Design of a Phase-Locked Loop (PLL) Using GPDK045 CMOS Technology

By

Nerob Kumar Mohonto
24371001

A project submitted to the Department of Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
M. Engg in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering
Brac University
February 2025

Declaration

It is hereby declared that

1. The project submitted is my own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Nerob Kumar Mohonto
24371001

Approval

The project titled "Design of a Phase-Locked Loop (PLL) Using GPDK045 CMOS Technology" submitted by

Nerob Kumar Mohonto (24371001)

of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M. Engg EEE on 6th February 2025.

Examining Committee:

Supervisor:
(Member)

Touhidur Rahman, PhD
Professor, Department of EEE,
Brac University

Internal Expert Examiner:
(Member)

MG Sorwar Hossain, PhD
Professor, Department of EEE,
Brac University

Departmental Head:
(Chair)

Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of EEE,
Brac University

Ethics Statement

In conducting this Phase Locked Loop project, I am committed to upholding the highest standards of integrity, responsibility and transparency. I have approached the project with honesty in all calculations, simulations and results, ensuring that the data accurately reflects my work without any misrepresentation. All third-party resources used have been properly cited, respecting intellectual property rights. I have carefully documented the design process to ensure transparency and accountability. In selecting components, I have considered both reliability and environmental impact, aiming for energy-efficient solutions. Ultimately, this project is carried out with the goal of contributing positively to the field of electronic design while adhering to ethical principles.

Abstract

A Phase-Locked Loop (PLL) is a feedback control system used to synchronize the frequency and phase of an output signal with a reference signal widely applied in communication systems, data recovery and frequency synthesis. The high-frequency operation and rapid switching of PLL circuits often result in substantial power consumption, posing challenges in designing for high-speed applications. This project addresses these challenges by developing a PLL that balances performance with power efficiency. The PLL architecture includes a Phase Frequency Detector (PFD) to compare the phase difference between input and feedback signals, a Charge Pump (CP) and Loop Filter (LF) to process this phase error into a stable control voltage and a Voltage-Controlled Oscillator (VCO) that adjusts output frequency accordingly. Performance metrics, including lock time, lock range, phase margin, gain margin, jitter, and output frequency range are analyzed and optimized for stability and efficiency. Using Cadence Virtuoso simulations, the project evaluates key trade-offs, producing a PLL design that maintains precise frequency control and low power consumption, ideal for high-performance electronic applications.

Keywords: Phase Locked Loop (PLL), Phase Frequency Detector (PFD), Charge Pump (CP), Loop Filter (LF), Voltage Controlled Oscillator (VCO), Frequency Divider (FD).

"For my teacher, who saw potential in a quiet student and fanned the spark into a flame."

Acknowledgement

I would like to express my deepest gratitude to all those who have supported me throughout this project. First and foremost, I extend my sincere thanks to my advisor professor Dr. Touhidur Rahman, Dr. MG Sorwar Hossain and Dr. Md. Mosaddequr Rahman for their guidance, insightful feedback and encouragement which helped me navigate the challenges I faced during the project. I am also grateful to Brac University for providing access to the necessary tools, resources and facilities that made this work possible. Additionally, I would like to acknowledge the contributions of various authors and researchers whose work in the field of phase-locked loop design provided valuable reference and inspiration.

Table of Contents

List of Tables

List of Figures

List of Acronyms

PFD Phase Frequency Detector

CP Charge Pump

LF Loop Filter

VCO Voltage Controlled Oscillator

FD Frequency Divider

Chapter 1

Introduction

1.1 Phase Locked Loop

High-performance digital systems rely on clocks to coordinate operations and maintain synchronization across functional units and integrated circuits (ICs) [1]. With each new generation of processing technology and processor architecture, clock frequencies and data rates continue to rise. Within these systems, phase-locked loops (PLLs) are used to generate precisely timed clocks, which are then distributed on-chip using clock buffers. The increasing clock frequencies present significant challenges in ensuring low uncertainty and low power consumption during clock generation and distribution [3]. This research explores innovative system-level and circuit-level techniques to minimize clock timing uncertainty while maintaining minimal power and area overhead.

As the number of internet-connected devices grows exponentially, the volume of data carried by the internet backbone is also increasing rapidly [4,7]. Among the various transmission media, optical fibers offer the highest bandwidth at a lower cost, making them an ideal solution for the internet backbone. However, the electronic interface often becomes a bottleneck in the design of high-speed digital systems. This, coupled with the need for rapid time-to-market, highlights the advantages of using flexible modules and macro cells in designs. In optical communication infrastructure, flexibility translates to features like programmable bit rates, which require a PLL capable of robust operation over a wide frequency range [8,9,12]. Such wide-range PLLs can support multiple protocols and applications, enhancing reusability and reducing development time [11].

1.2 PLL Fundamental

Phase-locked loops (PLLs) play a critical role in modern electronics by enabling the generation of precise on-chip clocks. These clocks are indispensable for a wide range of applications, including clock and data recovery in communication systems, clock generation in microprocessors, and frequency synthesis for various electronic devices [4,12]. The core concept of phase locking, which synchronizes an oscillator's phase and frequency with a reference signal, has remained largely consistent since its invention in the 1930s. However, as technology has advanced, the design and implementation of PLLs have become increasingly sophisticated and challenging [13,15].

Modern PLLs are expected to meet stringent performance requirements. One of the most critical factors is clock timing uncertainty, commonly referred to as jitter. In high-speed digital systems, even small amounts of timing error can lead to significant performance degradation. Similarly, power consumption is another major design constraint [8,9]. With the growing emphasis on energy efficiency in electronic systems, PLLs must deliver high performance while consuming minimal power. Additionally, as devices become more compact, the area occupied by PLLs on an integrated circuit must also be minimized.

This research focuses on addressing these challenges by exploring innovative approaches to PLL design specifically tailored for high-performance digital systems. The goal is to achieve an optimal balance between timing precision, power efficiency, and compactness, ensuring that PLLs can meet the demanding requirements of modern applications while maintaining reliability and robustness [8]. Through these efforts, PLLs continue to evolve, supporting the development of cutting-edge technologies across various industries.

1.3 PLL Definition

A phase-locked loop (PLL) is a closed-loop feedback system designed to maintain a fixed

phase relationship between its output clock and a reference clock. It continuously monitors and adjusts to phase variations within its bandwidth [5,7]. Additionally, a PLL can multiply a low-frequency reference clock (REF) to generate a high-frequency output clock (Vout).

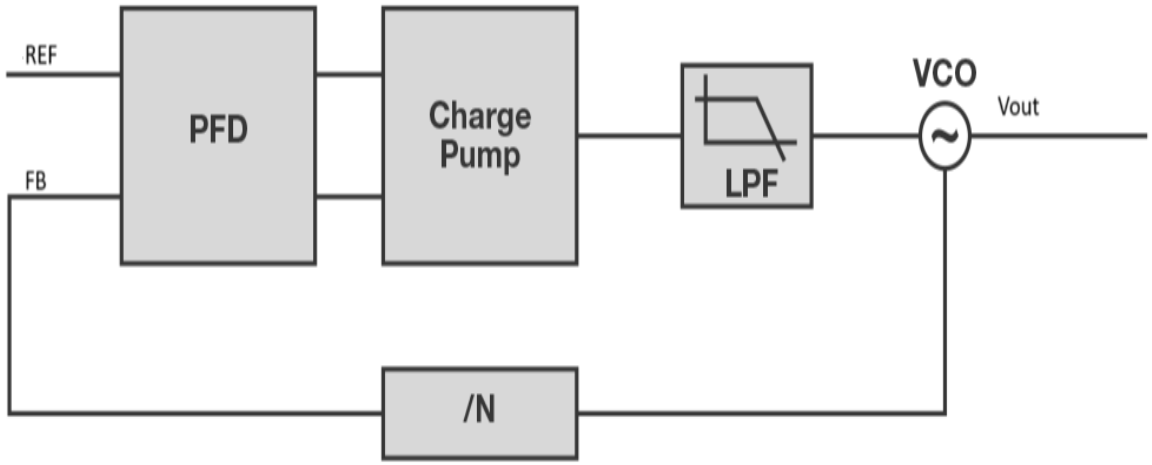


Figure 1.1: PLL basic block diagram

A PLL operates by comparing the phase of the feedback clock with the reference clock using a phase detector, which generates an error signal proportional to their phase difference. This phase error accumulates over time if there are small frequency mismatches [1]. The error signal is filtered through a low-pass filter to produce a control signal, which adjusts the oscillator's frequency to align the feedback clock phase (φ feedback) with the reference clock phase (φ ref) [8]. A frequency divider scales down the oscillator's output frequency to create the feedback clock. The system achieves phase lock when the feedback clock matches the reference clock in frequency, maintaining a fixed phase difference. The oscillator's frequency is (N) times the reference clock frequency since the feedback clock is a divided-down version of the oscillator's output. The PLL takes a lower frequency input, typically from a crystal oscillator and multiplies it to generate higher frequency clocks [2,5].

1.4 PLL Applications

PLL has several applications listed below [9,12];

I. Clock Generation:

PLLs are widely used in clock generation to produce stable, high-frequency clock signals from a low-frequency reference. They synchronize the internal clocks of digital systems, ensuring precise timing in microprocessors, GPUs, and FPGAs.

II. Frequency Synthesis:

PLLs generate a wide range of output frequencies from a single reference frequency by locking onto multiples or fractions of the reference. This is critical in RF communication systems, where precise frequency control is required for channel tuning.

III. Modulation and Demodulation:

PLLs are used for phase and frequency modulation/demodulation in communication systems. They lock onto the input carrier signal and track phase or frequency variations to extract transmitted information.

IV. Telecommunications and Networking:

PLLs ensure reliable synchronization of data streams by aligning clock signals in systems such as Ethernet, SONET, and optical networks, enabling seamless data transfer with minimal timing errors.

V. Radar and Sonar Systems:

In radar and sonar systems, PLLs are employed to generate precise local oscillator signals for

frequency mixing and to extract phase information, aiding in target detection and distance measurement.

VI. Motor Speed Control:

PLLs are used to regulate the speed of electric motors by comparing the actual motor speed (feedback) with the desired speed (reference) and adjusting the drive signal accordingly.

VII. Wireless Communications:

PLLs are integral in wireless communication systems for carrier frequency generation, channel selection, and synchronization. They enable stable operation in technologies like 5G, Wi-Fi, and Bluetooth.

Each application highlights the versatility of PLLs in ensuring frequency stability, synchronization, and precision across various fields.

This project mainly focuses on High Frequency Clock Generation and CPU-GPU synchronization.

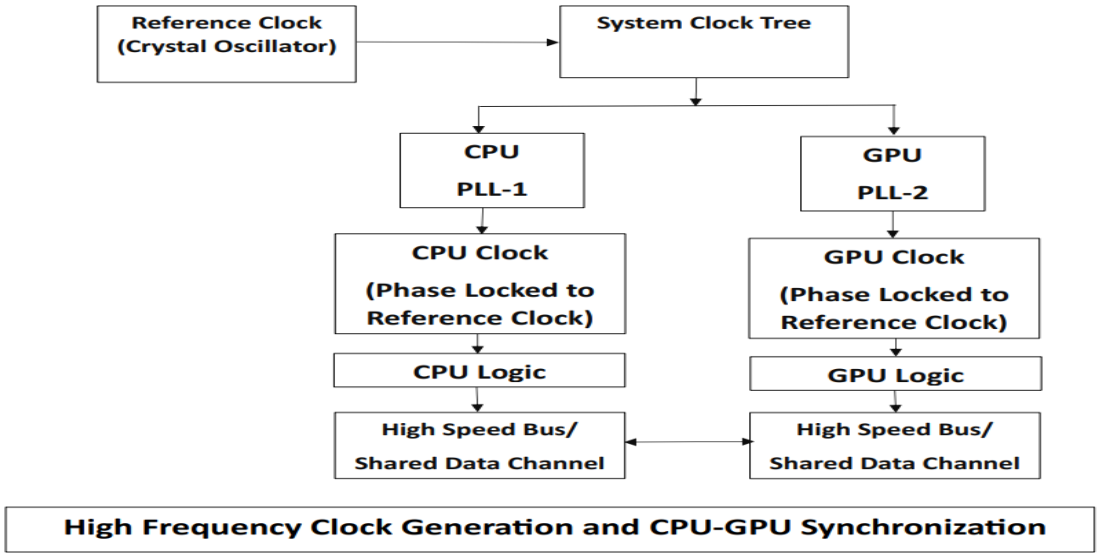


Figure 1.2: Clock Generation and CPU-GPU synchronization block diagram.

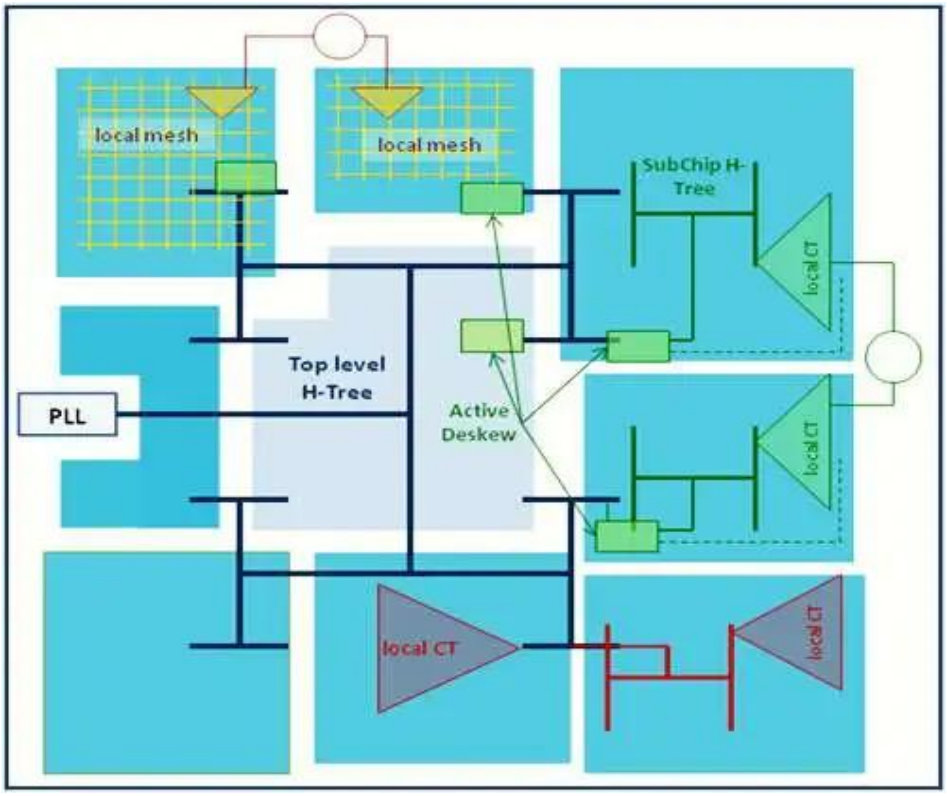


Figure 1.3: Clock tree distribution

1.5 PLL Types

There are several types of PLL based on their construction and purposes [4,7];

I. Analog PLL (APLL):

An Analog PLL uses fully analog components such as operational amplifiers, RC filters, and analog multipliers. It is known for its high performance in terms of frequency range and stability but is more sensitive to noise. APLLs are widely used in RF communications and clock recovery circuits.

II. Digital PLL (DPLL):

DPLLs are hybrid systems that use both analog and digital components. The phase detector and loop filter are often digital, while the VCO remains analog. They are robust to noise and

exhibit higher accuracy but may suffer from quantization errors. DPLLs are used in medium-speed systems like audio and video processing.

III. All-Digital PLL (ADPLL):

ADPLLs are fully digital implementations of PLLs, employing digital circuits for all components, including the phase detector, loop filter, and digitally controlled oscillator (DCO). They offer advantages such as CMOS integration, low power consumption, and scalability. ADPLLs are ideal for processors, SoCs, and wireless communication systems [11].

IV. Charge Pump PLL (CP-PLL):

CP-PLLs use a charge pump as part of the loop filter to eliminate steady-state phase errors. This configuration offers improved loop performance and better phase accuracy. CP-PLLs are commonly used in frequency synthesizers and clock distribution networks.

V. Fractional-N PLL:

This type of PLL uses a fractional divider in the feedback path, enabling fine frequency resolution by locking onto fractional multiples of the reference frequency. While it provides precise frequency control, it is prone to quantization noise. Fractional-N PLLs are used in broadband communication systems.

VI. Integer-N PLL:

An Integer-N PLL uses an integer divider in its feedback path, producing output frequencies that are integer multiples of the reference frequency. It has a simpler design and lower noise but provides coarser frequency steps. Integer-N PLLs are employed in fixed-frequency RF systems and clock generation applications.

Each type of PLL serves specific applications and balances trade-offs between complexity, precision, noise tolerance, and scalability.

In this project a charge pump integer N PLL have been developed.

Type	Core Technology	Main Advantage	Main Limitation	Applications
APLL	Analog	High-frequency performance	Noise sensitive	RF, communications, clock recovery
DPLL	Mixed-signal (mostly digital)	Robust to noise	Higher quantization error	Audio/video processing, medium-speed systems
ADPLL	Fully digital	CMOS integration, low power	Complex DCO design	Processors, SoCs, wireless communication
CP-PLL	Analog with charge pump	Eliminates phase errors	Current mismatch issues	Frequency synthesizers, clock networks
Fractional-N PLL	Fractional divider	Fine frequency resolution	Quantization noise	RF systems, broadband communication
Integer-N PLL	Integer divider	Simpler design	Coarse frequency steps	Fixed-frequency RF systems, clock generation

Table 1.1: Comparison between different types of PLL

1.6 PLL Components

A phase-locked loop (PLL) consists of several key components that work together to achieve phase and frequency synchronization. The phase frequency detector (PFD) compares the phase of the reference clock with the feedback clock and generates an error signal proportional to their phase difference [1,3,8]. This error signal is processed by a loop filter (LF) to remove high-frequency noise, producing a smooth control signal. The filtered signal drives the voltage-controlled oscillator (VCO) [6], which adjusts its output frequency based on the control voltage. The VCO output is then scaled down by a frequency divider to generate the feedback clock [1,6]. Together, these components form a closed-loop system that aligns the feedback clock with the reference clock in both phase and frequency.

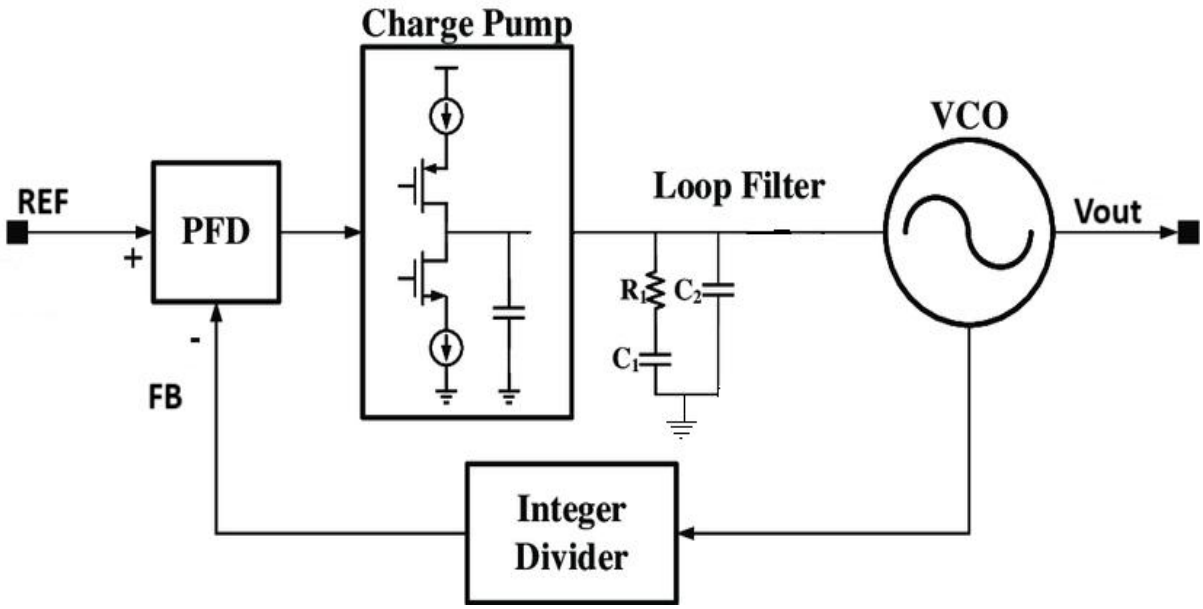
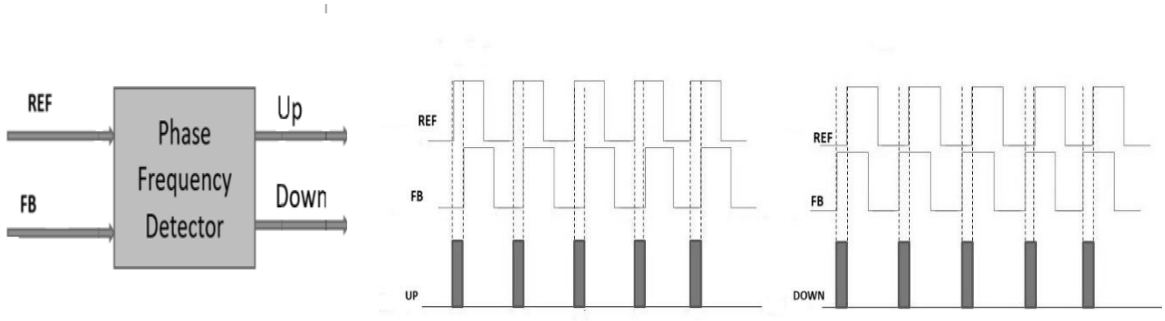


Figure 1.4: Individual Blocks in PLL Diagram

1.6.1 Phase Frequency Detector

The phase frequency detector (PFD) measures the phase difference between two input signals and generates an error signal proportional to that difference. However, when there is a significant frequency mismatch, a standard phase detector may fail to indicate the correct direction of the phase error. This can lead to rapid accumulation of phase error, causing oscillations between >180° and <180° from one cycle to the next [1]. As a result, the average output of the phase detector lacks meaningful frequency information and provides no useful phase data. Since the PFD is insensitive to input frequency differences, the PLL may struggle to achieve lock during startup when the oscillator's frequency divided by (N) is significantly different from the reference frequency. This issue is known as the PLL's inadequate acquisition range. To address this limitation, a phase-frequency detector (PFD) is employed, which can detect both phase and frequency differences. The operation of a PFD is illustrated in Figure 03 for two scenarios: (b) when the reference clock leads feedback clock, and (c) when the

reference clock lags feedback clock. In both cases, the DC components of the PFD outputs, UP and DOWN, provide crucial information about the phase or frequency difference [1,3].



(a) (b) (c)

Figure 1.5: (a) Phase frequency detector (b)Reference clock leads (c) Feedback clock leads

1.6.2 Charge Pump and Loop Filter

The charge-pump circuit consists of two switches controlled by the UP and DN signals from the Phase-Frequency Detector (PFD). It injects or removes charge from the loop filter capacitor C*CP*, forming an integrator in combination with the capacitor. This integrator converts the UP and DN pulses into an average voltage, which adjusts the frequency of the Voltage-Controlled Oscillator (VCO). Since the VCO itself acts as an integrator, the charge-pump PLL inherently has two poles at the origin in its loop gain, making the system unstable. To stabilize it, a zero is introduced at ωz= $\frac{1}{RCCP}$ by adding a resistor *R* in series with C*CP*. While the PFD, charge-pump and filter are often represented as a linear continuous-time system, in reality, the PFD operates as a pulse modulator, driving the charge pump based on the pulse width, which corresponds to the input phase difference Δϕ. This results in a non-linear phase response due to the cyclical nature of phase and the discrete, sampled phase information at the reference clock frequency [1,10].

1.6.3 Current Starved Voltage Controlled Oscillator

Voltage Controlled Oscillator is said to be the heart of the Phase Locked Loop. A Current Starved Voltage Controlled Oscillator (VCO) is a type of ring oscillator where the current available to each inverter stage is controlled by a voltage input, V*in*. This design enables precise control of the oscillation frequency by varying the current flowing through the inverter stages. The VCO consists of multiple inverting stages connected in a loop to create oscillations. Each stage is "starved" of current using current source transistors controlled by V*in*. The oscillation frequency is determined by the charging and discharging rates of the capacitors in the inverter stages, which depend on the available current [1,6,10].

The oscillation frequency, f*osc* is given by $\frac{1}{2N \cdot t_{delay}}$ (1.1)

Where, N is number of stages in the ring oscillator and t*delay* is delay per stage, which depends on the current set by V*in*.

1.6.4 Frequency Divider

A Frequency Divider is a circuit that reduces the input signal frequency by a fixed factor N, to produce a lower output frequency. It is commonly used in Phase-Locked Loops (PLLs) for generating feedback signals. Digital frequency dividers use flip-flops or counters to divide the input clock frequency by powers of two [1,15].

The relationship is given by: f*out*= $\frac{fin}{N}$ (1.2)

Where f*in* is the input frequency,

f*out* is the output frequency

N is the division factor.

1.7 Noise and Power Considerations in PLL

The primary objective in designing a PLL for high-performance digital systems is to produce an output clock with minimal timing uncertainty. This uncertainty arises due to device mismatches and noise sources within the system. Device mismatches result in a static phase shift in the PLL output clock, which can be mitigated through careful layout design and increasing device sizes. While skew is less critical than jitter, as it can be compensated for due to its static nature, dynamic noise introduces random phase shifts which are harder to manage. Jitter is caused by sources such as thermal noise, flicker noise and power-supply or substrate noise, making it a key challenge in PLL design [3,4,7,8,9].

Types of Noise in PLL:

1. Phase Noise: Represents random fluctuations in the signal phase, typically caused by intrinsic device noise (e.g., thermal or flicker noise) or disturbances in the power supply. It is measured in dBc/Hz and is critical in frequency-domain applications like communication systems and RF circuits. High phase noise reduces spectral purity, which can interfere with adjacent channels and degrade signal-to-noise ratio.

2. Jitter: Refers to variations in the timing of signal edges, often caused by noise, interference, or crosstalk. It is measured in picoseconds (ps) or nanoseconds (ns). Jitter directly affects the accuracy of high-speed digital systems, such as data recovery and timing synchronization in processors and communication links. Excessive jitter can lead to data errors and degraded system performance.

Spur: Unwanted spectral components in the PLL output arise due to imperfections like power supply coupling, mismatched components, or non-linearities in the loop. Spurs degrade spectral purity, which can introduce interference in sensitive systems and compromise overall signal quality, particularly in communication and radar systems.

Charge Pump Leakage Current: Leakage current in the charge pump introduces errors in the loop filter voltage, leading to instability or inaccuracies in phase and frequency control. This can prevent the PLL from achieving proper locking or maintaining stability, especially in low-power or high-precision applications. Managing leakage current is crucial to ensure consistent performance.

Chapter 2

Architecture of PLL

Phase-locked loops (PLLs) are classified based on design and application into types like Analog PLLs (APLLs) for traditional systems, Digital PLLs (DPLLs) combining digital and analog features and All-Digital PLLs (ADPLLs) for high programmability. Charge Pump PLLs (CP-PLLs) offer precise phase control, while Fractional-N PLLs enable fine frequency synthesis, and Integer-N PLLs provide simpler designs with lower complexity [3,5].

Each type has distinct characteristics and is suited to specific applications, such as frequency synthesis, clock generation and data recovery. Considering the various features and performance trade-offs, this project focuses on the Charge Pump Integer-N PLL as it combines the simplicity of integer division with the enhanced control provided by a charge pump. This makes it ideal for high-performance digital systems requiring reliable clock generation and synchronization [6,15].

2.1 Phase frequency Detector

The Phase-Frequency Detector (PFD) plays a vital role in Phase-Locked Loop (PLL) systems by comparing the phase difference ($\Delta\phi$) between the reference clock (f_{REF}) and the feedback clock (f_{FB}) [1,3]. In addition to phase comparison, the PFD detects any frequency mismatch (Δf) between these two signals. Based on the phase and frequency differences, the PFD generates appropriate control signals, either UP or DOWN, which are used to adjust the Voltage-Controlled Oscillator (VCO). These adjustments aim to minimize both $\Delta\phi$ and Δf, ensuring the PLL achieves and maintains phase and frequency lock [1,3,15].

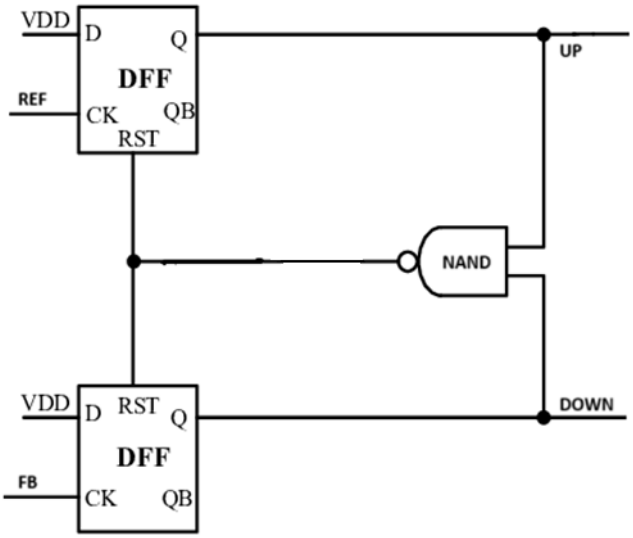


Figure 2.1: Traditional PFD

The PFD in the circuit consists of:

Two D Flip-Flops:

The first flip-flop is triggered by the reference clock (f_{REF}).

The second flip-flop is triggered by the feedback clock (f_{FB}).

NAND Gate:

The NAND gate resets the flip-flops when both UP and DOWN signals are detected, preventing further outputs.

The reset mechanism ensures that the PFD is ready for the next phase comparison.

Mathematical Representation of PFD Operation

The phase difference between the reference and feedback signals is expressed as:

$$\Delta\phi(t)=\phi_{REF}(t)-\phi_{FB}(t) \quad (2.1)$$

where:

$\phi_{REF}(t)$ is the phase of the reference clock.

$\phi_{FB}(t)$ is the phase of the feedback clock.

The frequency difference between the two signals is given by: $\Delta f=f_{REF}-f_{FB}$ (2.2)

The output of the PFD depends on $\Delta\phi$ and Δf:

If $f_{REF}>f_{FB}$ or $\phi_{REF}>\phi_{FB}$, the UP signal is detected.

If $f_{REF}<f_{FB}$ or $\phi_{REF}<\phi_{FB}$, the DOWN signal is detected.

Operational States

State 1 (UP = 0, DN = 1): Feedback Clock Leads ($\Delta\phi<0$), the DOWN signal is detected.

This causes the charge pump to decrease the control voltage of the VCO, thereby decreasing the feedback clock frequency (f_{FB}).

Case 2 (UP = 0, DN = 0): Phase Lock ($\Delta\phi=0$), both UP and DOWN signals are low. The PLL remains in a locked state.

State 3 (UP = 1, DN = 0): Reference Clock Leads ($\Delta\phi>0$) , the UP signal is detected.

This causes the charge pump to increase the control voltage of the VCO, thereby increasing the feedback clock frequency (f_{FB}).

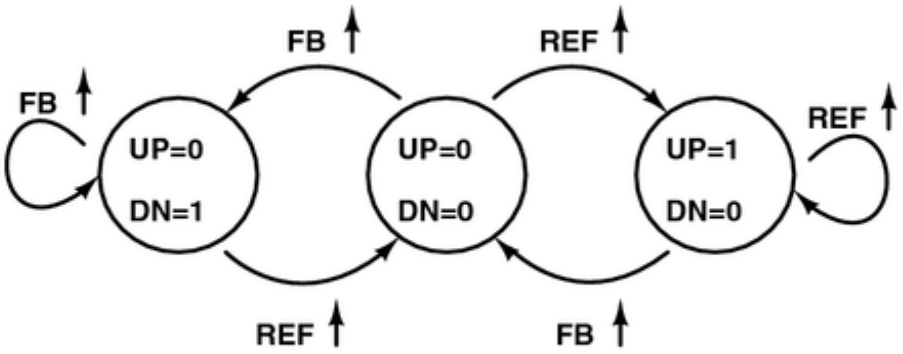


Figure 2.2: State diagram of PFD

2.1.1 Edge Triggered D-Flipflop

A D Flip-Flop is a type of sequential logic circuit that stores and transfers data based on a clock signal. It is one of the most commonly used flip-flops in digital electronics for edge-triggered operations.

The D Flip-Flop has:

Input: Data (D), Clock (CLK), optional Set and Reset.

Output: Q (current state), $\overline{Q}$ (complementary state).

Operation:

On the clock's rising edge the value of the D input is captured and transferred to the Q output.The output remains constant until the next clock edge, making it ideal for synchronizing data in sequential circuits.

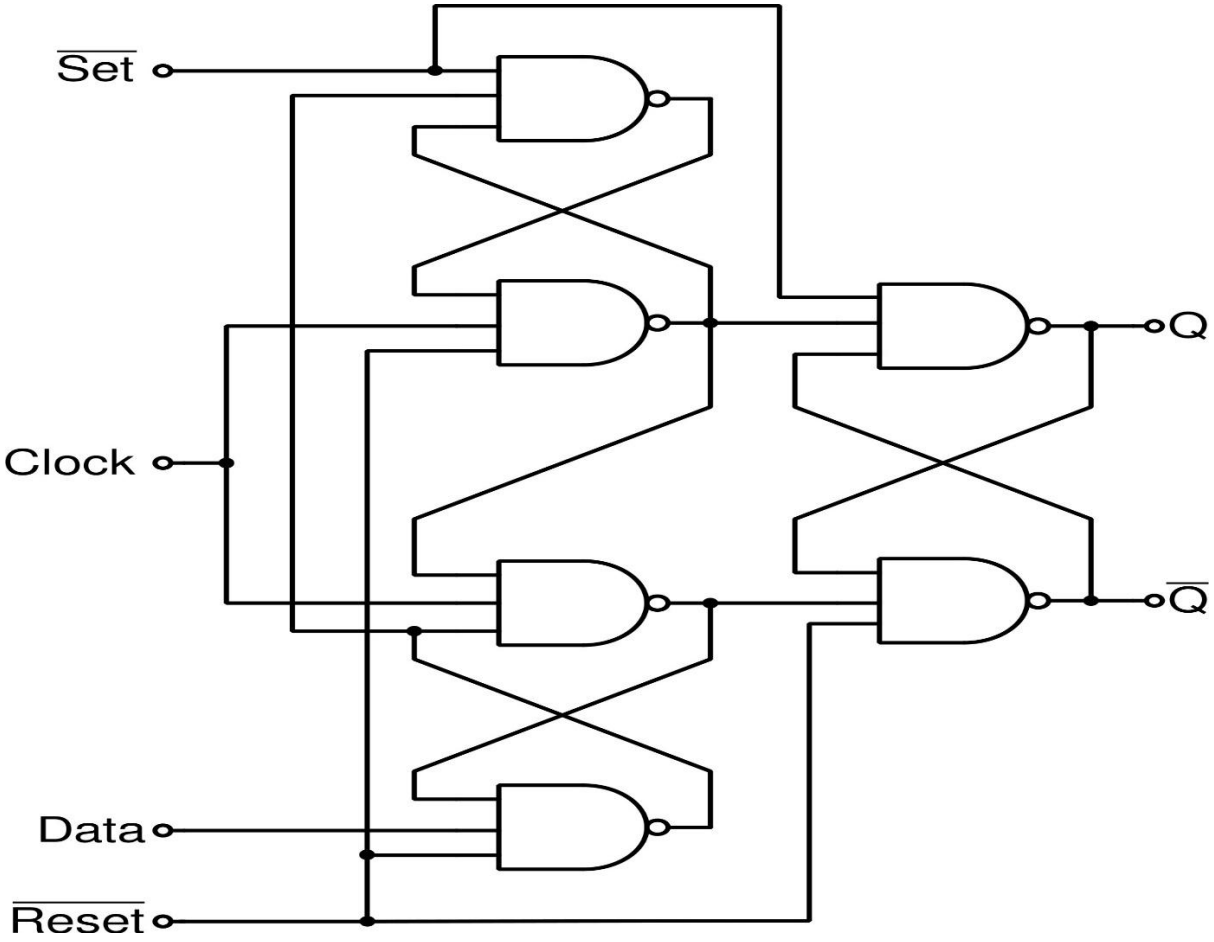


Figure 2.3: Asynchronus D Flip Flop with set and reset

Truth Table:

Clock (CLK)	**Data (D)**	**Q (Output)**	**$\bar{Q}$ (Output)**	**Description**
↑ (Rising)	0	0	1	Q follows D (0 stored)
↑ (Rising)	1	1	0	Q follows D (1 stored)
No Edge	X	Qprev	$\bar{Q}$prev	Output retains previous state

Table 2.1: D Flip Flop truth table

2.1.2 NAND Gate

A NAND gate is a fundamental digital logic gate that performs the inverse of the AND operation. It outputs a logic LOW (0) only when all of its inputs are HIGH (1); otherwise, it outputs a logic HIGH (1).

Boolean Expression:

$$Y=\overline{A.B}$$

Where A and B are the inputs, and Y is the output.

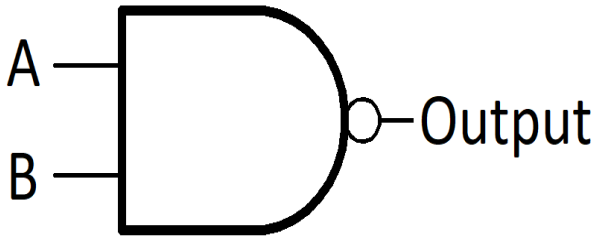


Figure 2.4: NAND Gate

Truth Table:

A	**B**	**Output ($Y=\overline{A.B}$)**
0	0	1
0	1	1
1	0	1
1	1	0

Table 2.2: NAND Gate truth Table

Reset Mechanism:

Signal associated with phase frequency detector;

Reference clock signal: Input clock signal to the upper flip-flop.

Feedback clock signal: Input clock signal to the lower flip-flop.

UP (Q_{REF}): Output of the upper flip-flop, goes high when REF rises.

DOWN (Q_{FB}): Output of the lower flip-flop, goes high when FB rises.

RESET: Signal generated by the NAND gate, resets the flip-flops when both Q_{REF} (UP) and Q_{FB} (DOWN) are high simultaneously.

D Flip-Flop Behavior:

The two D flip-flops in the PDF operate based on their clock inputs and reset signals as well:

Upper Flip-Flop (triggered by REF):

$$Q_{REF}(n+1)=D_{REF}\cdot\overline{RESET} \quad (2.3)$$

Where, Q_{REF}: Output of the upper D flip-flop.

Lower Flip-Flop (triggered by FB):

$$Q_{FB}(n+1)=D_{FB}\cdot\overline{RESET} \quad (2.4)$$

Where, Q_{FB} is the Output of the lower D flip-flop.

NAND Gate Logic:

The NAND gate combines Q_{REF} and Q_{FB} namely UP and DOWN signal to generate the RESET

signal:

$$RESET=\overline{Q_{REF}.Q_{FB}} \quad (2.5)$$

The PDF will not reset when either Q_{REF}=0 or Q_{FB}=0. PDF will reset when Q_{REF}=1 and Q_{FB}=1

The phase frequency detector will reset when both the UP and DOWN signal is high simultaneously.

Operation Phases of reset mechanism:

Rising Edge of REF:

At the rising edge of the upper D flip flop is triggered. In this state Q_{REF}=1 and Q_{FB}=0 so *RESET=1*. The UP signal goes high and no reset occurs.

Rising Edge of FB:

At the rising edge of the lower D flip flop is triggered. In this state Q_{REF}=1 and Q_{FB}=1 so *RESET=0.* The DOWN signal goes high and reset occurs in the PDF.

Timing Analysis:

Let, t_{REF} is the time of the rising edge of Reference clock

t_{FB} is the time of the rising edge of Feedback clock

t_{RESET} is the time when reset occurs.

Time difference between the Reference and feedback clock is:

$$\Delta t = t_{FB} - t_{REF} \quad (2.6)$$

The reset occurs at t_{RESET} immediately after both Q_{REFQ} and Q_{FB} are high.

The mathematical equation is expressed by;

$$t_{RESET}=\max(t_{REF},t_{FB})+t_{NAND}\_delay \quad (2.7)$$

2.2 Charge Pump

The charge pump is a critical component of a Phase-Locked Loop (PLL). It works in tandem with the phase-frequency detector (PFD) and the loop filter to control the voltage-controlled oscillator (VCO), ensuring the PLL achieves and maintains a locked state [1].

The PFD compares the phase and frequency of the reference clock and the divided output clock from the VCO. The PFD generates two output signals: UP and DOWN, which indicate whether the VCO frequency needs to increase or decrease to align with the reference clock.

When reference clock leads the feedback clock, the UP signal is high indicating that the VCO frequency need to be increased. When the feedback clock leads the reference clock the DOWN signal is high, indicates the VCO frequency need to decreases [1,3,8].

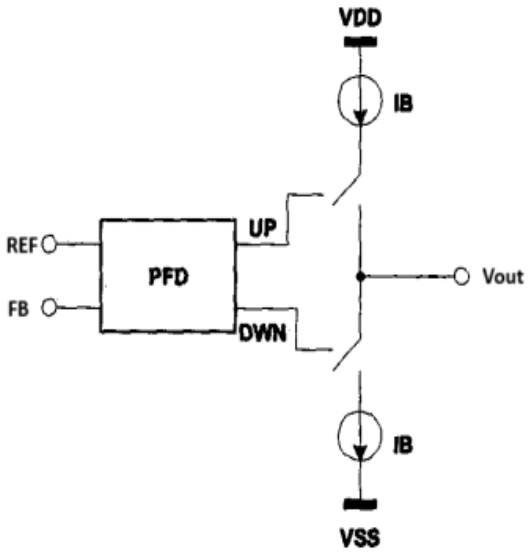
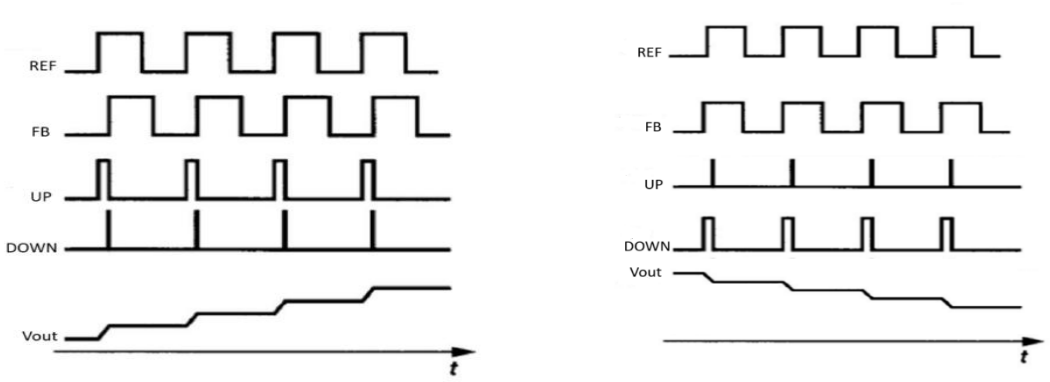


Figure 2.5: Charge Pump placed after PDF



(a) (b)

Figure 2.6: Charge Pump output (a) refecence clock leading feedback clock, charging occurs (b) feedback clock leading refecence clock, discharging occurs

2.2.1 Loop Filter

A loop filter is a crucial element in a Phase-Locked Loop (PLL) that processes the error signal produced by the phase detector to adjust the frequency of the Voltage-Controlled Oscillator (VCO). It is specifically designed to eliminate high-frequency noise and shape the dynamic characteristics of the PLL by establishing its bandwidth, stability, and transient response.

The loop filter functions as a low-pass filter, permitting the low-frequency error signal to pass through while attenuating undesired high-frequency components that may cause instability in the system or impair its performance [1,8].

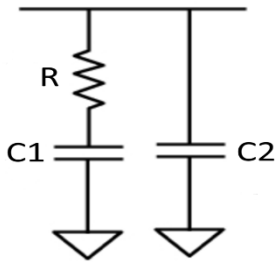


Figure 2.7: Second-order passive loop filter

The transfer function F(s) relates the input current Icp (charge pump output) to the output voltage Vout applied to the VCO. It is given as:

$$F(S)=\frac{Vout(S)}{Icp(S)} = \frac{\frac{1}{C_1}+Rs}{s(1+\frac{C1}{C2})} \quad (2.8)$$

Zero is Contributed by the resistor *R* and capacitor C1 and Poles are Contributed by the capacitors C1 and C2.

The zero frequency *fz* is given by:

$$Fz=\frac{1}{2\pi RC1} \quad (2.9)$$

The filter has two poles.

The first pole fp_1 is given by:

$$fp1 = \frac{1}{2\pi R(C1+C2)} \quad (2.10)$$

The first pole fp_2 is given by:

$$fp2=\frac{1}{2\pi RC2} \quad (2.11)$$

2.3 Current Starve Voltage Controlled Oscillator

A Current Starved Voltage-Controlled Oscillator (CSVCO) is a critical component in Phase-Locked Loop (PLL) circuits. It generates a periodic signal whose frequency is directly controlled by an input voltage. The "current-starved" architecture limits the current flowing through the oscillator's delay stages, providing fine control over the oscillation frequency [1,3].

This makes the CSVCO ideal for applications requiring precise frequency synthesis, such as clock generation in digital systems, RF communication and signal modulation [6,7].

Circuit Description:

The CSVCO typically consists of a ring oscillator made of inverters with additional current-limiting transistors that "starve" the inverters of current. The current starving voltage-controlled oscillator circuit consists of few major terms i.e current mirroring mechanism, current starving mechanism, inverter stages and control voltage. These work together to generate an oscillation frequency.

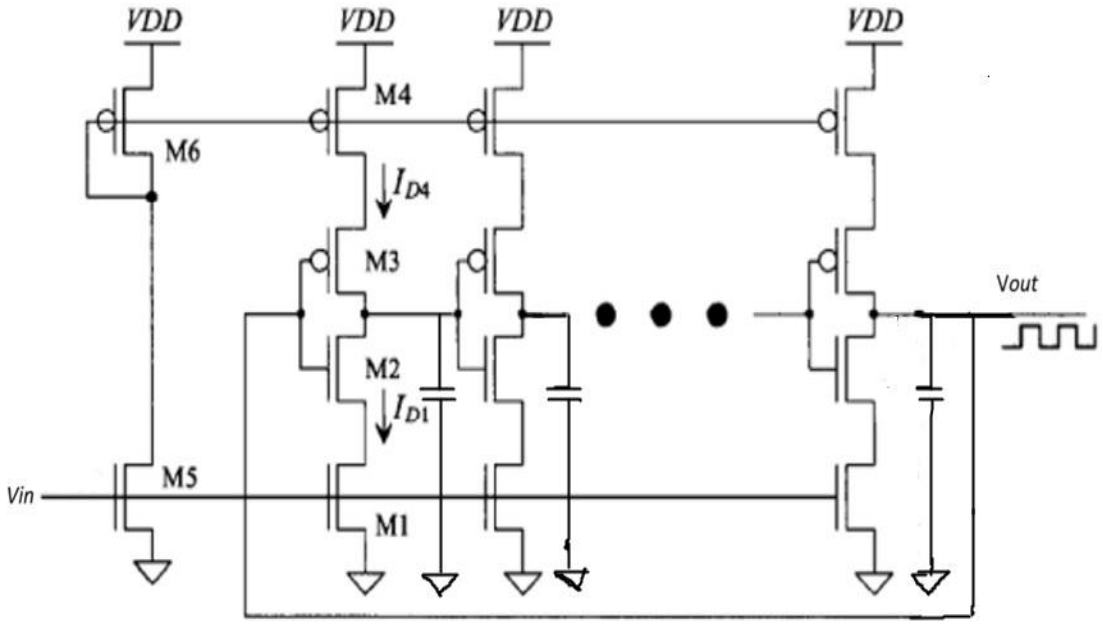


Figure 2.8: Current Starve Voltage-Controlled Oscillator

Current Mirroring Mechanism:

A PMOS and a NMOS form a current mirroring mechanism. PMOS sources current from VDD and this current will be copied across all the PMOS in the current starving mechanism. The amount of current flow to ground through NMOS is controlled by the control voltage V*in*. The

amount of current flow to the ground is copied by all the NMOS in the current starving mechanism.

Current Starving Mechanism:

The current starving mechanism is a technique used in Voltage-Controlled Oscillators to precisely control the oscillation frequency by limiting the current flowing through the inverter stages and it controls the current sink to ground as well. This mechanism ensures that the delay and therefore the frequency is directly controlled by the input control voltage V*in*. Each inverter consists of a pair of PMOS placed above inverter and NMOS placed below inverter, which from the current starving mechanism.

Inverter Stages:

In a Current-Starved Voltage-Controlled Oscillator (CSVCO), the core of the oscillation is built using a series of inverter stages. These stages act as delay elements and their switching speed is directly controlled by the current-limiting transistors, enabling frequency modulation based on the control voltage.

A PMOS and a NMOS form an inverter. The PMOS transistor is responsible for pulling the output node high when the input to the inverter is low. When the input to the inverter is low at the logic level, the PMOS conducts and pull the output towards VDD, creating the output logic high.

When the input to the inverter is high at logic level, the NMOS conducts and sink the output to ground, creating output logic low.

Each inverter stage has a phase shift of 180 degree, which ensures that the signal feedback from the output to the input remain in phase with the original signal.

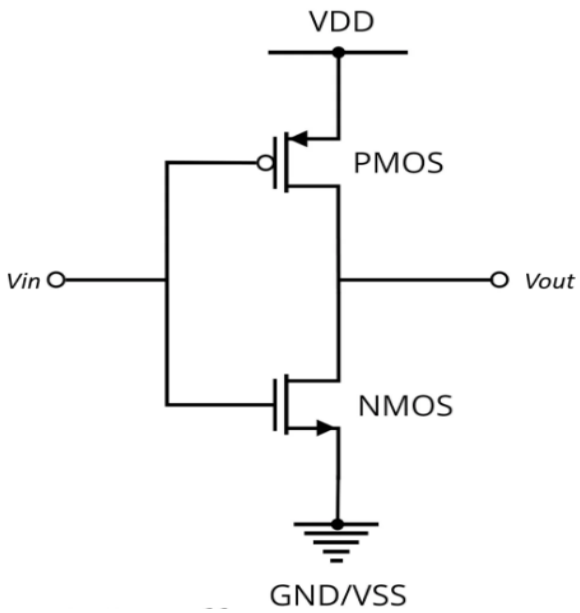


Figure 2.9: Inverter

Control Voltage:

The control voltage V*in* is the most critical parameter in a Current-Starved Voltage-Controlled Oscillator. It directly regulates the oscillator's frequency by controlling the amount of current flowing through the inverter stages. When the control voltage V*in* is high, the conductivity of the MOSFETs increases, leading to higher current flow, which allows the capacitors in the inverter stages to charge and discharge faster. This results in a higher oscillation frequency. Again, When the control voltage V*in* is low, the conductivity of the MOSFETs decreases, leading to lower current flow, which allows the capacitors in the inverter stages to charge and discharge slowly. This results in a lower oscillation frequency.

A ring oscillator requires an odd number of inverter stages to oscillate. The propagation delay through each stage depends on how fast the output node charges and discharges, which is controlled by the limited current. By varying the control voltage V*in*, the current changes, thus adjusting the delay and the overall oscillation frequency.

Center Frequency:

The center frequency in a Voltage-Controlled Oscillator (VCO) is the frequency at which the VCO naturally oscillates when the control voltage applied to it is at its nominal or midpoint value. This frequency serves as a reference point for the VCO's operation and defines the middle of its tuning range.

The center frequency expressed as;

$$f_{center} = \frac{f_{min}+f_{max}}{2} \tag{2.12}$$

where; f_{min} = minimum oscillation frequency.

f_{max} = maximum oscillation frequency.

The center frequency is crucial in PLL design because it determines the baseline frequency around which the VCO can be adjusted to lock onto the input signal.

2.3.1Mathematical Analysis

Oscillation Frequency:

The frequency of a CSVCO is inversely proportional to the delay of each stage in the ring oscillator.

$$f_{osc} = \frac{1}{2.N.t_{delay}} \tag{2.13}$$

Where, N is the number of stages in the CSVCO

t_{delay} is the delay in every stage

Delay in Every Stage:

The delay in the CSVCO is caused by the time taken by each inverter to switch one stage to another. The inverter switching is not instantaneous. Inverting operation takes small amount of time for each inverter to;

a. Charge or discharge the internal capacitance of the MOSFET

b. Pull the output either VDD or GND.

For a ring oscillator each inverter adds a propagation delay.

$$t_{delay} = \frac{C\ load.\ VDD}{I\ control} \tag{2.14}$$

Where, C*load* = Load capacitance of each inverter stage.

VDD= Supply voltage.

I*ctrl* = Control current limited by the current mirror.

The delay in a ring oscillator is cumulative.so,

$$\text{Total delay} = N.\ t_{delay} \tag{2.15}$$

Where, N = number of stages

T*delay* = delay in every stage.

The oscillation period is twice of total delay.

$$T_{osc} = 2.(N.\ t_{delay}) \tag{2.16}$$

Frequency-Voltage Relationship:

The frequency and voltage relationship in a CSVCO is given by;

$$fosc \propto \frac{(\mathrm{V}in-Vth)^2}{\mathrm{C}load\ .\ VDD} \tag{2.17}$$

where, V*in* = control voltage.

V*th* = Threshold voltage of the MOSFET.

C*load* = Load capacitance of each inverter stage.

VDD= Supply voltage.

This shows that the frequency has a quadratic dependence on control voltage V*in* allowing for fine frequency tuning.

A typical three stage CSVCO provide frequency range about 4 Ghz. The targeted frequency is 2Ghz for this project. To maintain the desired frequency a capacitor is placed after every inverter stage. The capacitors increase the load capacitance at each inverters output node. It provides an additional delay in every inverting stage.

Charging and discharging time of a capacitor is a function of current flowing through the inverter and capacitor itself expressed by;

$$\tau = \mathrm{R.C} \tag{2.18}$$

Where, τ is the time constant.

R is the internal resistance of mosfet

C is the internal capacitance of mosfet

Since the capacitor increases the time, it takes to charge or discharge the output node, this slow down the switching speed of each inverter of each inverter stage. As a result, oscillation frequency decrease and maintain the desired oscillation frequency. In this project 2Ghz is maintained for high frequency application.

The oscillation frequency is given by;

$$fosc = \frac{1}{2.N.R.C} \qquad (2.19)$$

Where, N is the number of stages in the CSVCO

R is the internal resistance of MOSFET

C is the internal capacitance of every stages

2.4 Frequency Divider

A frequency divider is an electronic circuit that reduces the frequency of an input signal *fin* by an integer factor. It is commonly implemented using sequential logic components like D flip-flops. The circuit shown in the image is a integer frequency divider using D flip-flops [1,3,9].

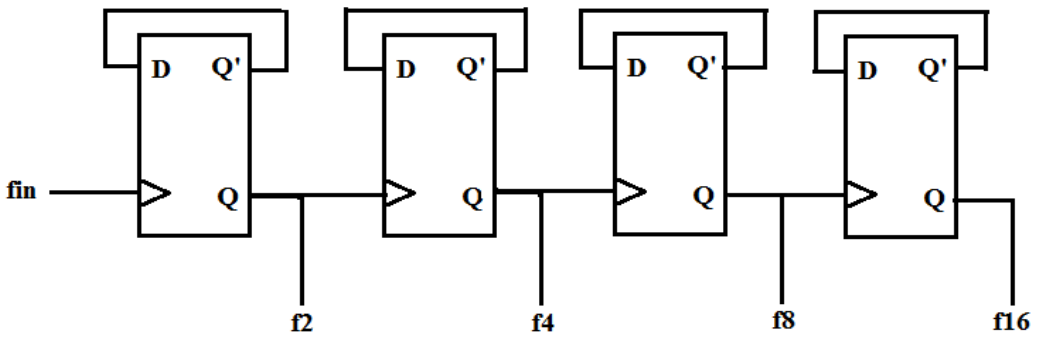


Figure 2.10: Frequency divider circuit (divide by 16)

The frequency divider depicted in the image comprises four D flip-flops linked in series, with each one set up to divide by 2 [1]. The initial flip-flop receives the input signal fin, and its output switches at a frequency that is half of the input signal's frequency. Each of the following flip-flops continues to halve the frequency at every stage. The output of the *n-th* flip-flop $f2^n$ has a frequency that is $fin/2^n$.

Mathematicaly;

For an input signal frequency of *fin:*

In the First D Flip-Flop:

Output frequency $f2=\frac{fin}{2}$ (2.20)

Time period $T2=2Tin$, where $Tin=\frac{1}{fin}$

In the Second D Flip-Flop:

Output frequency $f4=\frac{fin}{2^2}$ (2.21)

$$=\frac{fin}{4}$$

Time period $T4=4Tin$

In the Third D Flip-Flop:

Output frequency $f8=\frac{fin}{2^3}$ (2.22)

$$=\frac{fin}{8}$$

Time period *T8=8Tin*

In the Fourth D Flip-Flop:

Output frequency $f16 = \frac{fin}{2^4}$ (2.23)

$$= \frac{fin}{16}$$

Time period *T16=16Tin*

The generel expression for output frequency is;

$$fout = \frac{fin}{2^n} \quad (2.24)$$

Chapter 3

Circuit Design and Results

3.1 Design Environment

The design and simulation of the Phase-Locked Loop (PLL) were carried out in the Cadence Virtuoso environment utilizing the GPDK 45nm CMOS technology. This design environment offers a comprehensive platform for custom analog and digital circuit design, simulation, and verification.

The Cadence Virtuoso Design Framework is a comprehensive platform for integrated circuit design, widely used for analog, mixed-signal, and RF circuits. In this project, the Virtuoso Schematic Editor was utilized to design key PLL components, including the Phase Frequency Detector, Charge Pump, Loop Filter, Voltage-Controlled Oscillator, and Frequency Divider. The Virtuoso Layout Suite supported the physical design process, ensuring compliance with Design Rule Check (DRC) and Layout vs. Schematic (LVS) verification. Circuit behavior was thoroughly analyzed using Spectre, the integrated simulation engine, for both time-domain and frequency-domain performance evaluation.

GPDK 45nm Technology:

The Generic Process Design Kit (GPDK) 45nm is a technology library developed by Cadence Design Systems to support the design and simulation of integrated circuits (ICs) using a 45-nanometer (nm) CMOS process. In GPDK 45nm technology, the 45nm refers to the channel length (or gate length) of the MOSFET, which is the horizontal distance between the source and drain regions under the gate.

The Generic Process Design Kit (GPDK) 45nm was utilized for transistor-level design and simulation, providing accurate models for MOSFETs, resistors, capacitors, and inductors optimized for 45nm technology. It included comprehensive design rules to ensure manufacturable layouts with proper transistor sizing and spacing. Additionally, the advanced 45nm node enabled scalable designs that support high-performance and low-power circuit implementations.

3.2 Design Procedure

The design of a Phase-Locked Loop (PLL) involves multiple stages, including circuit design, simulation, and layout verification.

1. Specification Definition:

Define the PLL design requirements:

I. Input Reference Frequency

II. Output Frequency Range

III. Lock Time and Jitter Performance

IV. Power Consumption and Area Constraints

2. Schematic Design in Cadence Virtuoso

I. Phase Frequency Detector (PFD): Design using flip-flops and logic gates to detect phase and frequency differences.

II. Charge Pump (CP): Implement current sources/sinks controlled by PFD outputs for charging/discharging the loop filter.

III. Loop Filter (LF): Design a second-order RC filter to convert charge pump pulses into a stable control voltage.

IV. Voltage-Controlled Oscillator (VCO): Use a Current-Starved Ring Oscillator to generate frequency output based on control voltage.

V. Frequency Divider: Design a divide-by-N circuit by D flip-flop to scale down the VCO frequency for feedback.

3. Simulation and Verification

Use Spectre for functional and transient simulations.

I. PFD: Verify phase/frequency detection capability.

II. Charge Pump: Check current matching and pulse behavior.

III. Loop Filter: Analyze frequency response and stability.

IV. VCO: Simulate frequency tuning vs. control voltage.

V. Overall PLL: Simulate locking behavior, phase noise, and jitter performance.

4. Design Optimization

I. Tune transistor sizes and bias currents for better performance.

II. Adjust loop filter parameters to improve stability and reduce jitter.

III. Ensure that the design meets all specifications under different process corners voltage, and temperature variations.

5. Final Verification

I. Conduct different comprehensive analysis for yield estimation.

II. Validate the PLL's operation across Process, Voltage and Temperature variations.

III. Ensure that the design is fully optimized for performance, power, and area.

6. Perform Performance Analysis

I. Lock Time: Time for PLL to stabilize after startup or frequency changes.

II. Lock Range: Frequency range where the PLL maintains lock.

III. Phase Margin and Gain Margin: Stability indicator, typically >45° and Ensures stability against component variations.

IV. Jitter: Minimize timing variations for precision

3.2.1 Design of Phase frequency Detector

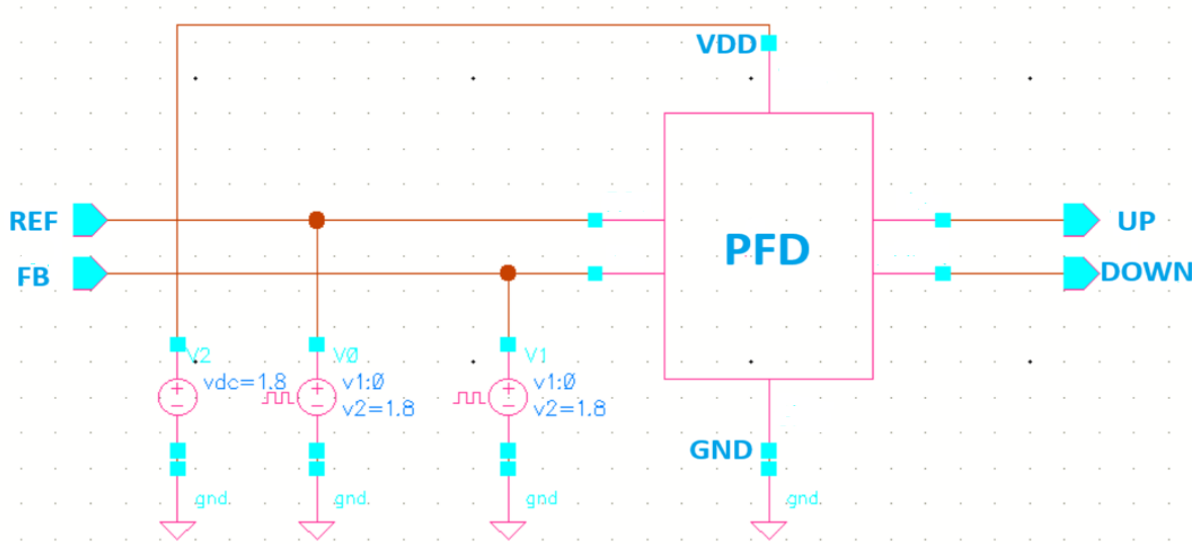


Figure 3.1: Phase frequency detector

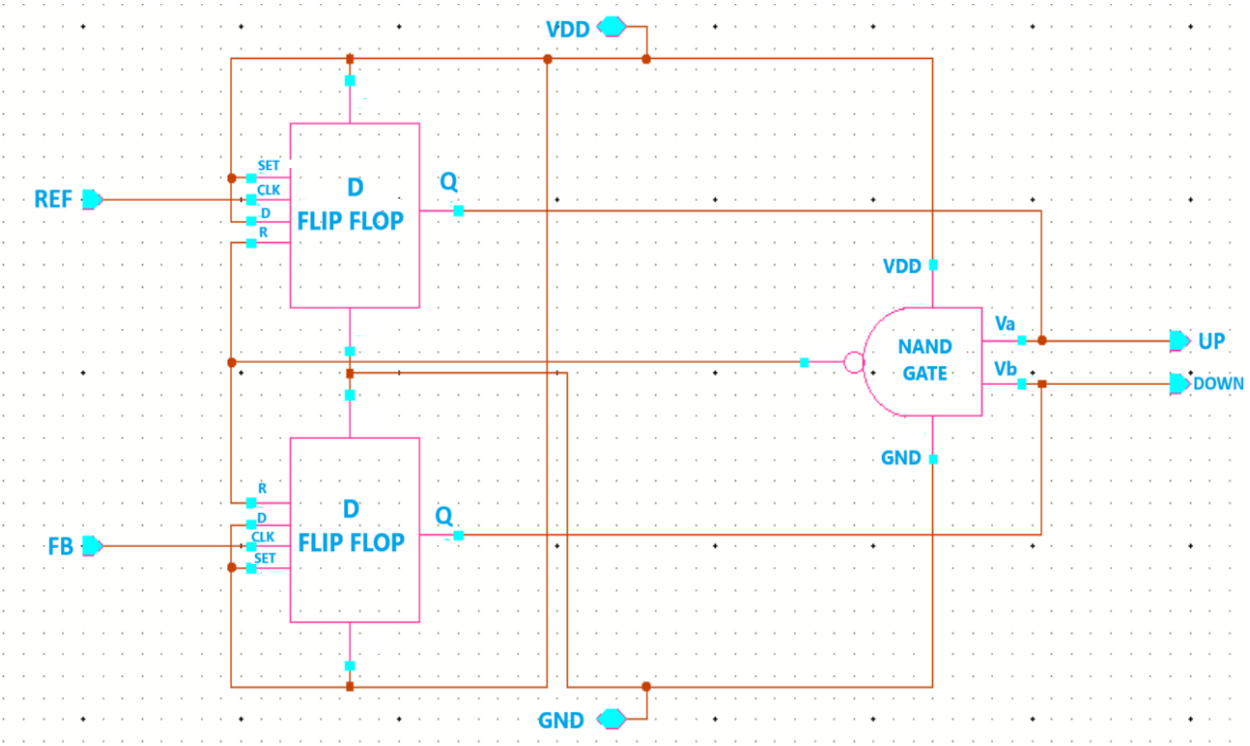


Figure 3.2: Phase frequency detector circuit diagram

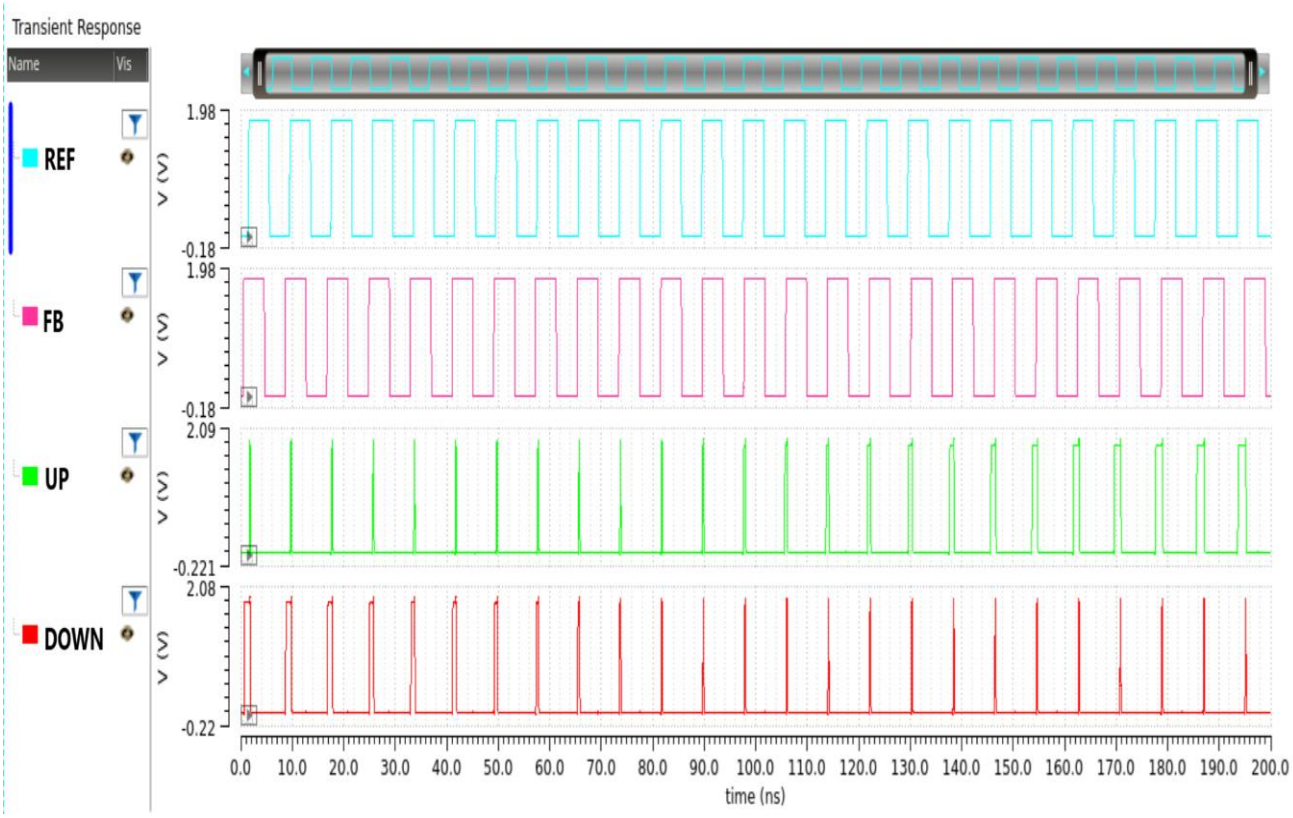


Figure 3.3: PFD output (UP and DOWN sianal)

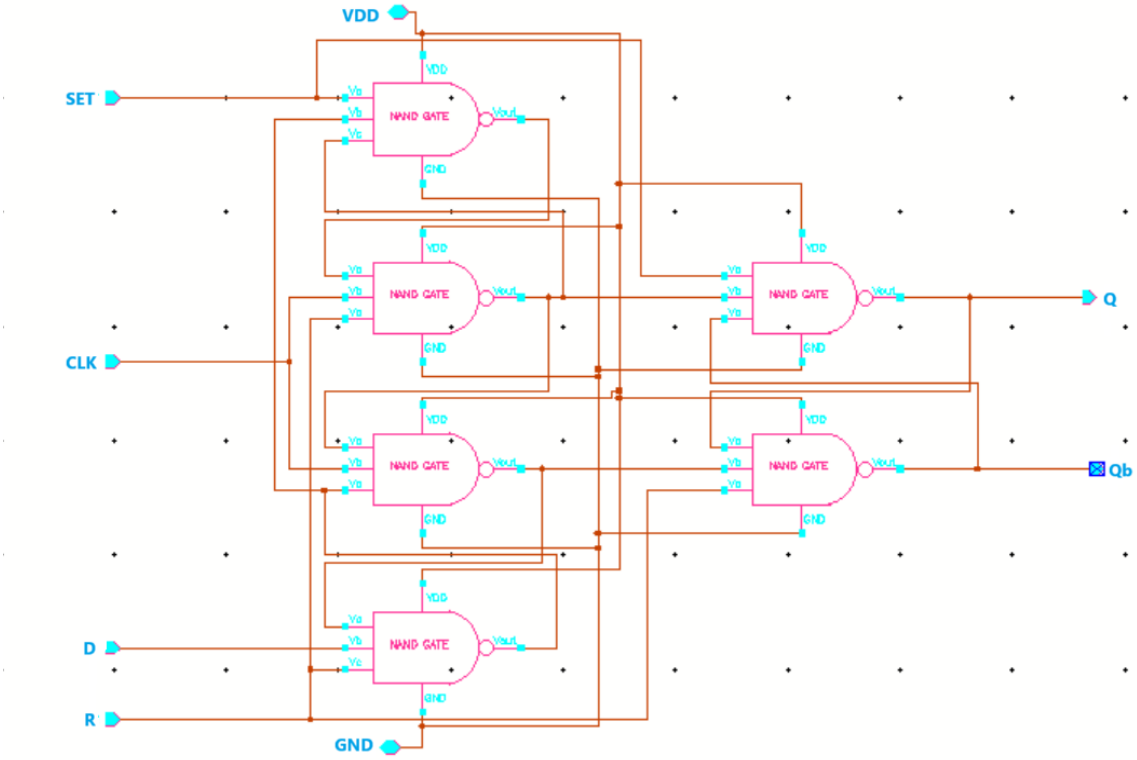


Figure 3.4: Edge triggered D flip-flop with set and reset

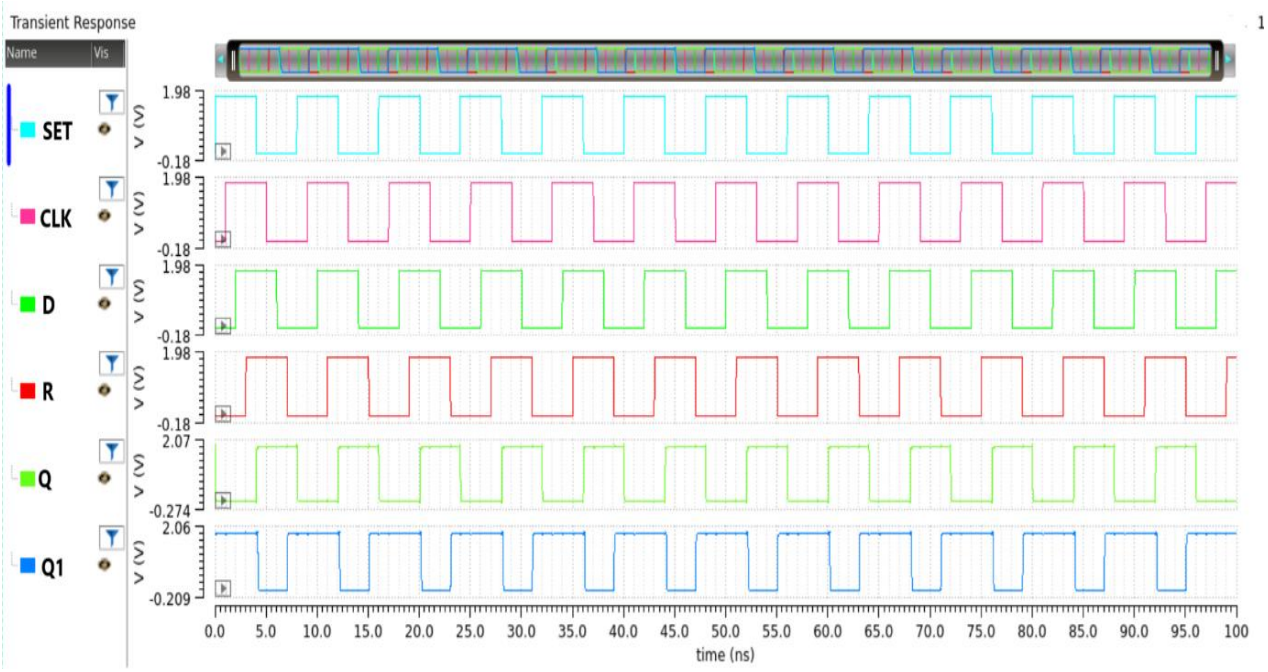


Figure 3.5: Edge triggered D flip-flop output

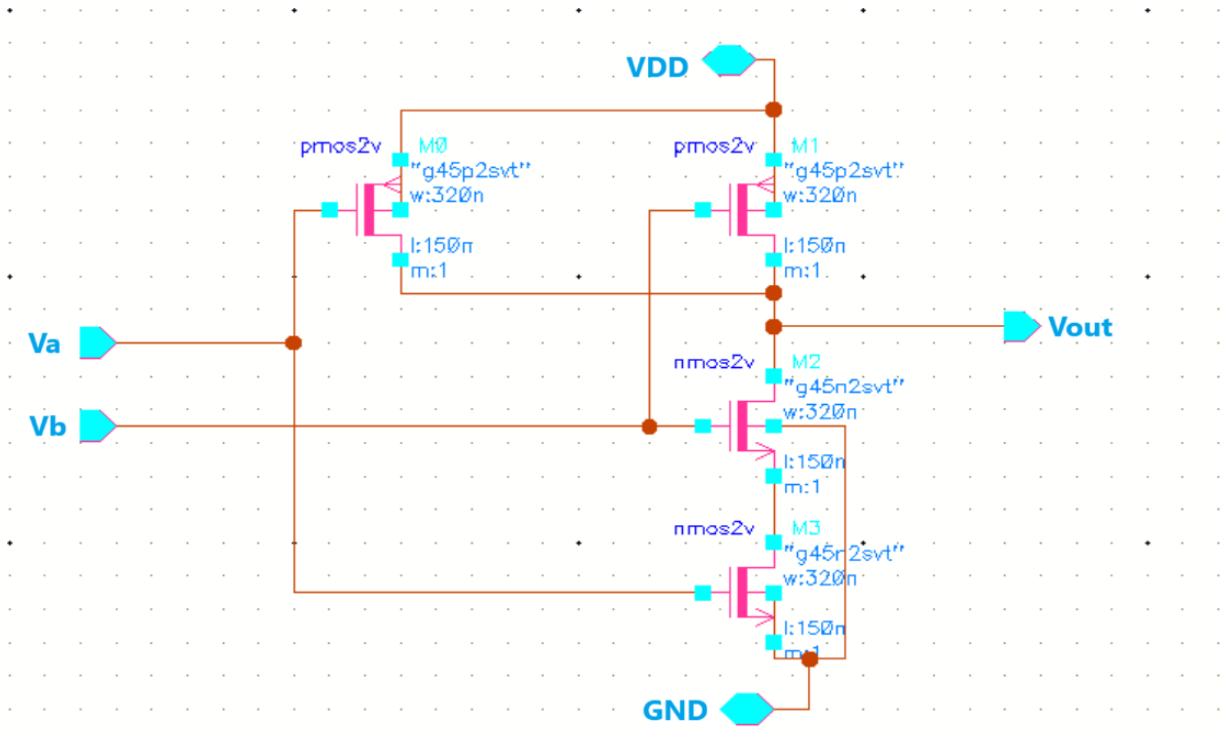


Figure 3.6: 2 Input NAND gate

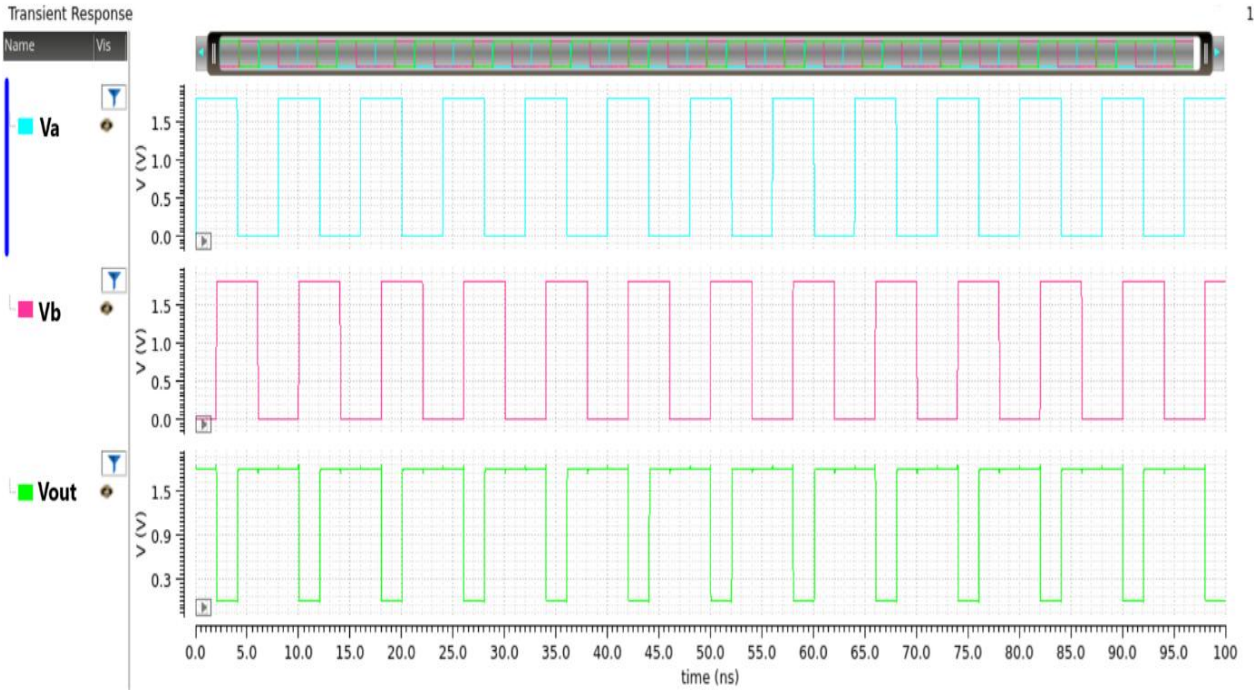


Figure 3.7: 2 input NAND gate output

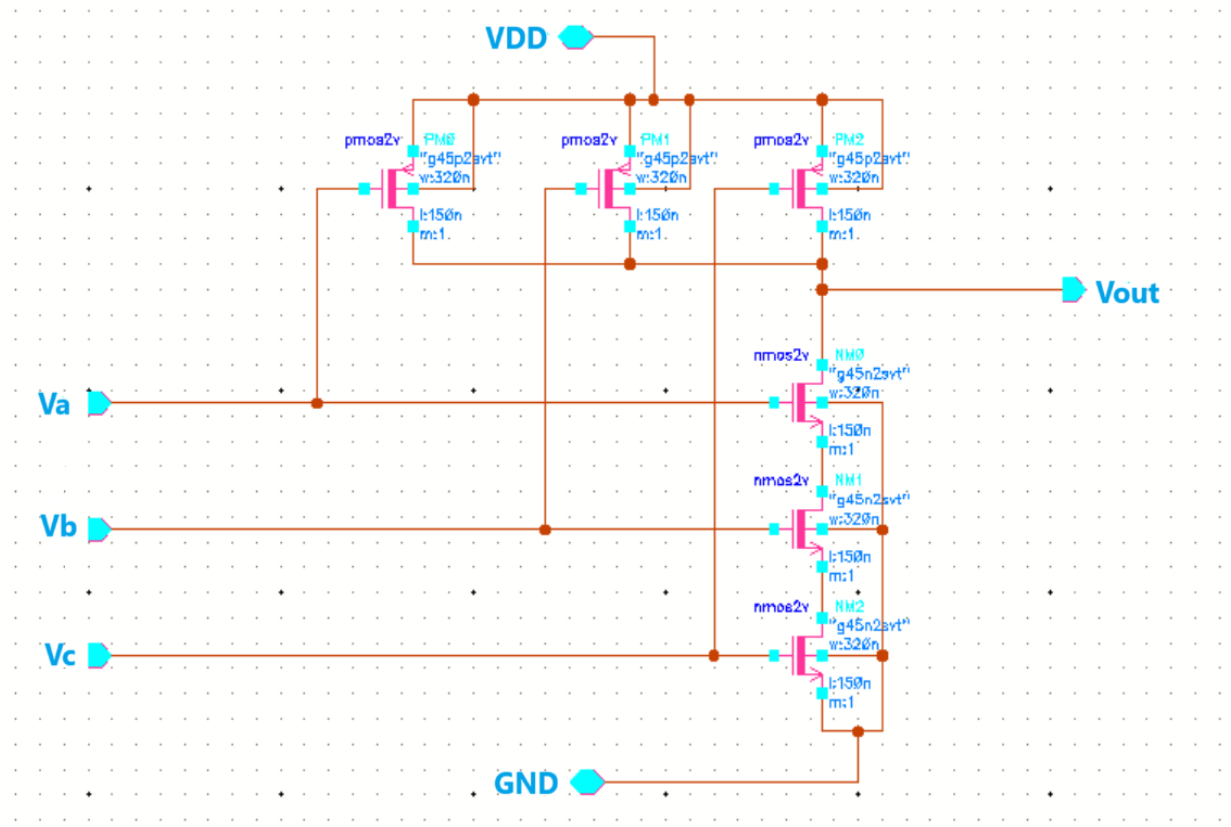


Figure 3.8: 3 Input NAND gate

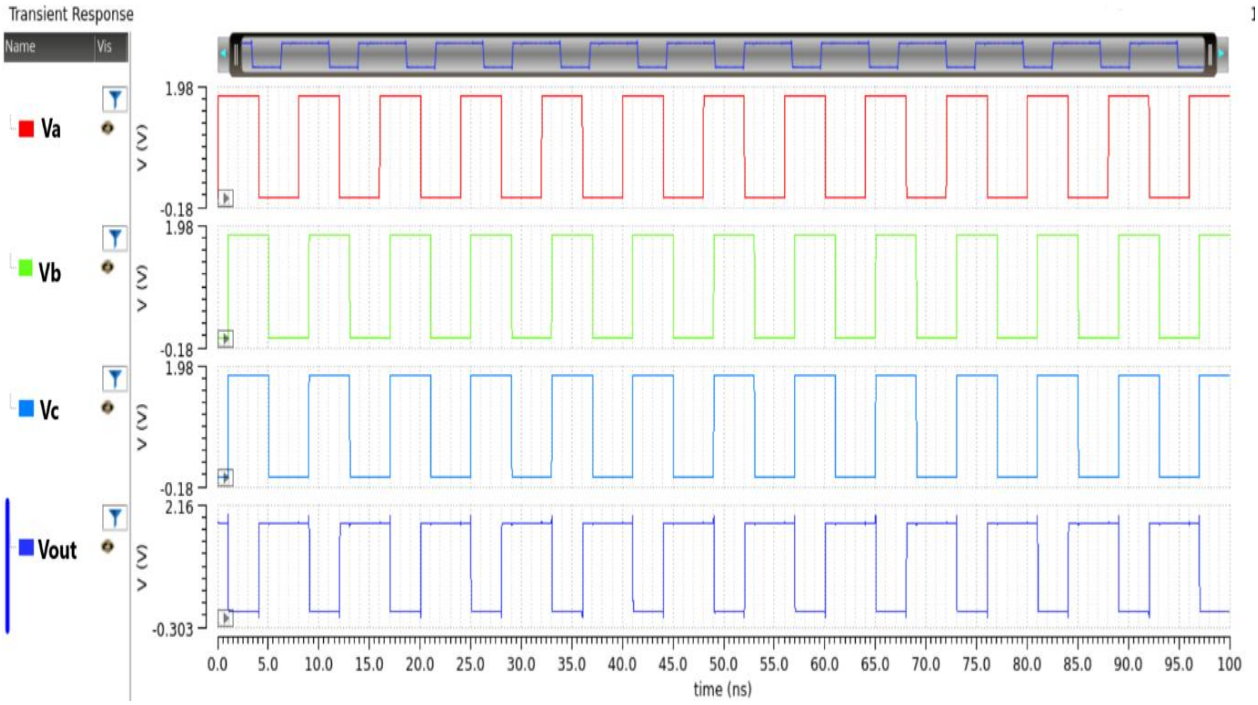


Figure 3.9: 3 input NAND gate output

3.2.2 Design of Charge Pump and Loop Filter

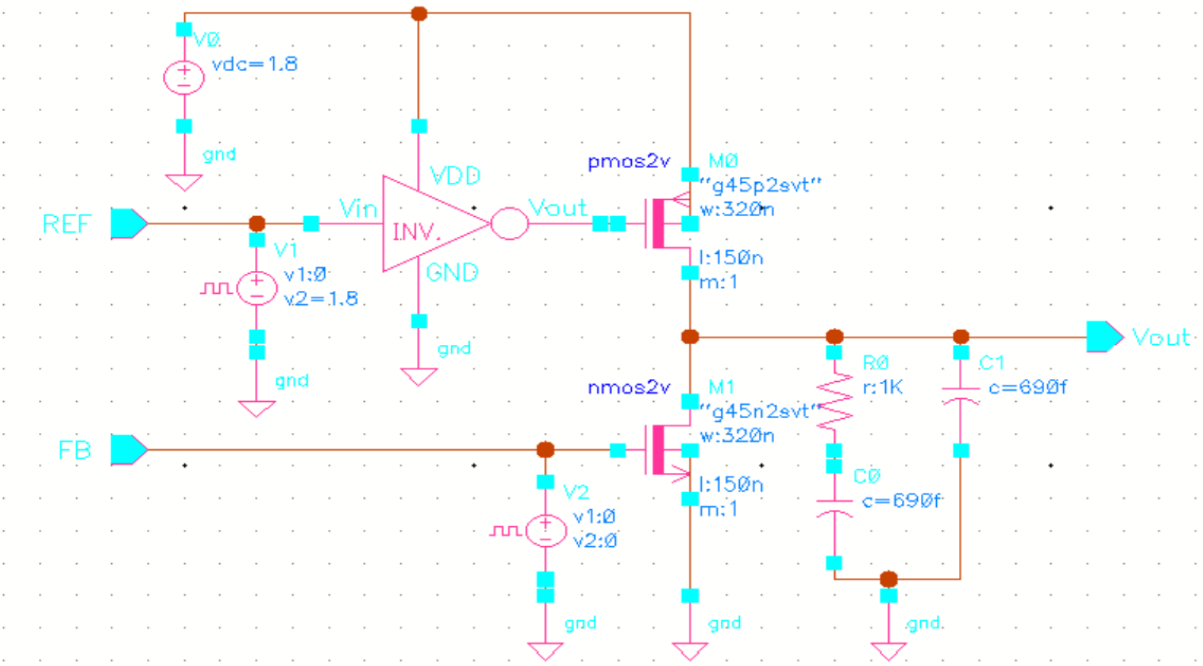


Figure 3.10: Traditional charge pump

The traditional charge pump follows a single-ended design, where a PMOS transistor sources current while an NMOS transistor sinks current. This architecture is simple but comes with certain limitations.One major drawback is higher charge sharing issues, primarily caused by direct switching. The charge injection and clock feedthrough effects can lead to degraded linearity, affecting overall performance. Additionally, current mismatches between the PMOS and NMOS transistors result in different UP and DOWN currents, leading to imbalance. Another concern is the larger dead zone, which occurs when transistor delays are mismatched. In such cases, the PLL may fail to respond to small phase errors, leading to slower lock times and increased jitter. This design suffers from higher reference spurs, as the non-differential architecture does not effectively cancel out switching noise and leakage currents. This results in increased spurs at the reference frequency, which can degrade the overall phase noise performance of the PLL.

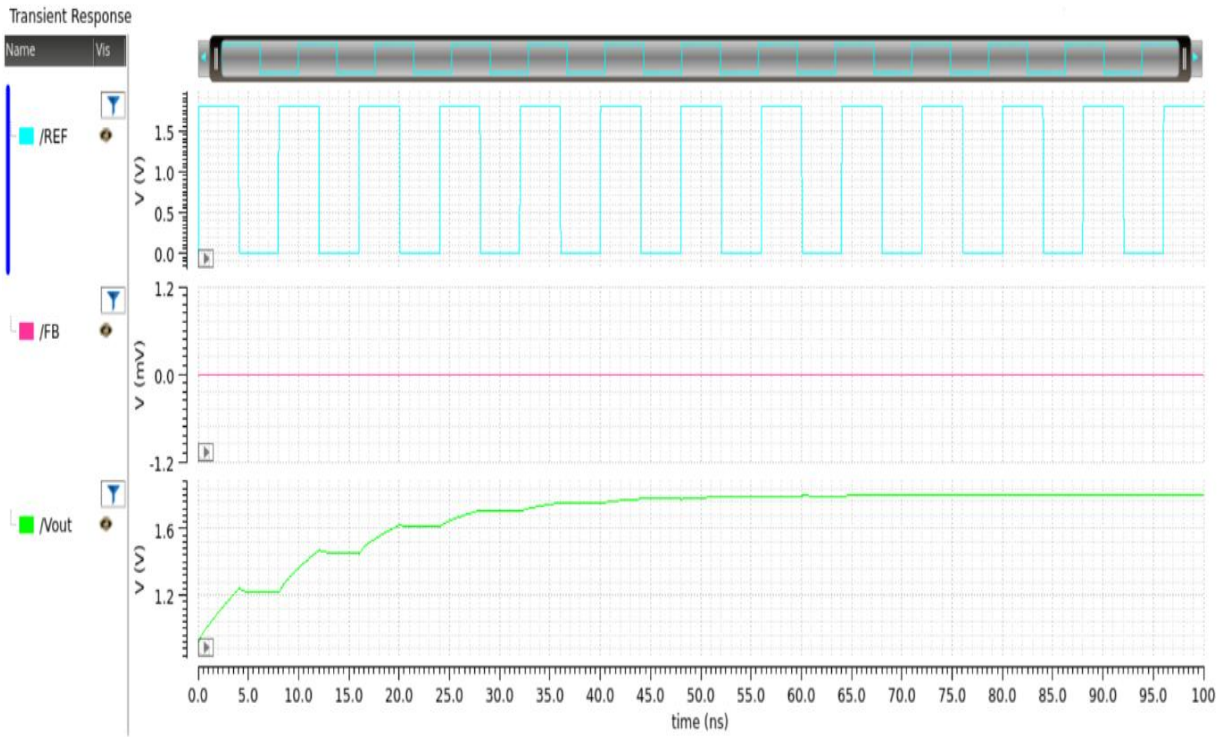


Figure 3.11: Traditional charge pump Charging

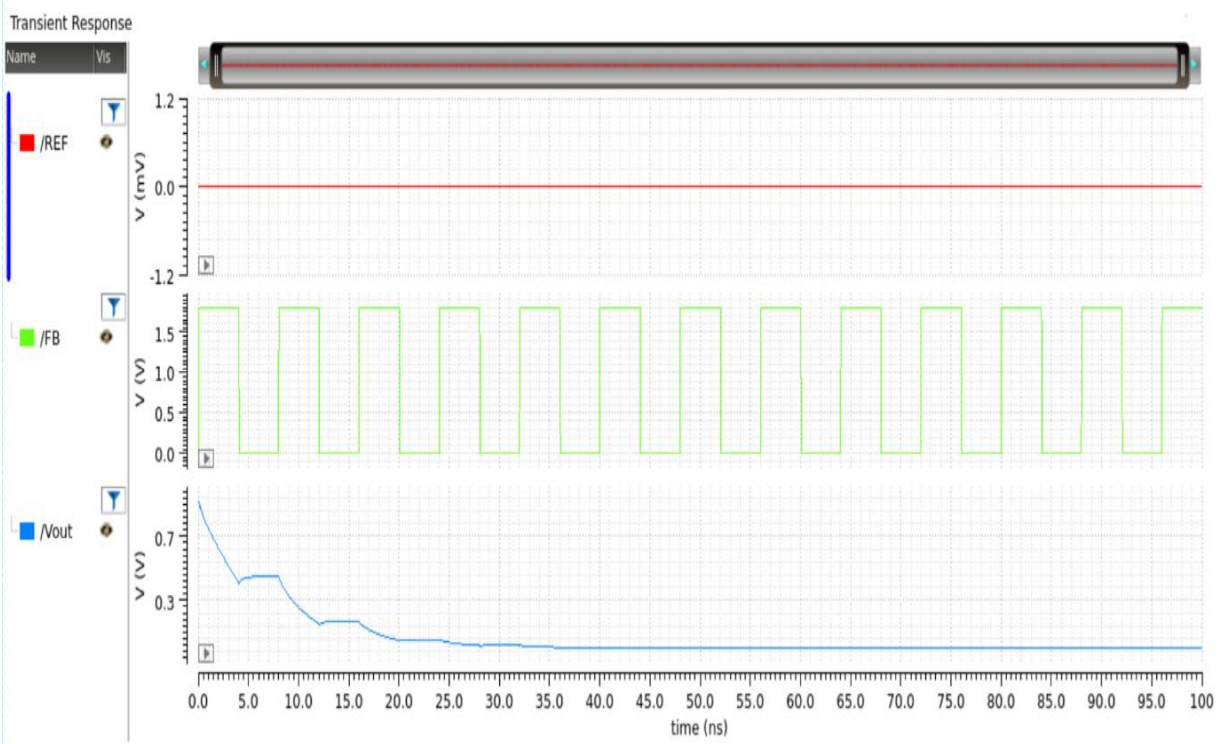


Figure 3.12 Traditional charge pump Discharging

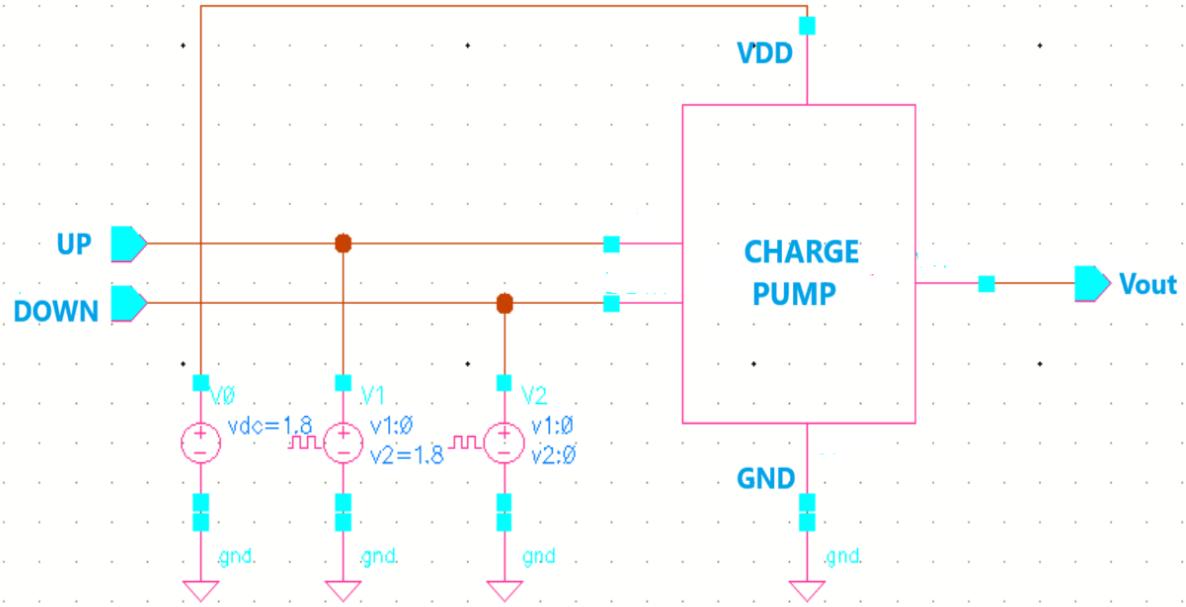


Figure 3.13: Charge pump

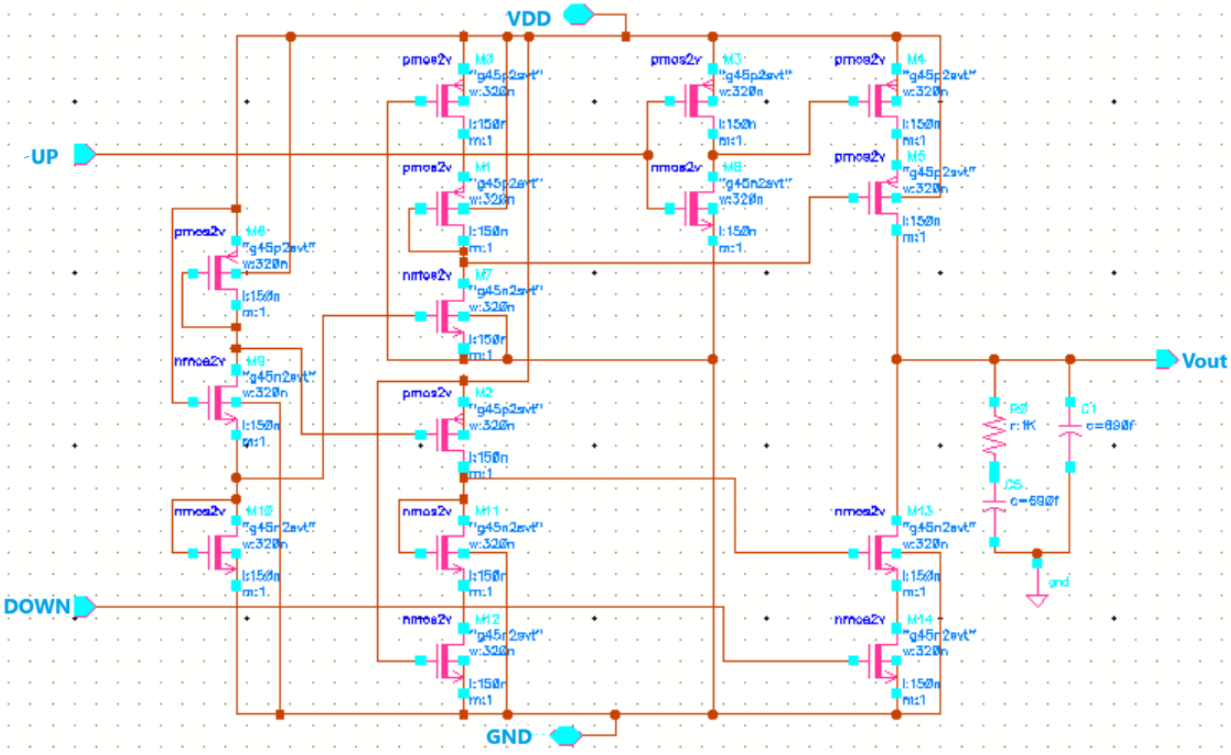


Figure 3.14: Charge pump circuit diagram integrated with loop filter

The adopted charge pump for this project follows a differential structure, utilizing two PMOS transistors act as current sources (UP path) and two NMOS transistors act as current sinks (DOWN path) to create a fully differential architecture. This design ensures better symmetry in both sourcing and sinking currents, improving overall charge balance. One of its key advantages is improved current matching, achieved through mirrored PMOS and NMOS transistors. This approach helps to balance charge injection, thereby reducing charge-sharing effects and contributing to lower phase noise in the PLL system.

Additionally, this design significantly reduces charge injection, as the more symmetrical ON-OFF timing enhances phase error correction. This improvement effectively minimizes the dead zone, ensuring that even small phase differences receive appropriate correction. Another major benefit is the lower reference spurious tones, as the differential architecture effectively minimizes reference spurs caused by mismatches and switching transients. This results in better spectral purity and improved PLL performance.

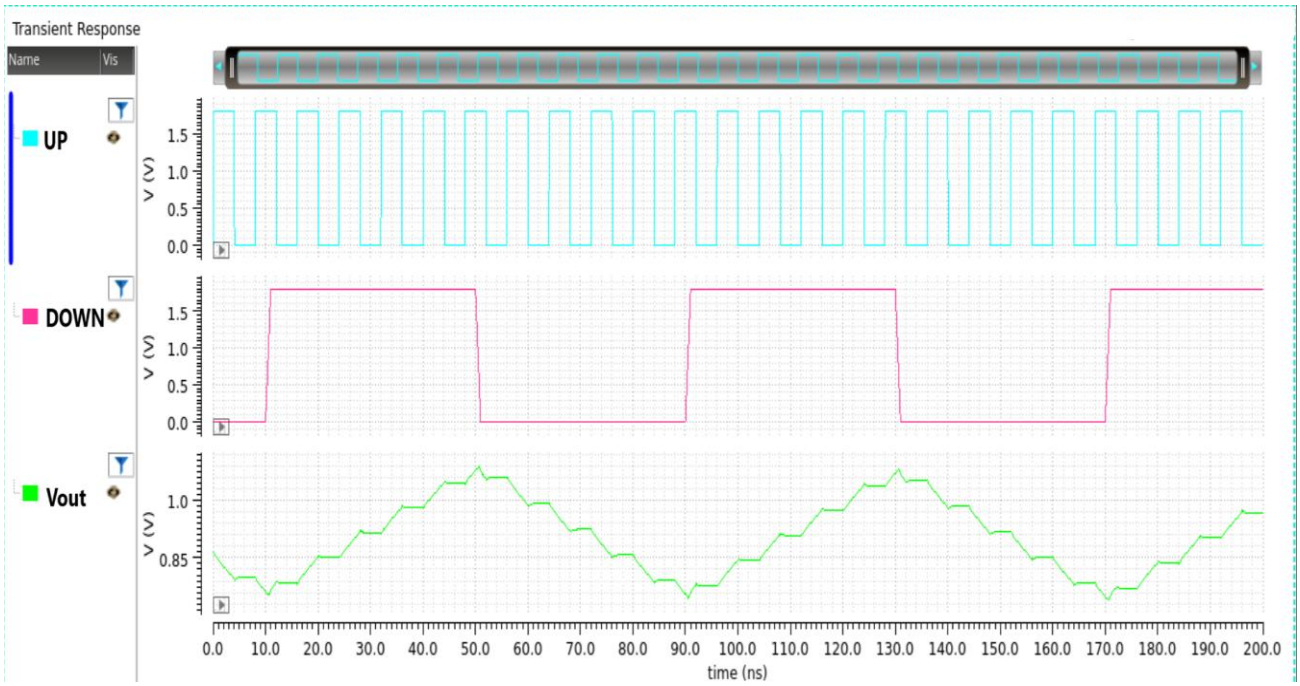


Figure 3.15: Charge Pump Output (Charging and Discharging)

3.2.3 Design of Current Starve Voltage Controlled Oscillator

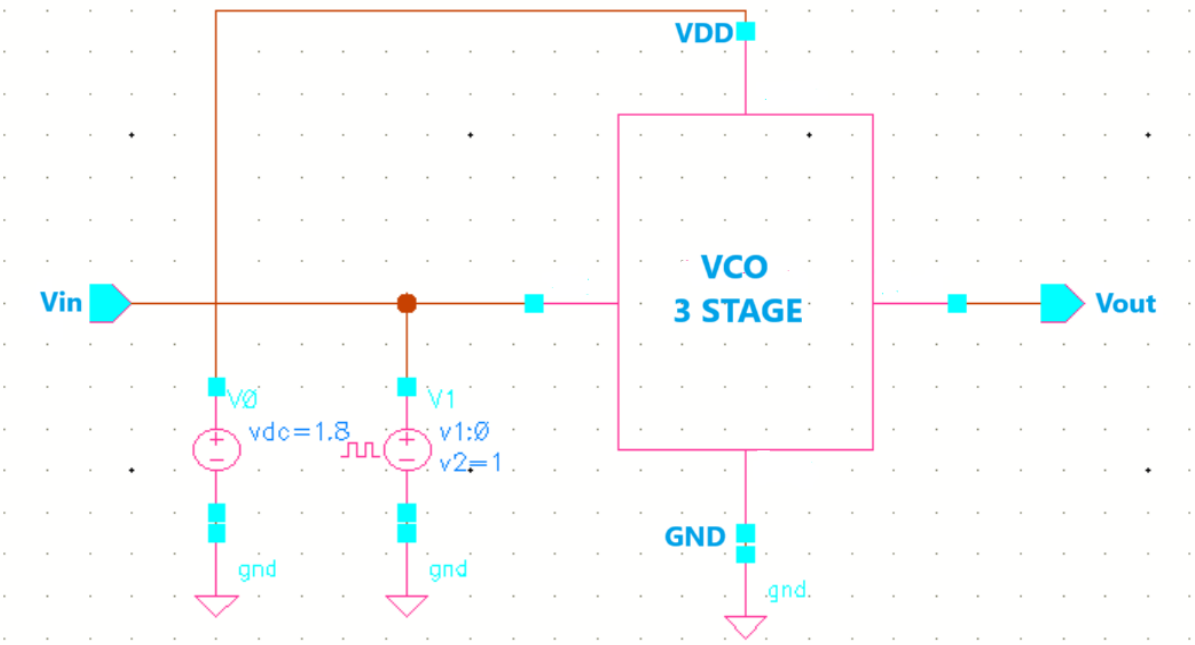


Figure 3.16: Current Starve Voltage Controlled Oscillator

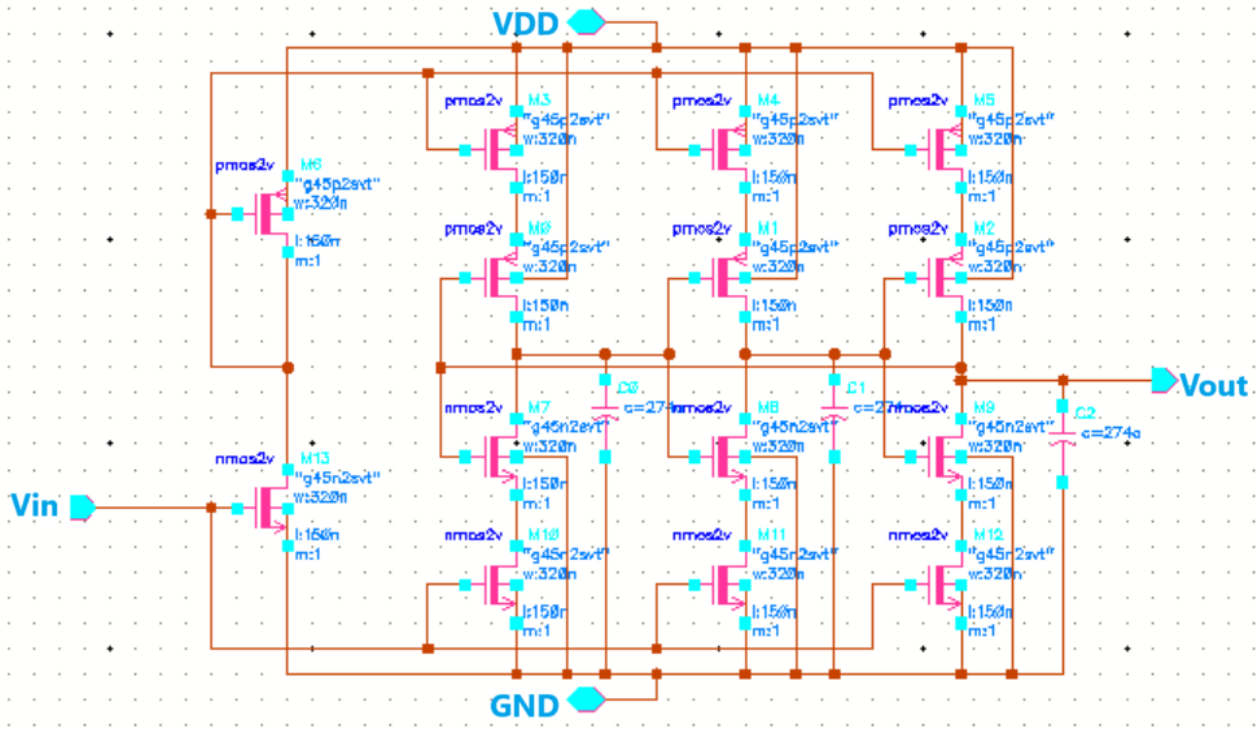


Figure 3.17: Current Starve Voltage Controlled Oscillator circuit diagram (3 stages)

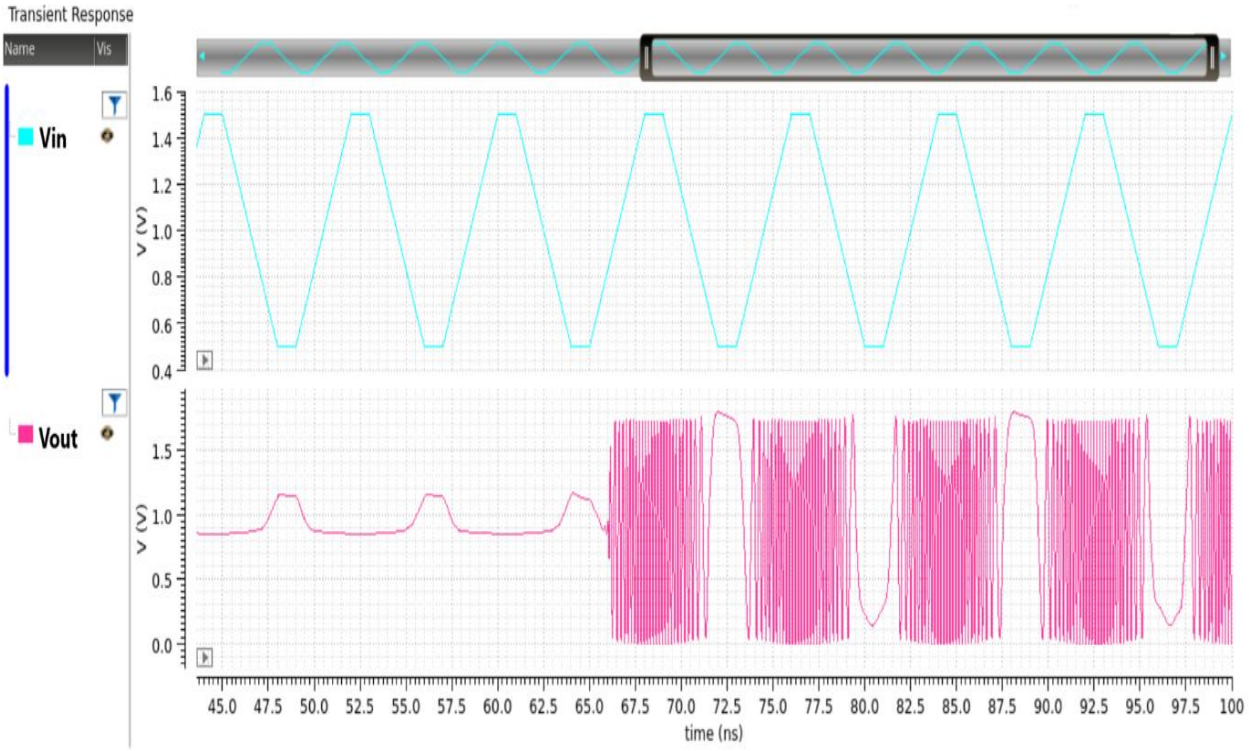


Figure 3.18: CSVCO output

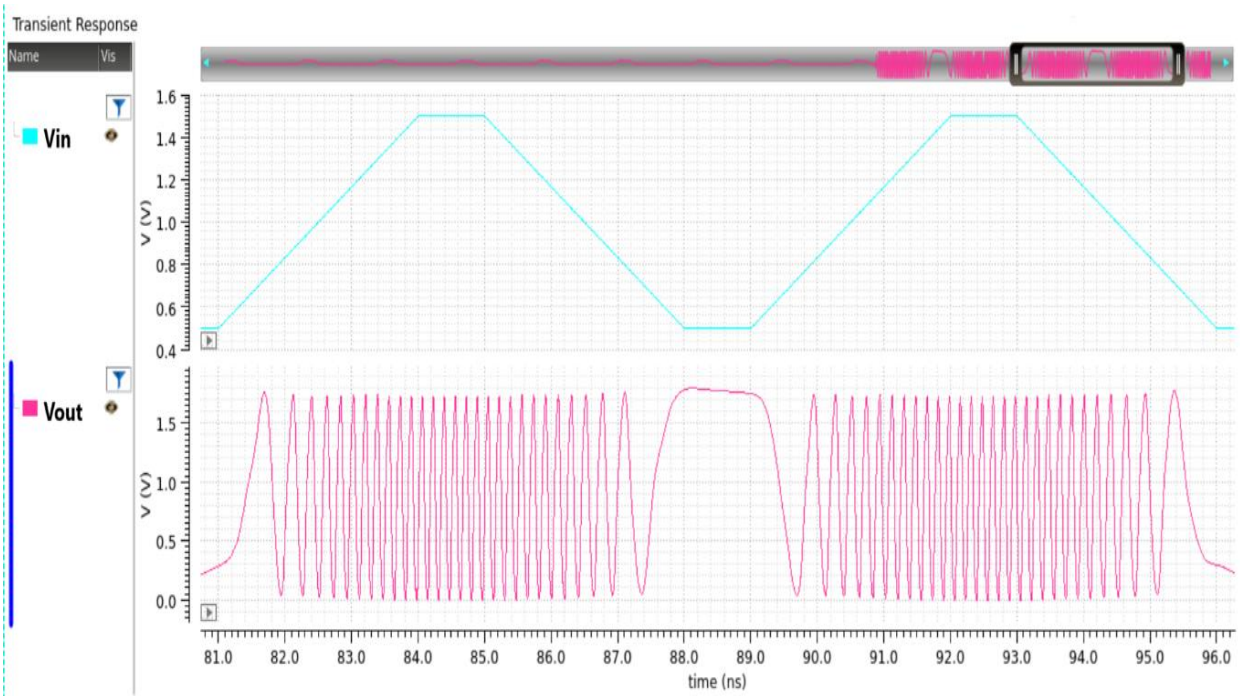


Figure 3.19: CSVCO output (zoomed view)

Voltage vs frequency table:

Voltage	Frequency
0.6	0.125
0.7	0.403
0.8	1.002
0.9	1.492
1.0	2.00
1.1	2.325
1.2	2.527
1.3	2.670
1.4	2.763
1.5	2.839

Table 3.1: CSVCO voltage vs frequency table

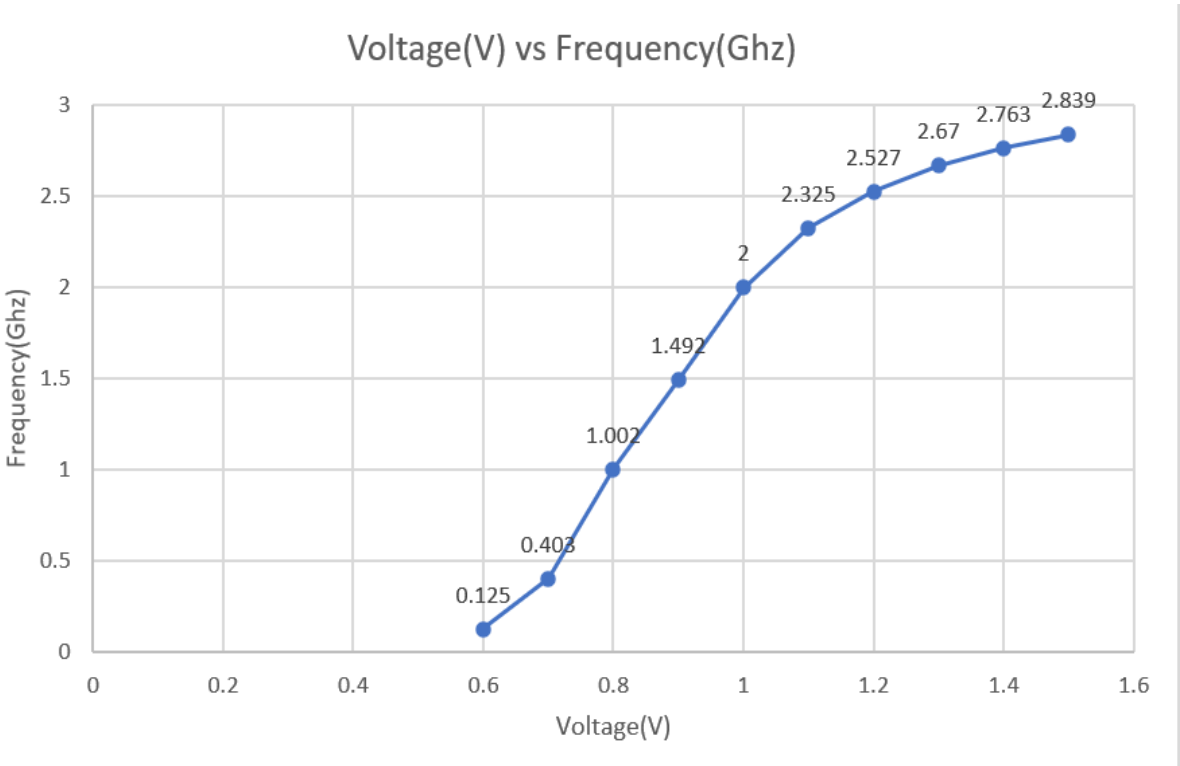


Figure 3.20: Voltage vs Frequency plot

3.2.4 Design of Frequency Divider

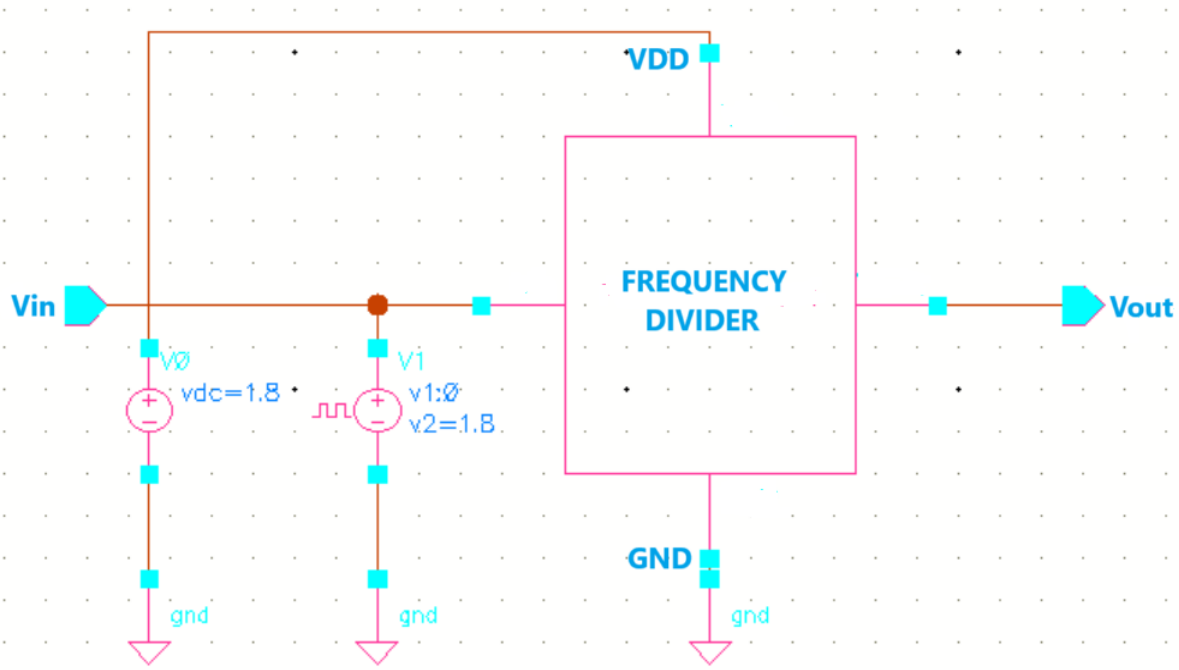


Figure 3.21: Frequency Divider

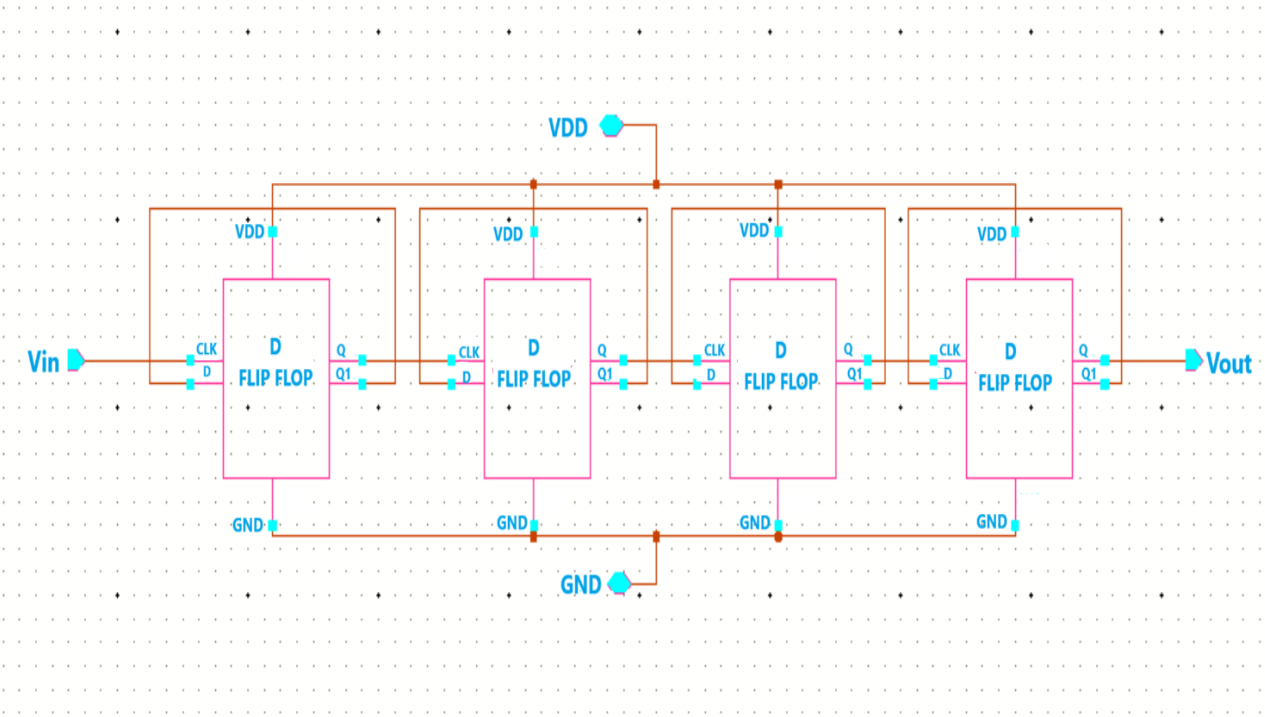


Figure 3.22: Frequency Divider circuit diagram

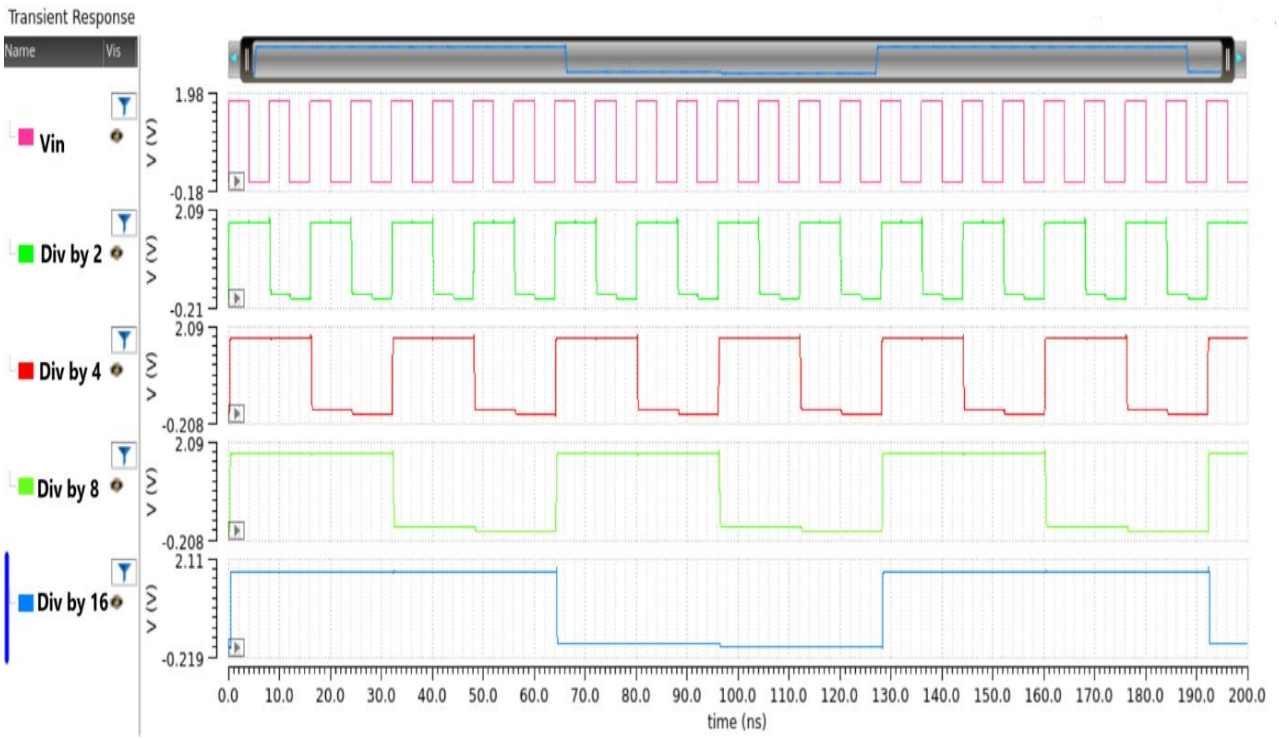


Figure 3.23: Frequency Divider Output (Vin is divided by a factor 16)

3.2.5 Phase locked Loop

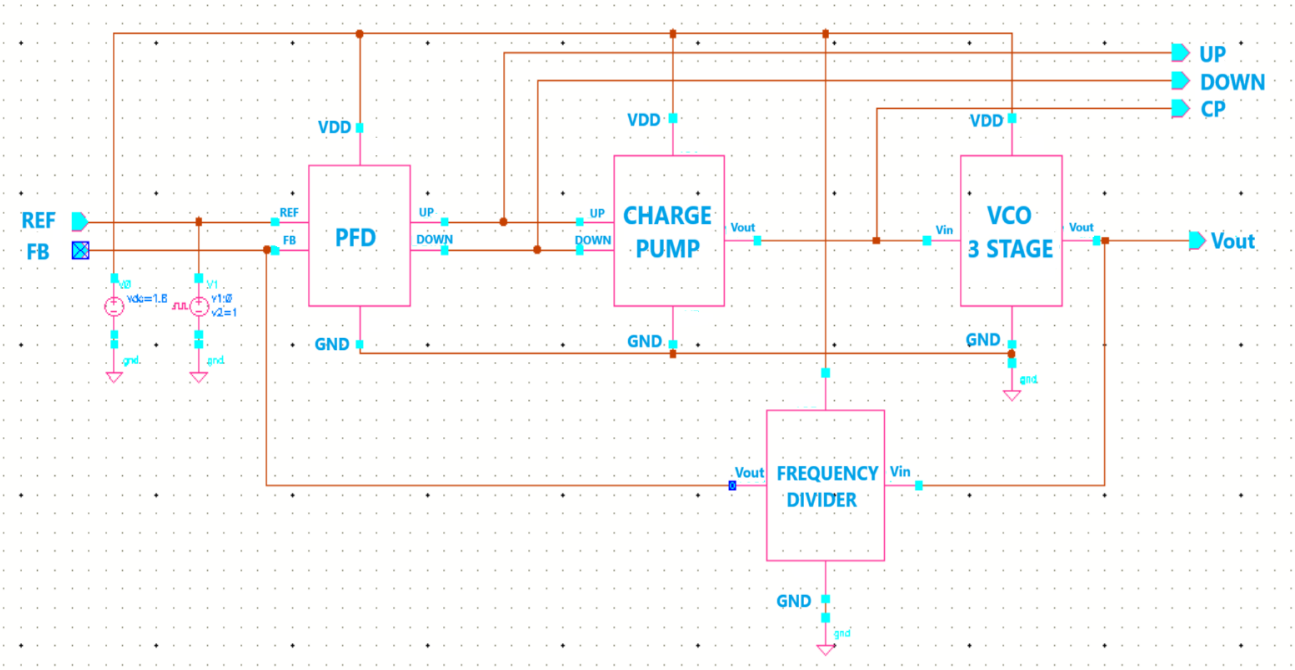


Figure 3.24: Phase Locked Loop Circuit Diagram

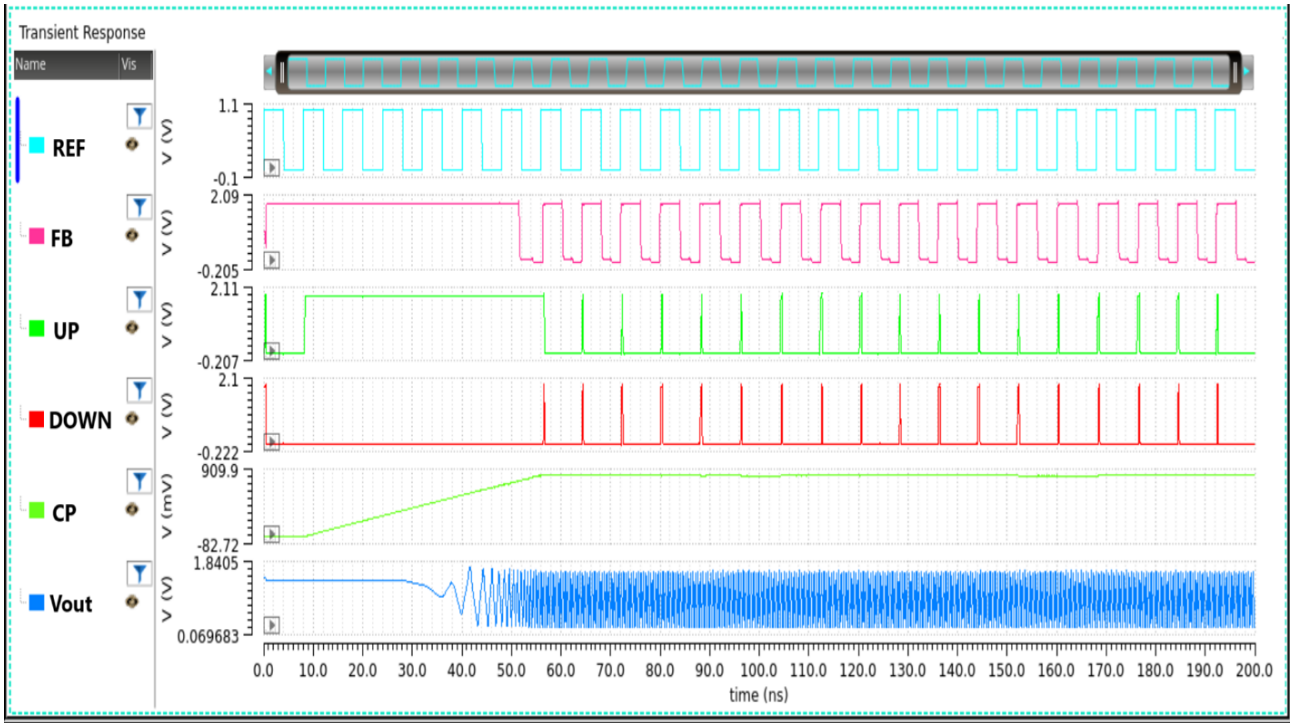


Figure 3.25: PLL output

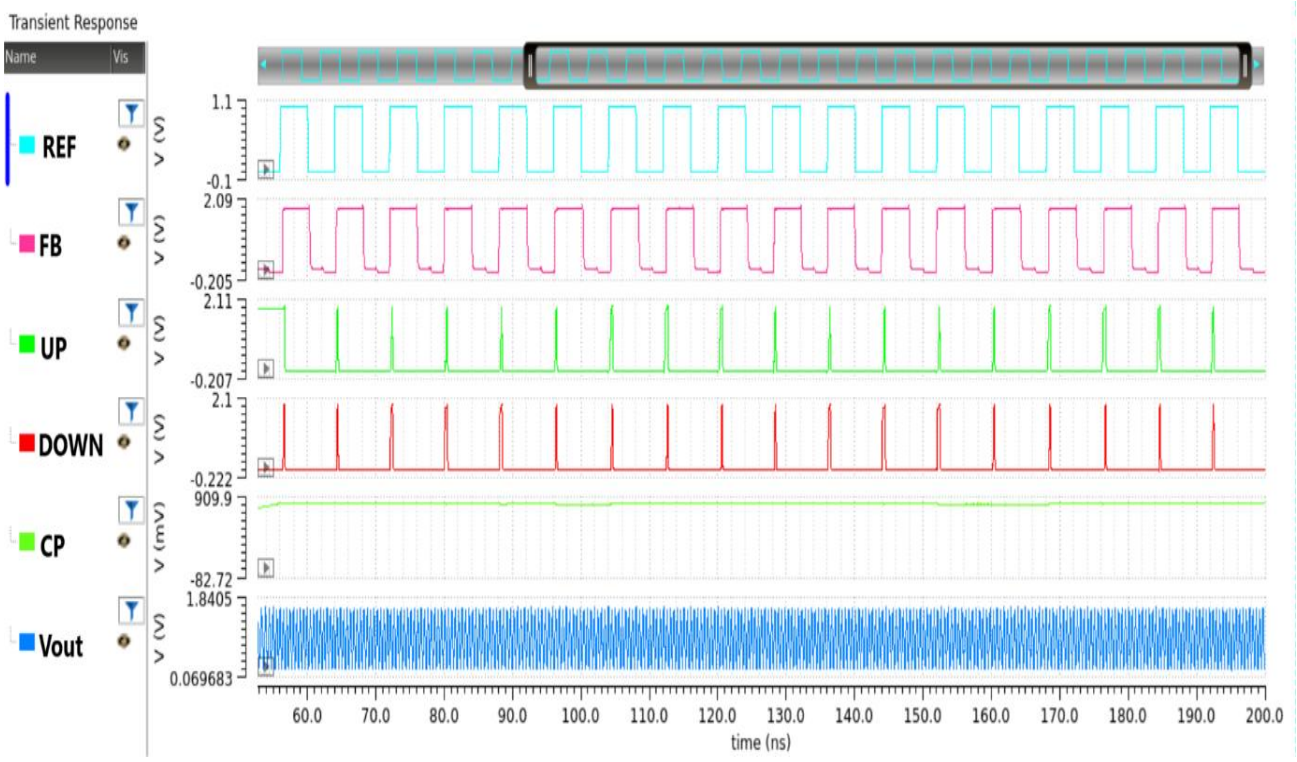


Figure 3.26: PLL output (Locked state)

Chapter 4

Performance Analysis

4.1 Lock Time

Lock Time t_{lock} is the time the PLL takes to transition from an unlocked state to a locked state, where the output signal matches the frequency and phase of the reference signal.

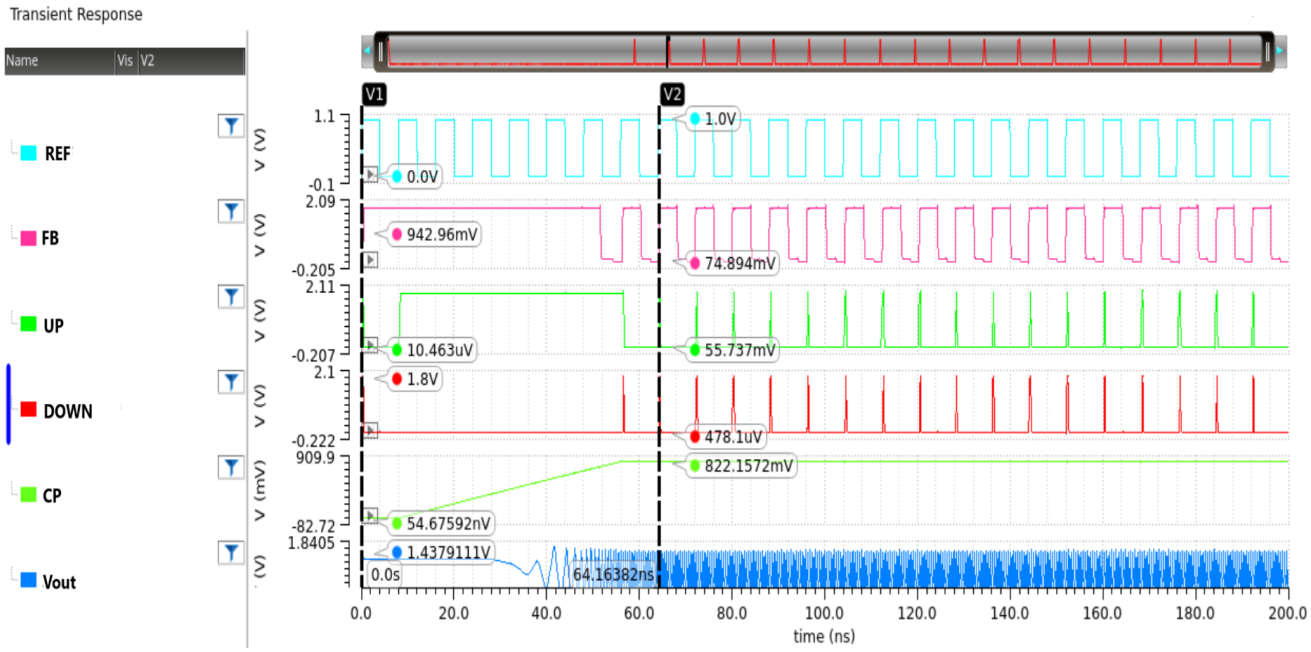


Figure 4.1: PLL lock time

Lock time t_{lock} = (64.16382 - 0) ns

= 64.16382 ns

4.2 Lock Range

The frequency range over which the PLL can achieve phase lock with the reference signal. It is the range till what the PLL can maintain a locked state.

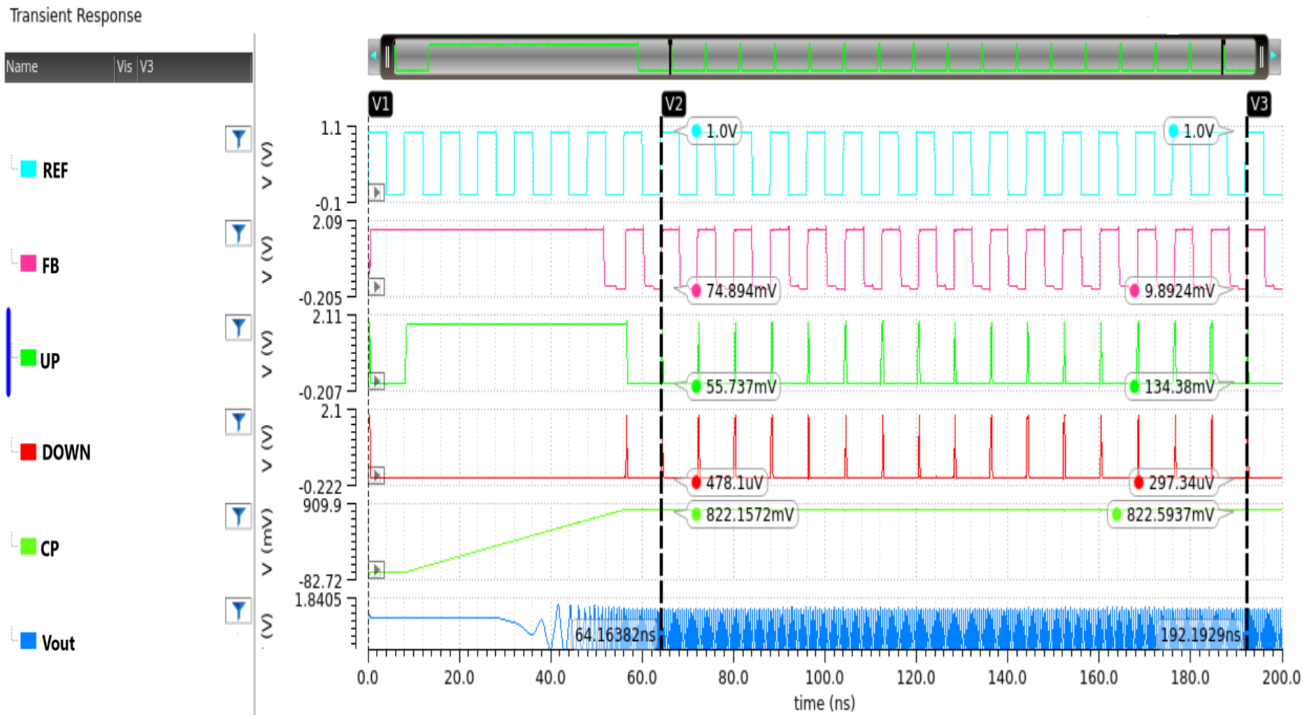


Figure 4.2: PLL lock range

Lock range = (192.1929- 64.16382) ns

=128.02908 ns

4.3 Phase Margin and Phase margin

Phase Margin:

The amount of phase shift that can be introduced in the phase locked loop before it become unstable.

Gain Margin:

The amount of gain that can be increased before the phase locked loop become unstable.

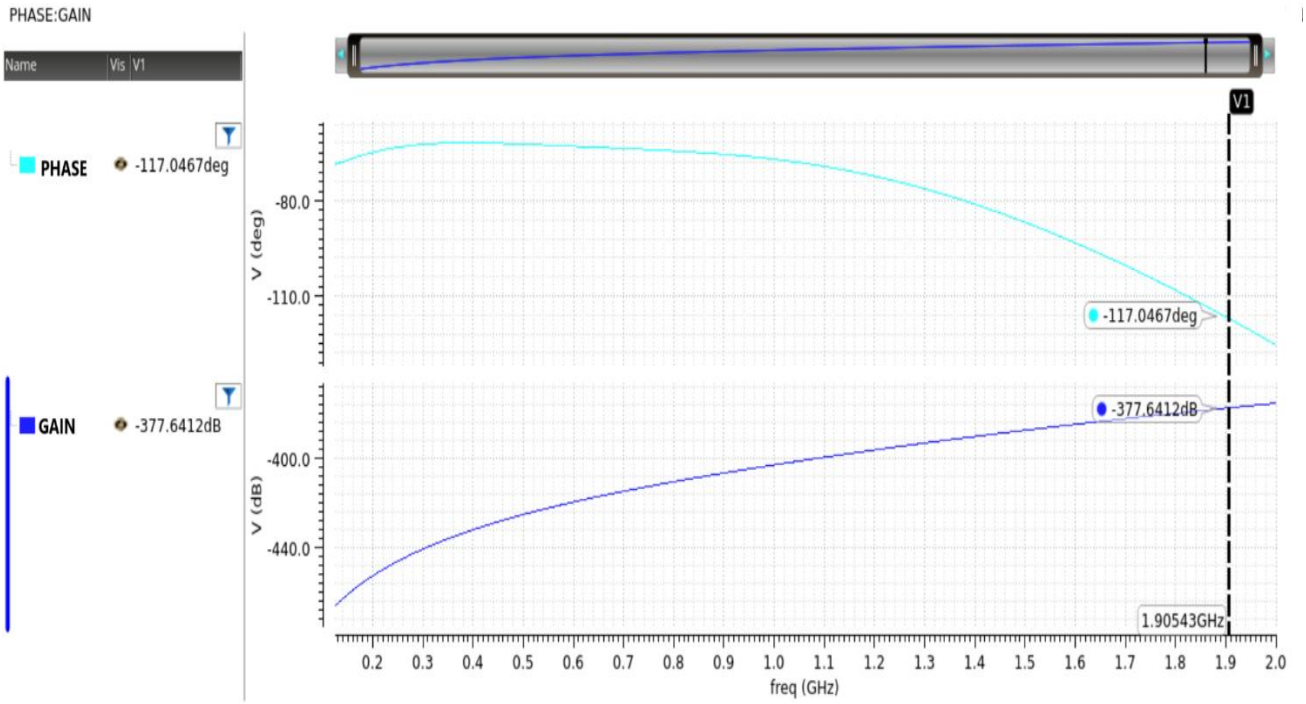


Figure 4.3: Phase Margin and Gain Margin

Phase margin = 180+ Phase at ωgc degree (4.1)

= 180+ (-117.0219) degree

= 62.9781 degree

GM (dB) = −(Gain at Phase Crossover Frequency(dB)) (4.2)

= - (-377.6477) dB

= 377.6477 dB

4.4 Jitter

Jitter refers to the small, unwanted variations in the timing of a signal's transitions from their ideal positions in time. It is the amount of timing variation in the PLL output signal.

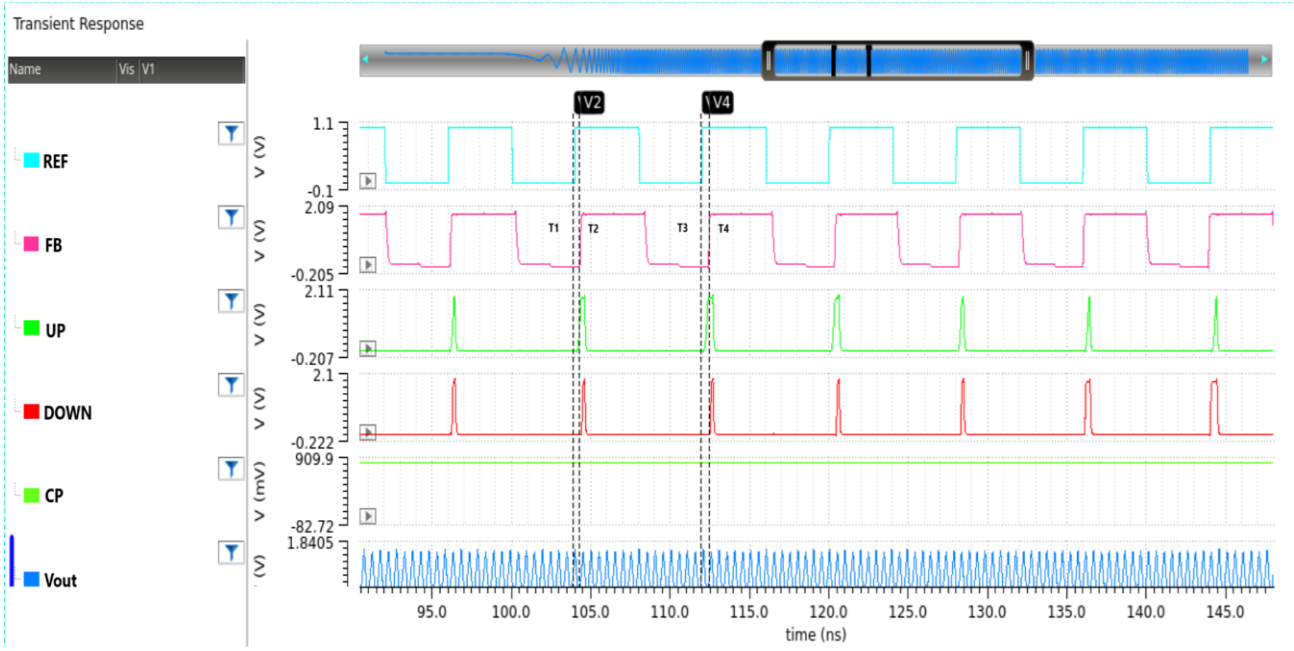


Figure 4.4: jitter analysis

Here, T1 = 103.9114 ns

T2 = 104.3079 ns

T3 = 111.9747 ns

T4 = 112.4373 ns

Timing deviations; t_1 = (104.3079 - 103.9114) ns

= 0.3965 ns

t_2 = (112.4373 - 111.9747) ns

= 0.4626 ns

Peak to peak jitter:

The difference between maximum and minimum jitter value is the pak to peak jitter.

Peak to peak jitter = $t_2 - t_1$ ns (4.3)

= (0.4626 - 0.3965) ns

= 0.0661 ns

Mean Jitter

Mean jitter is the avarage of all jitter values.

Mean jitter = $\frac{\sum(jitter\ values)}{number\ of\ jitter\ values}$ ns (4.4)

= $\frac{0.3965+0.4626}{2}$ ns

= 0.42955 ns

RMS Jitter

It is the statistical measure of the standard deviation of the timing variations from the mean value.

RMS Jitter = $\sqrt{\frac{\sum(jitter\ values)^2}{number\ of\ jitter\ values}}$ ns (4.5)

= $\sqrt{\frac{(0.3965)^2+(0.4626)^2}{2}}$ ns

= 0.4308 ns

Performance Analysis Summary:

Key Parameters	Obtained Values	Standard Values
Lock Time	64.16382 ns	100ns to 1ms
Lock Range	128.02908 ns	As long the PLL runs after achieving lock states.
Phase Margin Gain Margin	62.9781 degree 377.6477 dB	Phase margin should be between 30 to 70 degrees. [16] Gain margin must be a positive value.
Jitter	Peak to Peak: 0.0661 ns Mean Jitter: 0.42955 ns RMS Jitter: 0.4308 ns	Mean Jitter: 1 ms RMS Jitter: 10 ms

Table 4.1: Performance Analysis Summary

Chapter 5

Conclusion

In this project, the design and analysis of a Phase-Locked Loop (PLL) were successfully implemented to achieve reliable frequency synthesis and phase synchronization. The PLL demonstrated its ability to lock onto an input reference signal and generate a stable output frequency with minimal jitter. Key components, including the phase-frequency detector, charge pump, loop filter, and voltage-controlled oscillator, were carefully designed and optimized to ensure robust performance under various conditions. The successful operation of the PLL validates its application in critical systems such as communication networks, clock generation circuits, and frequency synthesizers. Further improvements in advanced techniques, such as fractional-N synthesis or digital PLLs, could enhance the functionality and performance, paving the way for use in next-generation systems.

References

1. B.S., Premananda & T.N., Dhanush & Parashar, Shreyas & Bharadwaj, D.. (2021). Design and Implementation of High Frequency and Low-Power Phase-locked Loop. U.Porto Journal of Engineering. 7. 70-86. 10.24840/2183-6493_007.004_0006.

2. Fast Design Exploration of Nanoscale Circuits, Saraju P. Mohanty and Elias Kougnianos NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu/) University of North Texas, Denton, TX – 76207, USA

3. Talwekar, R.H. & Limaye, Shyamkant. (2012). Design of High-Performance Phase Locked Loop for UHF Band in 180 nm CMOS Technology. Research Journal of Applied Sciences, Engineering and Technology. 4. 4582-4590.

4. Sefraoui, Hanane & Salmi, Khalid & Ziyyat, Abdelhak. (2022). Basic Concepts of a Phase-Locked Loop Control System. International Journal of Online and Biomedical Engineering (iJOE). 18. 25-37. 10.3991/ijoe.v18i13.33419.

5. Aly Onsy, Rana. (2022). A Delta-Sigma Fractional-N frequency synthesizer. 10.13140/RG.2.2.20434.63680.

6. Kulkarni, Madhusudan & Bhat, Nagaraj. (2014). Analysis and Design of 1GHz PLL for Fast Phase and Frequency Lock.

7. D. Abramovitch, "Phase-locked loops: a control centric tutorial," Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), Anchorage, AK, USA, 2002, pp. 1-15 vol.1, doi: 10.1109/ACC.2002.1024769.

8. Patel, Nilesh & Modi, Gunjankumar & Gandhi, Priyesh & Naik, Amisha. (2017). Design and Analysis of Phase Locked Loop and Performance Parameters. International Journal of Microelectronics Engineering. 3. 01-10. 10.5121/ijme.2017.3301.

9. Juarez-Hernandez, Esdras & Diaz-sanchez, A. & Tlelo-Cuautle, Esteban. (2003). A 1.35 GHz CMOS wideband frequency synthesizer for mobile communications. 3. II-292 . 10.1109/ISCAS.2003.1205964.

10. T. Oura, Y. Hiraku, T. Suzuk and H. Asai, "Modeling and simulation of phase-locked loop with Verilog-A description for top-down design," The 2004 IEEE Asia-Pacific Conference on Circuits and Systems, 2004. Proceedings., Tainan, Taiwan, 2004, pp. 549-552 vol.1, doi: 10.1109/APCCAS.2004.1412820

11. R. B. Staszewski and P. T. Balsara, "Phase-domain all-digital phase-locked loop," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 52, no. 3, pp. 159-163, March 2005, doi: 10.1109/TCSII.2004.842067.

12. D. Xie, D. Zhang and P. Gao, "Research on phase-locked loop control and its application," 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, 2016, pp. 818-821, doi: 10.1109/ITNEC.2016.7560475.

13. M. D. Sudara, V. S. Wijesinghe, D. M. Serasinghe, J. G. D. A. Thilakaratne and S. Thayaparan, "Implementation and analysis of fast locking 5GHz phase locked loop," 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2016, pp. 16-20, doi: 10.1109/ISCAIE.2016.7575029.

14. Yan Guo and Guang Zeng, "A digital phase locked loop based on frequency self-adaptive," 2015 IEEE 2nd International Future Energy Electronics Conference (IFEEC), Taipei, 2015, pp. 1-5, doi: 10.1109/IFEEC.2015.7361596.

15. S.Anjaneyulu,J.Sreepavani,K.Pramidapadma,N.Varalakshmi,S.Triven,' "Phase Locked Loop Design for Fast Phase and Frequency Acquisition," International Research Journal of Engineering and Technology (IRJET)

16. Barrett, C. (1999) *Fractional/Integer-N PLL Basics*. Texas Instruments Technical Brief SWRA029.

Appendix A.

crystal oscillator: A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating piezoelectric crystal (commonly quartz) to generate a precise and stable frequency.

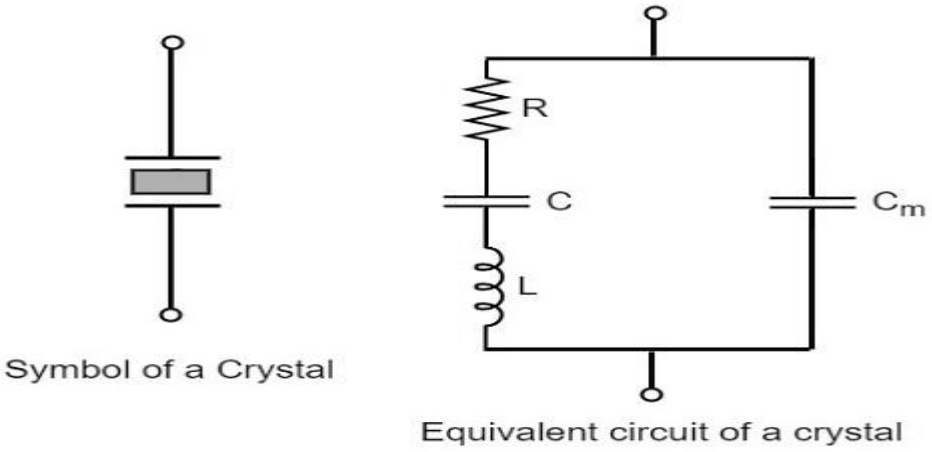


Fig: Crystal oscillator symbol and circuit diagram

Appendix B.

The parameter values of the current starve voltage-controlled oscillator are:

f*osc*	oscillation frequency of CSVCO	2 Ghz
N	number of stages	3
R	internal resistance of MOSFET	304.14 KΩ
C	internal capacitance of every stage	274a F

The parameter values of the frequency divider are:

fout	output frequency of frequency divider	125 Mhz
f*in*	input frequency of frequency divider	2 Ghz
2^n	n is the number of D flip-flop	4