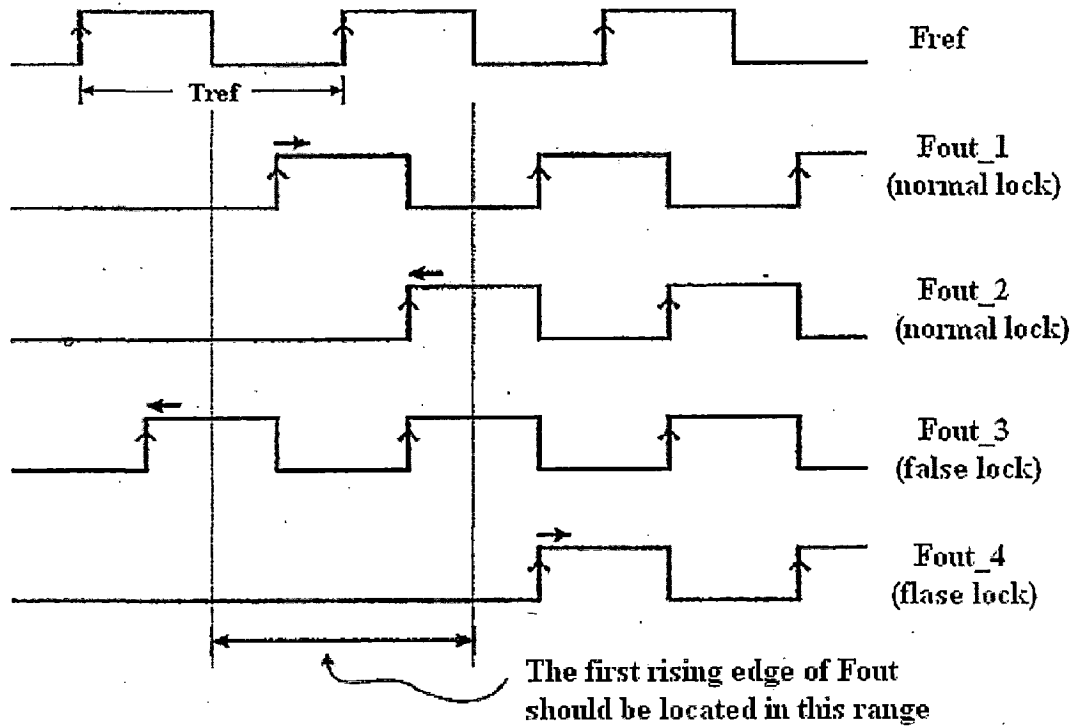


Fig. (2.16) Conditions to avoid false locking

From the above relations, the range of stuck free clock period should satisfy following relation.

$$\text{MAX}(T_{VCDL_{MIN}}, 2/3 \times T_{VCDL_{MAX}}) < T_{ref} < \text{MIN}(T_{VCDL_{MAX}}, 2 \times T_{VCDL_{MIN}}) \quad (2.6)$$

If the target reference signal satisfies the above condition, the DLL works perfectly without any harmonic or false locking. It can be seen that the above stuck free condition is satisfied for a very

narrow range of reference signal period and thus it corresponds to a small operating frequency range.

Various architectures have been proposed to solve the false locking problem and to have a wide operating frequency range. DLL by Foley and Flynn [8] utilizes a self correcting scheme, wherein a lock detect circuit is incorporated. It detects when the DLL is locked, or is attempting to lock to an incorrect delay and can bring the DLL back into a correct locked state.

H Chang and et al [9] proposes a different architecture to obtain a wide range. It uses a phase selector circuit and a start controlled circuit to solve locking problems and keep the latency of one clock cycle. The VCDL unit in its architecture contains large number of delay cells. Before the DLL begins to lock the phase selector circuit chooses the appropriate delay cell output to be the feed back signal (F_{out}). So, the number of delay cells utilized in VCDL can vary with varying input reference frequency and a large operating frequency range can be obtained. The mentioned architecture has a theoretical operating range from $1/(3T_{Dmin})$ to $1/(NT_{Dmax})$, where T_{Dmin} and T_{Dmax} are the minimum and maximum delay of the delay cell respectively.

One of the latest architecture to obtain a wide range, incorporating a frequency selector circuit is proposed by Cheng and Lo [10-11]. The VCDL in this architecture consists of a multi-controlled delay cell, whose delay is controlled by the frequency range selector circuit. Thus depending on the input reference frequency the delay of the delay cell and hence the delay of the VCDL is varied so as to satisfy equ(2.6).

2.5 Stability Analysis of DLL

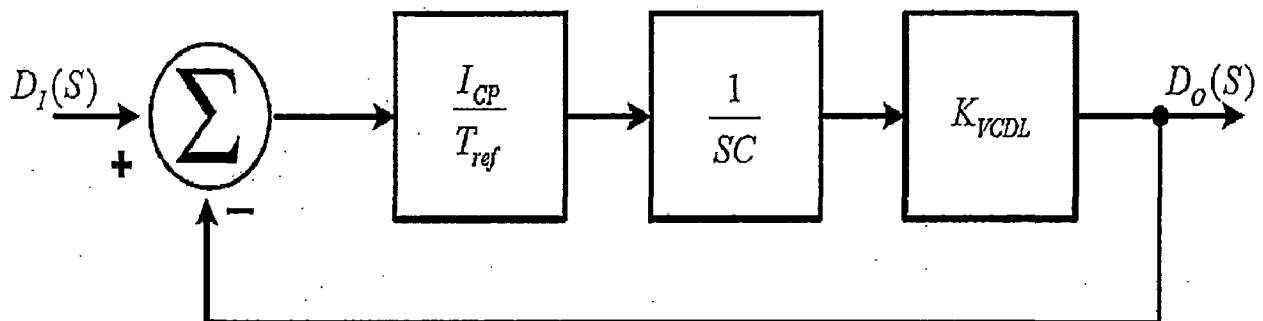


Fig. (2.17) Linear mathematical model of DLL

DLL is a negative feed back control system and hence its stability can be analyzed by studying, its dynamics. Fig. (2.17) shows the linear mathematical model of DLL with transfer function of each block. The summer represents PD, I_{CP} is the CP current, T_{ref} is the reference clock period, C is the capacitor of the loop filter and K_{VCDL} is the gain of VCDL. The analysis is done with the aid of continuous time approximation [12] where sampling nature of PD is ignored. In this approximation the discrete current pulses are approximated to a time averaged continuous current. When the loop is in steady state locked condition, the S-domain transfer function from input to output is given by

$$G_1(s) = \frac{D_o(s)}{D_i(s)} = \frac{1}{1+s/\omega_N} \quad (2.7)$$

Where,

$$\omega_N (\text{loop bandwidth}) = \frac{I_{CP} \cdot K_{VCDL}}{C \cdot T_{ref}} \quad \text{rad/s} \quad (2.8)$$

$D_o(s)$ - output delay

$D_i(s)$ - input delay

We can see from equ(2.7) that DLL is a 1st order system and is inherently stable. Since transfer function is inherently stable a wider loop bandwidth (B.W.) can be used. This allows a faster acquisition time.

However the continuous time approximation is valid only when loop B.W is about a decade or more below reference frequency [13]. Therefore, relation (2.9) must be satisfied for stability and should be taken care of in design of DLL.

$$\frac{\omega_N}{\omega_{ref}} = \frac{I_{CP} \cdot K_{VCDL}}{C} \leq \frac{1}{10} \quad (2.9)$$

2.6 Critical Issues of the Delay Locked Loop

In DLL design there are some critical issues, which should be taken care of by the designer. Some of the major issues are:

- Lock time
- Jitter performance
- Loop Bandwidth

The lock time describes how fast the DLL can attain lock from an initial unlocked state. It is dependent on the loop bandwidth of DLL. If the loop bandwidth is large the DLL attains lock fast. On the other hand if the loop bandwidth is small DLL takes long time to attain lock.

The loop bandwidth also affects the jitter performance of the DLL. When the DLL is in a locked state, the output signal is not ideal as there is a random variation in the clock edges from its ideal position as shown in fig. (2.18). This corresponds to timing jitter of the output and in frequency domain it results in phase noise.

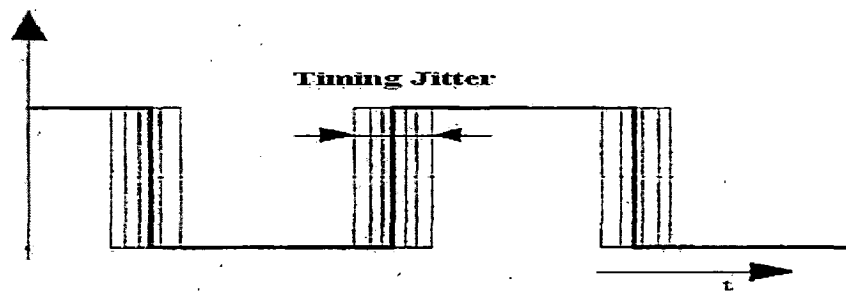


Fig. (2.18) Timing jitter

This is due to the presence of various noise sources in DLL. The most important noise sources are

- Noise in the input reference signal
- Noise generated internally in VCDL, caused by delay elements (random thermal noise) or induced in those elements by power supply or substrate noise.

For the noise due to reference signal, the linear model can be redrawn as shown in fig.(2.19)

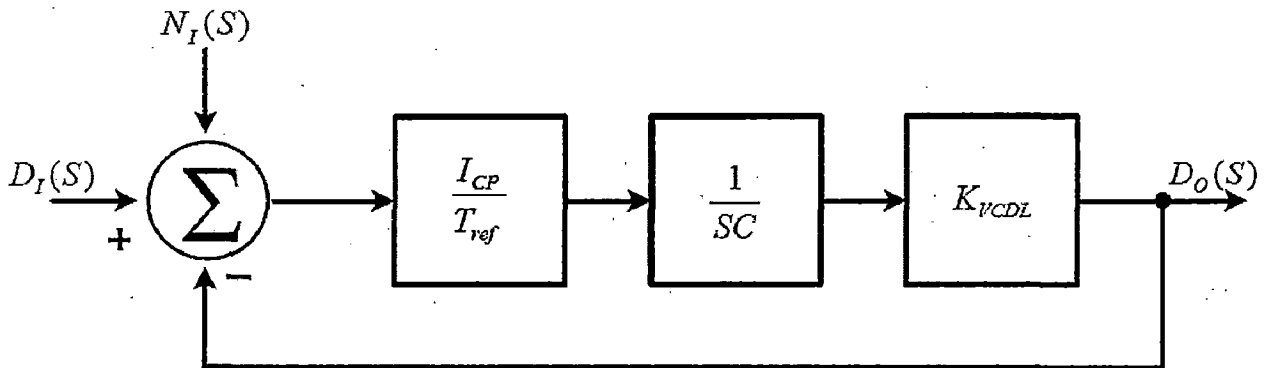


Fig. (2.19) Linear model of DLL with noisy source

The transfer function from output to input noise is

$$G_2(s) = \frac{D_o(s)}{N_i(s)} = \frac{\omega_N}{\omega_N + s} \quad (2.10)$$

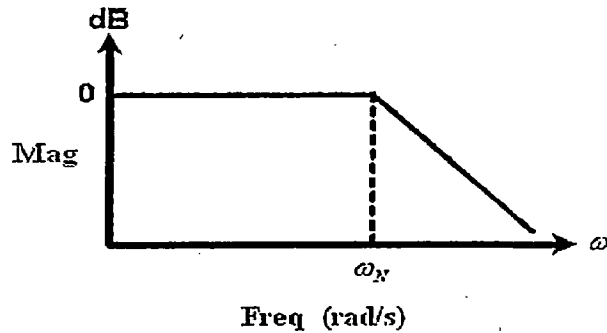


Fig. (2.20) Bode diag. of equ. (2.10)

From the Bode diagram shown in fig.(2.20) we can say that DLL characteristics has a low pass nature for noise due to input reference signal. Therefore the loop bandwidth should be set as small as possible in order to eliminate noise due to input reference signal.

For the noise internally generated in VCDL stage the linear model of DLL can be redrawn as shown in fig.(2.21)

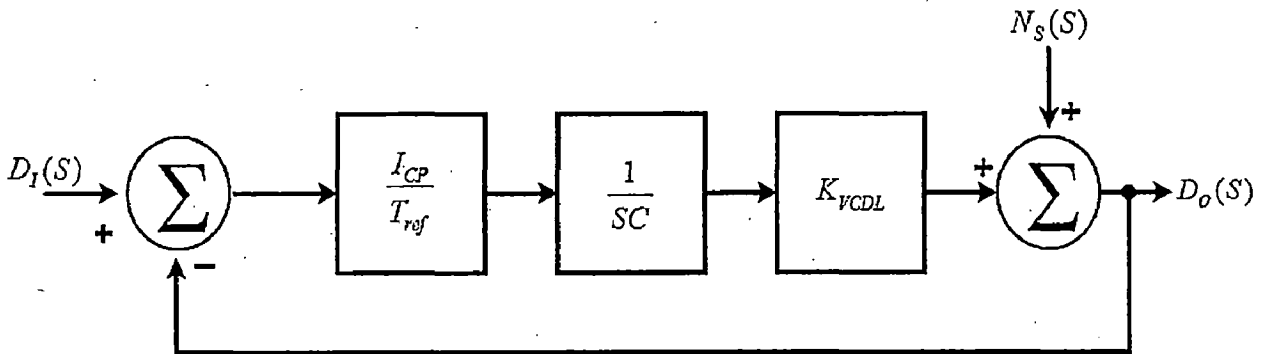


Fig.(2.21) Linear model of DLL for noisy VCDL

The transfer function for output to noise is

$$G_3(s) = \frac{D_o(s)}{N_s(s)} = \frac{s}{\omega_N + s} \quad (2.11)$$

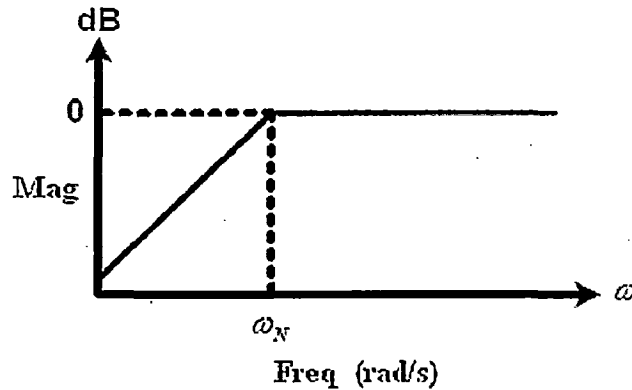


Fig. (2.22) Bode diag. of equ. (2.11)

From the Bode diagram shown in fig.(2.22) we can see that DLL characteristics has a high pass nature for noise due to VCDL stage. Therefore, if we want to minimize output jitter due to VCDL stage, the loop bandwidth should be made as wide as possible.

Importance of Loop bandwidth

As seen from the above discussion, loop bandwidth is one of the very important parameter in DLL design. It should be large if fast locking is desired or internally generated VCDL noise is to be suppressed. The loop bandwidth should be low if suppressing jitter due to input signal is of prime importance. Generally, the input reference signal to the DLL is jitter free as it comes from a stable crystal oscillator. Therefore, the loop bandwidth is made larger to enable fast locking and to minimize jitter due to VCDL stage. Also, it is to be ensured that the loop bandwidth chosen should satisfy equ(2.9) for stability of DLL.

2.7 Applications of DLL

DLL finds various applications in the synchronous circuits and also in the field of communication. Some of its applications are

2.7.1 Application in clock distribution networks in synchronous circuits.

The purpose of clock distribution network in synchronous circuits is to distribute a global reference clock signal to various digital circuits present in the chip. These clock signals at various digital circuits are used to define a time reference for movement of data within the

system. An accurate design of a clock distribution network is necessary to ensure that critical timing requirements are satisfied and no race conditions exist during synchronous communication of data between various digital circuits. Therefore, the relative clock skew between them should be ideally zero, but it is never the case .

Clock skew is caused by different RC delays of clock interconnects along different clock routes and varying delays of clock buffers due to layout variations, transistor sizes, temperature and process variations. Clock skew penalizes the overall performance and also poses maximum frequency limitation. As clock frequency increases, with the advancement of technology, clock skew minimization in clock distribution network becomes more significant. DLL and PLL are widely used in clock distribution network for clock skew minimization. Work in [14] shows the effectiveness of DLL over PLL in skew minimization. Fig. (2.23) shows a simple conceptual clock distribution network, incorporating a DLL for the purpose of skew minimization. The feedback in each tile adjusts the control voltage of VCDL such that buffered output is locked in phase to the global input clock. The feedback loop compensates for all the process , temperature and layout variation.

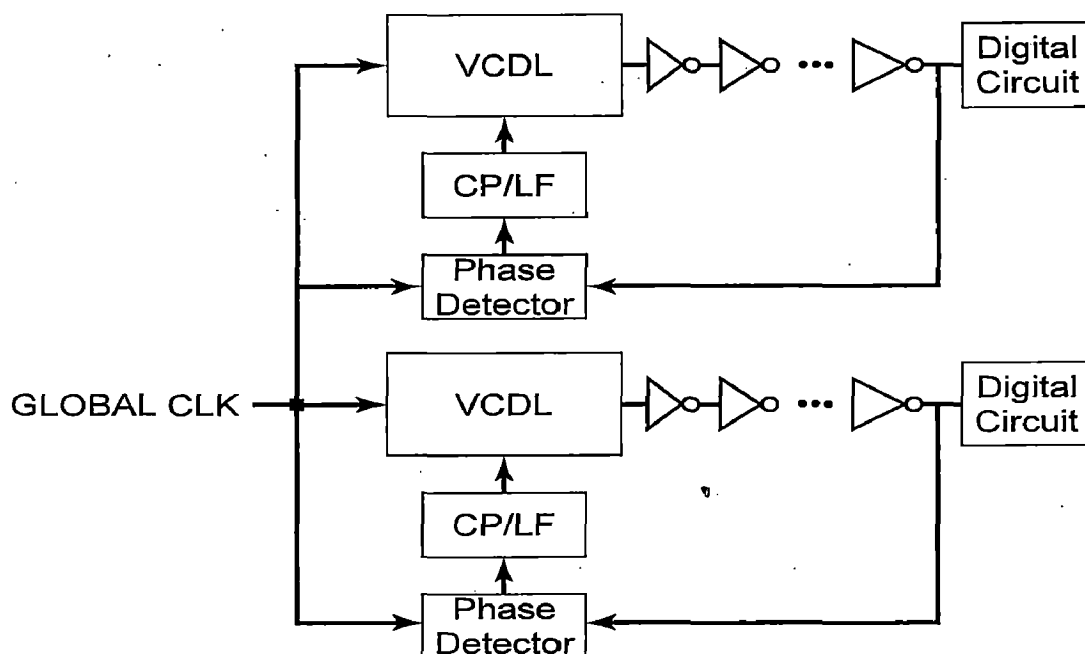


Fig (2.23) Conceptual clock distribution network [15]

2.7.2 Application in Digital Testing (Built in Self Test (BIST)) [16]

Since DLL, in locked state, produces equally spaced multiphase clock signal, it could be used in digital testing of timing sensitive specifications like I/O set up, hold time etc that require precise delayed clock signal. Conventionally the job of delay generation is done by external test equipment, which becomes very expensive as frequency of signal increase. Built in self test (BIST) methodology, where in some circuit blocks are built on chip (dedicated for test purpose), is becoming more popular owing to its cost effectiveness. Since DLL produces stable timing delays (at outputs of delay cells of VCDL) a DLL based BIST can be used for digital testing.

2.7.3 Application in frequency/clock multiplication

A detail of DLL as a frequency/clock multiplier is explained in next chapter.

Chapter 3

DLL Based Frequency/Clock Multiplier

Frequency multiplication is one of most important application of DLL. It can be used as a clock generator in high speed microprocessor and also as a frequency synthesizer in the field of communication.

As the technology node scales down, the demand for high frequency clock in microprocessor increases. Due to interconnection problem it is difficult to launch a high frequency clock, generated off chip, to the microprocessor. Therefore, a DLL is made on chip to serve as a clock generator from an externally connected crystal oscillator [17-18].

Also, in high speed serial I/O link a high frequency on chip clock multiplier is required at both transmitter and receiver side. The noise contribution by this clock multiplier should be minimum as it limits the bit rate. Again, a DLL is preferred for clock multiplication as it offers better jitter performance than conventional methods [19].

In the field of communication the growing demand of wireless communication systems, like cordless, cellular phones etc. for voice and data have led to the increase in the level of integration of RF transceivers. It led to the implementation of all the RF functions in CMOS technology because of its low cost and high level of integration. All the applications consists of a RF local oscillator (LO) block to down convert entire RF band to an intermediate frequency(IF). For an integrated transceiver the phase noise and spurious tone of this LO block is very critical and should be kept low. A DLL based local oscillator consisting of DLL as a frequency multiplier has shown good performance in terms of low phase noise and spurious tones [20].

Conventionally, a Phase Locked Loop (PLL) is used as a frequency multiplier for all the above mentioned applications. A DLL based frequency multiplier offers several advantages over PLL in generating a stable high frequency signal from a low frequency crystal oscillator.

In this chapter, application of DLL as a frequency multiplier and its advantages over PLL based frequency multiplier is explained in detail. It is followed by the introduction and discussion of the architectures of DLL based frequency multiplier developed in the recent past.

3.1 Basic of DLL based frequency multiplier

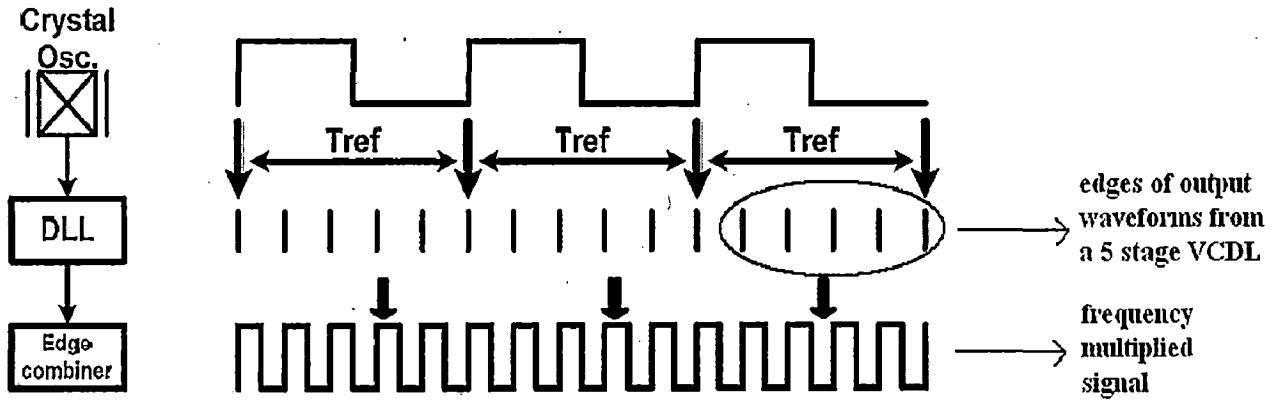


Fig.(3.1) Conceptual block diagram of DLL based frequency multiplier with waveforms

The basic of DLL based frequency multiplier can be explained with the help of fig. (3.1). It consists of a signal source (a crystal oscillator here), a DLL followed by an edge combiner. The objective of the DLL-based frequency multiplier is to produce a low-phase-noise high frequency signal by taking advantage of the inherently low jitter of a low-frequency crystal oscillator reference. When the DLL attains lock with the jitter free crystal oscillator reference, it generates a family of waveforms whose edges are well-controlled and evenly spaced within one period of the crystal oscillator. These family of waveforms are the multi phase output signals of DLL, obtained from VCDL delay cells. These edges form a pattern of higher-frequency transitions, and using the edge combiner, the crystal frequency is multiplied to realize the desired high frequency signal. The over all phase noise of the output signal, and hence timing jitters present in the output signal, in this case is much lower than conventional PLL based multipliers .

The frequency of the output signal depends on the number of delay cells in VCDL. The output frequency depends on the number of delay cells present in VCDL. If number of delay cells N in VCDL is odd the output frequency is given by [8][20] :

$$F_{out} = F_{ref} * N \quad (3.1)$$

where F_{out} is the frequency of the output signal and F_{ref} is the frequency of input reference signal. In this case each rising edge of delay cell output results in one oscillation of the output waveform.

If N is even, the frequency of oscillation is given by [17]:

$$F_{out} = F_{ref} * N/2 \quad (3.2)$$

In this case, the rising edge of odd numbered waveform causes a positive transition of the output signal and the rising edge of even numbered waveform causes a negative transition of the output signal.

Operation

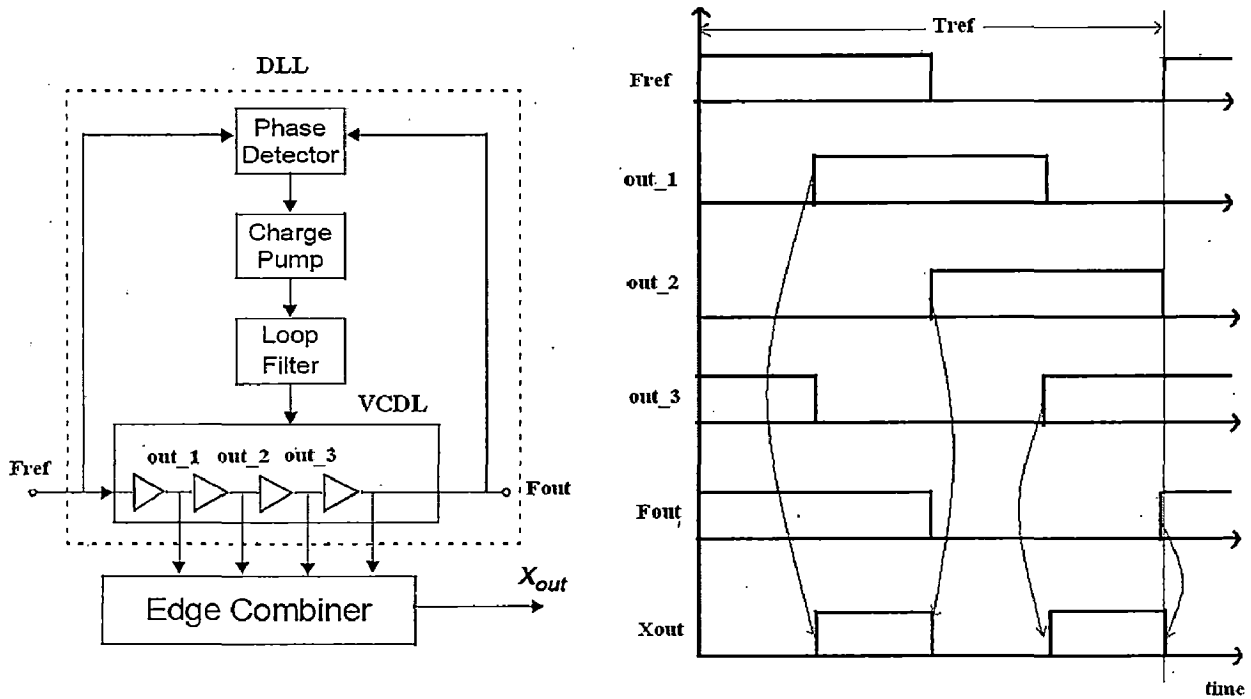


Fig. (3.2) (a) DLL as frequency multiplier (b) output waveforms

The operation of DLL based frequency multiplier can be explained with the help of fig. (3.2). Fig. 3.2 (a) shows a DLL along with edge combiner. As an example, the VCDL unit consists of only four delay stages. Fig. 3.2(b) shows the waveforms at outputs of various delay cells and the final multiplied frequency. During the start of the operation the PD detects the phase difference between the output and the input and accordingly directs the CP to either decrease or increase the

control voltage. The delay of the delay cells vary accordingly such that in locked state, the input (F_{ref}) and the output (F_{out}) of the delay chain are in phase. In the locked state, the outputs of delay cells generate waveforms with edges that are evenly spaced within one period of the reference signal. As seen from fig.3.2(b) output of the first delay cell (out_1) is delayed by $T_{ref}/4$ units w.r.t the reference signal. The output of the second delay cell (out_2) is delayed w.r.t out_1 by same amount and so on. Thus we get a family of waveforms whose edges are evenly placed within one period of the reference signal.

In order to generate the multiplied clock signal the edge combiner consists of a circuitry, which causes a positive transition from the rising edge of the odd numbered delay cell output and a negative transition from the rising edge of even numbered delay cell output. As there are even number (four) of delay cells, the output multiplied frequency is twice the input reference frequency. Even if output waveforms of delay cells don't have 50% duty cycle, the multiplied signal always has a duty cycle of exactly 50%. This is because all the rising edges of the waveforms are evenly placed, and these rising edges are only responsible for transitions.

3.2 Comparative study of PLL/DLL based frequency multipliers

In this section a comparative study of PLL and DLL based frequency multipliers and various advantages of DLL based multiplier are presented. We will start with a brief overview of PLL based frequency multiplier.

3.2.1 Overview of PLL based frequency multiplier

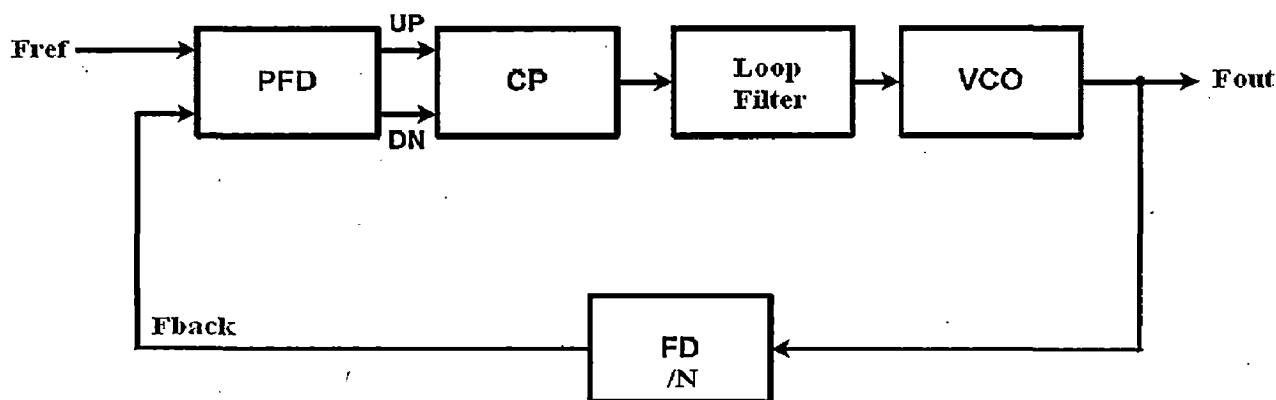


Fig.(3.3) Block diagram of PLL based frequency multiplier

The diagram in fig.(3.3) shows the block diagram of PLL based frequency multiplier. It consists of a phase frequency detector (PFD), a charge pump (CP), a loop filter, a voltage controlled oscillator (VCO) and a frequency divider (FD). The feedback signal F_{back} is compared to external signal F_{ref} by the PFD, which generates either UP or DN signal, depending on the phase and frequency difference between the inputs. According to signals of UP and DN, the CP will charge or discharge the loop filter to vary VCO output frequency. As we will shortly see, the loop filter required here is complex and is of minimum 2nd order. If the FD is divided by N circuit, the in the locked state:

$$F_{ref} = F_{back}$$

$$F_{back} = F_{out}/N$$

Therefore,

$$F_{out} = N * F_{ref}$$

Linear model of PLL [12]

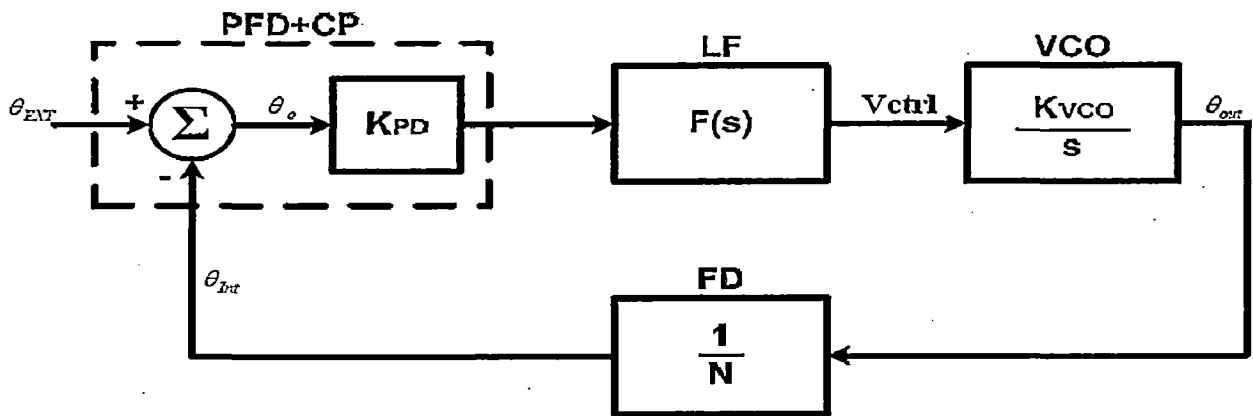


Fig.(3.4) Linear model of PLL

The linear mathematical model of a PLL is shown in fig.(3.4). Even though PLL is a highly non linear system, it can be approximated to a linear system using continuous time approximation, wherein the sampling nature of the PFD is ignored [12].

In the fig.(3.4) K_{PD} is the gain of PFD+CP combination. $F(s)$ is the transfer function of the loop filter. K_{VCO}/s is the transfer function of VCO.

When PLL is in locked state, the phase transfer function is given by:

$$H(s) = \frac{\theta_{out}(s)}{\theta_{Ext}(s)} = \frac{K_{PD}K_{VCO} F(s)}{s + K_{PD}K_{VCO} F(s)/N} \quad (3.3)$$

If a single capacitor (C) is used as a loop filter (like DLL) the transfer function becomes.

$$H(s) = \frac{\theta_{out}(s)}{\theta_{Ext}(s)} = \frac{K_{PD}K_{VCO} / C}{s^2 + K_{PD}K_{VCO} / CN} \quad (3.4)$$

As seen from equ.(3.4) the transfer function of the closed loop system has two poles on imaginary axis and hence unstable. It would result in un-damped oscillations at the output and the output will never settle.

A zero added to the above system (by adding a resistor (R1) in series to loop filter capacitor (C1)) makes the system stable, but cause sudden jumps on the voltage control line, which can further result in potential overload of VCO.

Therefore, to avoid it (sudden jumps), another capacitor (C2) is added in parallel to the series combination of resistor and loop filter capacitor, as shown in fig.(3.5).

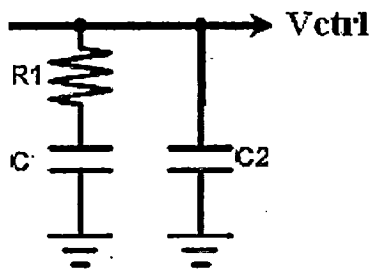


Fig.(3.5) Loop filter of PLL

Thus, a PLL requires a minimum of second order loop filter to ensure its stability.

3.2.2 Comparison of PLL and DLL based frequency multiplier

Advantages of DLL based frequency multiplier over PLL based

Simple system

As seen from the previous discussion, a PLL requires a minimum of 2nd order loop filter to ensure stability. In PLL, the VCO act like an ideal integrator, and thus presents a pole at origin. Due to this the closed loop transfer function is of minimum 3rd order and hence the PLL becomes a highly complex system. As the order of a system increases, it becomes more and more difficult to maintain its stability and hence hard to design. Moreover, since the loop filter is generally made up of passive components, the area occupied by them increases as they grow in number. This makes the loop filter of PLL costly to integrate on chip.

On the other hand, in DLL a simple capacitor in the loop filter is sufficient enough to ensure its stability and VCDL unit just provides gain (unlike VCO in PLL which provide a pole at origin). This makes DLL a 1st order system, which is inherently stable. Also, as the loop filter consists of only single capacitor, it occupies less area and hence easier to integrate on chip.

Better jitter performance [21]

DLL has been shown to have a better jitter performance than PLL. This is because in DLL, jitter does not accumulate over many clock cycles, which is the case in PLL.

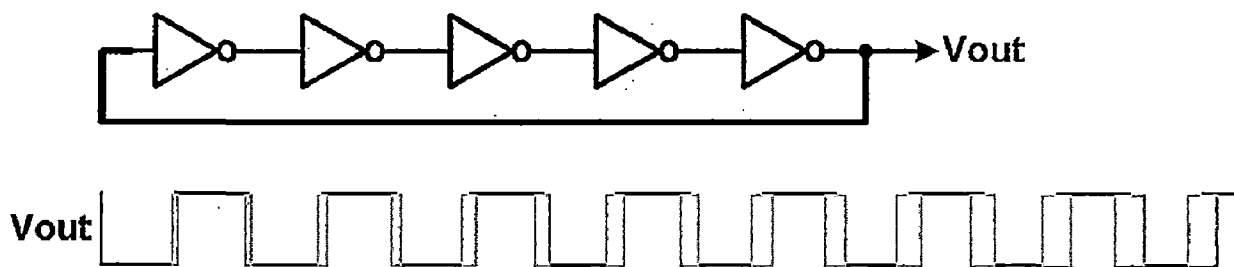


Fig.(3.6) Ring oscillator and jitter accumulation in its output waveform

Ring oscillator is the most commonly used configuration of VCO, incorporated in PLLs. As shown in fig.(3.6) it consists of a chain of odd number (five in our e.g.) of inverters, with the output fed back to the input. This configuration is very prone to random timing errors (jitters at

the output) caused by power supply or substrate noise or due to random thermal noise in the delay cells. The jitter per cycle of oscillation is determined by the sum of timing error contribution of each inverter stage in the ring. Since the output is connected back to the input, the timing jitter at the end of each oscillation is the starting point of the next. Thus, the random timing error of the output signal is the sum of timing errors of all previous oscillations and hence there is a phenomenon of jitter accumulation. This can be seen in fig.(3.6). When this ring oscillator, is incorporated in PLL-based frequency multiplier, the output is directly taken from the output of the VCO, and the VCO becomes the major source of noise. In this the jitter accumulation factor becomes an inverse function of loop bandwidth i.e. wider the loop bandwidth lesser is the jitter accumulation and vice versa. This is because, in wide loop bandwidth PLL, the PLL is fast enough to issue a corrective action (being negative feedback) and thereby removing the jitters at the output. However, loop bandwidth is constrained by practical considerations to a value several orders of magnitude lower than the reference frequency (equ.(2.9)).

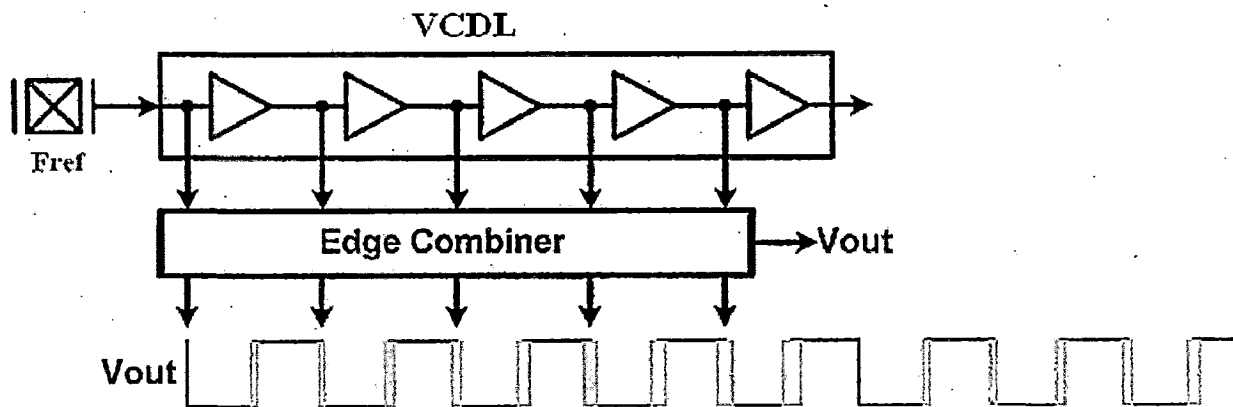


Fig.(3.7) Jitters in output waveform of edge combiner of DLL

A DLL incorporates a VCDL, which consists of a delay chain. In DLL, VCDL becomes the major source of noise and hence timing jitters at the output, but here the random timing jitter accumulates only within a single delay chain cycle. The timing error in one cycle of the delay chain does not affect the next cycle, because the waveform that triggers the next oscillation is the reference clock waveform (from crystal oscillator), which is stable and jitter free. This provides an excellent long-term jitter performance, or, equivalently a low close-in phase noise. When

DLL is used as a frequency multiplier, each output edge from DLL contains the timing uncertainties accumulated from previous stages within the same reference oscillation period. Therefore the jitter accumulation is limited within one reference period. This can be seen from fig.(3.7) where jitter accumulated disappears after fifth cycle of the output multiplied signal.

Lower Power Consumption [17]

For a given output frequency (multiplied signal) a PLL based frequency multiplier consumes more power than its DLL counterpart. The reason for this is as follows. A PLL requires a divide by two circuit at the output, in order to have a 50% duty cycle. Thus if a frequency of f units is desired at the output, the various units of PLL (VCO) works at $2f$ units of frequency. Whereas in a DLL, the multiplied output is inherently of 50% duty cycle. In DLL, the VCDL always work at the same frequency that of the reference signal, which is always lower than the desired multiplied signal. The lower frequency of operation of DLL corresponds to its lower power consumption.

Disadvantages of DLL based frequency multiplier over PLL based

Limited locking range

The locking range of DLL is limited and is given by equ. (2.6). This corresponds to a limited operating frequency range of the input reference signal. Therefore a DLL is reference signal dependent. This is not the case with PLL. Theoretically, the reference frequency can go to any value, but practically it is limited by various factors like loop bandwidth (for stability), VCO tuning range etc.

Poor variable multiplication

Unlike PLL, which can used to generate range of multiples of input frequency by changing the divider value, a DLL is used only for a fixed multiple of input frequency. Even if it is made programmable to generate various multiples of input frequency, the range of multiples is very limited.

3.3 DLL based Frequency/Clock Multiplier in Recent Past

Many DLL based frequency/clock multiplier have been proposed in recent past. Some of them are discussed below:

1. Foley and Flynn [8] have proposed a DLL based clock synthesizer and tunable oscillator, which generates an output with frequency exactly nine times the input reference frequency. The VCDL consists of nine delay stages. The frequency multiplier unit is shown in fig.(3.8). Multiplication process takes place in two steps. In the first step three clock signals $ck(1,2,3)$ are generated from nine output phases $\Phi(1...9)$. $\Phi 1, \Phi 4$ and $\Phi 7$ generate $ck1$, $\Phi 2, \Phi 5$ and $\Phi 8$ produce $ck2$ and $\Phi 3, \Phi 6$ and $\Phi 9$ produce $ck3$. These three signals $ck1, ck2$ and $ck3$ are phase shifted by one ninth of the reference clock period. In the second step these clocks are then combined in an AND-OR structure to produce a differential output clock, $ck+$ and $ck-$ having nine times the reference clock frequency. External load resistors set the output swing and match the output impedance to that of the test equipment. The disadvantage of this clock generator is that its multiplication factor is fixed (nine times).

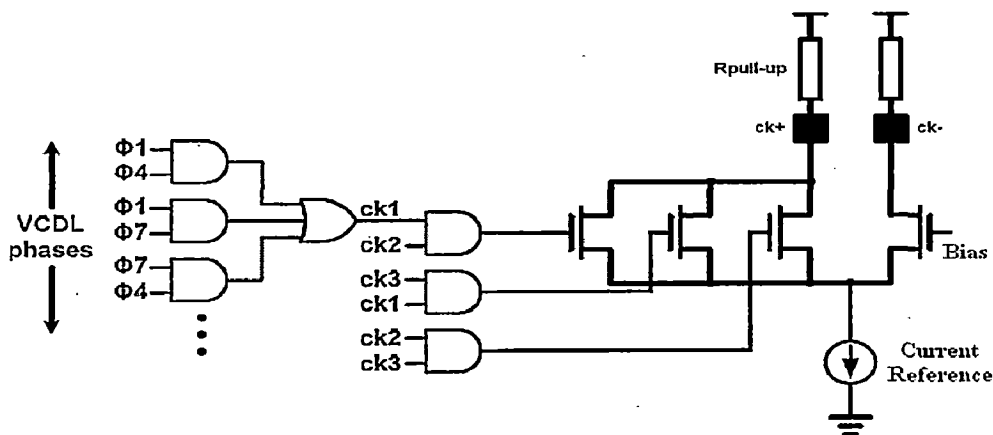


Fig.(3.8) clock multiplier in [8]

2. Chulwoo Kim *et al* [17] have proposed a low power small area DLL based clock generator. The frequency multiplier is shown in fig.(3.9). “ Ai ’s” are the output waveforms from delay cells. Qbd is internally generated signal and Clk is the output signal. At every rising edge of Ai , Qbd and the output clock signal (Clk) toggles. Thus output clock frequency can be expressed as $Freq_{OutputClk} = Freq_{Ref} \times N/2$, where $Freq_{Ref}$ is the frequency of input reference signal.

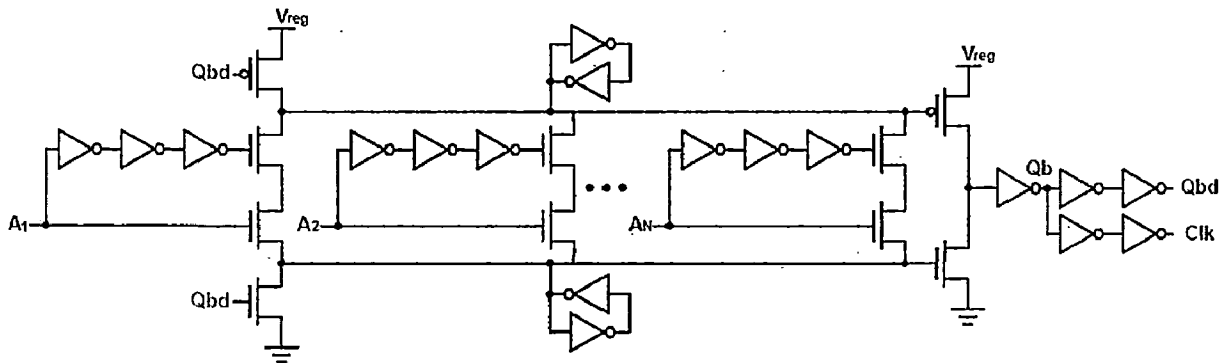


Fig.(3.9) frequency multiplier in [17]

The multiplication factor can also be made programmable by the architecture proposed. It consists of MUXs, which controls the number of delay cell waveforms k ($k \leq N$) to be fed to frequency multiplier and also controls the second input of phase detector (first being F_{ref}) coming from one of the delay cell output, thereby making the multiplication factor variable, but a wide tuning range of VCDL is necessary to cover all multiplication factors required.

3. Jin-Han Kim *et al* [22] have proposed a CMOS DLL-Based Clock Generator for Dynamic Frequency Scaling. The block diagram of the proposed architecture is shown in fig.(3.10)

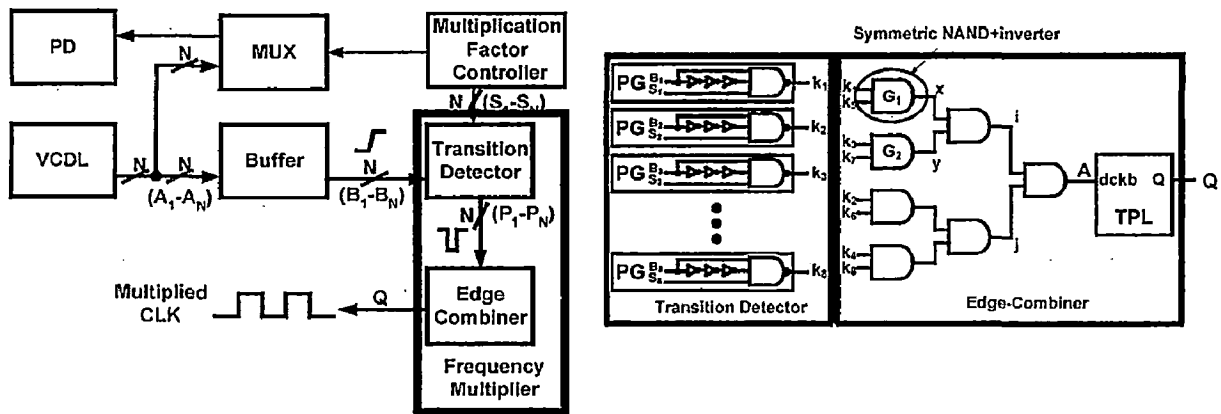


Fig.(3.10) Block diagram of clock generator in [22]

The multiplication factor in the proposed architecture is programmable. If VCDL has N delay cells, the output clock frequency is given by $F_{req_{OutputClk}} = F_{req_{Ref}} \times (M/2)$, (M =integer, $1 < M \leq$

N) , where Freq_{Ref} is frequency of reference signal. The transition detector generates a negative pulse for rising edge of each delay cell output. As a result a negative pulse appears at A for every rising edge of each delay cell output, causing TPL (Toggle Pulsed latch) output to toggle. Thus a multiplied clock with 50 % duty cycle is obtained at Q. The multiplication factor (M/2) can be dynamically chosen by multiplication factor controller and it depends on which of the last bit (delay cell output) selected by MUX is fed back to the phase detector. For e.g. if a multiplication factor is 3 is desired, the MUX selects the output of 6th delay cell to be fed back to phase detector

4. DLL based frequency multiplier can also be used in local oscillator of communication system. A local oscillator using a DLL based frequency multiplier for PCS (Personal Communication System) applications, proposed by Chien and Gray [20] has shown good performance in terms of phase noise and spurious tones. This is due to inherent advantage of DLL having good jitter performance as compared to conventional methods. The proposed circuit has some disadvantages. Firstly, the frequency multiplication factor is fixed and cannot be changed. Secondly, the edge combiner utilizes an LC tank circuit, which occupies large area and consumes large power. Wang *et al* [23] have proposed a CMOS local oscillator (maximum frequency of 1.2 GHz) for wireless application using a programmable DLL based frequency multiplier. The frequency of the output clock is 8X to 10X of an input reference frequency. It removed LC tank circuit from the edge combiner (purely digital here) as used in [18], resulting in drastic reduction in power dissipation and active area.

Chapter 4

Design of Delay Locked Loop for multiphase clock generation

In this chapter, the design of delay locked loop for multiphase clock generation is discussed. First of all each unit is separately designed and finally all the units are integrated, and the performance of the designed DLL is observed. The various units to be designed are:

- Phase Detector (PD)
- Charge Pump (CP)
- Voltage Controlled Delay line (VCDL)

All the units are designed at 180 nm technology node and simulations were carried in T-SPICE, with a reference signal of 200 MHz. The PD and VCDL units are designed with a power supply of 1.8 V. The CP unit is designed with 3.3 V power supply, as this can give maximum VCDL-control voltage compliance which helps to reduce the gain of VCDL. This helps to reduce the phase noise and spurious tones at the output.

4.1 Design of Phase Detector

The phase detector is the foremost unit of a delay locked loop. Its purpose is to detect phase difference between the two signals applied at the input. A tri-state phase detector, mentioned in chapter 2, has an advantage of better linearity range from -2π to $+2\pi$ and is the most widely used phase detector in delay locked loop. Due to this we incorporate a tri-state phase detector in our delay locked loop. The most important design considerations for a tri-state phase detector are:

- Dead Zone
- Power Consumption
- Maximum frequency of operation

Dead zone, resulting in undetectable phase difference between the inputs, can be minimized by designing the circuit such that there exist a finite reset pulse width, when the two input signals are exactly in phase. The reset pulse width should be wide enough to just turn on the charge

pump. Also, it should not be too wide, as in that case it would limit the maximum frequency of operation (equ.(2.1)) and cause more ripples on the control voltage line (due to inherent current mismatch). The power consumption can be kept minimum by using minimum number of transistors and using lower supply voltage.

4.1.1 Conventional Phase Detector

Fig. (4.1) shows one of the conventional phase detector, which is implemented using static CMOS technology. As seen, this PD consists D type flip flop made up of NAND gates.

This PD has severe limitations. It consist of large number of NAND gates, which corresponds to an excessively large number of transistors and thereby requiring a large chip area. Also, large number of transistors results in large power consumption.

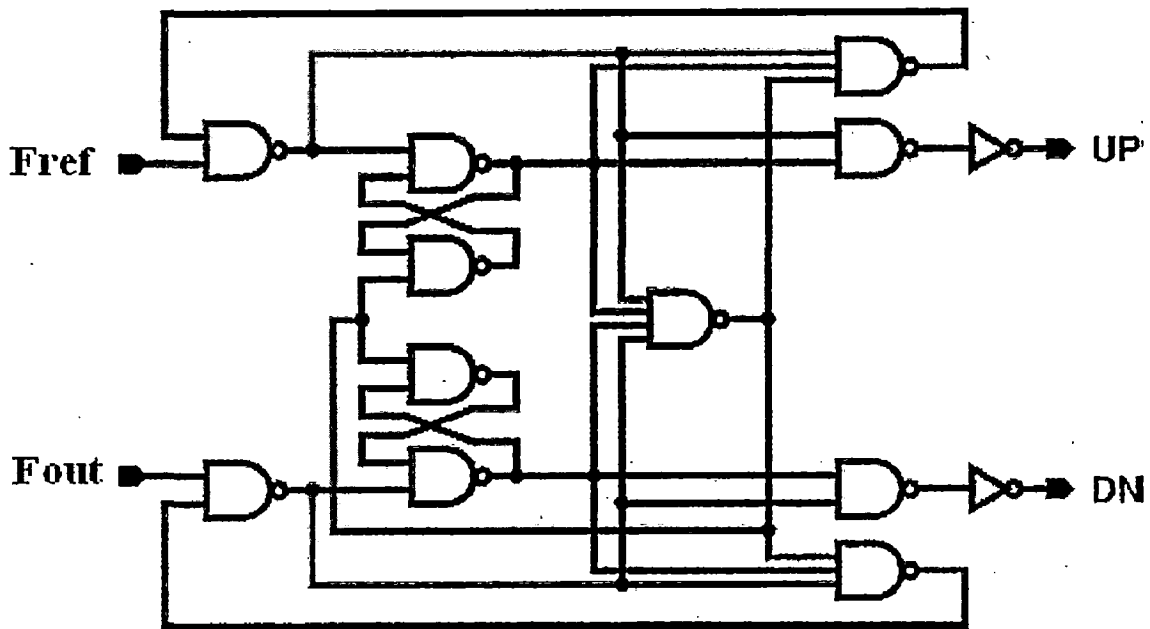


Fig. (4.1) Conventional phase detector using NAND gates

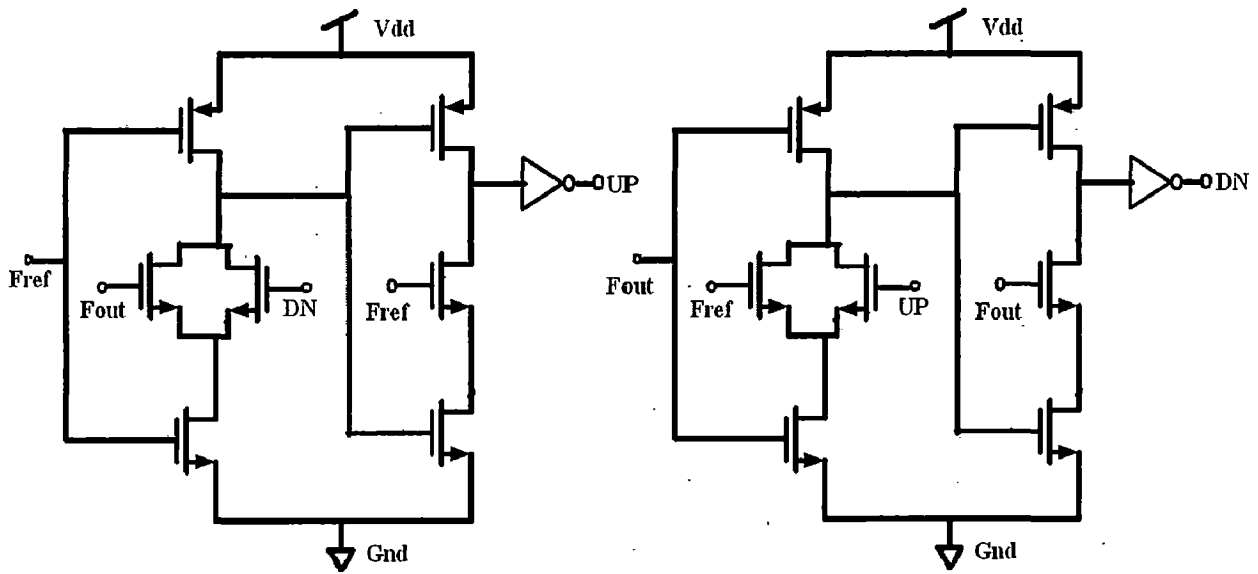


Fig. (4.2) Pre-charge type phase detector [24]

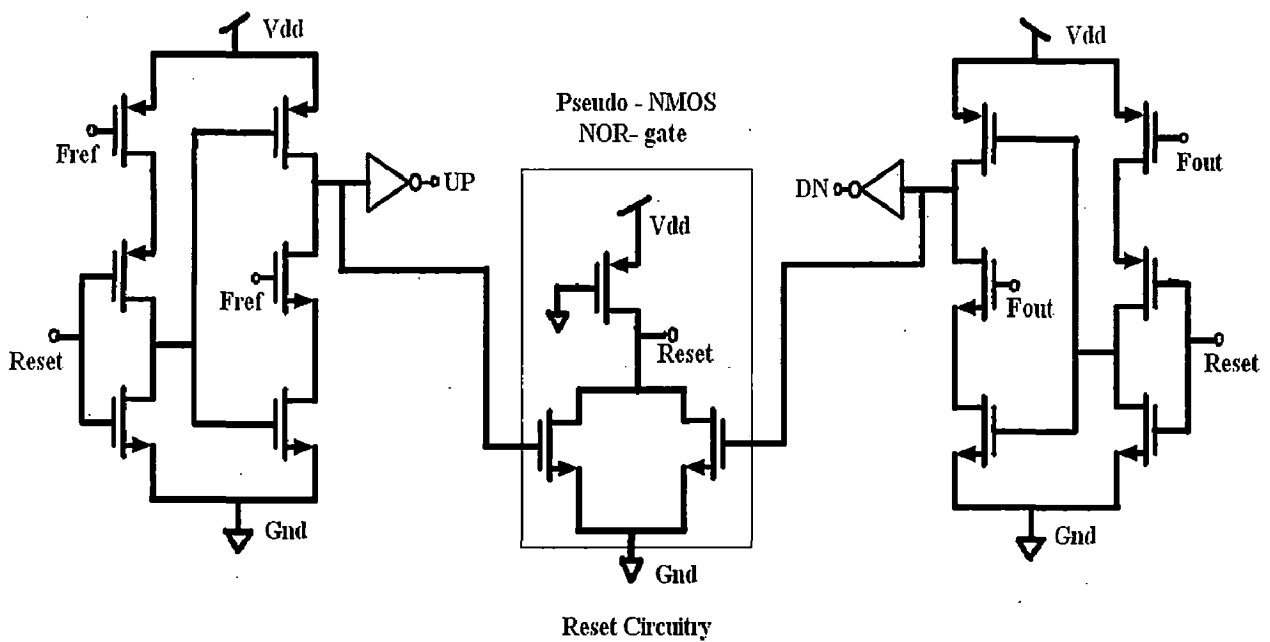


Fig. (4.3) Phase detector using TSPC type D flip flop [25]

In order to overcome its limitation a pre-charge type (PT) phase detector was proposed by Notani *et al* [24] as shown in fig(4.2). It worked on dynamic CMOS technology, utilizing the voltage generated at various nodes. However, the error detecting range of this phase detector was limited to $-\pi$ to $+\pi$.

This limitation is removed by phase detector proposed by Lee *et al* [25]. It again works on dynamic CMOS technology and uses a TSPC-type D flip flop structure. It has good performance characteristic in terms of dead zone and provides a linearity range of -2π to $+2\pi$, but requires an additional reset circuitry consisting of a pseudo-NMOS NOR gate. The pseudo-NMOS NOR gate makes the operation fast, but consumes static power when the NOR gate output is low and also the logic becomes ratioed.

4.1.2 Phase Detector Incorporated in Proposed DLL

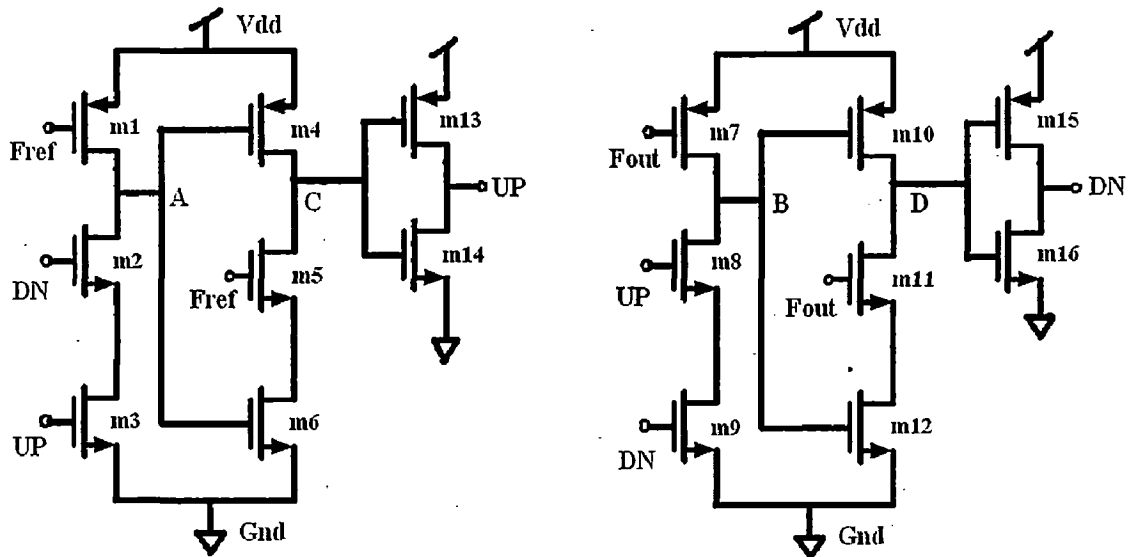


Fig.(4.4) Phase detector in proposed DLL

The phase detector incorporated in the proposed DLL is shown in fig.(4.4). It works on dynamic CMOS technology and is similar to the previously mentioned PD. It removes the reset circuitry which was required by the PD of fig.(4.3), without sacrificing on its performance.

Working

Initially both F_{ref} and F_{out} are low, therefore the transistors $m1$ and $m7$ are on (P-MOS). Due to this node A and node B gets charged to V_{dd} and hence switching off transistors $m4$ and $m10$. This also results in switching on of N-MOS transistors $m6$ and $m12$.

Consider the case when F_{ref} lead F_{out} in phase. When rising edge of F_{ref} (leading in phase) arrives, it causes N-MOS transistor $m5$ to turn on and hence creating a low resistance path from node C to ground ($m6$ already on). Thus, node C discharges to ground potential and thereby causing UP to go high instantaneously. This high UP signal switches on $m3$ and $m8$. The circuit will remain in this state, even if F_{ref} goes low. This is because node C is prevented from being pulled up as $m4$ is off (due to charges at node A).

Now, when rising edge F_{out} (lagging in phase) arrives DN signal goes high by similar mechanism as explained before. When this DN signal becomes sufficiently high, transistors $m2$ and $m9$ are turned on causing a low resistance path for node A and node B. Therefore, node A and node B discharges to ground potential, which results in P-MOS transistors $m4$ and $m10$ turning on. Due to this node C and node D are pulled up and causing UP and DN signal to go low. Exactly the reverse happens when F_{out} leads F_{ref} . When both F_{ref} and F_{out} are in phase, only reset pulses are obtained at UP and DN signals.

Thus, the above PD circuit utilizes the UP and DN signal directly for the reset operation, and hence eliminating the need for extra reset circuitry. The reset pulse generated is of sufficient width such that dead zone is almost completely eliminated. The PD consists of minimum number (16) of transistors. Proper relative sizing of various transistors is required in order have rail to rail swings. The combined strength of $m2, m3$ ($m8, m9$) should be more than $m1$ ($m7$). This is because, there exists a contention if F_{out} arrives when F_{ref} is already low (in case of F_{ref} leading F_{out}). Node A (B) must have a stronger pull down path than pull up, in order to get discharged. Similarly strengths of $m4$ and $m10$ should be weak in order to prevent UP and DN signal to go low until node A and node B are completely discharged to ground potential. The strengths of $m5$ ($m11$) and $m6$ ($m12$) should be high in order to generate instant response from the rising edge of F_{ref} (F_{out}).

4.2 Simulation Output And Results of Incorporated Phase Detector

The phase detector shown in fig.(4.4) is simulated in T-SPICE at 180 nm technology node ,1.8 V power supply and with 200 MHz input signals. The simulations were carried out for varying phase difference between the input signals (F_{ref} and F_{out}).

1. Fig. (4.5) shows the simulation output for the case when F_{ref} is leading F_{out} . The rising edge of F_{ref} sets the UP signal, this UP signal is then reset by the rising edge of F_{out} . A short duration reset pulse is seen on DN signal when the reset operation takes place.

2. Fig. (4.6) shows the simulation output for the case when F_{out} is leading F_{ref} . The rising edge of F_{out} sets the DN signal, this DN signal is then reset by the rising edge of F_{ref} . A short duration reset pulse is seen on UP signal when the reset operation takes place.

3. Fig. (4.7) shows the simulation output for the case when F_{ref} and F_{out} is exactly in phase. As seen, only short duration reset pulse are obtained at UP and DN signal. These pulses are essential in order to eliminate dead zone. Also, the reset pulses are shown on a magnified scale. They have a duration of around 0.2 ns. Therefore, the maximum frequency of operation of the incorporated phase detector is 2.5 GHz (equ(2.1)).

4. Fig. (4.8) shows the characteristic of converting phase difference (between input signals) into average loop filter current by designed PD via designed charge pump (next section) for I_{cp} (peak current) = 80 μ A. It is plotted with the help of simulation results obtained. We can see that characteristic is linear in region -5 ns to +5 ns, which corresponds to a phase difference of -2π to $+2\pi$ for 200 MHz reference signal. The linearity range is seen to be slightly less than -2π to $+2\pi$, due to finite reset pulse width [3] as mentioned before. We see that even for a extremely small phase difference (order of ps) between the two inputs, there is net current flowing. Thus, dead zone is almost eliminated.

5. Table (4.1) gives the summary of the performance of designed phase detector.

The performance of the designed phase detector is satisfactory. It has a linearity range of -2π to $+2\pi$, the average power consumption is 0.8 mw and the reset pulses are wide enough to eliminate the dead zone completely.

1. F_{ref} leading F_{out}

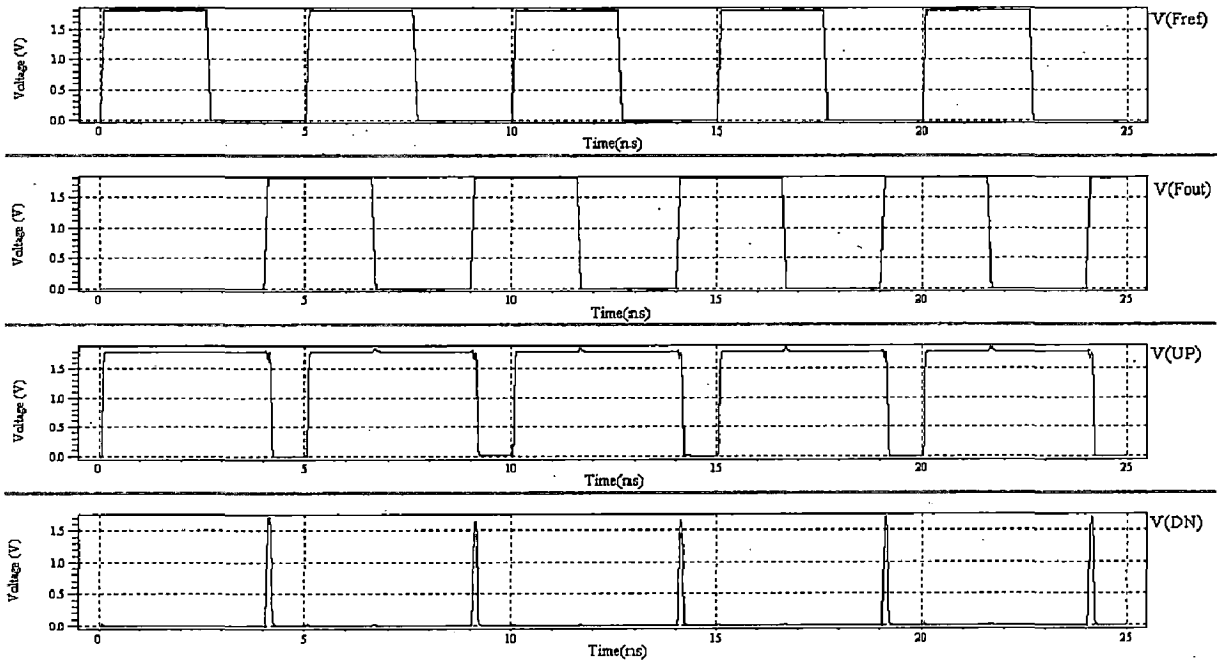


Fig.(4.5) Simulated waveforms for F_{ref} leading F_{out}

2. F_{out} leading F_{ref}

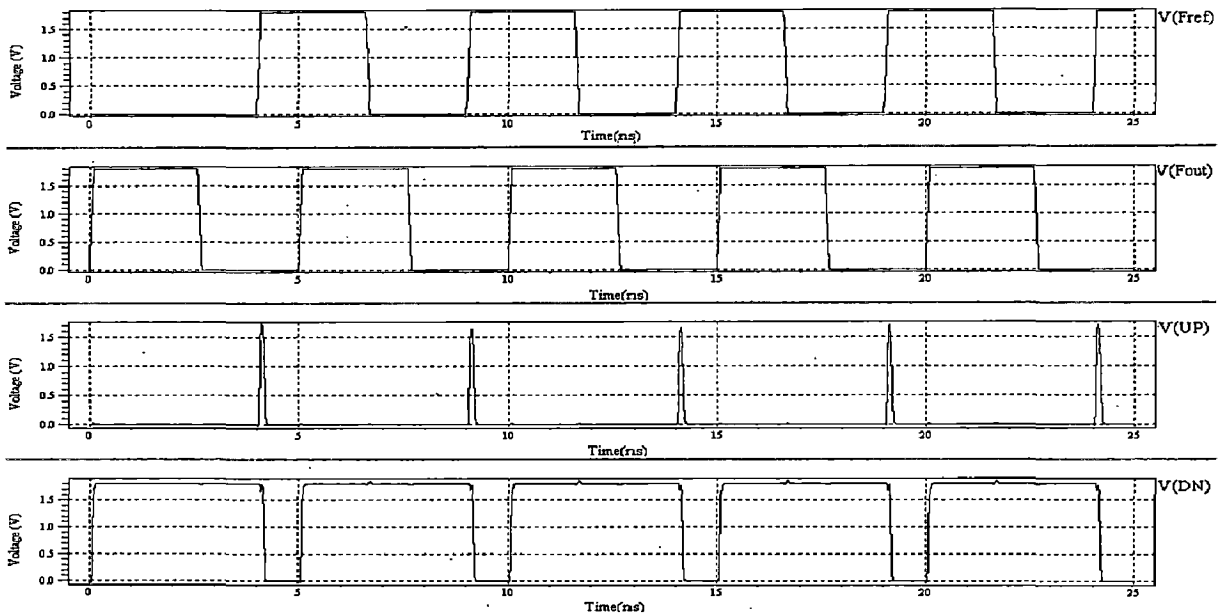


Fig.(4.6) Simulated waveforms for F_{out} leading F_{ref}

3. F_{ref} and F_{out} in phase

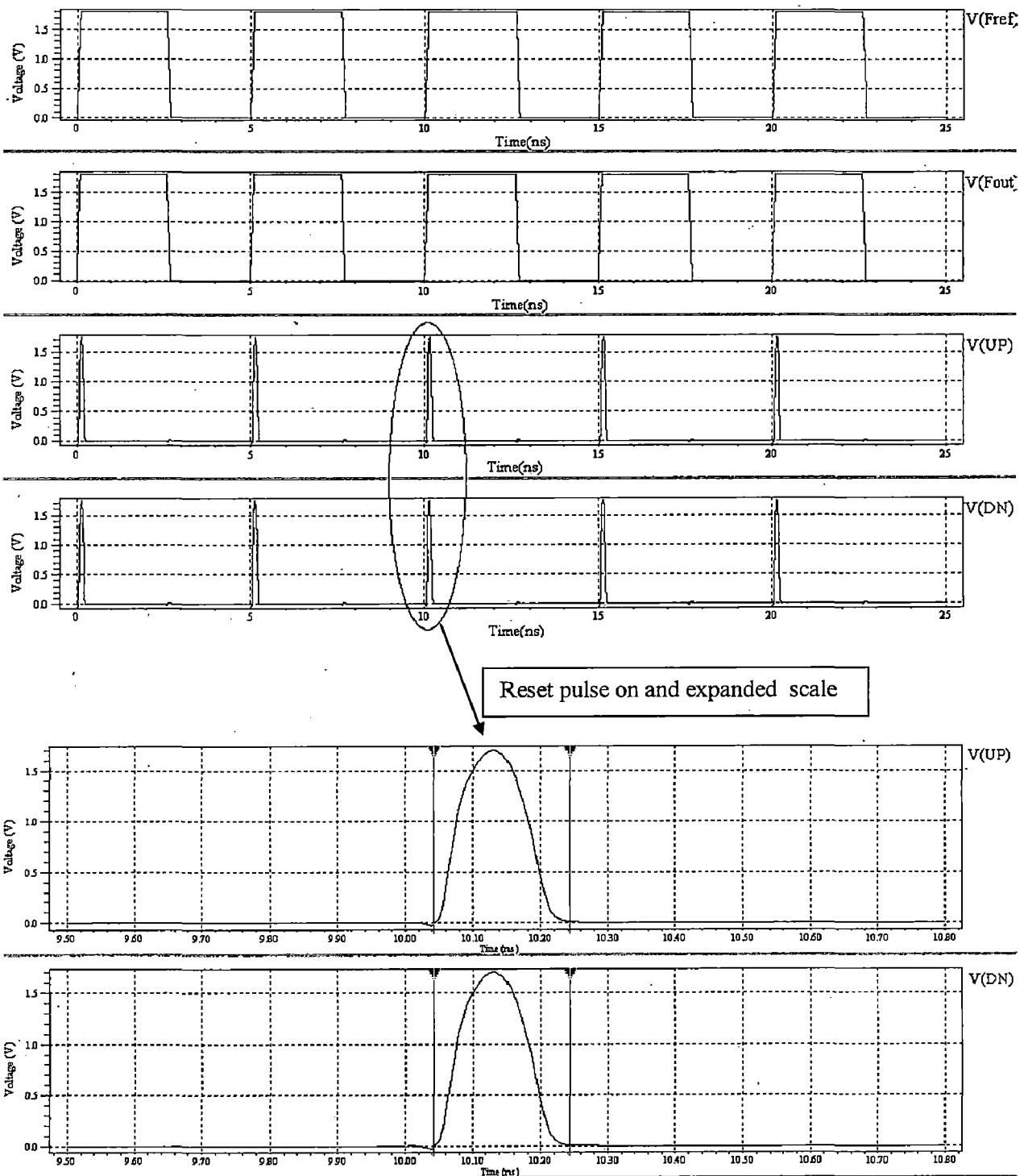


Fig.(4.7) Simulated waveforms for F_{ref} in phase with F_{out}

4. Phase Characteristics

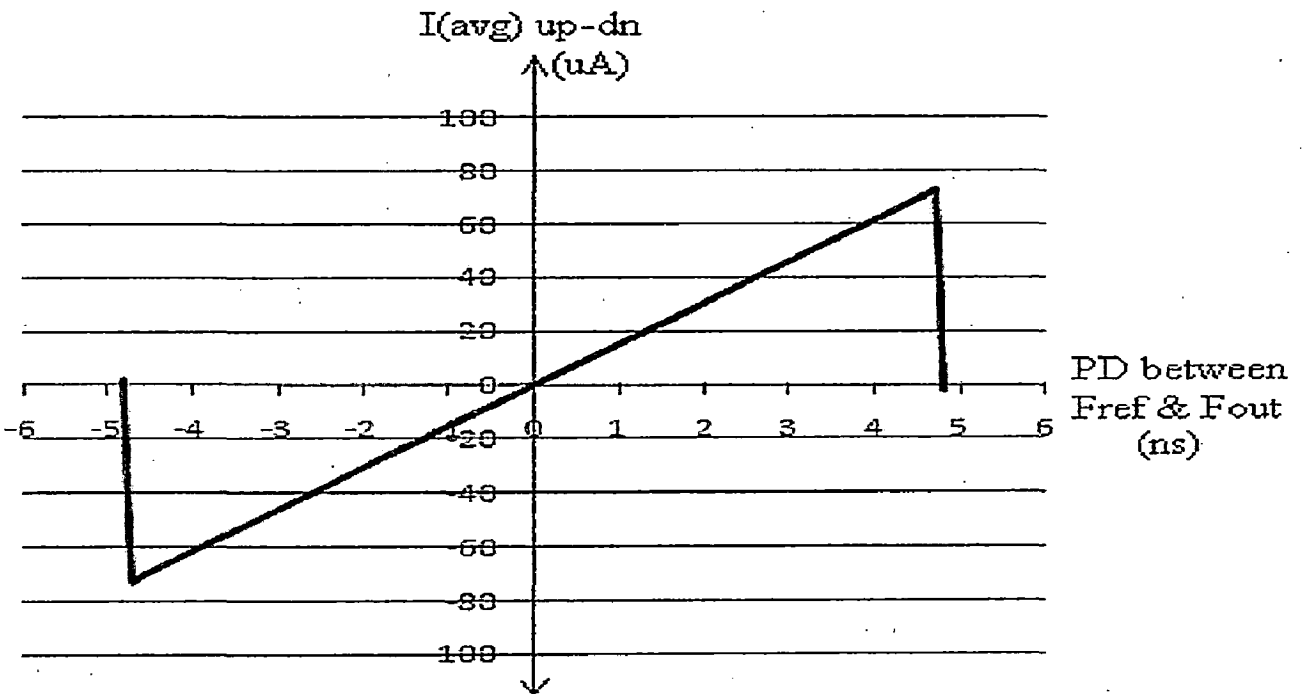


Fig. (4.8) Phase characteristics of designed phase detector

5. Summary of Designed PD

Table (4.1) Summary of designed phase detector

Technology	180 nm technology node
Power supply(Vdd)	1.8 V
No. of Transistors	16
Reset pulse width	0.2 ns
Maximum frequency of operation	2.5 GHz
Average Power Consumption	0.8 mw
Linearity range	-2π to $+2\pi$
Dead Zone	≈ 0

4.2 Design of Charge Pump (CP)

The charge pump is the second unit of DLL. It is responsible for generating necessary control voltage to the VCDL unit. It utilizes the UP and DN signal provided by the phase detector, in order to pump in or pump out the charge from the loop filter capacitor. When DLL is in locked state, the control voltage is constant with some inherent ripples.

The important design considerations for an effective charge pump circuit are

- There should be minimum peak current mismatch
- The charge sharing phenomenon should be avoided
- The effect caused by channel charge injection and clock feed through should be minimized
- Output voltage range provided should be large

Minimum peak current mismatch ensures minimum static phase error between the two input signals (F_{ref} and F_{out}) in locked state. The reason for this is as follows. In locked state the phase detector generates equal UP & DN pulses and the control voltage (V_{ctrl}) is supposed to be constant. This means that quantity of charge Q_{charge} and the one of discharge $Q_{discharge}$ must be equal and given by equ.(4.1)

$$Q_{charge} = I_{UP} * T_{UP} = Q_{discharge} = I_{DN} * T_{DN} \quad (4.1)$$

where T_{UP} and T_{DN} are duration of UP and DN pulses in locked state, or charging and discharging time in one cycle. If I_{UP} and I_{DN} are different, then there has to be some difference between the duration of UP and DN pulse in order to satisfy equ.(4.1) : The DLL, thus creates a static phase difference between its inputs in order to maintain the difference between the duration of UP and DN pulse in accordance with equ.(4.1).

The charge sharing, charge injection and clock feed through phenomenon results in periodic ripples and sudden jumps on the control voltage line. This results in generation of spurious tones and jitters at the output. This becomes critical when DLL is used as a frequency multiplier in a local oscillator of a transceiver [20]. These non ideal effects can be reduced by suitable circuit level manipulations.

The output voltage range provided by charge pump should as large a possible. This helps to keep the VCDL gain to a lower value (for a given operating frequency range), which further helps to decrease the sensitivity of VCDL to sudden variations on control voltage line.

4.2.1 Conventional Charge Pump

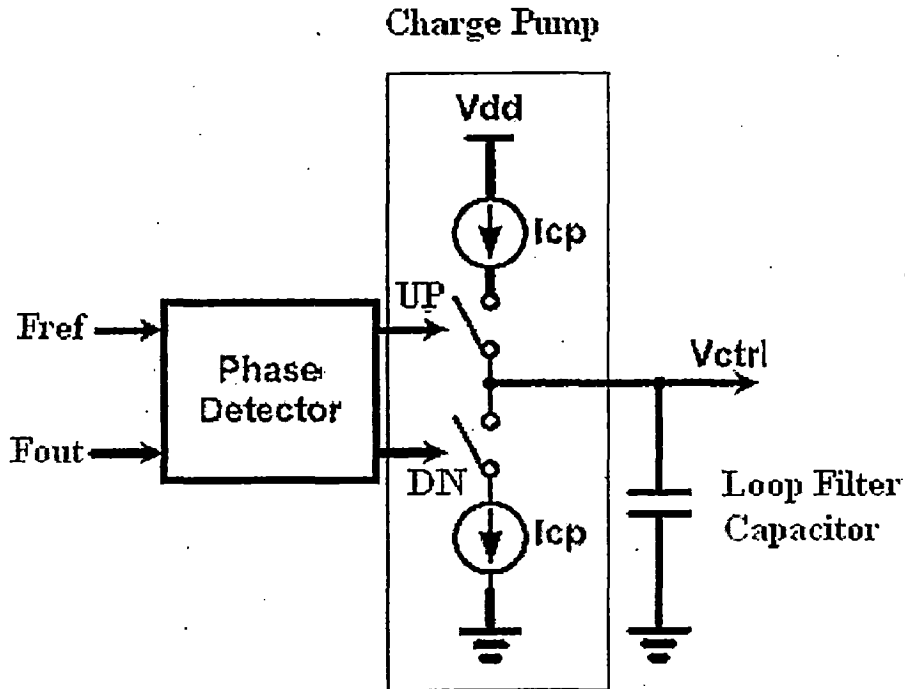


Fig.(4.9) Conceptual circuit of charge pump

Fig .(4.9) shows the conceptual circuit of the charge pump. It consists of two switched current sources. The current sources are switched by UP/DN pulses from the phase detector. When there is an UP pulse the charge pump pumps in charge to the loop filter capacitor. Similarly, when there is a DN pulse the charge pump pumps out the charge from the loop filter capacitor. This results in a staircase waveform during the transient period. When the DLL is locked, the output voltage remains constant.

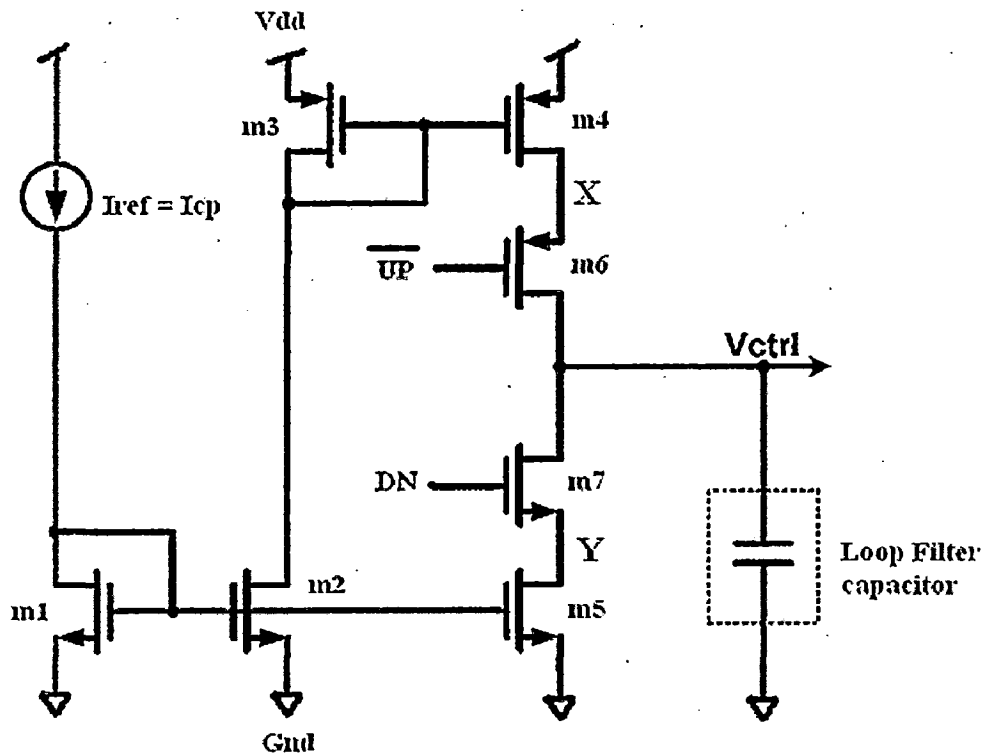


Fig.(4.10) Conventional Charge pump

The conventional charge pump is based on the conceptual circuit of fig.(4.9) and is shown in fig. (4.10). The transistors m6 and m7 acts as switch and transistors m4 and m5 act as a constant current source. The desired value of current is mirrored from I_{ref} via current mirror configuration formed by m1,m2,m5 and m3,m4.

However, the circuit of fig. (4.10) shows many non ideal effects like charge sharing, clock feed through and channel charge injection. These phenomenons tend to produce ripples on control voltage line, which is transformed in phase noise and spurious tones in VCDL output. The charge sharing is due to balancing of charge between node X, node Y and output voltage Vctrl, when the equal duration UP/DN pulse arrives (during locked state of DLL). Channel charge injection occurs due to flow of part of the channel charge from the channel to the output load capacitor (loop filter capacitor), at the drain terminal. This happens when the MOS switch, in on condition operating in triode region, is suddenly turned off. The magnitude of voltage error is given by [26]:

$$\Delta V = \frac{k Q_{ch}}{C} = \frac{k WL C_{ox} (V_{gs} - V_t)}{C} \quad (4.2)$$

Where k is the fraction of channel charge moving to drain, Q_{ch} is the channel charge, C_{ox} is the oxide capacitance, V_{gs} is the gate to source voltage, V_t is the threshold voltage and C is the output capacitance (loop filter capacitance)

Clock feed through results due to non ideal behavior of MOS switches, which presents a parasitic gate to drain capacitance. These error causes sudden jumps on the control voltage line. The magnitude of the error caused by clock feed through is given by [26]:

$$\Delta V = \frac{(V_{dd} - V_{ss}) C_{par}}{C + C_{par}} \quad (4.3)$$

Where V_{dd} and V_{ss} are the high and low values of UP/DN signal, C_{par} is the parasitic gate to drain capacitance. In order to minimize the error due to clock feed through, the circuit should be designed carefully enough such that the parasitic capacitance C_{par} is made to be a small fraction of the loop filter capacitance C .

There are some circuit level techniques to suppress the effects due to above mentioned non-idealities. For eg. in order to suppress the effects due to charge sharing an OP AMP is connected in unity gain configuration [27] and to suppress the effect due to channel charge injection and clock feed through 'dummy switches' are connected, which creates opposite effect to the one created by UP/DN signal, hence minimize sudden jumps [28].

Chang and Kuo's model [29] provides one of the best solution for charge pump design. The circuit eliminates all the non ideal effects completely by isolating the output node from the switching transistors and thus making the output is free from sudden jumps. At 180 nm technology node this circuit shows significant mismatch in pump up and pump down currents. This is due to Channel Length Modulation (CLM) effects which becomes prominent at short channel lengths. The current mismatch further results in static phase error between the inputs when DLL is in locked state.

4.2.2 Charge Pump Incorporated in Proposed DLL

Fig.(4.11) shows the block diagram of charge pump incorporated in proposed DLL. It consists of

- Pump up circuit
- Pump down circuit
- 1:1 current convertor

The circuit is designed for a pump up and pump down current of 80uA. In order to maximize the output voltage range a power supply of 3.3 V is used.

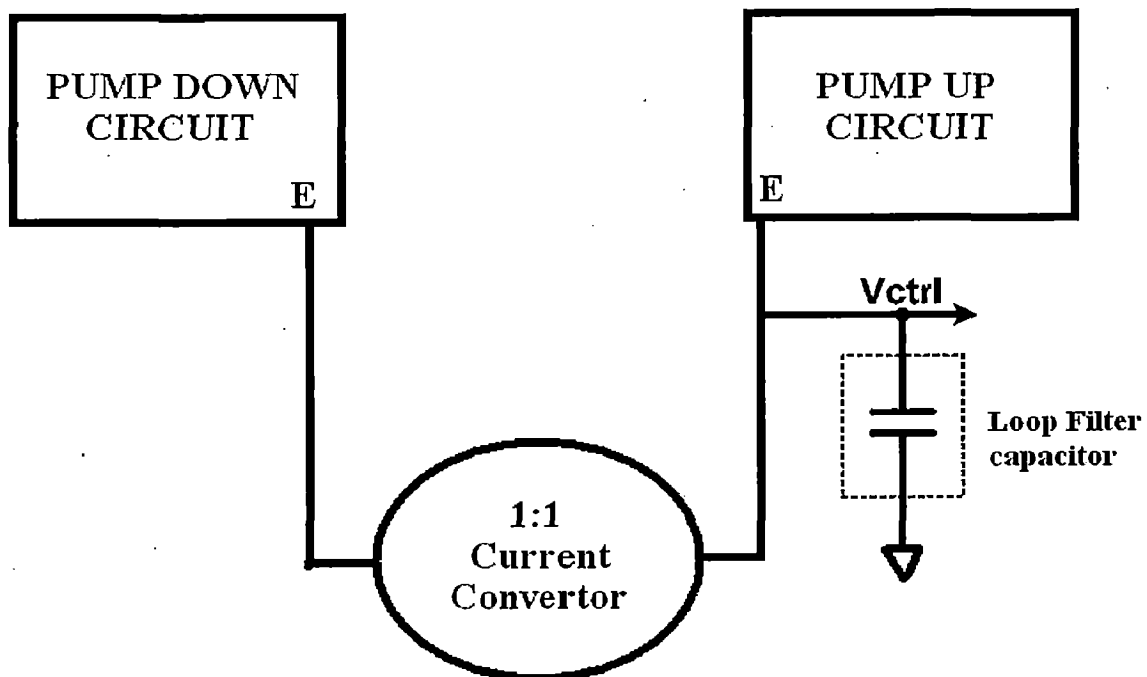


Fig. (4.11) Block diagram of charge pump incorporated in proposed DLL

The pump up and pump down circuit are symmetric. The working can be explained from pump up circuit shown in Fig.(4.12). It uses only NMOS devices as switches and the output node is well isolated from the switching transistors and thus avoiding various non idealities mentioned before.

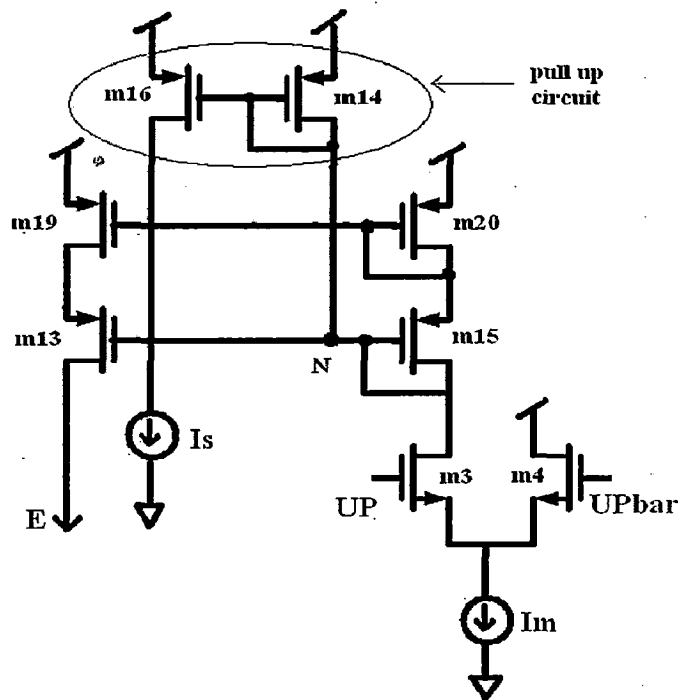


Fig.(4.12) Pump up circuit

It consists of a differential input pair, current mirror load, bias current sources I_m and I_s and finally a pull up current mirror. When UP is high, the bias current I_m is steered through m3. The difference between I_m and I_s flows through m15 and is mirrored to m13 and finally to node E. The current mirror load comprising of m15, m13, m19 and m20 forms a cascode configuration. This configuration increases the output impedance and hence suppresses the CLM effects that become prominent at short channel lengths. Due to this the mismatch between pump up and pump down currents is significantly reduced.

When UP goes low, the current through m3 and hence through m13 start to become zero. To make this operation instantaneous a pull-up circuit comprising of m16, m14 and I_s is inserted. If there is no pull up circuit, there will be a long time constant to charge up the node N and therefore m15 and hence m13 will take time to switch off when UP goes low. Thus with the pull up circuit, when UP goes low, m16 mirrors I_s to m14 and m14 pulls up the gate of m15 (node N) to V_{dd} so that m15 and hence m13 turns off within short time. A low voltage wide swing cascode current mirror is used for 1:1 current converter as it provides larger output swing. The full schematic of the charge pump circuit incorporated in proposed DLL is shown in Fig. (4.13).

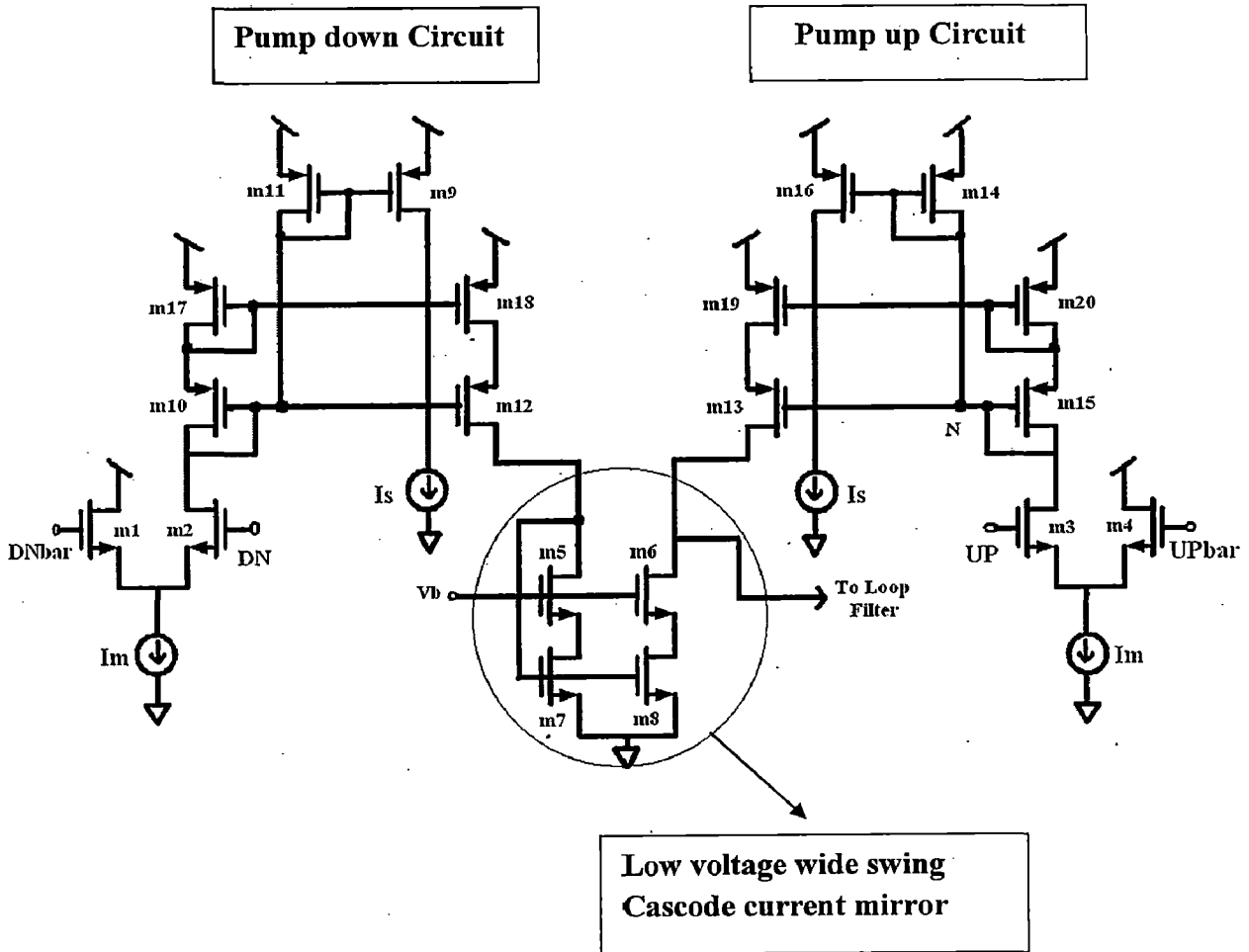


Fig. (4.13) Complete charge pump incorporated in proposed DLL

Design of Low Voltage Cascode Current Mirror [6]

A low voltage cascode current mirror is used for 1:1 current convertor. Its schematic diagram is shown in fig. (4.14).

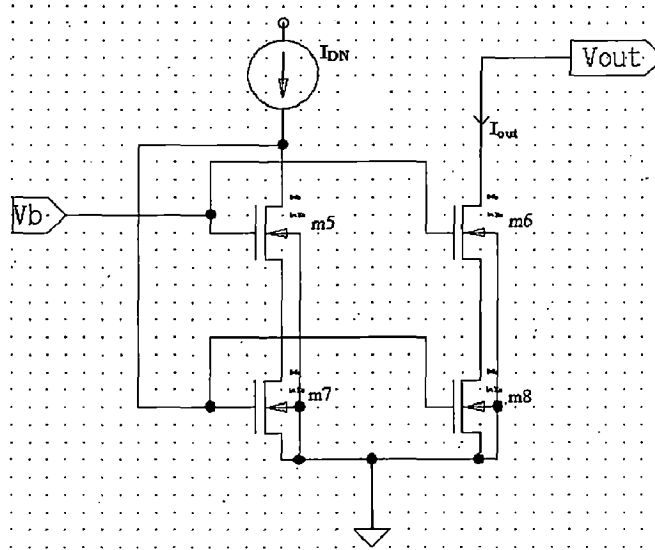


Fig.(4.14) Schematic diagram of low voltage wide swing cascode current mirror

It mirrors the pump-down current to discharge node V_{out} . Using the structure, we can accurately mirror the pump-down current. The reason for including M5 is to decrease the drain-source voltage and lower the channel-length modulation effect of M7 so that it is matched to the drain-source voltage of M8. Therefore, the output current I_{OUT} is more accurately match the input current I_{DN} . Also, it provides wide output swings as compared to other cascode structures. To design this current mirror, it is important not to make M7-M8 go into triode region. To achieve this V_b is chosen such that

$$V_{GS5} + (V_{GS7} - V_{TH7}) \leq V_b \leq V_{GS7} + V_{TH5}$$

$$\text{A solution exists if } V_{GS5} - V_{TH5} \leq V_{TH7} \quad (4.4)$$

We choose an appropriate value of W/L of all four transistors to be 8/0.35 and calculate the value V_b for minimum headroom and fulfilling the necessary condition .

$$I_{DN} = 80\mu A$$

$$I_{DN} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 \quad (\text{neglecting CLM}) \quad (4.5)$$

$$\mu_n C_{ox} = 8.87 \times 10^{-5} \text{ A/V}^2 \quad (\text{model parameter})$$

$$V_{TH} = 0.4 \text{ V}$$

substituting values in equ. (4.5) we get , $V_{GS}=0.68 \text{ V}$

since W/L of all transistors are chosen to be equal ,their overdrive voltages ($V_{GS}- V_{TH}$) are also equal. With the value of $V_{GS} =0.68 \text{ V}$ condition of inequality (4.4) is satisfied

$$(0.68-0.4=0.28 \leq 0.4)$$

To have minimum voltage head room at the output

$$V_b = V_{GS5} + (V_{GS7} - V_{TH7})$$

Since $V_{GS5} =V_{GS7} =V_{GS} =0.68\text{V}$, $V_b = 0.96\text{V} \cong 1 \text{ V}$

The SPICE simulated output of wide swing current mirror for pulsed input of $80\mu\text{A}$ with designed parameter values is shown in fig.(4.15) from which we see that output current exactly follows the input current with negligible delay.

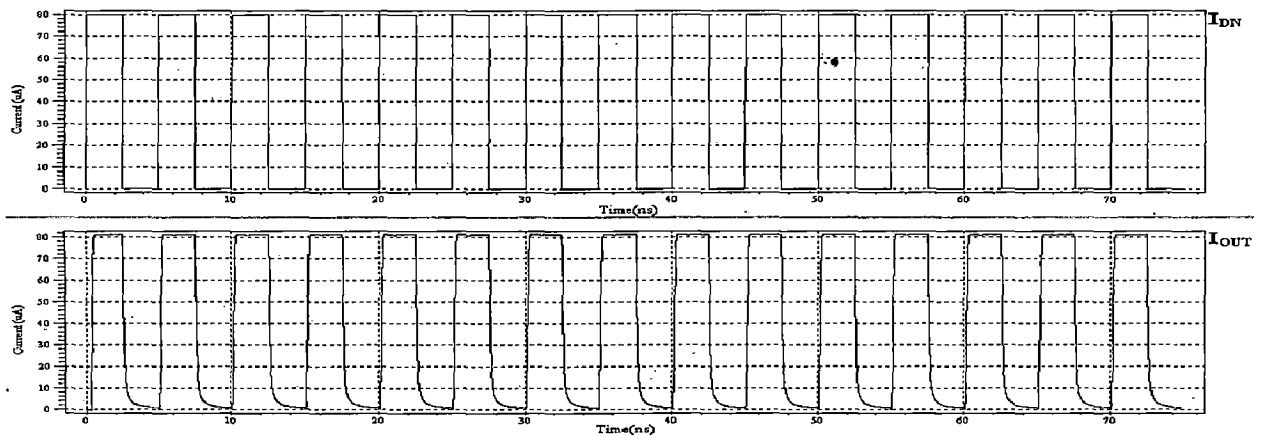


Fig.(4.15) Simulated waveforms of low voltage wide swing cascode current mirror

4.2.3 Simulation output and results of incorporated charge pump

1. Output voltage range

Lower limit

The lower limit of the output voltage is set by low voltage wide swing cascode current mirror shown in fig (4.14). Since V_b is 1V , the minimum value the output can go, in order to maintain $M6$ and $M8$ in saturation is given by

$$V_{out} \geq V_b - V_{TH}$$

Or $V_{out} \geq 1-0.4$

Or $V_{out} \geq 0.6V$

Upper limit

The upper limit of output voltage is set by node N of fig. (4.16) of the pump up circuit. The output voltage should not cross this upper limit as it will drive the cascode current mirror transistors (m13 and m19) out of the saturation region. The spice simulated plot of node N during pump up operation is shown in fig. (4.16).

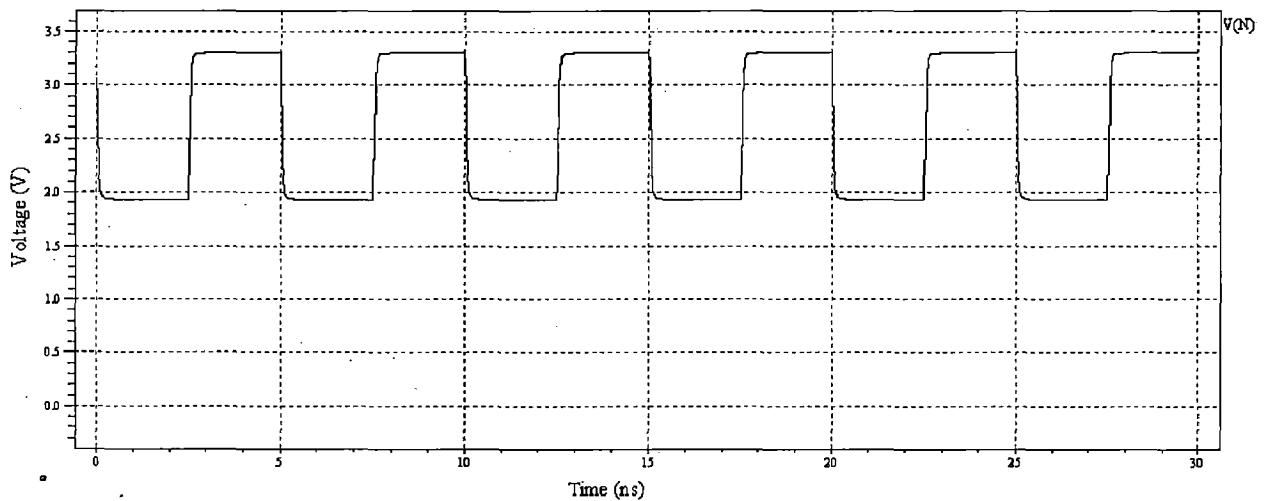


Fig.(4.16) Simulated plot for voltage at node N

In order to ensure that cascode transistors work in saturation region

$$V_{out} \leq V_{Nmin} - V_{TH}$$

$$V_{Nmin} = 1.9V \text{ (from simulated plot)}$$

$$V_{TH} = -0.42 \text{ (model parameter value)}$$

Therefore,

$$V_{out} \leq 1.9 + 0.42 \text{ or}$$

$$V_{out} \leq 2.32 V$$

Therefore output voltage range for proper operation is given by

$$0.6 \leq V_{out} \leq 2.32 V$$

2. Pump up and Pump down current

Pump up current

The simulated plot of pump up current during pump up operation is shown in fig. (4.17)

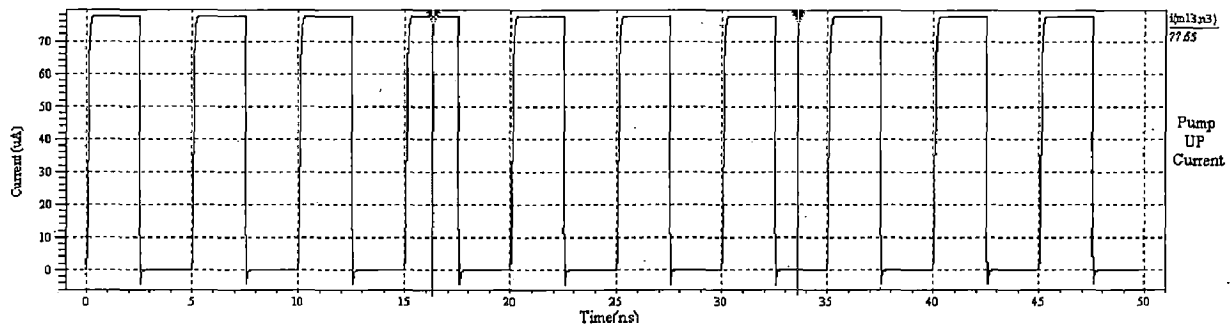


Fig.(4.17) Simulated plot of pump up current

Pump down current

The simulated plot of pump down current during pump down operation is shown in fig. (4.18)

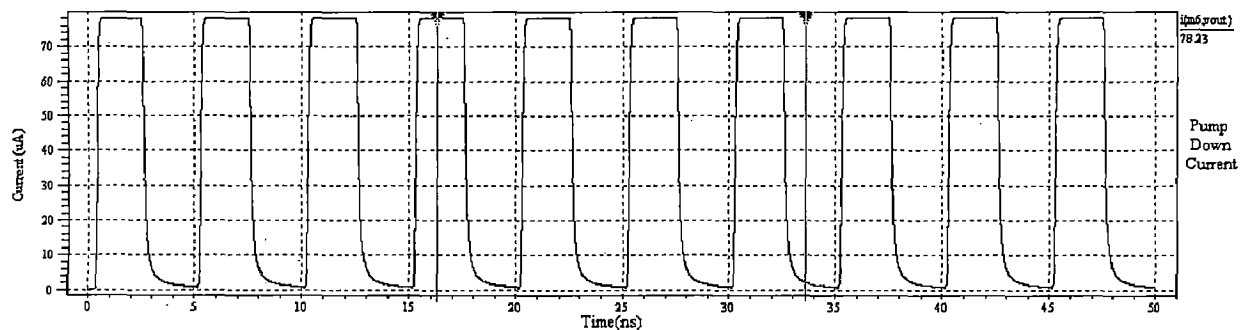


Fig.(4.18) Simulated plot of pump up current

Peak Current mismatch

Peak Pump up current = 77.65uA

Peak Pump down current = 78.23uA

$$\% \text{ Peak Current Mismatch} = \frac{78.23 - 77.65}{78.23} \times 100 = 0.74$$

3. Pump up and pump down operation

To verify the pump up and pump down operation, a test capacitor of 1 pf was connected at the output node. External UP/DN pulses were applied with a frequency of 200 MHz and 50% duty cycle. The simulation outputs of the operation are as follows.

Pump up operation

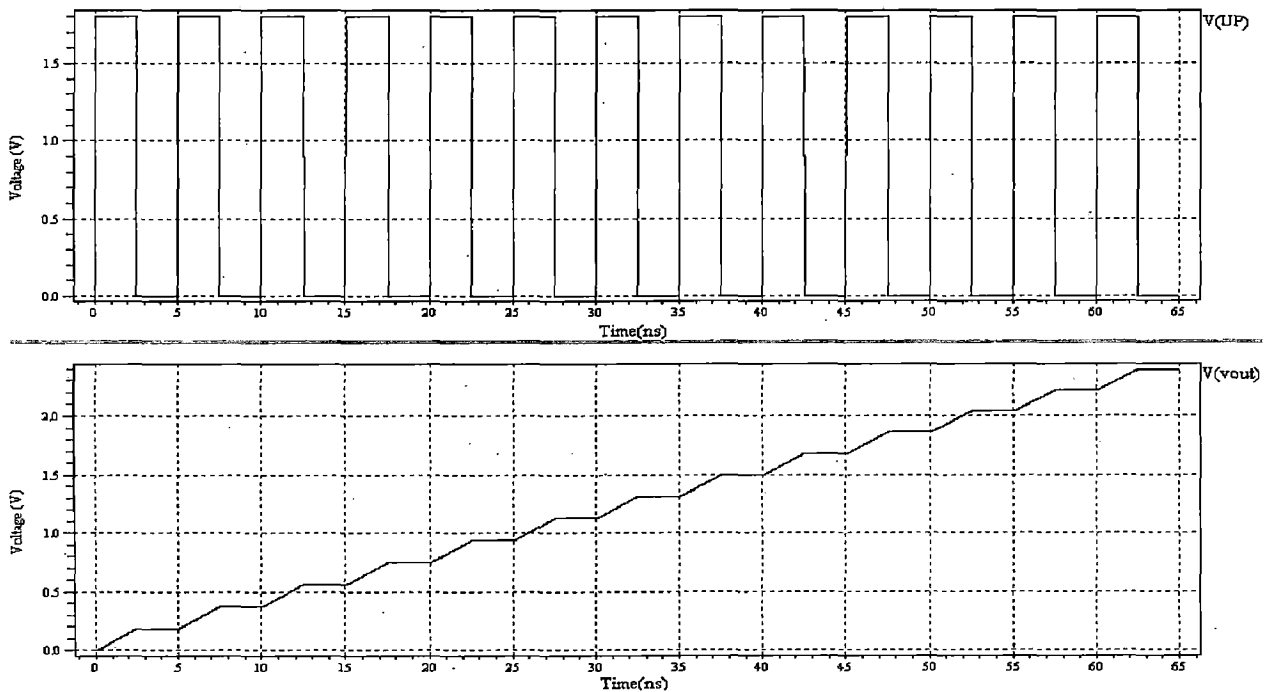


Fig.(4.19) Simulated plot of pump up operation

As seen from fig. (4.19), when UP is high, the test capacitor gets charged by a constant current source ($\approx 80 \mu\text{A}$), thereby generating an upward ramp during charging period. When UP goes low the test capacitor holds the charge and maintains a constant output voltage. Thus, we see a staircase like waveform. The output will rise steadily till 2.32 V, when the cascode current mirror transistor m13 moves in linear region, resulting in decrease in charging current and hence decrease the output voltage increment. Also, it can be seen that there are no sudden jumps on the output voltage, thus the effects of clock feed through, channel charge injection and charge sharing are eliminated.

Pump down operation

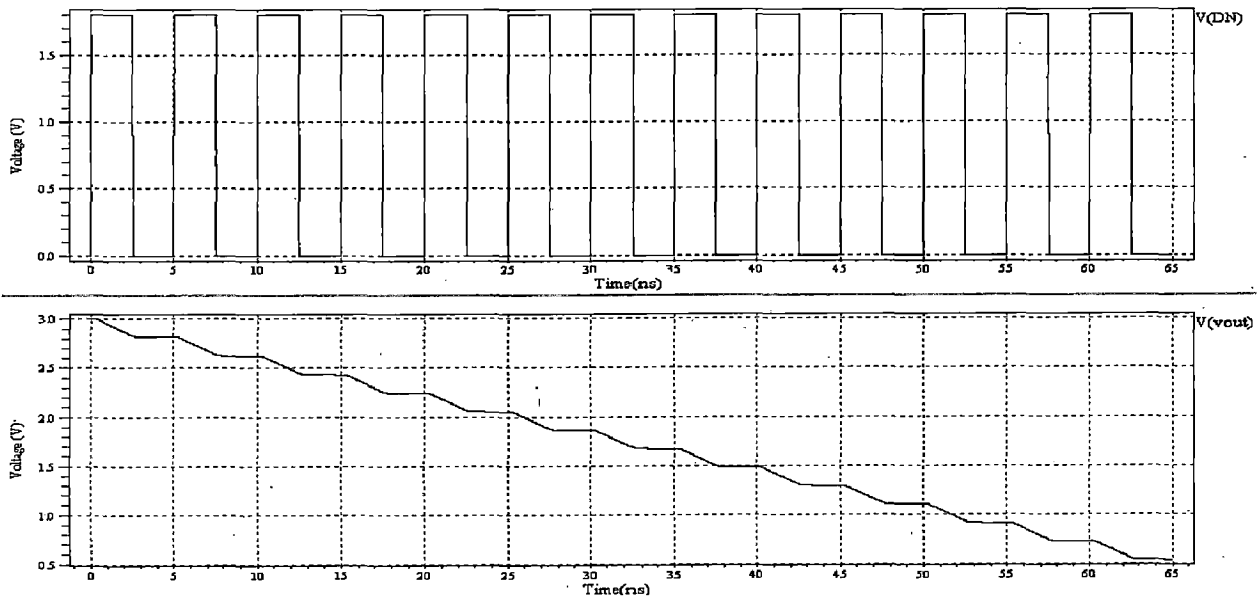


Fig.(4.20) Simulated plot of pump down operation

As seen from fig.(4.20), the test capacitor generates a downward ramp when DN is high (discharging period), and holds the charge when DN is low. Thus, we see a staircase like waveform. The output will fall steadily till 0.6 V, when the low voltage wide swing cascode current mirror transistor m6 moves in linear region, resulting in decrease in discharging current and hence decrease the output voltage decrement.

4. Summary of designed Charge Pump

Table (4.2) Summary of designed charge pump

Technology	180nm technology node
Power supply	3.3V
No. of Transistors	20
Average Power Consumption	0.88 mw
Peak Current (Icp)	≅ 80uA
Peak Current Mismatch	0.74%
Timing Mismatch of Current Pulse	0.15 ns
Output Voltage Range	0.6V- 2.32V

4.3 Design of Voltage Controlled Delay Line (VCDL)

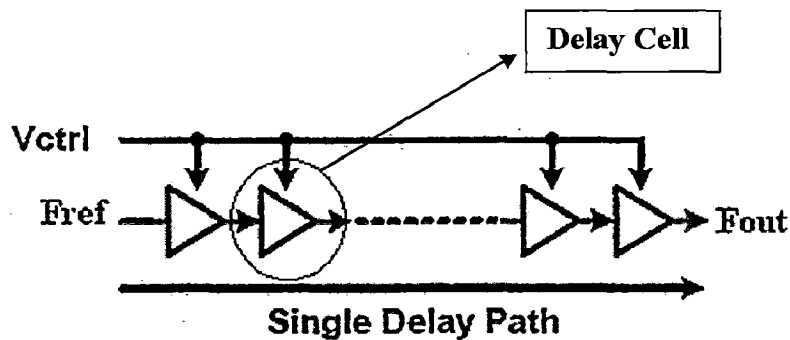


Fig.(4.21) Conceptual diagram of VCDL

The voltage controlled delay line is the final unit of a DLL. Besides PD and CP, the VCDL is one of the most critical blocks, within a DLL. This is because operating frequency range of DLL depends on the minimum and maximum delay provided by the VCDL (equ(2.6)). Improper designing can lead to false locking or harmonic locking as explained in chapter 2. The VCDL provides necessary delay to the input reference signal, depending on the control voltage (V_{ctrl}) applied to it. When the DLL is locked, the output of the VCDL is delayed w.r.t the input by exactly one period. The VCDL consists of several stages of delay cells as shown in Fig.(4.21). Each delay cell provides an equal delay to the signal at its input. There are two most commonly used configuration of a delay cell. They are:

- Differential delay cell
- Single ended delay cell

The most popular differential delay cell is the source coupled differential delay cell with replica bias [13] and the most common single ended delay cell is the delay cell based on current starved inverters [1] which consists of two cascaded CMOS inverters (thus forming buffer) whose charging and discharging currents and hence the delays are controlled by the control voltage (V_{ctrl}). Both the configurations have their respective advantages and disadvantages. The former (source coupled differential delay cell) have better immunity to common mode noise, low sensitivity to supply noise i.e. better dynamic supply noise rejection. Also, it employs an ideal

tail current source which results in good immunity to static supply noise. The later (CMOS inverter delay cell) occupies much smaller area and does not require a level conversion circuit as required by differential delay cell . They also consume much lower power as there is no static power consumption, which is present in differential delay cell. The power consumption in and N-stage CMOS buffer delay line is approximately $2/N$ times less than that of differential source coupled delay line [17]

4.3.1 VCDL Incorporated in Proposed DLL

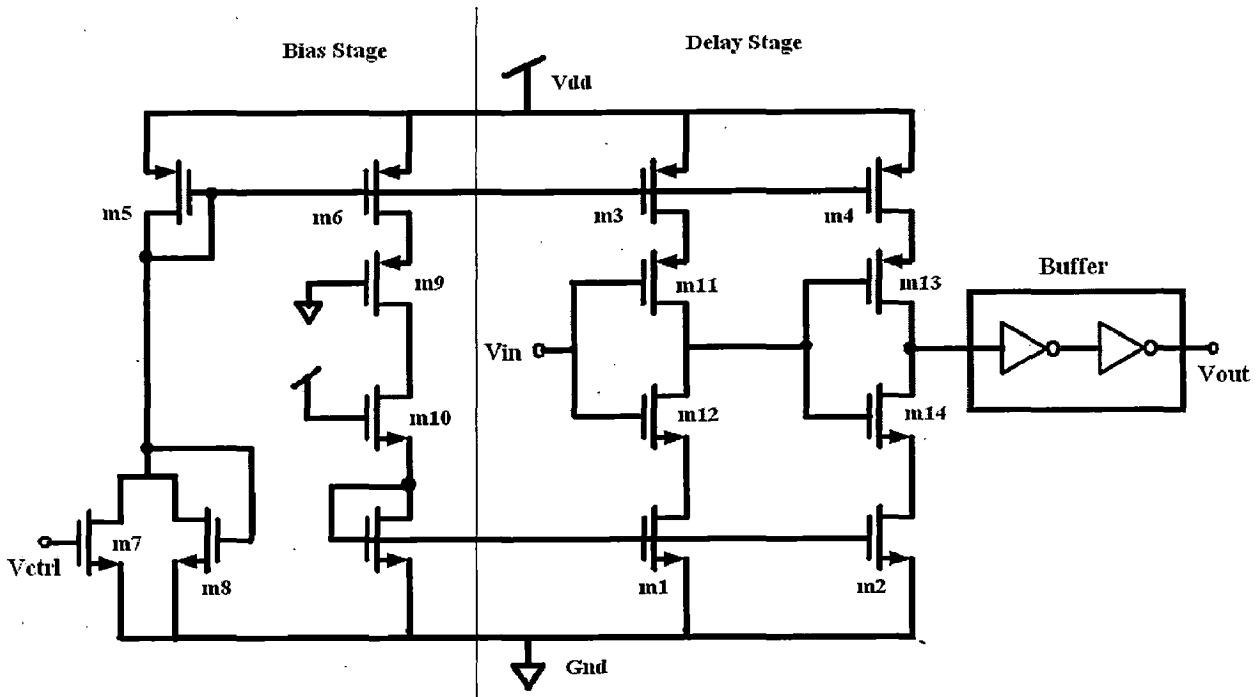


Fig (4.22) Bias stage and delay stage in VCDL of proposed DLL

The VCDL incorporated in proposed DLL is of single ended type delay cell, based on current starved inverter configuration. The VCDL consists of a bias stage and eight stages is delay cell. It works with a power supply of 1.8 V. The delay stage along with the bias stage is shown in Fig. (4.22) and it is similar to the one used by Chen and Lo [30]. The delay cell propagation delay depends of current source (m3,m4) and current sink (m1,m2). The bias stage and V_{ctrl} controls the voltage to current sink and current source and thus controlling the delay. The larger the V_{ctrl} ,

the larger is the current and lesser is the delay. As seen a buffer circuit (comprising of two static inverters) is connected at the output of each delay cell. Without the buffer circuit the output has large rise and fall times. A large rise and fall times are unsuitable for applications like frequency multiplication. Fig. (4.23) shows the simulated output of first two delay stage with and without the buffer. As seen presence of buffer circuit, not only shortens the rise and fall time but also smoothens the output waveform. Measures to avoid false locking as given by equ.(2.2)-(2.5), must be taken care of while fixing the minimum and maximum delay of VCDL. The minimum and maximum delay of VCDL depends on the maximum and minimum value of V_{ctrl} respectively, which in turn depends on the output voltage range provided by the charge pump.

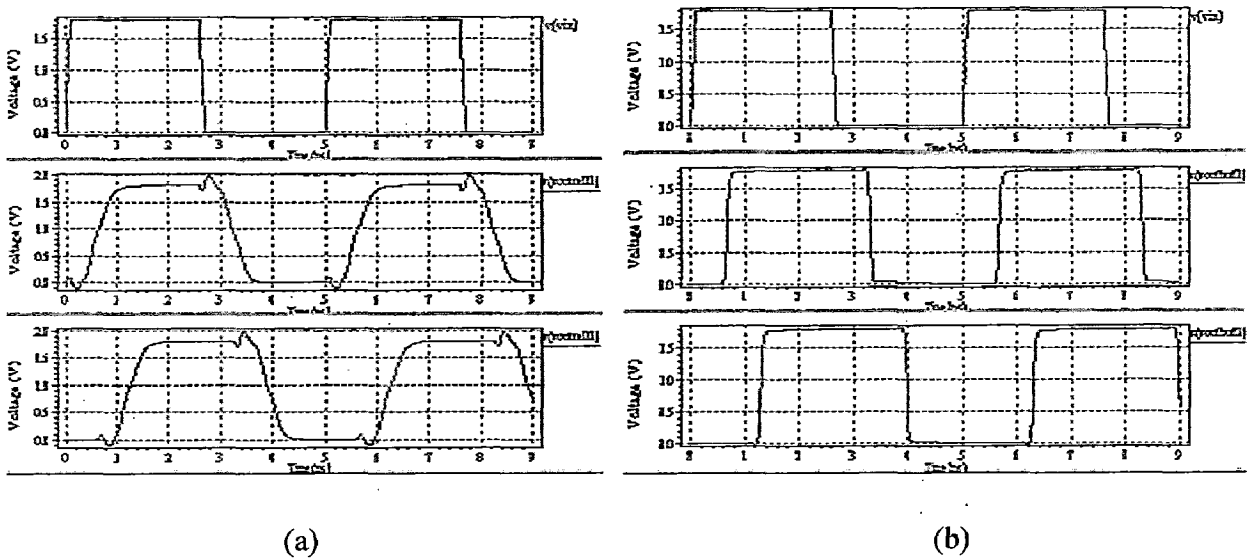


Fig.(4.23) Output of first two delay cells (a) without buffer (b) with buffer

4.3.2 Simulation Output and Results of Incorporated VCDL

1. Transfer characteristics

The simulated transfer characteristics of the incorporated VCDL is shown in fig. (4.24). It is a plot of delay provided by VCDL vs the control voltage V_{ctrl} . We can see that as the control voltage V_{ctrl} decreases, the current decreases and hence the delay increases until it (delay) becomes constant when V_{ctrl} goes below the threshold voltage. When this happens, the diode connected transistor $m8$ provides constant current to the delay stage and hence the delay remains constant.

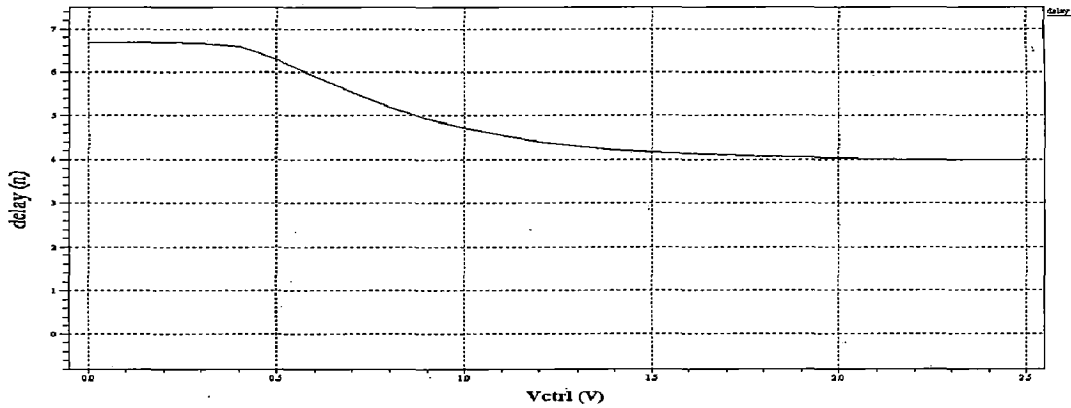


Fig.(4.24) Simulated transfer characteristics of designed VCDL

2. Verification of equ (2.4) and equ (2.5)

In order to prevent harmonic or false locking of DLL, the maximum delay and the minimum delay, of VCDL has got an upper and a lower boundary given by equ(2.4) and equ(2.5). The maximum and minimum delay provided by VCDL depends on the minimum and maximum values of control voltage (V_{ctrl}) respectively. Since output voltage provide by charge pump is 0.6V-2.32V, they correspond to the minimum and maximum value the control voltage can have.

Case 1 Fig. (4.25) gives the simulated output of VCDL for $V_{ctrl}= 0.6$ V (minimum value).

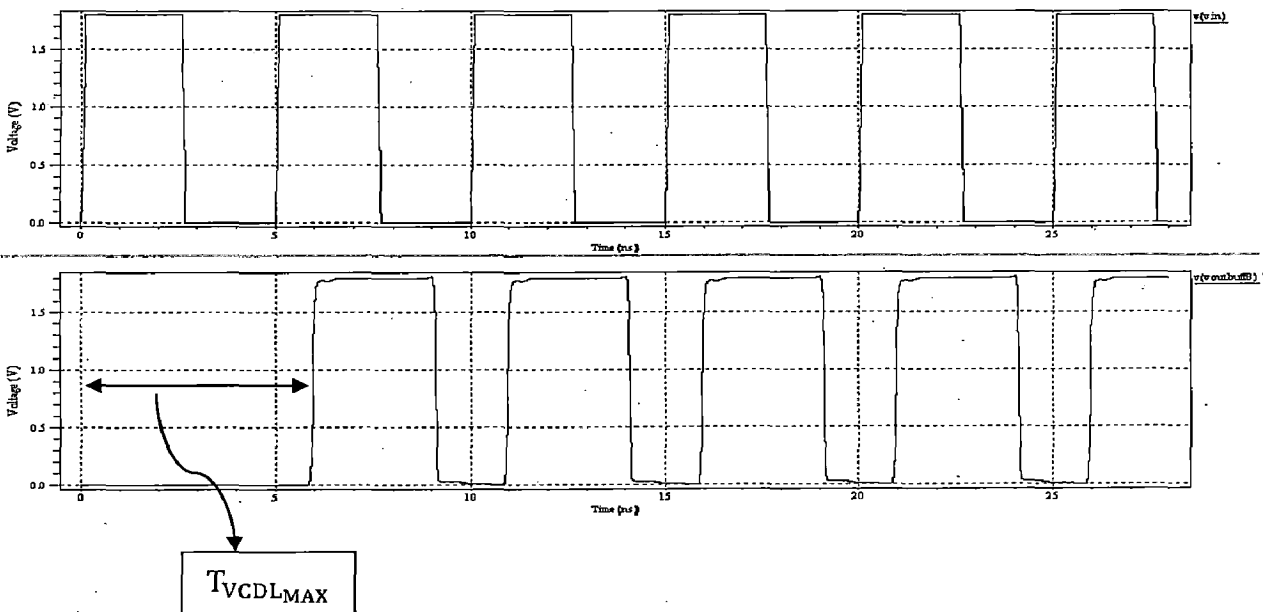


Fig.(4.25) Simulated output of VCDL for $V_{ctrl}= 0.6$ V

As seen the maximum VCDL delay ($T_{VCDL_{MAX}}$) is 5.9 ns. Thus, equ(2.5) is satisfied as

$$T_{ref} < T_{VCDL_{MAX}} < 1.5 T_{ref} \quad \text{where } T_{ref} = 5 \text{ ns (for 200 MHz ref. signal)}$$

Case 2 Fig. (4.26) gives the simulated output of VCDL for $V_{ctrl} = 2.32 \text{ V}$ (maximum value).

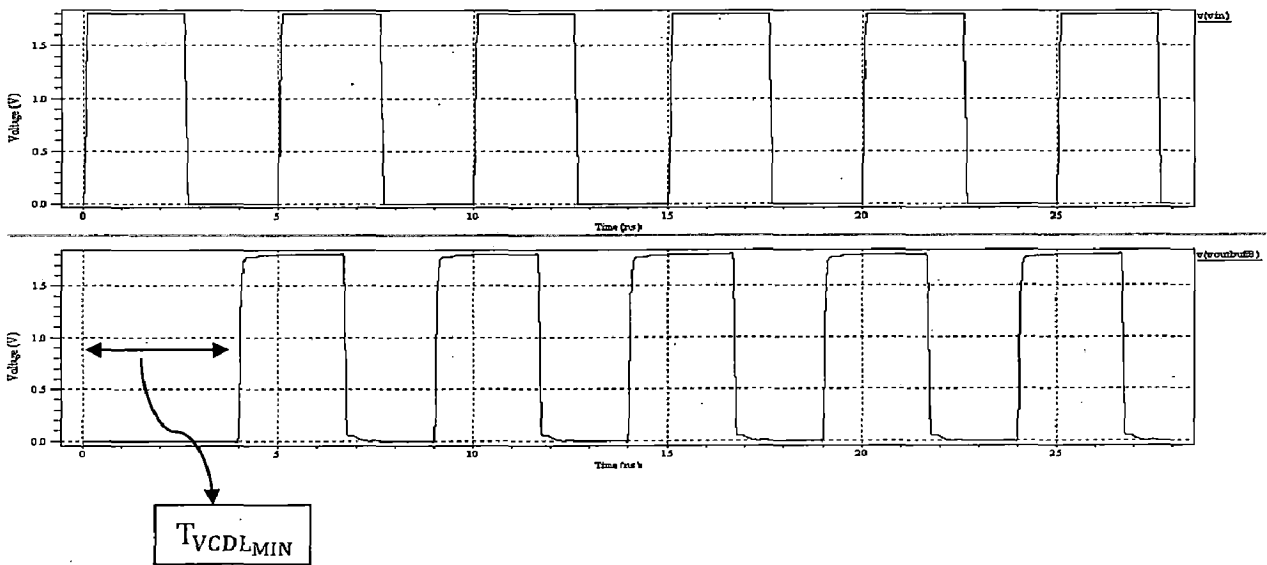


Fig.(4.26) Simulated output of VCDL for $V_{ctrl} = 2.32 \text{ V}$

As seen the minimum VCDL delay ($T_{VCDL_{MIN}}$) is 3.98 ns. Thus, equ(2.4) is satisfied as

$$0.5 T_{ref} < T_{VCDL_{MIN}} < T_{ref}$$

4.4 Loop Filter Capacitor

The value of loop filter capacitor is chosen such that equ. (2.9) is satisfied in order to maintain the stability of DLL. The equation is

$$\frac{\omega_N}{\omega_{ref}} = \frac{I_{CP} \cdot K_{VCDL}}{C} \leq \frac{1}{10}$$

$$\text{Therefore,} \quad C \geq 10 \cdot I_{CP} \cdot K_{VCDL}$$

Since, the transfer characteristics of VCDL shown in fig. (4.24) is non-linear, the value of K_{VCDL} (gain of VCDL) varies at various point. Therefore equ.(2.9) is made to satisfy for maximum value of K_{VCDL} , so that it can also be satisfied for any other value of K_{VCDL} .

In the operating range of DLL (0.6 V- 2.32 V), the maximum slope of curve in fig.(4.24) is seen at $V_{ctrl} = 0.6$ V and delay = 5.9 ns. The slope of the curve (i.e maximum gain of VCDL) at this point is found to be 2.43 ns/V. Substituting this value for K_{VCDL} in equ.(2.9) with $I_{cp} = 80$ μ A, we get:

$$C \geq 10 \times 80 \times 10^{-6} \times 2.43 \times 10^{-9} \text{ Farad}$$

$$\text{Or } C \geq 1.94 \text{ pf}$$

A capacitance value too large will elongate the lock time of the DLL, as it will make loop bandwidth small. A capacitance value too small will result in more ripples on the control voltage line, due to inherent current mismatch. Large ripple results in phase noise and spurious tones at the output. Thus, there exists a trade off.

A value of C as 2.5 pf, is chosen as it gave satisfactory results (through simulations) in terms of lock time and ripples on control voltage line.

4.5 Complete Circuit

All the units of DLL designed, are integrated in accordance to fig. (2.1).The input reference signal is of 200 MHz. The initial voltage of the loop filter capacitor is set to be equal to 2.32 V (maximum value the control voltage can have, as provided by charge pump). Thus, initially the VCDL provides minimum delay $T_{initial} = T_{VCDL_{MIN}}$, and therefore equ. (2.2) is satisfied. Now the DLL begins its operation, and moves towards acquiring lock.

4.5.1 Simulation result and output of complete circuit

The complete integrated circuit was simulated in T-SPICE at 180 nm technology node and with a reference signal of 200 MHz.

1. Locking range and Operating Frequency range of proposed DLL

The locking range and operating frequency range of DLL depends on maximum and minimum delay of VCDL, given by equ(2.6) which is as follows:

$$\text{MAX} (T_{VCDL_{MIN}}, 2/3 \times T_{VCDL_{MAX}}) < T_{ref} < \text{MIN} (T_{VCDL_{MAX}}, 2 \times T_{VCDL_{MIN}})$$

Since $T_{VCDL_{MIN}} = 3.98 \text{ ns}$ and $T_{VCDL_{MAX}} = 5.9 \text{ ns}$, the above equation reduces to :

$$\text{MAX}(3.98, 2/3 \times 5.9) < T_{ref} < \text{MIN}(5.9, 2 \times 3.98)$$

Therefore the locking range of DLL is given by :

$$3.98 \text{ ns} < T_{ref} < 5.9 \text{ ns}$$

The operating frequency range of DLL is given by

$$170 \text{ MHz} < F_{ref} < 252 \text{ MHz}$$

2. Locking Curve

Fig. (4.27) shows the simulated locking curve of the proposed DLL, along with waveforms of F_{ref} and F_{out} . As seen the control voltage (V_{ctrl}) decrease gradually from initial value of 2.32 V, thereby increasing the delay, until the delay becomes exactly equal to one reference period (5 ns). The DLL is then said to have attained lock and F_{out} and F_{ref} are exactly in phase. The locking control voltage is 0.876 V. As seen the lock time of DLL is less than 300ns. The average power consumption of DLL in locked state is 6.46 mw. The static phase error between F_{ref} and F_{out} is as low as 2 ps. The control voltage has a ripple of around 1.3 mv(p-p) as seen from fig.(4.28).

2. Multiphase Clock Generation

Fig.(4.29) shows the multiphase output of all the VCDL delay stages in the proposed DLL. We can see that all the successive waveforms are equally phased, and thus DLL in locked state results in multiphase clock generation. We can also see that, in locked state F_{out} and F_{ref} are exactly in phase. These multi-phased clock outputs can be used for various applications like, digital testing, frequency multiplication etc. as mentioned in chapter 2. The output F_{out} of DLL has a duty cycle variation of 58 %.

3. Summary

Table (4.3) gives the summary of the proposed DLL and table (4.4) gives the performance comparison of proposed DLL with prior similar designs.

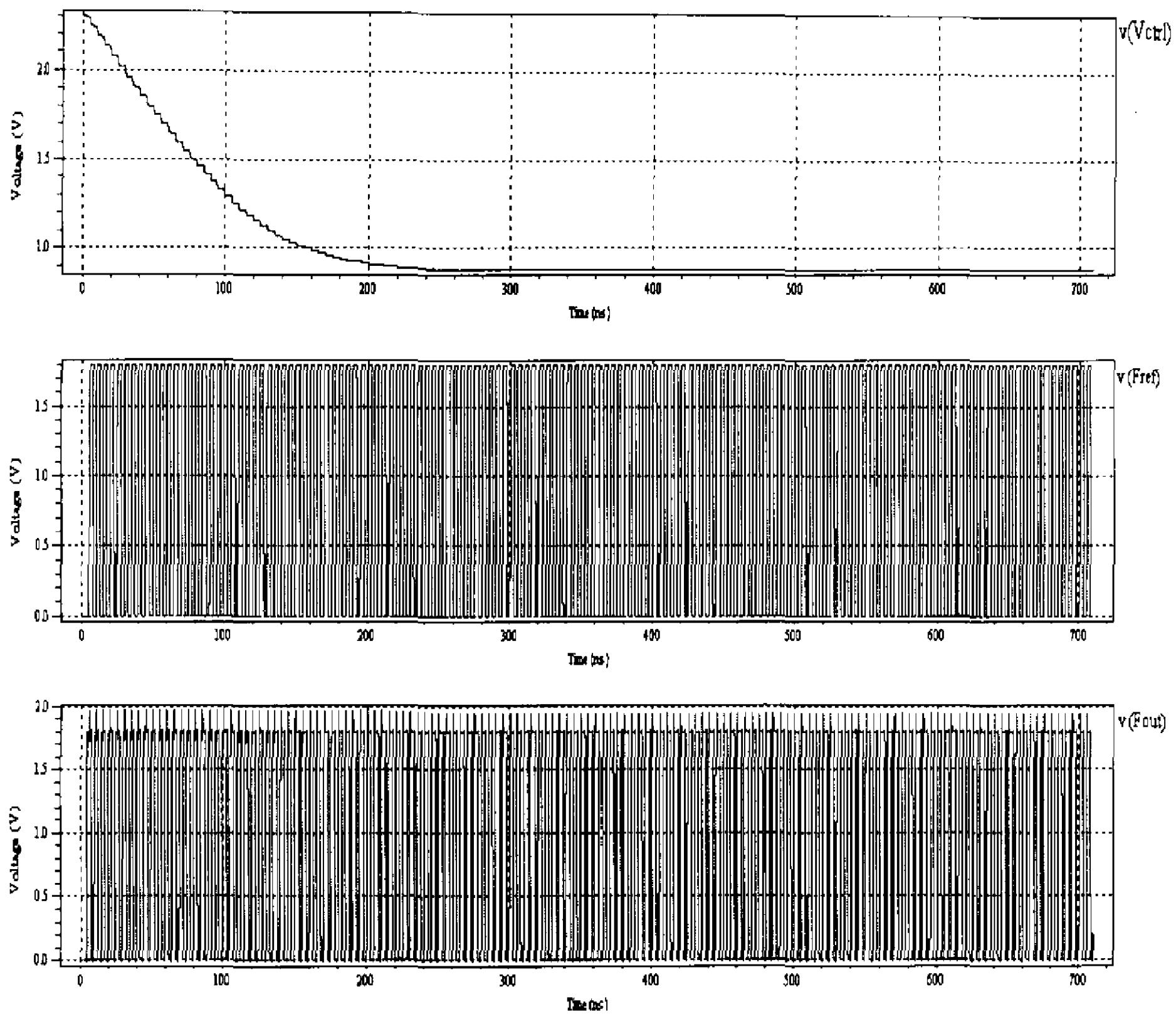


Fig.(4.27) Locking curve of proposed DLL along with Fref and Fout

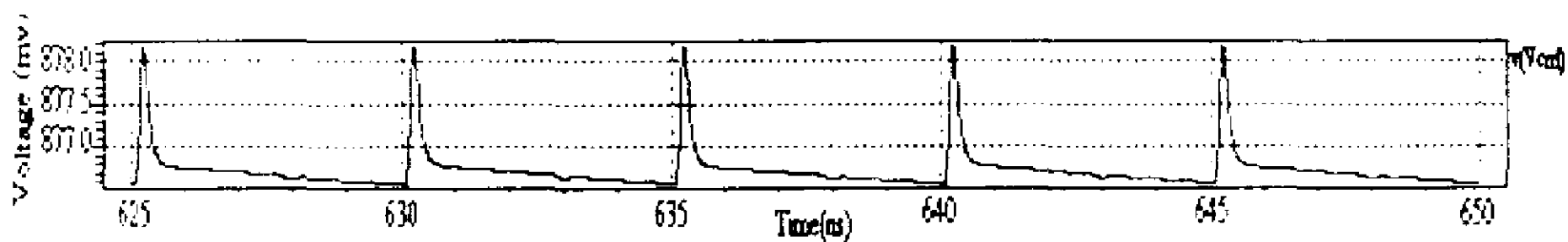


Fig.(4.28) Ripples on control voltage line

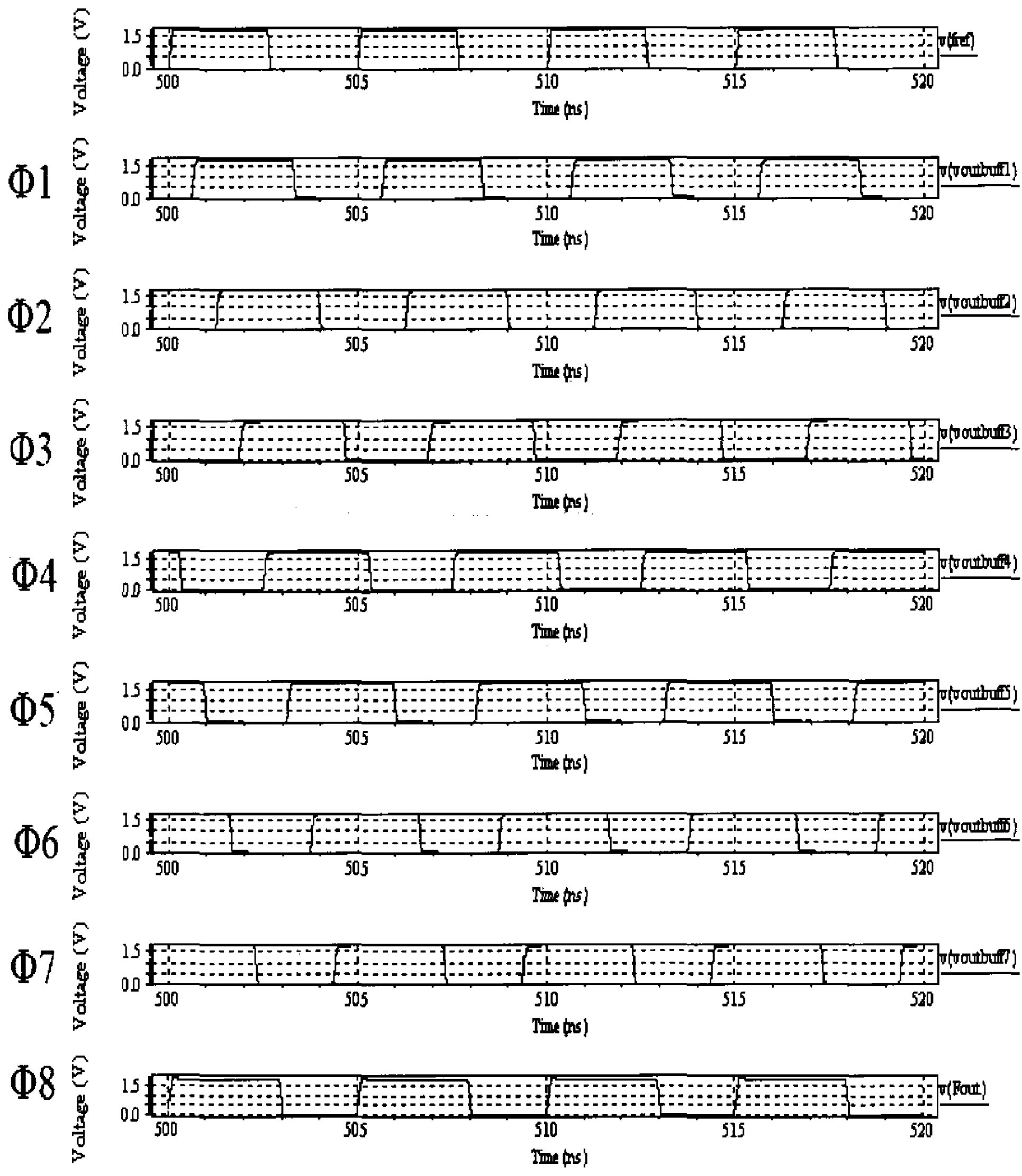


Fig.(4.29) Multiphase clock output of DLL in locked state

Table(4.3) Summary of proposed DLL

Technology node	180nm
Supply Voltage	1.8 V
Average Power Consumption	6.46 mw
Charge Pump Current	≈80 uA
% Current Mismatch	0.74
Input Frequency Range	170 MHz - 252 MHz
Lock Time	< 300 ns
Static Phase Error	≈2ps
Duty Cycle Variation	≈ 58%
Ripples at locked state	1.3 mV(p-p)
Control Voltage in locked state	0.876 V

Table(4.4) Performance comparison of proposed DLL with prior similar designs

	Frequency Range	Process	Power Supply	Power Consumption	Lock Time	Static Phase Error
[7]	62.5 MHz-250 MHz	0.35 um CMOS	3.3 V	12.6 mA @ 250 MHz	NR	< 40 ps
[9]	6 MHz-130 MHz	0.35 um CMOS	3.3 V	132 mw @130 MHz	1130 Clock cycles	NR
[10]	50 MHz-280 MHz	0.25 um CMOS	2.5 V	NA	90 cycles @ 50 MHz 45 cycles @ 280 MHz	38ps@50 MHz 7ps @280 MHz
[11]	32 MHz-320 MHz	0.25 um CMOS	2.5 V	15 mw @320 MHz	22 clock cycles (max)	9.9 ps @200 MHz
[30]	100 MHz-190 MHz	0.35 um CMOS	3.3 V	13.2 mw @100MHz 39.6 mw @ 190 MHz	43 clock cycles	NR
This work	170 MHz-252 MHz	0.18 um CMOS	1.8 V	6.46 mw @200 MHz	50 clock cycles @ 200 MHz	2 ps @ 200 MHz

NR- Not Reported

Chapter 5

Design of DLL based Programmable Frequency/Clock Multiplier

As seen in chapter 3, one of the major application of a Delay Locked Loop is that its multi-phased outputs can be used for frequency multiplication of the reference clock signal. It has several advantages over conventional PLL based frequency multiplier namely better jitter performance, simple system, low power consumption. One of its major disadvantage is that it has poor variable multiplication factor and hence can generate only fixed number of frequencies.

In this chapter we will use the DLL designed in the previous chapter for variable frequency multiplication. The proposed programmable frequency multiplier can generate outputs with multiplication factor of 1X, 2X and 4X. Since the DLL designed (previous chapter) has an operating frequency range of 170 MHz to 252 MHz, the proposed programmable frequency multiplier using DLL can generate frequencies in the range 170 MHz - 252 GHz (multiply by 1), 340 MHz – 504 MHz (multiply by 2), 680 MHz- 1 GHz (multiply by 4), with exactly 50 % duty cycle.

5.1 Principle of Frequency Multiplication

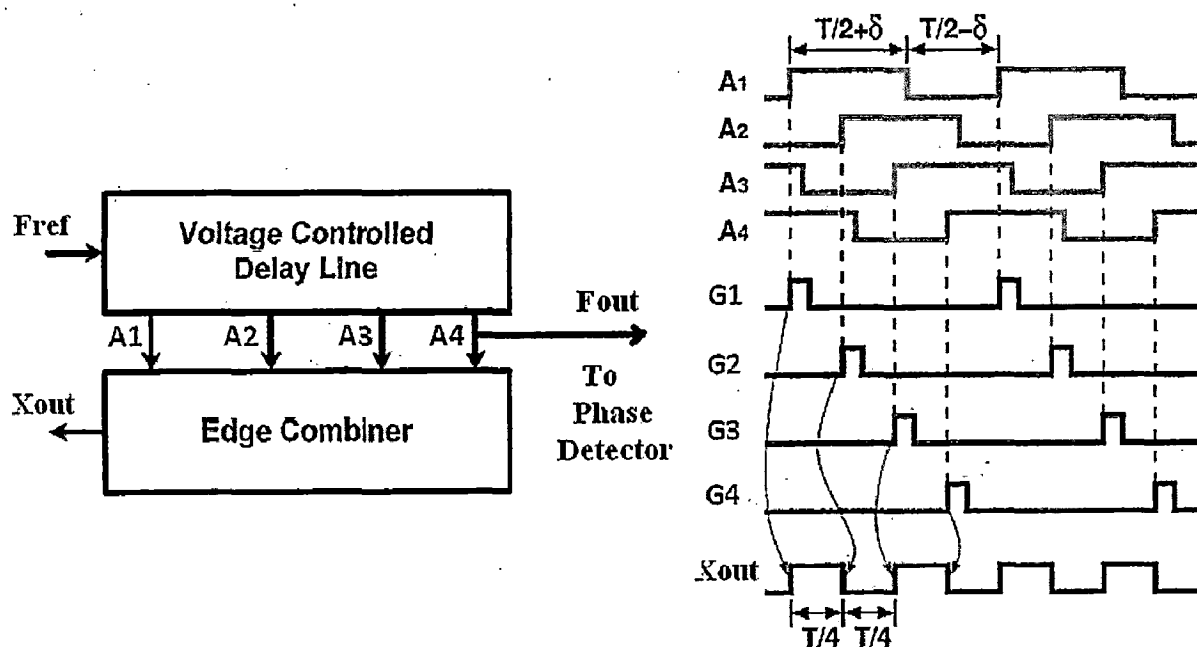


Fig.(5.1) Basic circuit of a frequency multiplier with 4 stage VCDL and its output waveforms

The principle of frequency multiplication by proposed frequency multiplier can be explained with the help of fig. (5.1). As an example let the VCDL of the DLL consists of only four delay stages. Let the output waveforms at these delay stages be A1, A2, A3 and A4. When the DLL is locked all the waveforms will be equally phased within one reference clock period as shown in fig.(5.1). The edges of these four waveforms are used for frequency multiplication.

The edge combiner consists of a circuitry that generates a glitch of short duration (G1, G2, G3, G4) at the rising edge of each output waveform and finally combines these glitches such that G1 and G3 results in rising edge and G2 and G4 results in falling edge of the multiplied output waveform. Thus for a 4 stage VCDL we get an output whose frequency is twice the reference frequency. Even though the outputs of delay stages (A1, A2, A3 and A4) does not have 50 % duty cycle, the multiplied output signal has exactly 50 % duty cycle. This is primarily because the glitches (G1, G2, G3, G4), which are responsible for rising and falling edge of output signal, are equally placed on time axis. In general, for an N stage VCDL (N-even number) the maximum multiplication factor is N/2.

5.2 Architecture of Proposed Programmable Frequency/Clock Multiplier

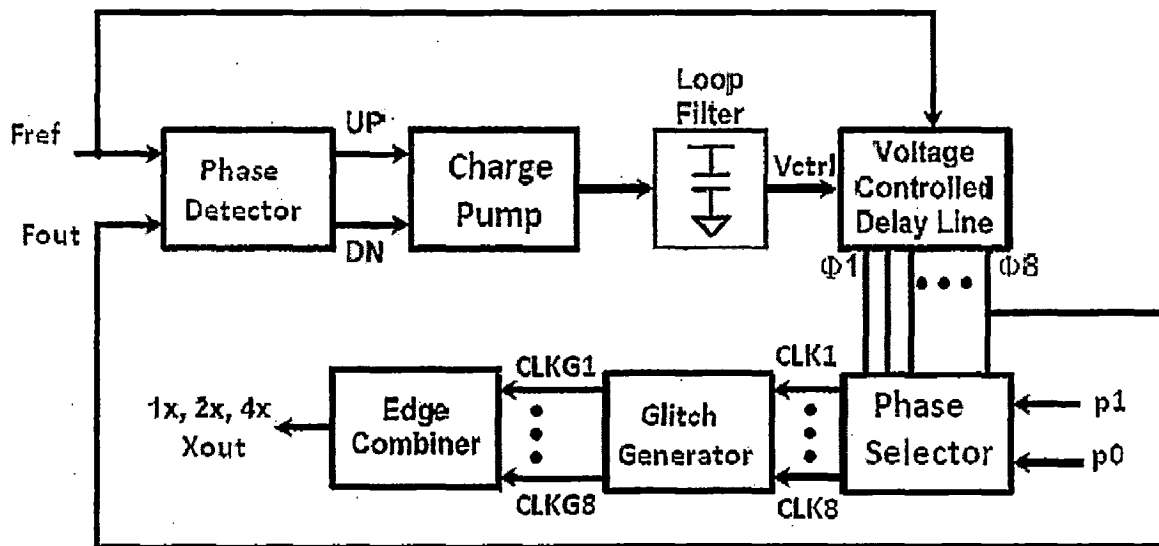


Fig.(5.2) Architecture of proposed programmable frequency/clock multiplier

Fig. (5.2) shows the architecture of the proposed programmable frequency multiplier/clock generator. It consists a DLL stage (comprising of phase detector, charge pump, loop filter and VCDL stage), a phase selector unit, a glitch generator and finally an edge combiner. The VCDL consists of eight delay stages (as designed in previous chapter). There are two external digital signals, known as phase select signals (p1,p0), which are given to the phase selector. Depending on the values of these signals the phase selector selects and sends appropriate phases to the glitch generator and thereby enabling frequency multiplication of the reference signal with multiplication factor of 1X, 2X and 4X. Following are the output frequencies for different values of phase select signals (p1,p0)

When (p1, p0) = (0, 0) $X_{out} = F_{ref}$

When (p1, p0) = (0, 1) $X_{out} = F_{ref} \times 2$ (multiply by two)

When (p1, p0) = (1, 0) $X_{out} = F_{ref} \times 4$ (multiply by four)

Where X_{out} is the frequency of the output signal

When (p1, p0) = (0, 0) the phase selector sends the phases Φ_1 and Φ_5 on two of its output lines (CLK(1...8)) to the glitch generator. The glitch generator generates glitches G1 and G5 on two of its output lines (CLKG(1...8)). The corresponding glitch G1 is used to generate a rising edge of the output signal and glitch G5 is used to generate the falling edge. Thus the output will have same frequency as that of the input.

When (p1, p0) = (0, 1) the phase selector sends the phases Φ_1 , Φ_3 , Φ_5 and Φ_7 to the glitch generator. The glitches G1 and G5 are used to generate a rising edge of the output signal and glitches G3 and G7 are used to generate the falling edge. Thus the output will have twice frequency as that of the input.

When (p1, p0) = (1, 0) the phase selector sends all the phases ($\Phi_1 \dots \Phi_8$) to the glitch generator. The glitches G1, G3, G5 and G7 are used to generate a rising edge of the output signal and glitches G2, G4, G6 and G8 are used to generate the falling edge. Thus the output will have four times the frequency as that of the input.

Fig. (5.3) describes all the above cases.

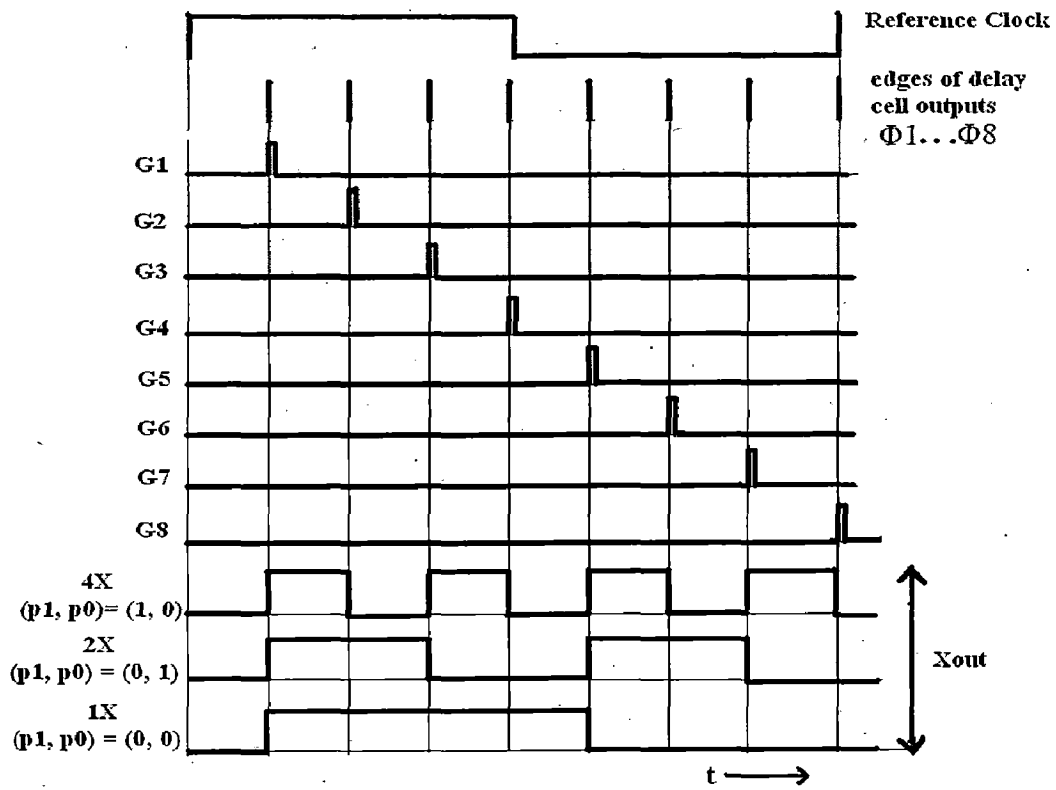


Fig.(5.3) Different cases of frequency multiplication

5.2.1 Glitch generator

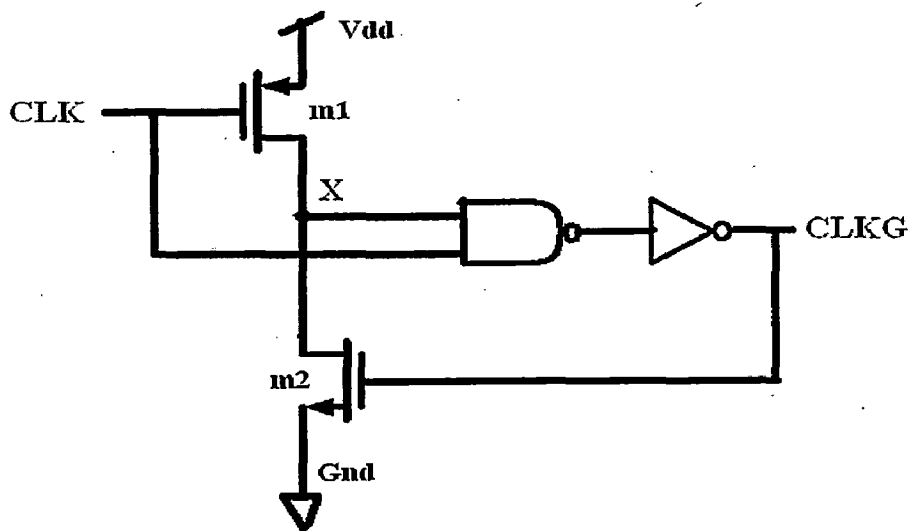


Fig.(5.4) Glitch generator [15]

Fig.(5.4) shows the glitch generator used in the programmable frequency multiplier. It generates a pulse of very short duration at the output when the CLK input undergoes a positive transition. Its working is as follows. Initially when clock $CLK = 0$, node X is pulled up to V_{dd} ($m1$ being a PMOS transistor). Therefore, initially (before rising edge of CLK) one of the inputs to the NAND gate is high (X) and other is low (CLK). This results in low output ($CLKG=0$). Now, when the rising edge of CLK arrives, there is a short period of time when both the inputs of NAND gate becomes high, causing $CLKG$ to go high. This in turn activates $m2$ ($m2$ being NMOS), pulling down X and eventually $CLKG$ goes low. The length of the pulse is controlled by delay of NAND gate and inverter. Fig.(5.5) shows the simulation output of the above glitch generator for a clock frequency of 200 MHz. We see that, there is a glitch obtained (as expected) at every rising edge of the clock.

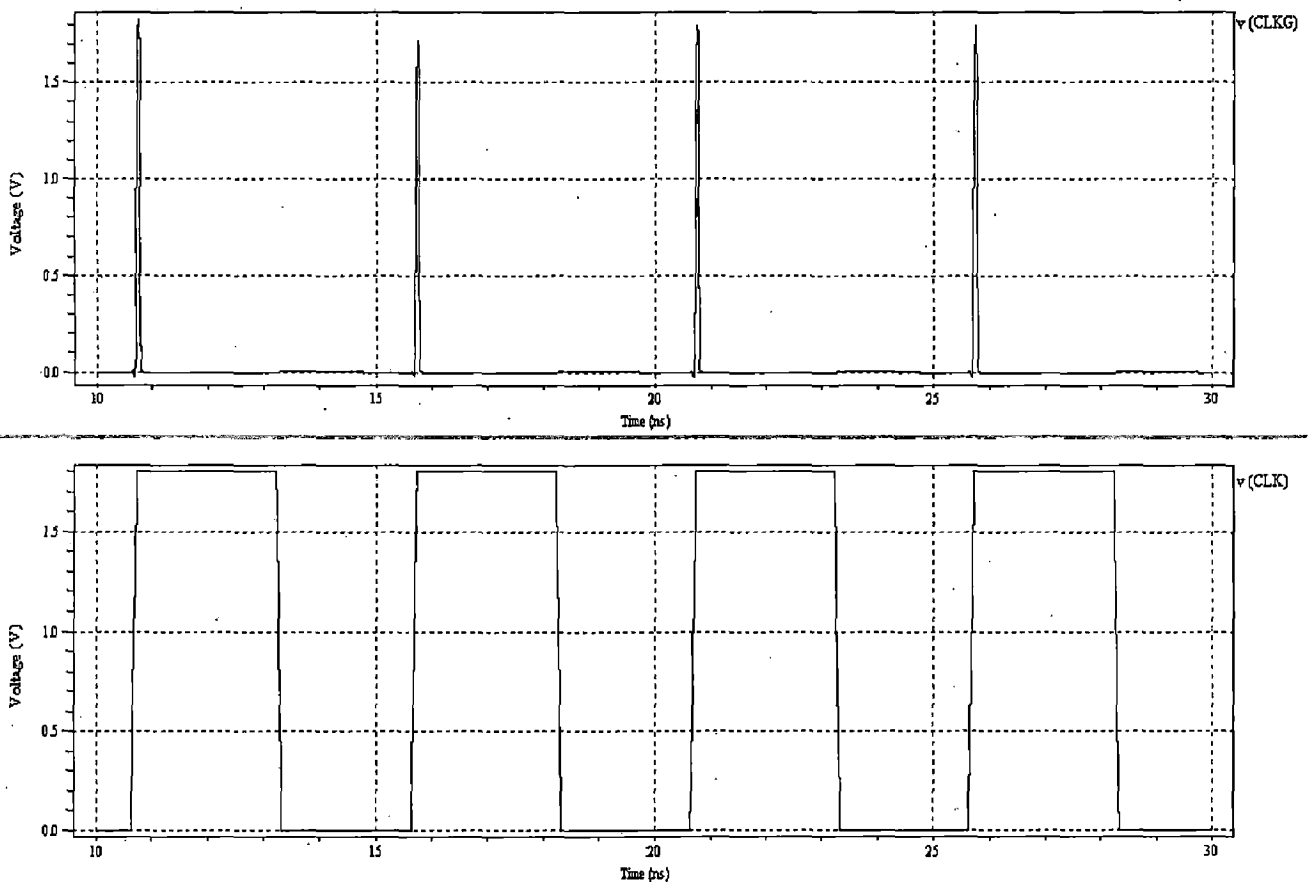


Fig.(5.5) Simulated waveform of glitch generator

5.2.2 Phase Selector and Edge Combiner

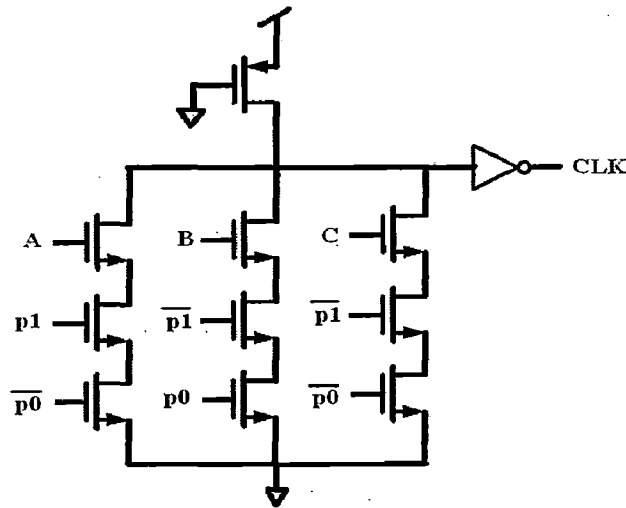


Fig.(5.6) Basic unit of phase selector

Table(5.1) A,B and C inputs for various cases

	p1	p0	CLK1 (CLKG1)	CLK2 (CLKG2)	CLK3 (CLKG3)	CLK4 (CLKG4)	CLK5 (CLKG5)	CLK6 (CLKG6)	CLK7 (CLKG7)	CLK8 (CLKG8)
A	1	0	$\Phi 1$ (G1)	$\Phi 2$ (G2)	$\Phi 3$ (G3)	$\Phi 4$ (G4)	$\Phi 5$ (G5)	$\Phi 6$ (G6)	$\Phi 7$ (G7)	$\Phi 8$ (G8)
B	0	1	$\Phi 1$ (G1)	$\Phi 3$ (G3)	$\Phi 5$ (G5)	$\Phi 7$ (G7)	Gnd	Gnd	Gnd	Gnd
C	0	0	$\Phi 1$ (G1)	$\Phi 5$ (G5)	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd

CLK (1...8) - output lines of phase selector

CLKG (1...8) – Output lines of glitch generator

$\Phi(1...8)$ – Outputs of eight delay cells

G(1...8) – Glitches of eight delay cells

The main function of phase selector is to choose the suitable phases and pass them to glitch generator. The edge combiner then utilizes these glitches to produce different multiplies of reference frequency. To generate the output clock the VCDL output phase $\Phi 1... \Phi 8$ are first selected in phase selector, corresponding glitches are produced in glitch generator and then combined in edge combiner which consists of S-R latches and OR gate. Fig.(5.6) shows the basic

unit of a phase selector. It is an implementation of 4 :1 (only three input considered here) MUX using pseudo-NMOS configuration. This unit is repeated eight times. The connections to A,B and C inputs are in accordance to table(5.1). CLK(1...8) corresponds to the output lines of phase selector and CLKG(1...8) corresponds to the output lines of corresponding glitch generator. G(1...8) denotes the glitches of corresponding phases $\Phi(1...8)$.

Fig.(5.7) shows the edge combiner incorporated in the proposed programmable frequency multiplier. It consist of four S-R latches and followed by a four input OR gate. The output lines of glitch generator CLKG(1...8) are connected to the latches as shown.

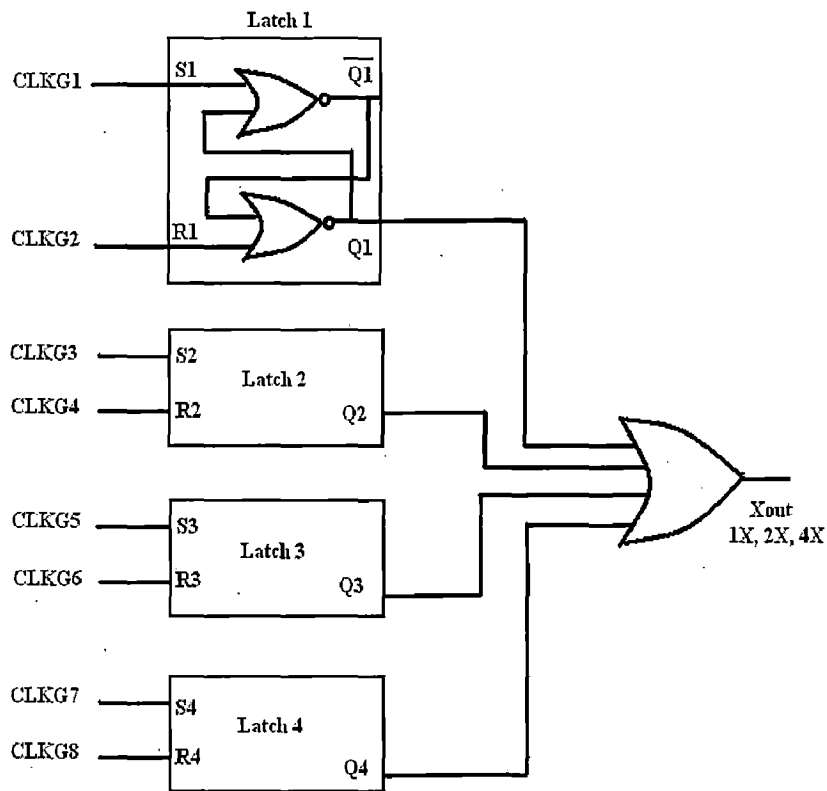


Fig.(5.7) Edge Combiner

Consider the case when $(p_1, p_0) = (1, 0)$. The A input of all the eight units of phase selector gets activated and therefore $CLK(1...8) = \Phi(1...8)$. These are passed to the glitch generator, which generates $CLKG(1...8)$. These glitches are then passed to edge combiner. Thus the final output X_{out} is 4 times the input reference frequency and 50 % duty cycle.

When $(p_1, p_0) = (0, 1)$, all the B inputs get activated. Thus phase selector passes phases such that

CLK(1...4) corresponds to Φ_1 , Φ_3 , Φ_5 and Φ_7 respectively and CLK(5...8) are grounded. Therefore CLKG(1...4) corresponds to G1, G3, G5, G7 respectively. In the edge combiner, only two latches (first two) are utilized as inputs to other latches are grounded. Thus the final output Xout is 2 times the input reference frequency and 50 % duty cycle.

When $(p_1, p_0) = (0, 0)$, all the C inputs get activated. Thus phase selector passes phases such that CLK(1,2) corresponds to $\Phi(1,5)$ respectively and CLK(3...8) are grounded. Therefore CLKG(1,2) corresponds to G(1,5) respectively. In the edge combiner, only one latch (first one) is utilized as inputs to other latches are grounded. Thus the final output Xout is same as the input reference frequency and 50 % duty cycle.

Thus a programmable operation can be achieved the proposed programmable frequency multiplier.

5.3 Simulation Output and Results of Programmable Frequency Multiplier

The proposed programmable frequency multiplier was simulated in T-SPICE, with a reference signal of 200 MHz. Simulations were carried for various values of (p_1, p_0) to obtain 4X, 2X and 1X multiplied output signal

1. Case 1 $(p_1, p_0) = (1, 0)$

Fig.(5.8) shows the simulation results for $(p_1, p_0) = (1, 0)$. As expected the frequency output signal (Xout) is 4 times the input reference frequency with 50% duty cycle. Since the input reference signal is of 200 MHz, the output signal (Xout) is of 800 MHz.

2. Case 2 $(p_1, p_0) = (0, 1)$

Fig.(5.9) shows the simulation results for $(p_1, p_0) = (0, 1)$. As expected the frequency output signal (Xout) 2 times the input reference frequency with 50% duty cycle. Since the input reference signal is of 200 MHz, the output signal (Xout) is of 400 MHz.

3. Case 2 $(p_1, p_0) = (0, 0)$

Fig.(5.10) shows the simulation results for $(p_1, p_0) = (0, 0)$. As expected the frequency output signal (Xout) is same as the input reference frequency with 50% duty cycle. Since the input

reference signal is of 200 MHz, the output signal (Xout) is also of 200 MHz. Table (5.3) gives the performance comparison of proposed clock multiplier with previous designs.

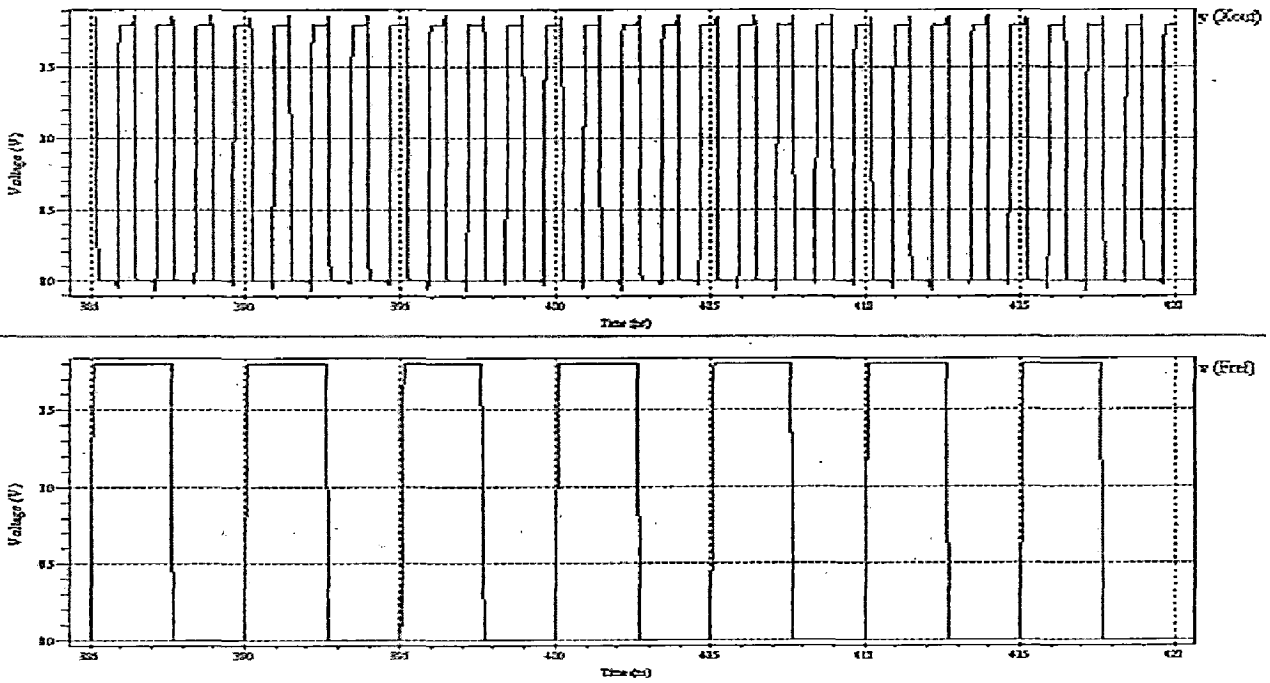


Fig.(5.8) Simulated output for $(p1, p0) = (1, 0)$, output frequency = 800 MHz

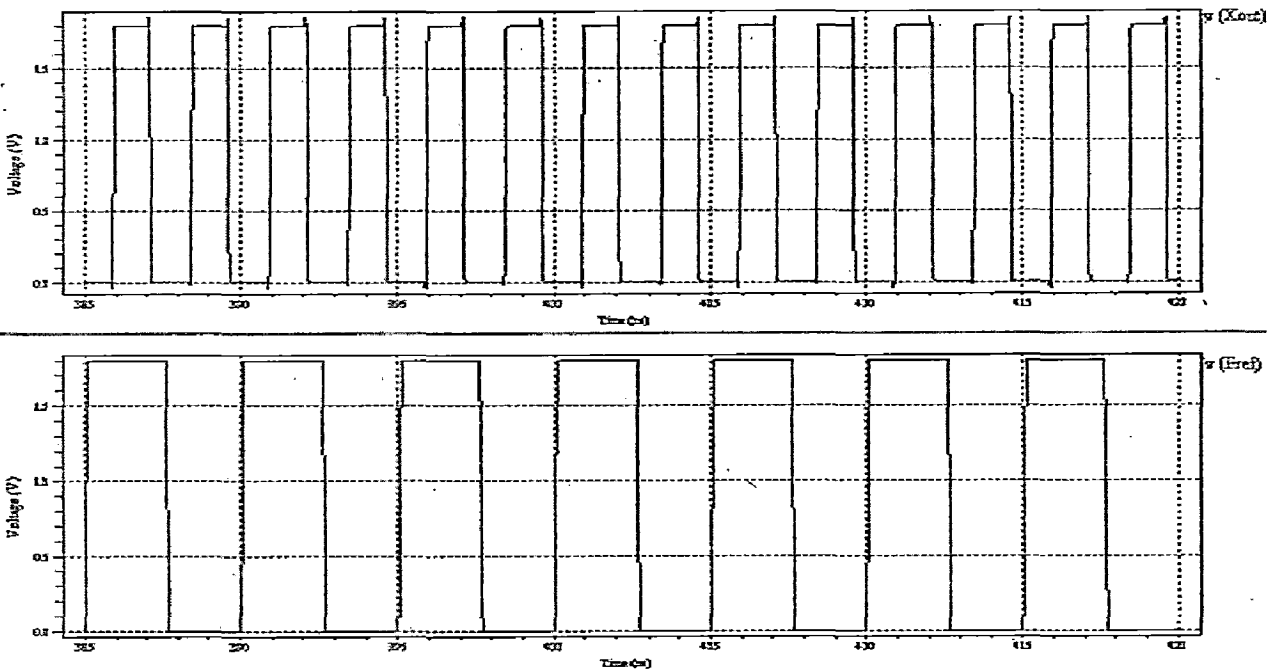


Fig.(5.9) Simulated output for $(p1, p0) = (0, 1)$, output frequency = 400 MHz

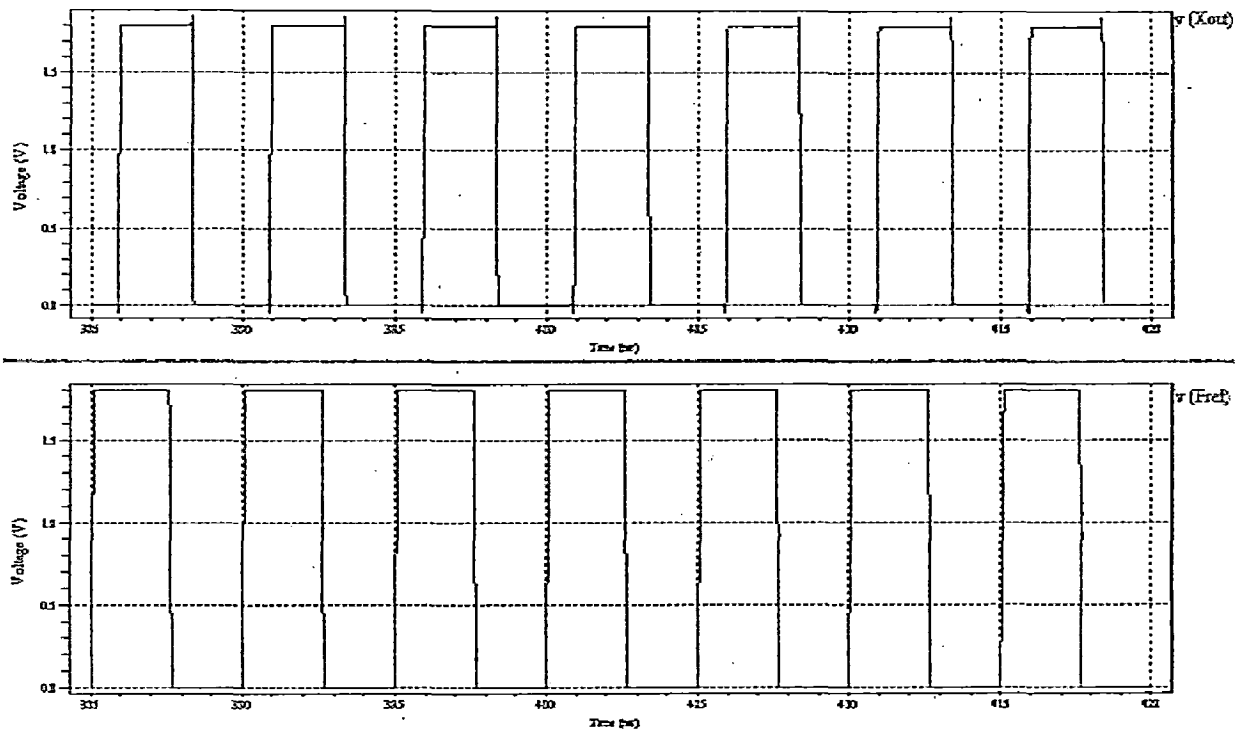


Fig.(5.8) Simulated output for $(p1, p0) = (0, 0)$, output frequency = 200 MHz

Table (5.2) Summary of proposed programmable frequency/clock multiplier

Technology	180 nm technology node
Power Supply	1.8 V
DLL Operating Frequency Range	170 MHz – 252 MHz
Frequency Multiplier Frequency Range	170 MHz – 252 MHz (multiply by 1) 340 MHz – 504 MHz (multiply by 2) 680 MHz – 1 GHz (multiply by 4)
Average Power consumption	7.99 mw @ 200 MHz (Fref=200 MHz, 1X) 8.91 mw @ 400 MHz (Fref=200 MHz, 2X) 9.65 mw @ 800 MHz (Fref=200 MHz, 4X)

Table(5.3)- Performance comparison with prior clock multiplier

	Power Consumption	Max. Freq.	Vdd	Process	Programmable
[8]	NA	1.6GHz	3.3V	0.5 um CMOS	NO
[17]	43 mw	1.1GHz	3.3V	0.35 um CMOS	YES
[19]	12 mw	2 GHz	1.8 V	0.18 um CMOS	YES
[20]	130 mw	900 MHz	3.3 V	0.35 um CMOS	NO
[22]	86.6 mw	1.8 GHz	3.3V	0.35 um CMOS	YES
[23]	23.2 mw	1.2 GHz	2.5V	0.25 um CMOS	YES
This work	9.65 mw	1 GHz	1.8 V	0.18 um CMOS	YES

Chapter 6

Conclusions

This thesis presents the work in two parts. In the first part the design of CMOS based Delay Locked Loop for multiphase clock generation is presented. In the proposed DLL, the previously proposed circuits for various units of DLL were modified for better performance at desired technology node.

It incorporates a high performance phase detector that requires minimum number of transistors and consumes as low as 0.8 mw of power. It provides a linearity range of -2π to $+2\pi$ and eliminates dead zone almost completely.

The charge pump unit incorporated in the proposed DLL gave satisfactory simulation results. Various non idealities (channel charge injection, clock feed through and charge sharing) that were present in conventional charge pump were eliminated. Therefore no sudden jump phenomenon was observed in output during pump up or pump down operation. The problem of current mismatch that was present before (due to channel length modulation (CLM) effect) was suppressed by using a cascode current mirror load. The current mismatch in charge pump incorporated is 0.74% (for charge pump current of 80 uA) , and the average power consumption is 0.88 mw.

The VCDL unit, incorporated an additional buffer circuit in order to reduce rise time and fall time of output waveforms, which is essential when DLL is used as frequency multiplier. The transfer characteristics of VCDL was satisfactory in the operating range provided by charge pump (0.6 V- 2.32 V). The conditions required to avoid false or harmonic locking were also satisfied.

The proposed DLL obtained by integrating all the units gave good simulation results. The operating input frequency range of the proposed DLL is 170 MHz to 252 MHz .The simulations were carried in T-SPICE at 180 nm technology node with 1.8 V power supply at 200 MHz input reference frequency. The lock time of DLL is less than 300 ns and the average power consumption in locked state is 6.46 mw. The static phase error of proposed DLL is as low as 2 ps. There is a slight duty cycle variation of 58% in the output signal. The proposed circuit (DLL)

s suitable for various applications in high speed systems due to advantages of low power, fast locking and low static phase error.

In the second part of the thesis, the multiphase outputs from VCDL (when DLL is in locked state) are utilized and a DLL based programmable frequency/clock multiplier, with a variable multiplication factor of 1X, 2X and 4X, is proposed. Simulations were carried for a reference frequency of 200 MHz and an output of 200 MHz, 400 MHz and 800 MHz were obtained. The average power consumptions were less than 10 mw in each case. The maximum output frequency that can be obtained by proposed programmable frequency multiplier is 1 GHz.

References

- [1] M. Johnson and E. Hudson, "A variable delay line PLL for CPU coprocessor synchronization," *IEEE J. Solid-State Circuits*, vol. 23, pp.1218-1223, Oct.-1988.
- [2] R. Best, *Phase-locked Loops*. 4th edition, McGraw-Hill, 1999
- [3] K. Lee, B. Park, H. Lee and M. Yoh, "Phase frequency detectors for fast frequency acquisition in zero-dead-zone CPPLLs for mobile communication systems", *IEEE , ESSCIRC '03.Proceedings of the 29th European* , pp 525-528 ,16-18 Sept. 2003
- [4] M. Soyuer, R. G. Meyer, " Frequency Limitation of a Conventional Phase-Frequency Detector", *IEEE J. Solid-State Circuits*, vol. 25, no. 4. pp1019-1022, Aug 1990.
- [5] W. Rhee, "Design of high-performance CMOS Charge Pumps in Phase-Locked Loops", *Circuits and Systems, Proceedings of IEEE international symposium, Orlando, May-Jun 1999*, vol. 2, pp. 545-548, July 1999.
- [6] B Razavi " *Design Of Analog CMOS Integrated Circuits*",McGraw Hill, 2002
- [7] Y. Moon, J. Choi, K. Lee, D. Jeong and M. Kim, "An All-Analog Multiphase Delay-Locked Loop Using a Replica Delay Line for Wide-Range Operation and Low-Jitter Performance", *IEEE J. Solid-State Circuits*, vol.35, no.3, pp. 377-384, Mar. 2000.
- [8] D. Foley and M. Flynn, "CMOS DLL-based 2-V 3.2-ps jitter 1-GHz clock synthesizer clock synthesizer and temperature-compensated tunable oscillator," *IEEE J. Solid-State Circuits*, vol. 36, no.3, pp. 417-423, Mar. 2001
- [9] H. Chang, J. Lin, C. Yang and S. Lin, " A Wide-Range Delay-Locked Loop With a Fixed Latency of One Clock Cycle", *IEEE J. Solid State Circuits*, vol.37, no.8, Aug-2002
- [10] K. Chen , Y. Lo, W. Yu and S. Hung, "A Mixed-Mode Delay-Locked Loop for Wide-Range Operation and Multiphase Clock Generation", *Proceedings of 3rd IEEE Int. Workshop on System on Chip for Real Time Applications*, pp. 90-93, 2003.

- [11] K. Chen and Y. Lo, "A Fast-Lock Wide-Range Delay-Locked Loop Using Frequency Range Selector for Multiphase Clock Generator", *IEEE Transactions on Circuits and Systems*, vol.54, no.7, pp. 561-565, July 2007
- [12] F. Gardner, "Charge-Pump phase-locked loops", *IEEE Trans. Comm.*, vol. 28, pp.1849-1858, Nov.1980
- [13] J. Maneatis, "Low-Jitter Process-Independent DLL and PLL Bases on Self-Biased Techniques", *IEEE Journal of Solid State Circuits*, vol. 31, no.11, pp. 1723-1732, Nov 1996.
- [14] Maheshwari S.Ravi, "Modeling and Simulation of Clock Distribution Networks using Delay-Locked Loops", *M.S Thesis, University Of Cincinnati*, July, 2006.
- [15] J Rabaey, A Chandrakasan and B Nilolic, *Digital Integrated Circuits : A Design Perspective*, 2nd Edition, Prentice Hall, 2005.
- [16] J. Cheng , "A Delay-Locked Loop for Multiple Clock Phases/ Delays generation" ,*Ph.D Thesis, Georgia Institute of Technology*, December, 2005.
- [17] C. Kim, I. Hwang, and S. Kang, "A low-power small-area +/- 7.28 ps jitter 1 GHz DLL-based clock generator," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1414–1420, Nov. 2002.
- [18] Aguiar, R.L.; Santos, D.M., "Oscillatorless clock multiplication", *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp.630-633, May 2001.
- [19] R. Rad, W. Dally, H. Ng, R. Senthinathan, M. Lee, R. Rathi, J. Poulton , "A Low-Power Multiplying DLL for Low-Jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips", *IEEE J. Solid and Circuits*, vol. 37, no.11, Dec. 2002
- [20] G. Chien and P.Gray, "A 900 MHz local oscillator using DLL based frequency multiplier technique for PCS applications," *IEEE J. Solid State Circuits*, vol.35, no.12, pp. 1996-1999, Dec-2000

- [21] B. Kim, T. Weigandt, and P. R. Gray, "PLL / DLL System Noise Analysis for Low Jitter Clock Synthesizer Design," *Intl. Symposium on Circuits and Systems (ISCAS)*, London, vol.4, pp.31-34 June, 1994.
- [22] J. Kim, Y. Kwak, M. Kim, S. Kim and C. Kim, "A 120-MHz–1.8-GHz CMOS DLL-Based Clock Generator for Dynamic Frequency Scaling," *IEEE J. Solid State Circuits*, vol. 41, no. 9, pp. 2077-2082 Sep 2006.
- [23] C. Wang, Y. Tseng, H. She and R. Hu, "A 1.2 GHz programmable DLL-based frequency multiplier for wireless applications," *IEEE transactions on VLSI systems*, vol. 12, no. 12, pp 1404-1408, Dec 2004.
- [24] H. Notani, H. Kondoh and Y. Matsuda, "A 622MHz CMOS Phase Locked Loop with pre-charge type CMOS Phase - Detector," *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, pp.129-130, 1994.
- [25] W. Lee, J. Cho, S. Lee, "A High Speed And Low Power Phase Frequency Detector And Charge Pump", *Design Automation Conference , Proc. of the ASP-DAC*, pp. 269-272, 1999.
- [26] L. Dai, and R. Harani, "CMOS switched-op-amp-based sample-and-hold circuit," *IEEE J. Solid-State Circuits*, vol.35, no. 1, pp. 109-113, Jan. 2000.
- [27] H. Yu, Y. Inoue, and Y. Han, "A new high-speed low-voltage charge pump for PLL applications," *Proceedings of 6th IEEE International Conference on ASIC (ASICON)*, Shanghai , China , p.435-438, Oct. 2005.
- [28] D. Sahu, "A Completely Integrated Low Jitter CMOS PLL for Analog Front Ends in System on Chip Environment", *Design Automation Conference*, Bangalore, pp. 360-365. July-Nov 2002
- [29] R. Chang and L. Kuo, "A New Low Voltage Charge Pump Circuit For PLL", *IEEE int. symposium on Circuits And Systems*, Geneva, pp. v701-v704, May 28-31,2000.
- [30] K.H. Chen and Y Lo., "A Fast-Lock DLL with Power-On Reset Circuit," *IEICE Trans. on Fundamentals*, Vol.E87-A No.9, pp.2210-2220, Sep. 2004.