
DESIGN AND SIMULATION OF DELAY LOCKED LOOP IN CADENCE VIRTUOSO

Major Project(EC498) Report

*Submitted in Partial Fulfillment of the Requirements for the Degree of
BACHELOR OF TECHNOLOGY*

in

Electronics and Communication Engineering

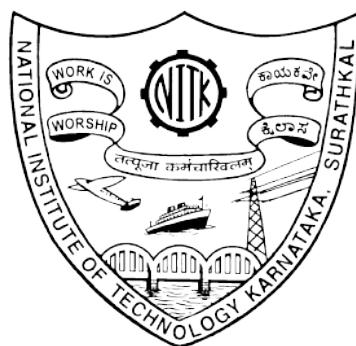
under

Prof. T. Laxminidhi

by

Name: Arvindh Ganesan , Soumik Dutta

Reg.No.: 211EC211, 211EC255



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025
November 2024

DECLARATION

We hereby *declare* that the Project work report entitled **Design and Simulation of Delay Locked Loop in Cadence Virtuoso** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** for the award of the Degree of Bachelor of Technology in Electronics and Communication engineering is a bonafide report of the work carried out by us. The material contained in this Project Work Report has not been submitted to any University or Institution for the award of any degree.

Arvindh Ganesan, Reg. No.- 211EC211

Soumik Dutta, Reg No.- 211EC255

Place: NITK Surathkal

Date:

CERTIFICATE

This is to certify that the Major Project Report entitled **Design and Simulation of Delay Locked Loop in Cadence Virtuoso** supervised by **Prof. Laxminidhi T.**, submitted by **Arvindh Ganesan(211EC211) and Soumik Dutta(211EC255)** as a record of the report presented by them is accepted as the Major Project Report Submission in partial fulfillment of the requirements for the award of Bachelor of Technology in the Department of Electronics and Communication Engineering , National Institute of Technology Karnataka, Surathkal, Mangaluru

Guide:

(Name and Signature with date)

Chairman - DUGC -

(Signature with date and Seal)

ABSTRACT

This project focuses on the design and implementation of a Delay Locked Loop (DLL), a crucial component in modern digital communication and high-speed integrated circuits. Delay-Locked Loops (DLLs) are negative feedback circuits that achieve phase alignment of an output signal to an input reference signal ensuring robust synchronization in systems such as clock distribution networks and clock-data recovery circuits without requiring an oscillator. In certain applications, DLLs are preferred over Phase-Locked Loops (PLLs) due to their inherent advantages, including reduced sensitivity to supply noise and lower phase noise. This report delves into the fundamental design principles of DLLs, highlighting their critical role in achieving precise timing and synchronization in various systems. DLLs are commonly employed in applications like clock generation, data synchronization and memory interfaces particularly in Double Data Rate (DDR) memory technology and SerDes such as USB, Ethernet and PCI, where low jitter and high reliability are critical. The developed DLL architecture is meticulously designed, taking energy efficiency and delay units into considerations, so that it can be effectively applied to applications requiring precise timing and phase control, enhancing overall system reliability.

Contents

1	Introduction	7
2	Motivation	7
3	Literature Review	8
4	Specifications	10
5	Description	11
6	Progress Timeline	24
7	Specific Modifications	25
8	Results and Discussion	27
9	Conclusion and Future Scope	44

List of Figures

1	Phase Frequency Detector Internal Schematic	11
2	Up-Down Counter Schematic	13
3	Delay Line Internal Schematic	14
4	Complete Schematic of the Counter type Delay-Locked Loop	15
5	Successive Approximation Register internal architecture	18
6	Schematic of the PIPO Shift Register	19
7	Complete Schematic of the SAR based Delay Locked Loop	20
8	Transistor level Schematics of NAND Gate and CMOS-Transmission Gate	21
9	Transistor level schematics of various components	22
10	Transistor level schematics of the D-Flip-Flops	23
11	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	28
12	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	29
13	Eye Diagram at 1GHz and associated measurements	29
14	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	30
15	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	31
16	Eye Diagram at 1GHz and associated measurements	31
17	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	32
18	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	33
19	Eye Diagram at 1GHz and associated measurements	33
20	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	34
21	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	35
22	Eye Diagram at 1GHz and associated measurements	35
23	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	36
24	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	37
25	Eye Diagram at 1GHz and associated measurements	37
26	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	38
27	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	39
28	Eye Diagram at 1GHz and associated measurements	39
29	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	40

30	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	41
31	Eye Diagram at 1GHz and associated measurements	41
32	DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps	42
33	Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock	43
34	Eye Diagram at 1GHz and associated measurements	43

1 Introduction

A Delay-Locked Loop(DLL) is a negative-feedback based circuit whose function is to synchronise the phase of two signals, which includes an input reference clock and the DLL output signal. In this project, we have implemented an all-digital DLL. The operation of a DLL relies on several critical components: the Phase Frequency Detector (PFD), the Up-Down Counter, the Delay Line, and the Successive Approximation Register(SAR), each playing a vital role in accurately receiving and sampling signals to achieve effective phase lock, even at certain high frequencies. The PFD compares the phase and frequency of the input reference and feedback signals, thereby delivering a clear picture about the lead or lag characteristics of the active signals. If a phase difference is detected, the PFD generates either an "up" or "down" signal to adjust the delay, ensuring the output remains phase-locked to the input, unaltered by any unintended generation of noise due to false signal edges during circuit operations.

The Up-Down Counter circuit translates these "up" and "down" signals from the PFD into a control mechanism for adjusting the delay in the system. Depending on the PFD's output, the counter increases or decreases its count, which in turn controls the delay line. The counter essentially functions as the digital equivalent of the integrator block used in an analog DLL. The delay line on its part, is responsible for introducing the necessary time delay in the output signal to match the phase of the input signal. By adjusting the delay incrementally based on feedback from the PFD and Up-Down Counter, the DLL achieves precise phase alignment, serving to the requirements of stability and accuracy in digital circuits.

The SAR-Register based DLL has the characteristic of being able to achieve faster lock-time than the basic counter type DLL.

2 Motivation

The motivation for choosing the Delay-Locked Loop(DLL) as the focus of our project stems from the increasing importance and rapid advancements in high-speed, low-noise, and reliable timing solutions in modern and future technologies. As digital systems evolve, they demand greater precision and efficiency, particularly in high-performance applications including DDR memory, microprocessors and high-speed communication networks. DLLs excel in these areas by offering phase alignment without the need for oscillators, reducing jitter and sensitivity to supply noise, hence ensuring more stable performance at higher frequencies. This project not only equips us with valuable skills for designing advanced systems but also contributes to the broader engineering efforts to develop cutting-edge solutions for the next wave of technological innovation. The decision on implementing a digital DLL instead of an analog design is based on the fact that analog designs suffer from multiple disadvantages such as Charge-Pump noise, Charge-Pump current source mismatch, effect of parasitics on the integrator precision and non-linearity in the delay-line transfer function.

3 Literature Review

A novel architecture for phase frequency detector (PFD) which eliminates the blind zone effect as well as the dead zone for a charge-pump phase-locked loop (CP-PLL) has been presented in [1]. This PFD is designed in 65 nm CMOS technology, and its functionality is verified across process, voltage and temperature variations. Achieved maximum frequency of operation (F_{max}) is 3.44 GHz which is suitable for high reference clocked fast settling PLLs. Proposed PFD consumes 324 uW power from 1.2 V supply at maximum operating frequency. The area occupied by proposed circuit layout is $322.612 \text{ } \mu\text{m}^2$.

A 2.5GHz, 30 mW, 0.03mm^2 , all-digital delay-locked loop (ADDLL) in 0.13nm CMOS technology is presented in [2]. The tri-state digital phase detector suppresses the dithering phenomenon and reduces the output peak-to-peak jitter for a counter-controlled digital DLL. The lattice delay unit has both a small delay step and a fixed intrinsic delay of two NAND gates. A modified successive approximation register-controller reduces the locking time and allows the DLL to track the process, voltage, temperature, and load variations. This ADDLL locks in 24 cycles and has a closed-loop characteristic. The measured peak-to-peak jitter is 14 ps at 2.5 GHz.

A successive approximation register-controlled delay-locked loop (SARDLL) has been described in [3] and fabricated in a 0.25nm standard n-well DPTM CMOS process to realize a fast-lock clock-deskew buffer for long distance clock distribution. This DLL adopts a binary search method to shorten lock time while maintaining tight synchronization between input and output clocks. The measured lock time of the proposed SARDLL is within 30 clock cycles at 100-MHz clock input. The power dissipation is 3.3 mW (not including off-chip driver's) at a 1.1-V supply voltage while the measured rms and peak-to-peak jitter are 11.3 ps and 95 ps, respectively.

A 1.5–3.3GHz, 7mW, all-digital delay-locked loop (ADDLL) designed in a UMC 130nm CMOS technology is presented in this paper [4]. The proposed DLL uses the modified successive approximation register to control a coarse delay line, which enables wider operating frequency range and small delay. The inverter-based fine delay line is controlled by an XOR-based up/down counter with dead-zone free phase detector to overcome the dead-zone problem of conventional phase detectors. The D-type flip-flops in the phase detector are modified to detect sub-ps level delay difference between the input and output clocks, so that a delay resolution of better than 1ps is achieved in the proposed design. The combination of both coarse and fine locking processes gives outstanding performance in terms of residual phase difference and output jitter. The overall design occupies 0.0077 mm² area. The experimental results show that the peak-to-peak and root mean square jitters are 12 and 1.629ps at 3.3GHz, respectively, while the input jitter is 2.6ps peak-to-peak and 612fs rms.

Delay-locked loops (DLLs) can be considered as feedback circuits that phase lock an output to an input without the use of an oscillator. This DLL has been described in [5]. In some applications, DLLs are necessary or preferable over phase-locked loops (PLLs), with their advantages including lower sensitivity to supply noise and lower phase noise. This article deals with fundamental DLL design concepts. The origins of DLLs can be traced

to a paper published in 1961[8]. The authors present the topology shown in Figure 1 as a “delay-lock discriminator” operating on random signals. The feedback loop consists of a controlled delay line, a multiplier acting as a phase detector (PD), and a low-pass filter. The use of DLLs in modern CMOS design evidently began with the work by Bazes in 1985[6] and Johnson and Hudson in 1988[7]

Unlike common delay lines, which are implemented in hybrid technologies, the Synchronous Delay Line is implemented in MOS in [6]. Thus the SDL obviates the need in certain applications for separate delay-line components, since it can be integrated directly into LSI or VLSI components implemented in common MOS technologies. The SDL was implemented for the first time in a commercial DRAM controller, in which it provided precision trigger pulses for the DRAM control signals. The SDL utilizes the system clock as a delay reference. The negative feedback that is intrinsic to the SDL design also makes it very insensitive to supply-voltage, temperature, and processing variations. A delay analysis predicts a linear relationship between the delays provided by the taps and the input reference clock. This linear relationship was confirmed experimentally, as was the low sensitivity of the SDL to temperature and voltage-supply variations. A closed-loop analysis defines the circuit parameters that determine stable and optimum operation.

A fully integrated phase-locked loop (PLL) is used to time-align the hi-Z/low-Z transitions of a CMOS CPU and its floating-point coprocessor (FPC), resulting in minimum timing difference (skew) between the two devices at their shared data bus, and decreasing the bus cycle time. This design is analyzed in [7]. The PLL circuit abandons the traditional voltage-controlled oscillator function, instead using a CMOS voltage-controlled delay line to improve noise immunity, ease loop stabilization, and permit dynamically adjustable clock periods. With the PLL enable, measured timing skew between the CPU and FPC is below 1 ns.

The delay-lock discriminator described in this paper [8] is a statistically optimum device for the measurement of the delay between two correlated waveforms. This new device seems to have important potential in tracking targets and measuring distance, depth, or altitude. It operates by comparing the transmitted and reflected versions of a wide-bandwidth, random signal. The discriminator is superior to FM radars in that it can operate at lower power levels; it avoids the so-called "fixed error," and it is free of much of the ambiguity inherent in such periodically modulated systems. It can also operate as a tracking interferometer. The discriminator is a nonlinear feedback system and can be thought of as employing a form of cross-correlation along with feedback. The basic theory of operation is presented, and a comparison is made with the phase-lock FM discriminator. Variations of performance with respect to signal spectrum choice, target velocity, and signal and interference power levels are discussed quantitatively. The nonlinear, "lock-on" transient and the threshold behavior of the discriminator are described. Performance relations are given for tracking both passive and actively transmitting targets. Results of some experimental measurements made on a laboratory version of the discriminator are presented.

4 Specifications

The Delay-Locked Loop has been designed in Cadence Virtuoso using the United Microelectronics Corporation UMC-65nm library. The circuit is designed to work at a power supply voltage of 1.8V.

Specifications of the Counter type DLL

- The steady state current consumption drawn from the supply is 6.909 mA
- The steady state power dissipation is 12.438 mW
- Reliable highest operating frequency is 2GHz
- There is a total of 950 transistors embedded into this design
- The width of the counter as well as the delay line is 5-bits.
- The delay due to each delay line stage is 13.26ps and the maximum delay due to the delay line is 411ps. The 5 stage Mux-tree that behaves as the delay select input, contributes to an additional delay to the DLL.

All the simulations have been performed at 125°C and with a 100mΩ resistor in series with the Voltage-Source, Vdd, to simulate power supply noise. These conditions were applied to ensure that the circuit is robust and is able to operate in realistic conditions.

Specifications of the SAR type DLL

- The steady state current consumption drawn from the supply is 5.947 mA
- The steady state power dissipation is 10.703 mW
- Reliable highest operating frequency is 2GHz
- There are 1197 transistors embedded into this design

All the simulations have been performed at 27°C and with a 100mΩ resistor in series with the Voltage-Source, Vdd, to simulate power supply noise. These conditions were applied to ensure that the circuit is robust and is able to operate in realistic conditions.

5 Description

In this project, we have designed the transistor-level circuit for each component. The key blocks utilized in this design include the Phase-Frequency Detector(PFD), the 5-bit Up-Down Counter and the 5-bit Delay Line and the 5-bit Successive Approximation Register and a PIPO shift register. Additionally, optimal number of buffers, inverters and multiplexers have also been incorporated wherever required to optimize the circuit's functionality and acquire smooth and stable waveforms at the output.

Phase-Frequency Detector

The Phase Frequency Detector(PFD) shown in **Figure 1** has been implemented based on the PFD shown in reference[1] with a minor modification. It is implemented using two D Flip-Flops with asynchronous reset pins. The PFD consists of two pulse inputs of varying delays fed to the clocks of the D Flip-Flops in the PFD circuitry - a Reference signal and the Feedback signal wired from the delay line. The circuit produces two outputs, an Up or Down signal, which are mutually exclusive of each other at a rising edge of one of the inputs. If the Feedback signal leads the reference, the UP signal is asserted and consequently, we increase the delay. Conversely, if the Reference Signal leads the Feedback, the DN signal is asserted and as expected, the delay is reduced. The D inputs of both the Flip-Flops are connected to the supply voltage Vdd(logic 1). For obtaining an UP signal, the Reference Signal acts as the Clock and the Feedback acts as the reset. Thus, if the Reference signal is leading the feedback signal, at the rising edge of reference, the UP signal will go high and at the rising edge of the feedback, the UP signal will go low. For obtaining a DN signal, the Feedback acts as the Clock and Reference signal acts as the reset. Thus, if the feedback signal is leading the Reference signal, at the rising edge of feedback, the DN signal will go high and at the rising edge of the reference, the DN signal will go low.

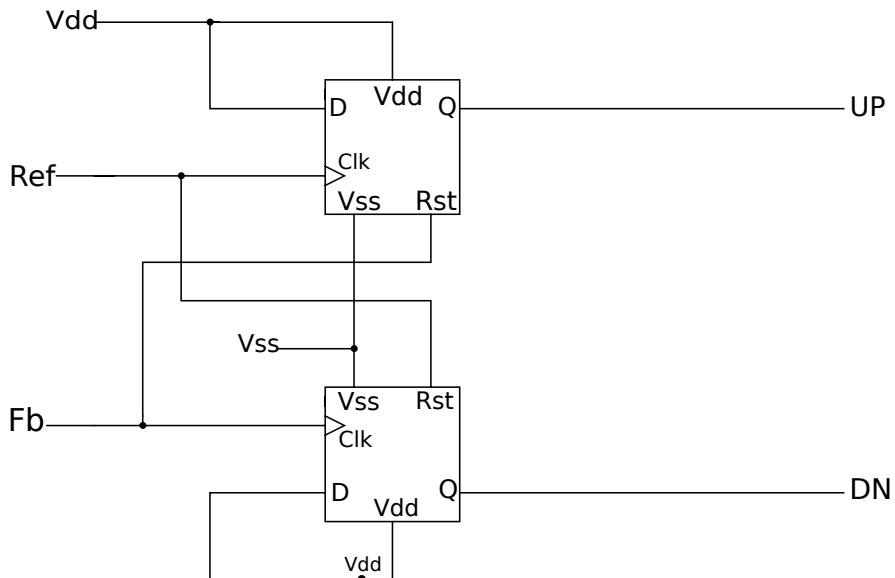


Figure 1: Phase Frequency Detector Internal Schematic

Up-Down Counter

The Up-Down Counter , whose schematic is given in **Figure 2** consists of a 5-bit Synchronous Counter using T Flip-Flops, in which we have inserted multiplexers between individual stages to control the direction of the count. We have used the XOR of the Up and Down signals from the PFD as a Clock signal for the individual Flip-Flops. Since the Up and Down Signals also control the direction of the count, they are connected to the Up and Down control signals of the counter. However, they do not directly control the Up and Down lines of the Counter directly, but through two T-Flip-Flops. This is to ensure that the Up and Down signals do not change at the same rate as that of the clock. Otherwise, this gives rise to metastability conditions, as the signal falls into the setup-and-hold window of the clock. To give further immunity to metastability, the output of the XOR gate is inverted and fed as the clock to the Flip-Flops. The addition of inverter is designed such that it delays the rising edge by half the clock period, thereby avoiding possible setup violations. The LSB bit and the next term has a very large fanout as it is connected to the innermost hierarchy of the Multiplexers in the Delay-Line. The large fanout causes very high rise-times when the Flip-Flop outputs are directly connected to the Delay-Line control inputs, leading to glitches at the output and thereby falsely triggering the PFD. To avoid this, we have added two buffers in parallel at the LSB and the next largest bit. Schematic of the Up-Down counter is shown in the next page.

Delay Line

The Delay Line consists of a series of buffers, where each buffer introduces a specific delay. The schematic of the Delay line is shown in **Figure 3**. In this design, we used thirty-one buffers to provide the corresponding delays for each combination of the 5-bit output that is drawn from the Up-Down Counter. By selecting appropriate tap points between the junctions of the buffers, we can control the delay for every combination by using the tap points for the selection bits as inputs to a tree-based NAND gate structure, which acts as a multiplexer. The LSB and the next term has a very large fanout as it is connected to the innermost hierarchy of the Multiplexers in the Delay-Line. The large fanout causes very high rise-times if the inverter output is directly connected to the NAND gate inputs, leading to glitches at the output and thereby falsely triggering the PFD. To avoid this, we have added two buffers in parallel at the LSB and the next largest bit at the output of the inverter.

We connect the individual components together as shown in **Figure 4**. The completed circuit acts as a negative-feedback loop that constantly tracks the phase of the Reference signal and adjusts the delay to an appropriate value such that the output clock is phase-aligned with the Reference.

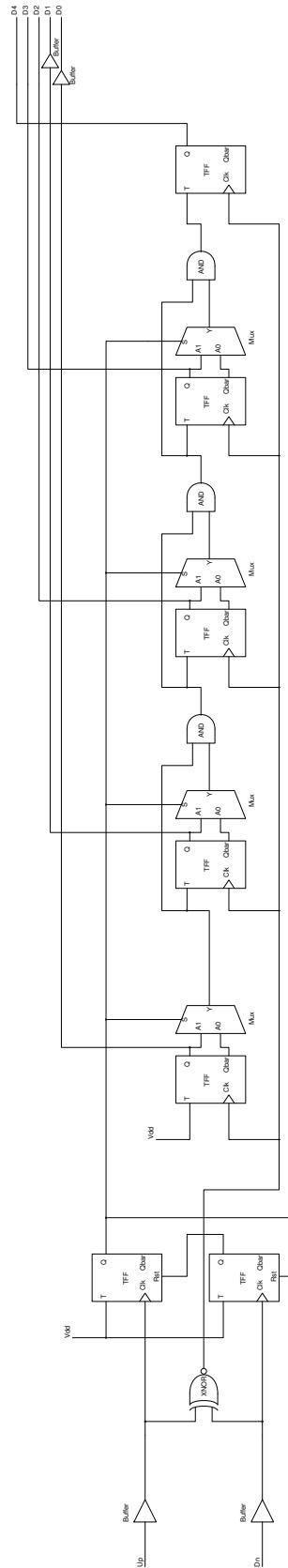


Figure 2: Up-Down Counter Schematic

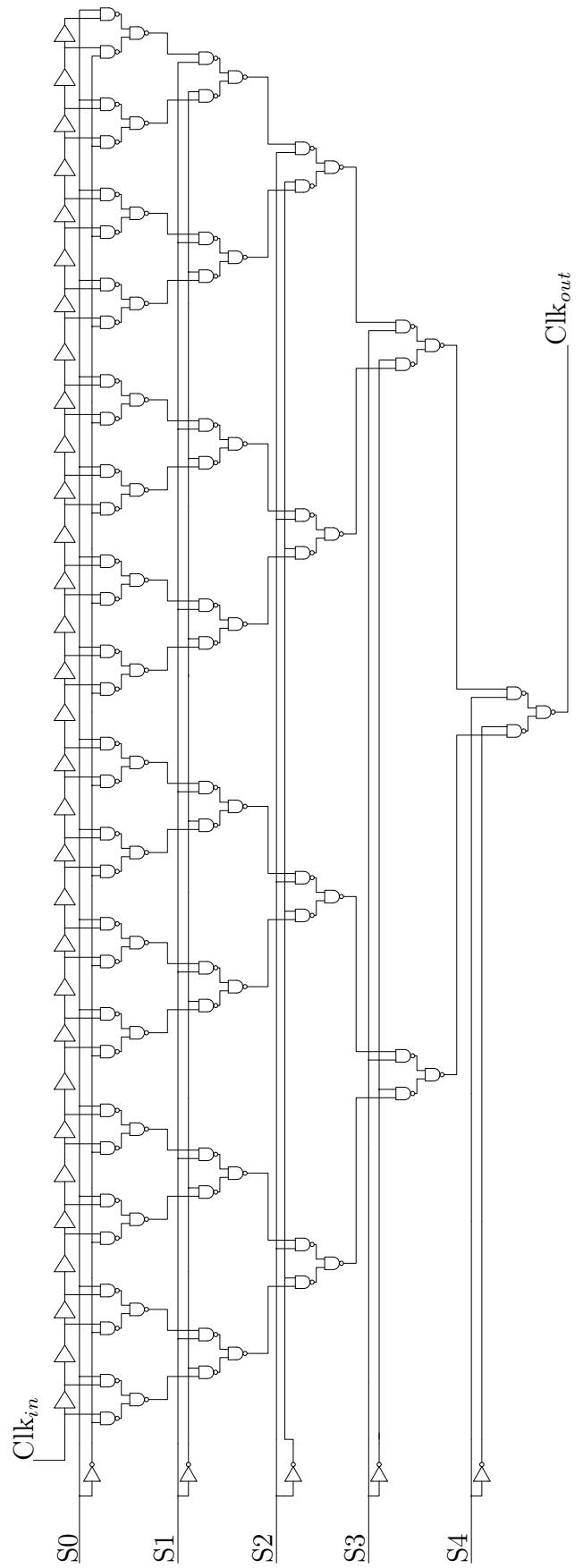


Figure 3: Delay Line Internal Schematic

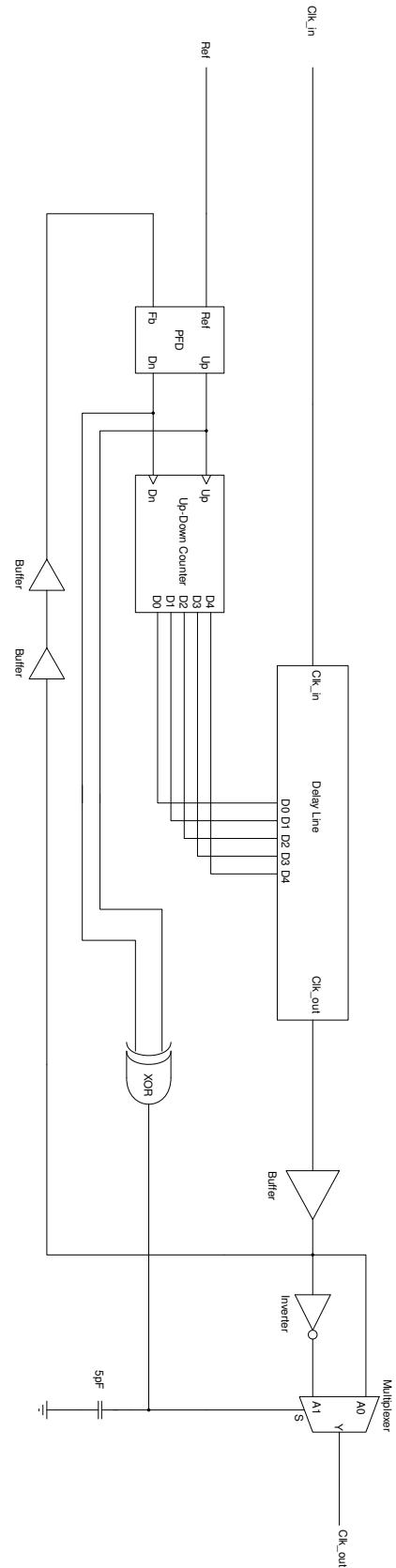


Figure 4: Complete Schematic of the Counter type Delay-Locked Loop

Whenever the Feedback leads the Reference, Up signals are generated by the PFD, and the counter correspondingly increments when Up signal is asserted. The counter keeps incrementing until we have reached an appropriate delay such that Up and Down signals are phase aligned or 180° out of phase. This is due to the architecture of the PFD. When the Reference and the Feedback signals are in phase, the output signals of the PFD, i.e. both the Up and Down signals are low. However, when the Reference and the Feedback signals are 180° out of phase, the Up and Down signals are two 180° out of phase signals. We have mentioned that the counter is clocked with the XOR of Up and Down signals. Due to this, when Up and Down signals are applied in the above mentioned manner, the input to the XOR Gate is a sequence of 10,01,10,01..... Due to this, the Clock remains at a constant high level, devoid of rising edges, and the Flip-Flops tend to maintain their state. It has been observed that the counter locks at 180° phase shift when the delay is less than half the period of the incoming Reference signal and at zero degree phase shift when the delay is more than half the period of the Reference signal. Due to this behaviour, we need two paths, one is the direct output of the delay-line, and another is an inverted version of the delay-line output. Depending on the lock state, whether 0° or 180° , we have a multiplexer, which selects the appropriate output and routes it to the Clock-Out line. The condition of the locked state is detected by an XOR Gate, which detects the lead or lag phase and uses that signal to feed to the select line of the multiplexer. Due to the presence of short transient signals at the output of the XOR Gate, we add a 5pF capacitor to stabilize the input to the Multiplexer. The subsequent diagrams show the internal transistor level schematics of the different components used in the DLL. All the circuits have been implemented using 65nm MOSFETs in static CMOS technology. The circuits have been optimized for minimum transistor count while giving maximum speed and efficiency. In this circuit, we have used XOR Gates of two different architectures, one composed of only static CMOS Pull-Up and Pull-Down network, is used for providing clock to the counter Flip-Flops. This is due to the high fanout of the Clock signal going to the Flip-Flops. Another XOR gate, composed of an inverter and a transmission-gate based multiplexer is used in the T-Flip-Flops. This XOR gate uses only 8 transistors as opposed to the 12 Transistor design of the previous static CMOS based XOR gate, thus reducing transistor count, leakage current, power consumption and improving efficiency.

Successive Approximation Register (SAR)

The successive approximation register works on the principle of Binary Search. We first turn on our MSB. Then, we observe our Input pin. If the effect due to MSB(in this case, delay) does not exceed the reference, the input remains low and we retain MSB as high and proceed to the next lower bit, turning that high. However, if the input pin gives a high input, indicating that the delay due to turning MSB high exceeded the required value, we set the MSB to 0 and proceed by turning the next significant bit high and observing the resultant effect. This cycle we continue till we reach the LSB.

The Successive Approximation Register (SAR) architecture, shown in **Figure 5**, is composed of 5 D-Flip Flops, five 2:1 multiplexers, a MOD 7 Counter, a 3:6 decoder and a gate-controlled logic to control the clock and reset signals. The SAR circuitry takes two inputs, In and Clk, where In is the up or down signal generated by the PFD and Clk is the clock pulse to synchronise the SAR with the DLL. The frequency of this Clk signal of is 4 times lesser than the clock frequency of the DLL. The SAR is designed such that it functions based on the following mechanism; The D Flip-Flops act as shift registers, and the values are given by the output of the multiplexer. The multiplexer based on the select line obtained from the decoder, either enables the shift operation of the D Flip-Flop or passes the inverted IN signal such that the currently selected bit is made 0. The MOD 7 counter is a free running counter that counts from 0 to 7 periodically until the phase lock is achieved. Seven states are needed as 1 state is required to set the initial values of all the bits, 5 states are needed to set the state of each bit(total 5 bits in this case) and another state to reset all the bits. A 3:6 decoder is followed by the counter to enable the clock pin as the counter increments its value. Initially, as the MSB has to be set to 1, one of the inputs of the first multiplexer is directly connected to Vdd, while the other Flip-Flops are under reset mode.

As the count increases, all the D Flip-Flops are automatically in active mode(i.e. clock is gated to them) and function based on the signal produced by the decoder. When the counter reaches to its last count, which in this case is 7, the counter immediately resets, the last bit of the decoder is sent to all the reset inputs of the Flip-Flops, thus forcing them into 0 state. Simultaneously, the output of the counter is tapped by a three input AND gate, which is directly wired to the reset of the first Flip-Flop, practically setting the MSB bit to 0. Based on the IN signal, the MSB and the MSB-1 bits are manipulated. The clock pin acquires its input through a gated logic consisting of CMOS AND gates, that assist in controlling the clock until locking. The decoder along with the OR gates act as the enable input for the clock, which is gated with the AND gate such that when the enable signal is high, the clock will be fed to the clock pin of the D Flip-Flops. The OR gate is exploited to set the currently selected bit and its lower bit to 1 when input is lower, suggesting that the delay is increased to lock with the reference signal. The AND gate takes the output of the OR gate and the clock signal, and when both the inputs are high, the clock is effectively activated.

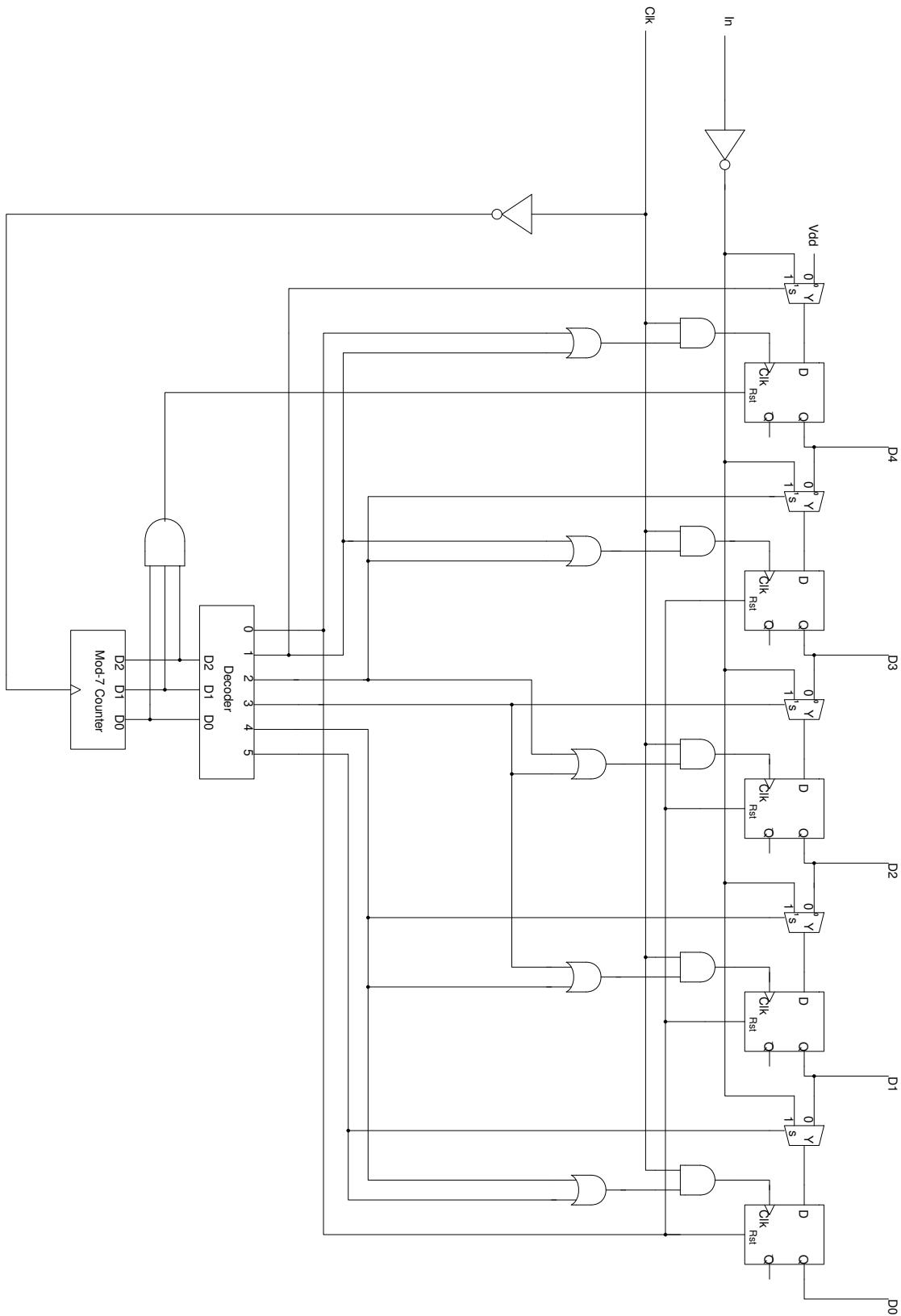


Figure 5: Successive Approximation Register internal architecture

Parallel In Parallel Out Shift Register(PIPO)

The Parallel In Parallel Out Shift Register(PIPO), shown in **Figure 6** is built using a series of five D-Flip Flops clocked using an XOR gate that takes the UP and DN outputs of the Phase Frequency Detector circuit. According to the functionality of the PIPO register, a parallel input is provided to the Flip-Flops, which produces a parallel output in just one clock cycle. The PIPO register takes its inputs from the SAR register, which provides the delay information for the locking phase based on its binary search mechanism. This information is passed to the delay line that interprets it by traversing till the required multiplexer stage based on the code, essentially achieving phase lock between the reference signal and DLL output at a faster and reliable rate. But the PFD poses a problem of generating low width pulses even after the occurrence of the locking stage. Naturally, the PFD has to settle to a constant HIGH or LOW value based on the locking phase of the signals. But because of minor propagation delays between the signals due to gated paths, these narrow pulses tend to disturb the locking stage, hence extending the lock times to further time intervals despite of already reaching the locking state. We counter this phenomenon by introducing hysteresis at the output of the XOR gate by adding a 100fF capacitor.

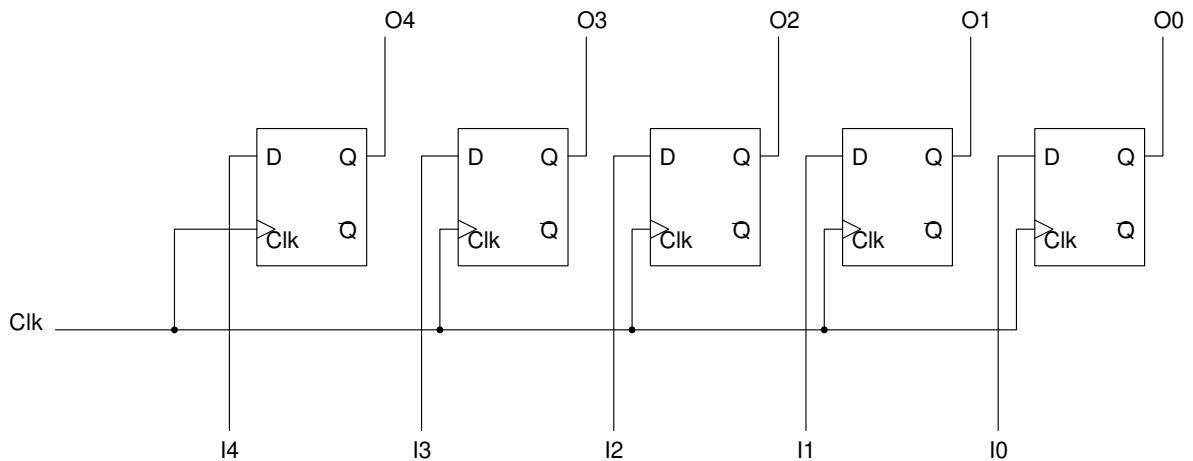


Figure 6: Schematic of the PIPO Shift Register

The complete SAR based DLL has been shown in **Figure 7**. Since the Up and Down signals at the output of the Phase-Frequency detector is not a constant signal but a very rapidly varying one, we feed the signals to a latch to hold them stable for the SAR register to operate correctly. Due to inherent delay between input to the delay-line and the response, to achieve proper timing characteristics, we have to divide the clock to the SAR Register(the reference signal acts as the clock in this case) by 4 to achieve correct operation. The clock is divided by 4 by using two T-Flip-Flops consecutively.

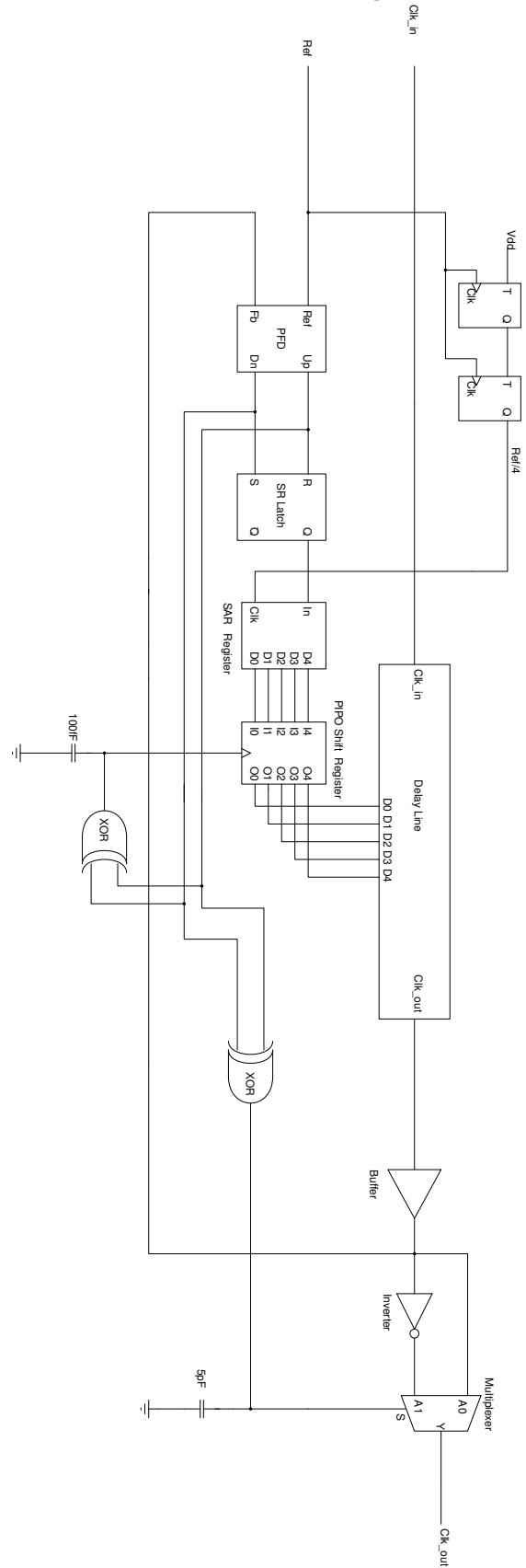


Figure 7: Complete Schematic of the SAR based Delay Locked Loop

We have to detect the position of the lock and disable the inputs to the delay-line from changing. This is done by register array between the SAR register and the delay line. The Parallel In Parallel Out(PIPO) register is clocked by the XOR of the Up and Down signals from the PFD. When locking is achieved, the output of the XOR gate becomes constant(either high or low depending on in-phase lock or 180° out-of-phase lock). Due to very sensitive nature of the PFD, even when the signals have locked in acceptable range the PFD still produces very narrow pulse, which can occasionally trigger the PIPO register, leading to instability or longer locking times. This is avoided by adding some hysteresis at the output of the XOR gate, by adding a 100fF capacitor at its output.

In order for the SAR based DLL to work correctly, we have to give the inputs in a definite sequence. First the power supply Vdd is turned on in the absence of any input. Then, after atleast 1ns, the Reference clock is turned on, followed by the input clock. Following this sequence ensures that the DLL works in a deterministic manner always.

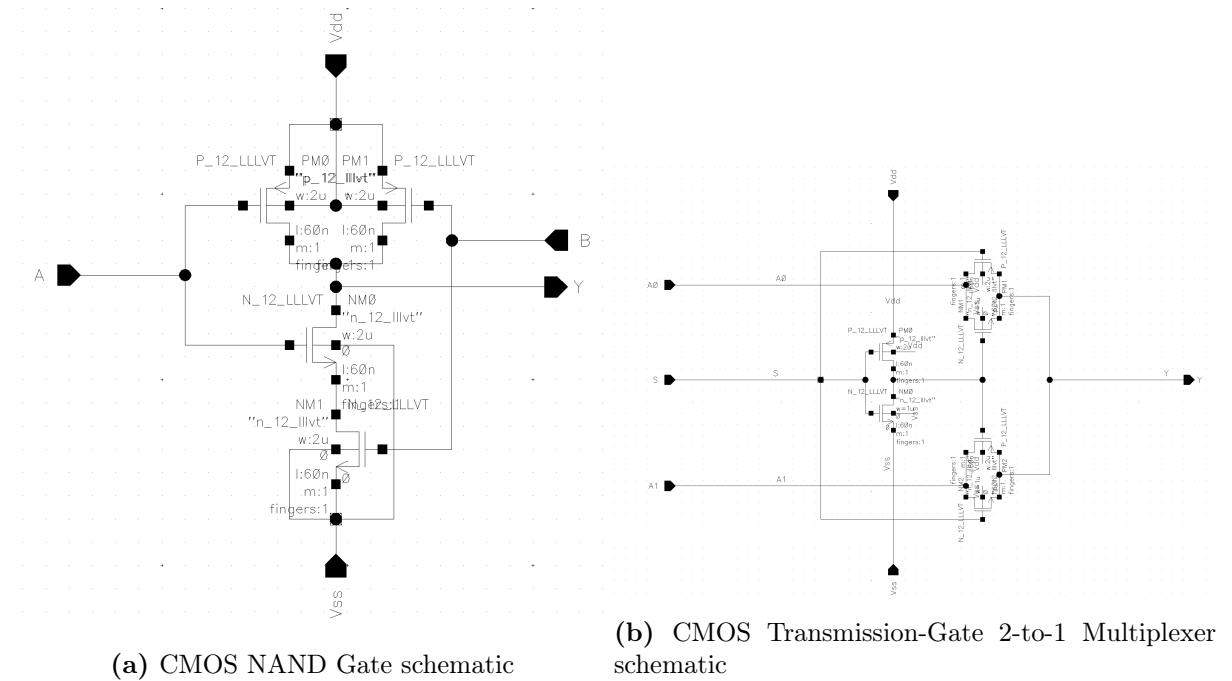
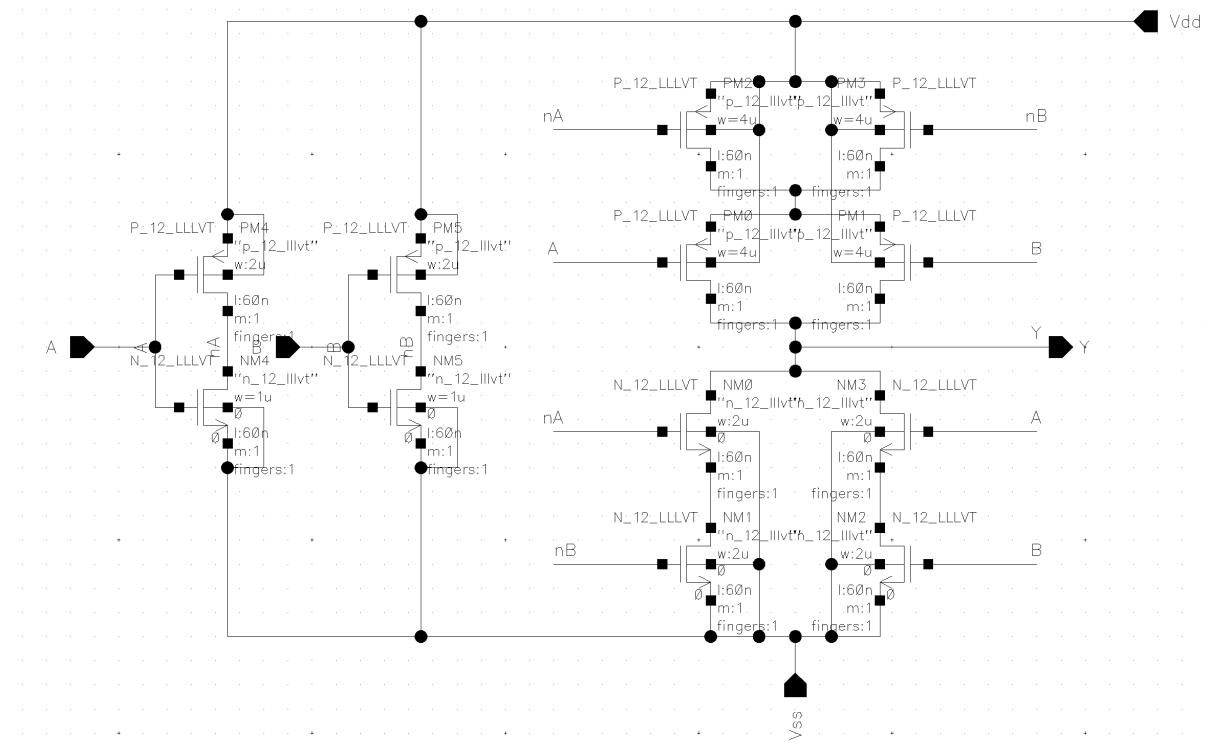
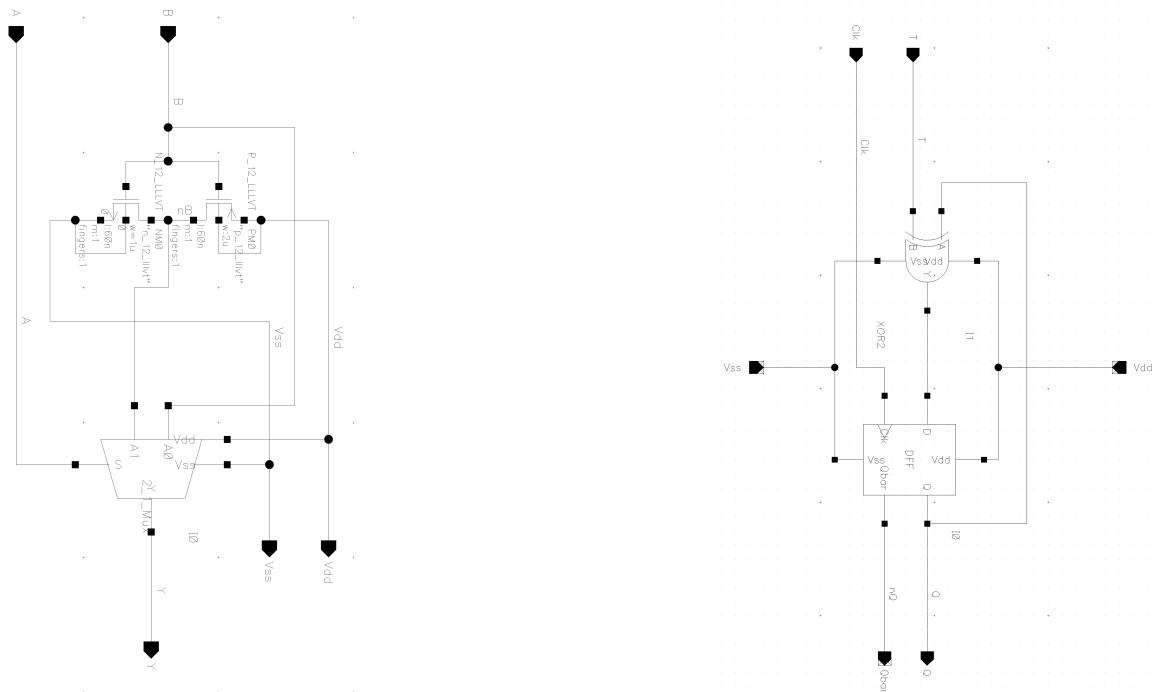


Figure 8: Transistor level Schematics of NAND Gate and CMOS-Transmission Gate



(a) CMOS XOR Gate



(b) CMOS Transmission-Gate based XOR Gate



(c) T-Flip-Flop using D-Flip-Flop and XOR Gate

Figure 9: Transistor level schematics of various components

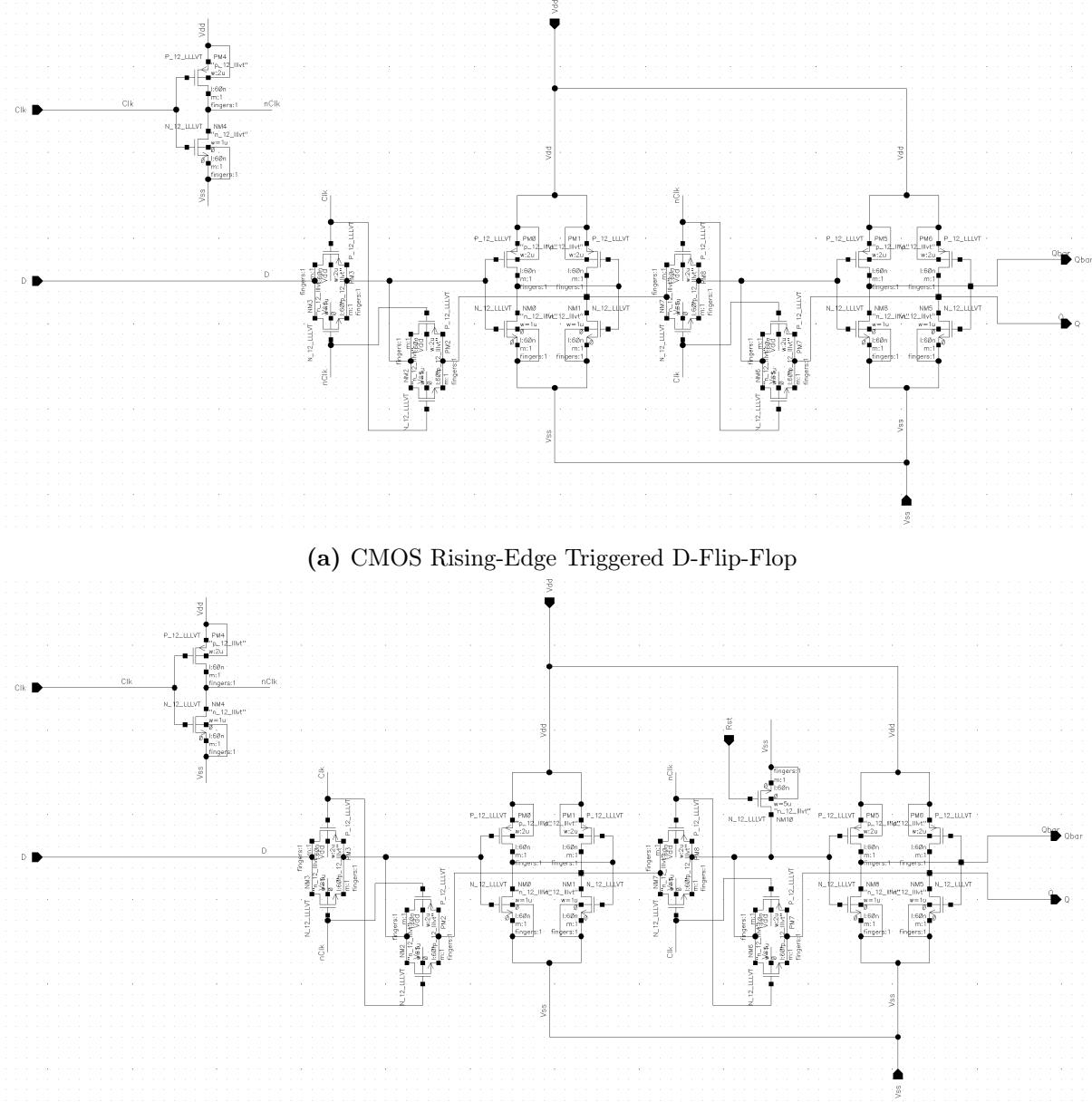


Figure 10: Transistor level schematics of the D-Flip-Flops

6 Progress Timeline

Table 1: Table of the Timeline

Weekly Timeline	Progress
Week 0: 05 August 2024 - 11 August 2024	Research paper work, mathematical derivations and architectural planning
Week 1: 12 August 2024 - 18 August 2024	Design and simulation of each component of the DLL on XCircuit to test and verify its functionality
Week 2: 19 August 2024 - 25 August 2024	Transistor-level optimizations of the delay line and up-down counter to improve output waveforms
Week 3: 19 August 2024 - 01 September 2024	Implementation of logic circuits and components into blocks for employing them into the overall design in Cadence Virtuoso
Week 4: 02 September 2024 - 08 September 2024	Integration of all the components and solving issues of inverted phase locking at output
Week 5: 09 September 2024 - 15 September 2024	Modifications to PFD circuit and upgrading to 5 bit counter and delay line
Week 6: 16 September 2024 - 22 September 2024	Simulation based on different time periods to analyze their shifts in locking phase
Week 7: 23 September 2024 - 29 September 2024	Adding frequency-noise parameters to analyze its effects and impact on the waveforms and determine lock period
Week 8: 30 September 2024 - 06 October 2024	Plotting of eye diagrams to visualize the effect of noise parameters and determine the thickness of the plot
Week 9: 07 October 2024 - 13 October 2024	Optimizing delay paths by adding buffers for balancing the output delay with the feedback path
Week 10: 20 October 2024 - 27 October 2024	Design of the Successive Approximation Register (SAR) by implementing a digital logic based on its functionality
Week 11: 28 October 2024 - 4 November 2024	Integration of the SAR Register into the DLL Circuit and optimization of the architecture such that it is free of all errors
Week 12: 5 November 2024 - 12 November 2024	Waveform observation by adding noise parameters and plotting the relevant figures for verifying the complete process flow of the DLL

7 Specific Modifications

There were several modifications that were performed on the design as we progressed through the implementation phase of each block of the DLL.

1. Phase Frequency Detector Circuit:

We had initially adopted the standard design of the PFD circuit that employs two D Flip-Flops with asynchronous reset pins. An additional AND gate was used to combine the non-inverted outputs of both the flip-flops, feeding the result back to the reset inputs of the flip-flops. A noticeable setback with this design were some significant spikes at the clock edges that were generated either on the UP or Down pins instead of an expected steady low output, which was predominantly introduced due to the delayed(AND gate propagation delay) assertion of the resets of their corresponding flip-flops. These short spikes were falsely recognised by the counter, which produced undesirable deviations in the counter waveforms, consequently affecting the further processes of the design. To suppress these unwanted spikes, the clock signals of both the flip-flops were routed to the reset inputs of the opposing flip-flops, such that as soon as the rising edge of the leading clock is detected, the other flip-flop is automatically forced into reset mode, hence unaffected by its own active clock edge. As a result, only one of the outputs, either an UP or a Down are active, while the other stays at a steady low output, conditioned to changes in their delays.

2. Upgrade to a 5 bit UP-DOWN Counter:

With reference to the research paper[4] used to implement the design of a DLL, the proposed UP-DOWN counter was a synchronous 4-bit counter. But due to the architecture of the PFD circuit as mentioned in the description of the Delay Line, the phases of the Delay Line output may lock in phase or out of phase with each other. Due to the architecture of the delay line, we get a non-zero delay when the input code to the delay line is 0. However, for locking, we might require 0° phase shift. We make the delay line longer so that for a certain input code and in the range of its operating frequency, we can get a 360° phase shift, which is equivalent to 0° phase shift. Due to the increased length of the delay-line, we had to increase the size of our counter also by 1-bit. To accommodate the control circuitry of the counter, we had initially fed the UP and DOWN outputs of the PFD to an XOR gate and directly connected it to the clocks of each of the T Flip-Flops. Using this logic, we had come across metastability conditions as the UP and DOWN signals change at the same time. To overcome this hurdle, an inverter was added and then fed to the clock, effectively giving $1/2$ clock period delay and avoiding setup violations.

3. Optimizing the Delay Line:

The Delay Line in the starting phase comprised of buffers, transmission gates and AND-gate based decoder that taps out different points in conjugation with the select inputs to provide the necessary delay for every possible input code combinations as it increases or decreases. The main problem faced here was the non-monotonic delay

brought by the transmission gates and multiplexer, which is undesirable as it introduces non-linearity and makes the behaviour of the circuit non-deterministic. To mitigate this issue, buffers were used along with the shift from a standard transmission-gate based architecture to a NAND-based multiplexer with selection bits, which resulted in an optimised design that uses minimal transistors and effective locking period.

4. Mitigating the effects of the 180° phase shift:

As discussed in the description of the Delay Line, it has been observed that the counter locks at 180° phase shift when the delay is less than half the period of the incoming Reference signal and at zero degree phase shift when the delay is more than half the period of the Reference signal. Due to this phenomenon, two paths were drawn from the Delay Line, the non-inverted and the inverted outputs, which are passed into a 2:1 multiplexer to select the appropriate output based on the differences in delays. To account for the select line, we used an XOR gate that takes the reference and feedback pulses as inputs and detects the lock state condition. Due to the presence short transient signals at the output of the XOR Gate, we add a 5pF capacitor to stabilize the input to the multiplexer. Following the improvements in the waveforms, we were met with another hurdle that came with the delays introduced by the XOR gate and multiplexer added to the output end. To overcome this problem, additional buffers were inserted at the feedback path so that delays are meticulously evened, and the effectiveness in the locking phase is restored.

5. Replacement of UP-DOWN Counter with a SAR Register:

The SAR register has been designed from scratch based on its functionality. The replacement of the UP-DOWN Counter originated from the idea of achieving phase locks at reduced time rates. As the input count increases, the usage of UP-DOWN Counter further increased the delay of the locking stage in the Delay Line, which does not occur in the SAR register model as we efficiently start searching from an intermediate value and provide the necessary delay depending on the behaviour of the input signal. The SAR register has been designed considering the ease of scalability, which can be decided based on the specification of the design. We can easily add or remove additional bits to the SAR register without any major change in the design. This can allow us to use the SAR register in other designs such as SAR-based ADC. We detect the lock stage by calculating the XOR of the Reference and the Output-clock. If the reference and the output clock are exactly aligned, its XOR is 0 and the Register array feeding clock to the SAR register gets disabled, retaining its state.

6. Detection of locked state with an XOR Gate

In the SAR based DLL, we are detecting the locked state based on the XOR of the Up and the Down signals(PFD outputs). In the event of locking, output of the XOR gate retains a stable value, holding the output of the Parallel-in Parallel out shift-registers feeding the delay line.

8 Results and Discussion

It has been observed that the DLL finds it difficult to lock at frequencies more than 2GHz in the presence of power-supply noise. The DLL has been tested to lock at frequencies of 1GHz at 125°C and in the presence of power-supply noise in the Vdd line. For reliable operation, proper decoupling capacitors must be used as near as possible to the Vdd line to sink any AC transient in the Vdd line to ground and provide reliable operation at high frequencies. The Successive Approximation Register (SAR) was simulated at a temperature of 27°C, with noise at the inputs and power supply noise.

In the figures shown below, we can observe the tracking and locking phase of the DLL at two different frequencies- 1GHz and 2GHz. We have included two examples at each frequency under two different conditions: one in which Reference signal is leading the Input-Clock signal, and the other in which the Input-Clock leads the Reference signal. For each observed condition, we have included the tracking behaviour, the steady state(locked) phase error between the output clock and the input reference, the Eye Diagram and the details of the Eye diagram crossover point showing the maximum jitter.

Though the plots, we can observe that the circuit tracks the phase of the reference signal, provides the necessary delay to the Input-Clock signal to keep it phase aligned with the reference signal.

Multiple modifications were introduced to encourage the aim of reduced transistor usage to conserve power while also meeting the vital requirements of obtaining functionally correct outputs for all valid input combinations. By employing the SAR-based DLL, we eliminated the extra delay of the UP-DOWN Counter circuit that produced a periodic 5-bit count until the lock phase was achieved. There were better results observed with the inclusion of the SAR Register, especially regarding the improved rate of phase locking and hence, power was optimally managed throughout the circuit and the design is hence proven to be efficient.

Figures of the outputs of the Counter type DLL

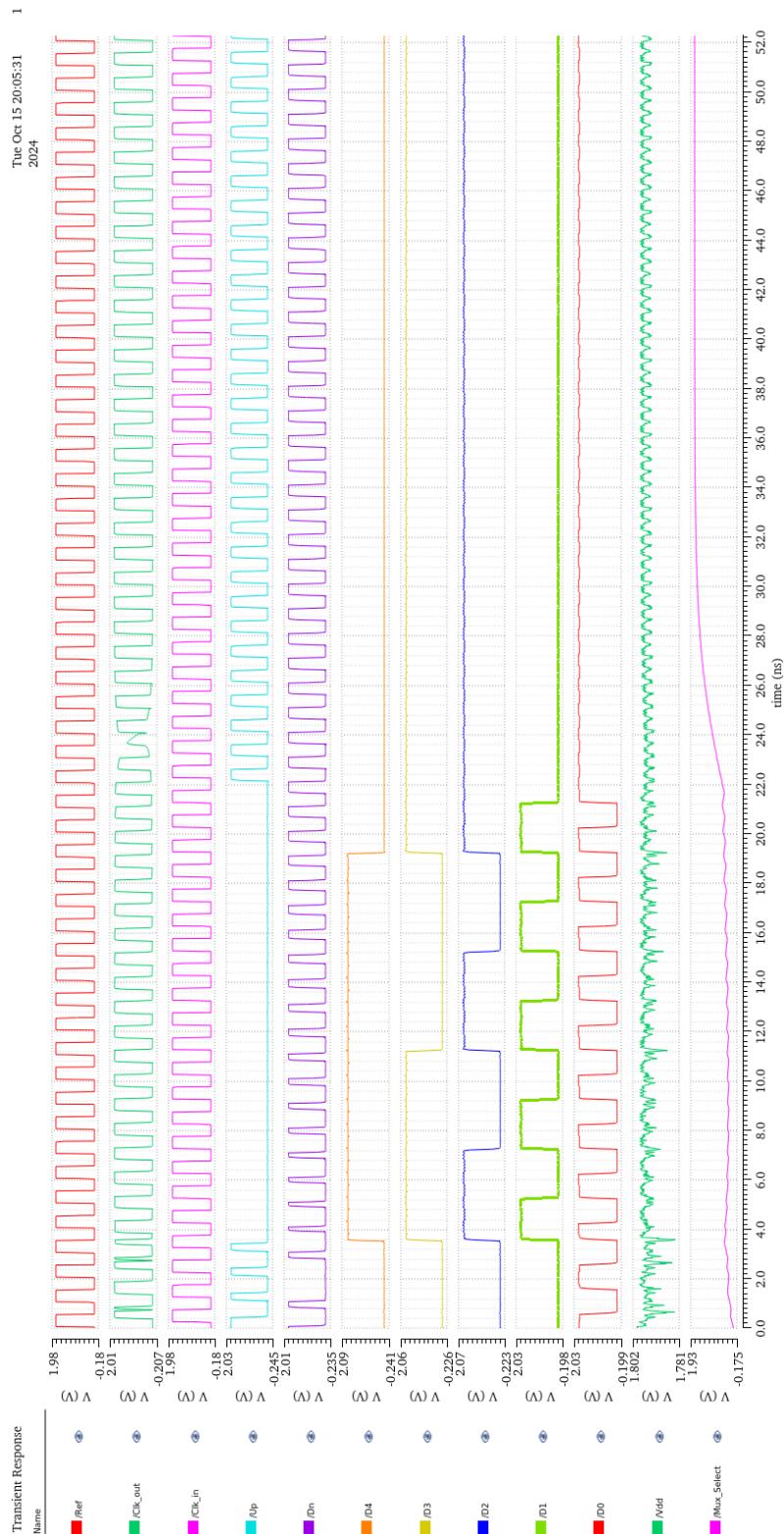


Figure 11: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

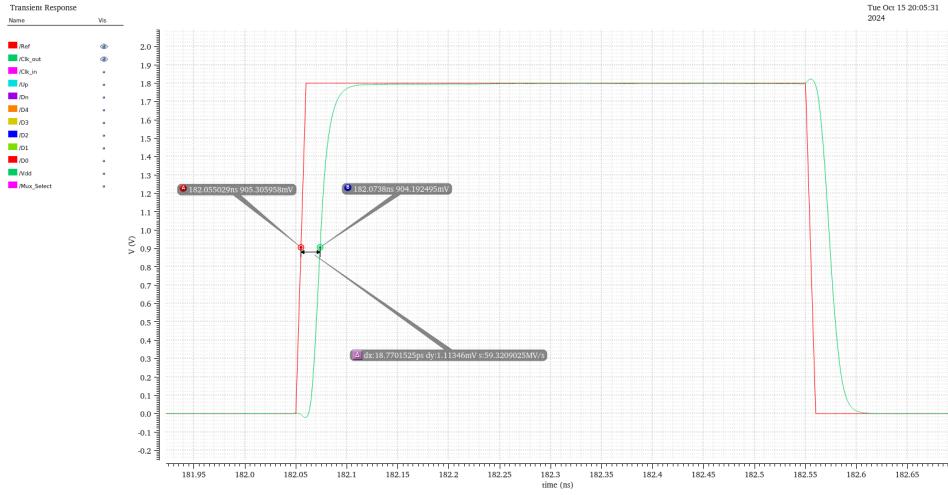
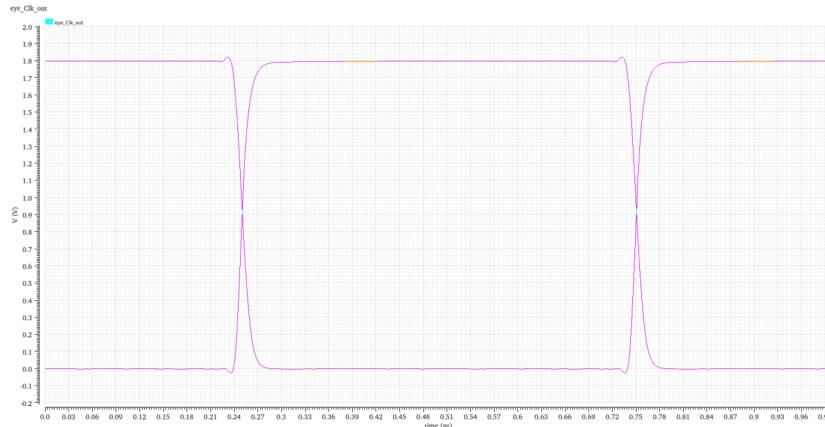
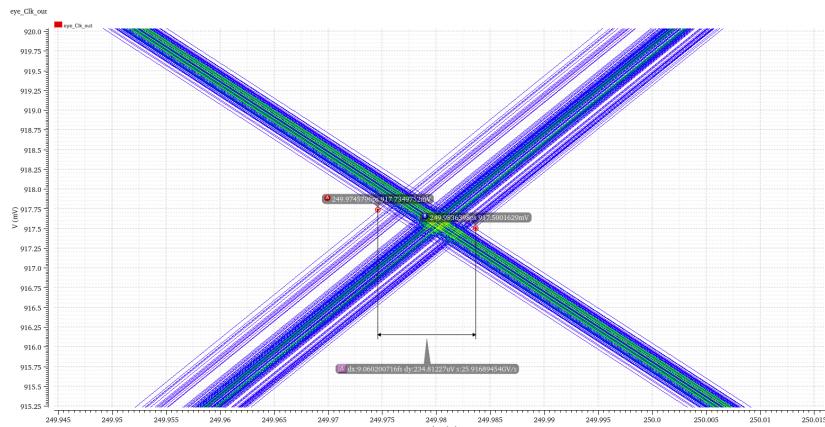


Figure 12: Steady State Phase Error of 18.77 ps at f = 1 GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at f = 1 GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at f = 1 GHz with Delay time of 350ps

Figure 13: Eye Diagram at 1GHz and associated measurements

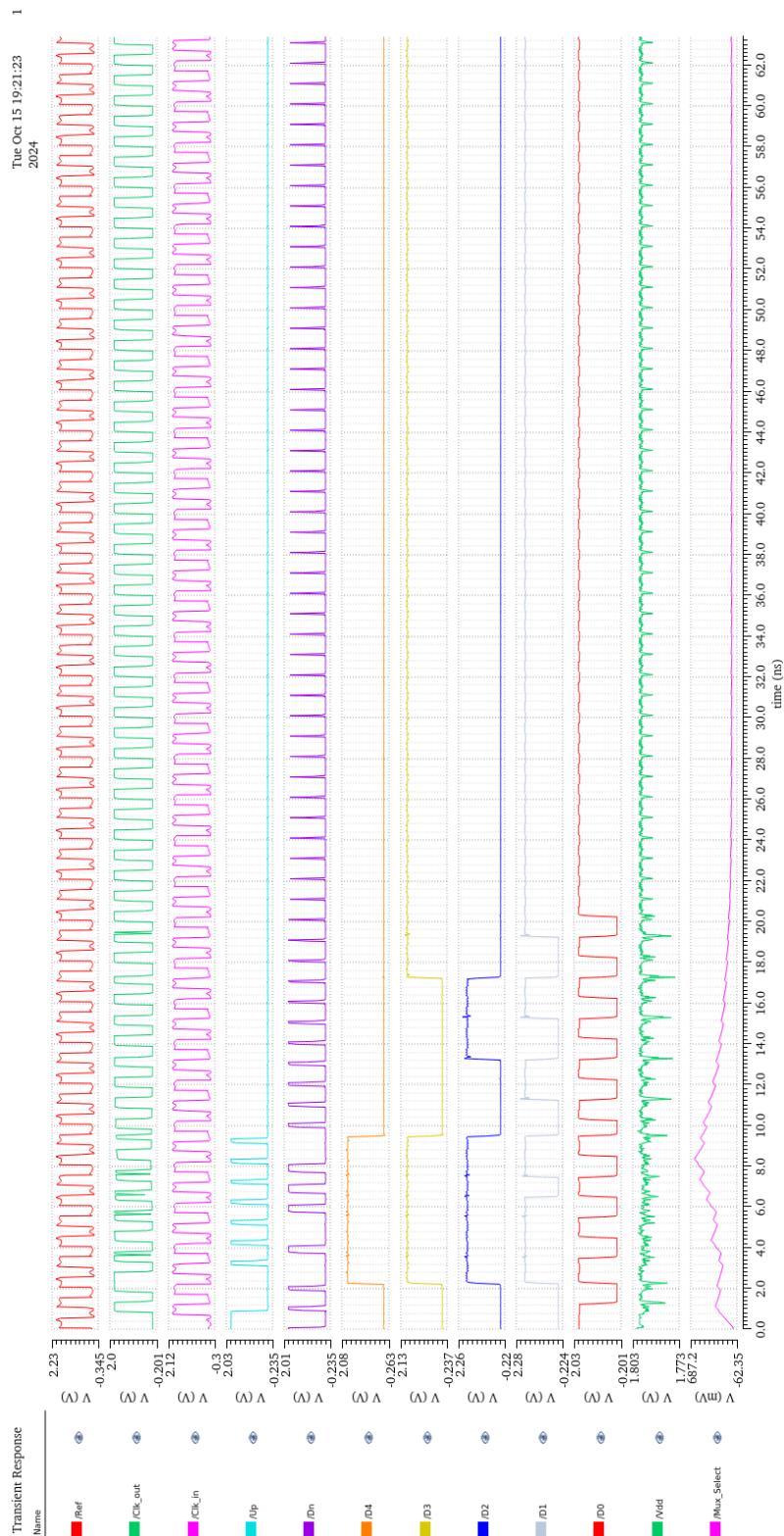


Figure 14: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

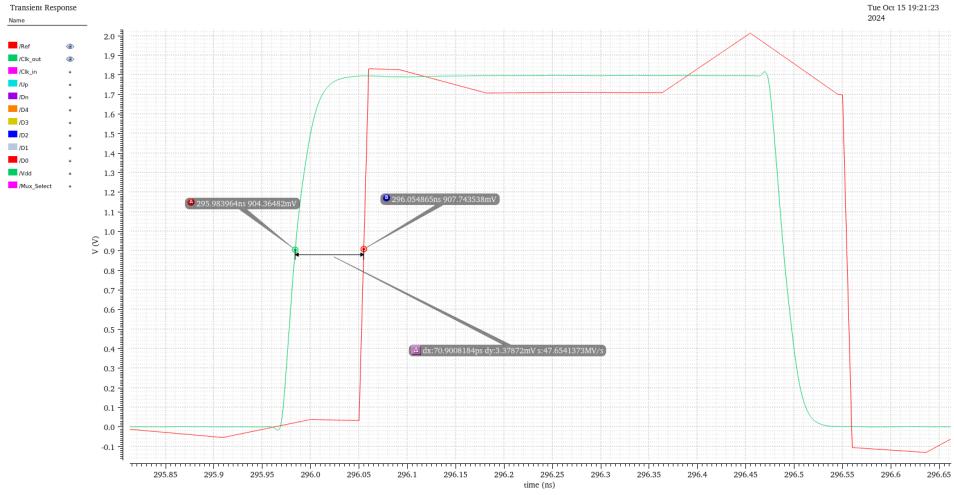
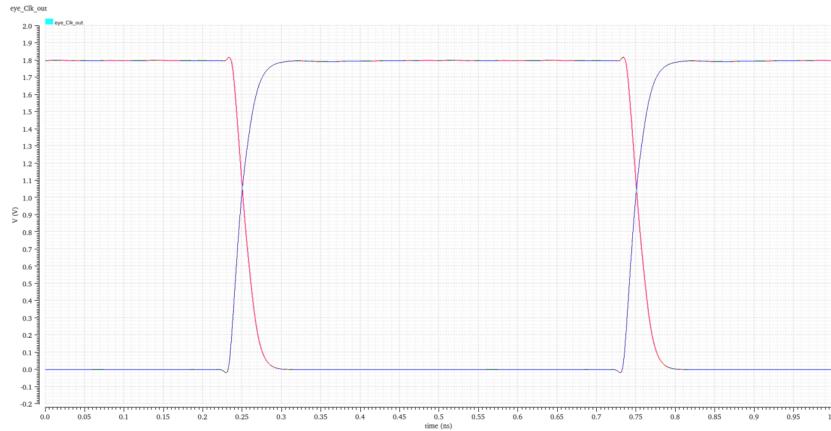
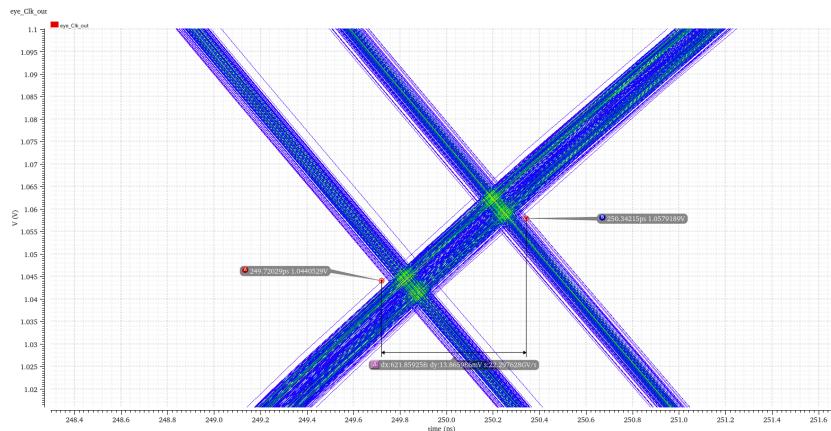


Figure 15: Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at $f = 1$ GHz to analyze effects of noise on circuit



(b) Cross-OVER Point with maximum jitter of 9.060 fs at $f = 1$ GHz with Delay time of 350ps

Figure 16: Eye Diagram at 1GHz and associated measurements

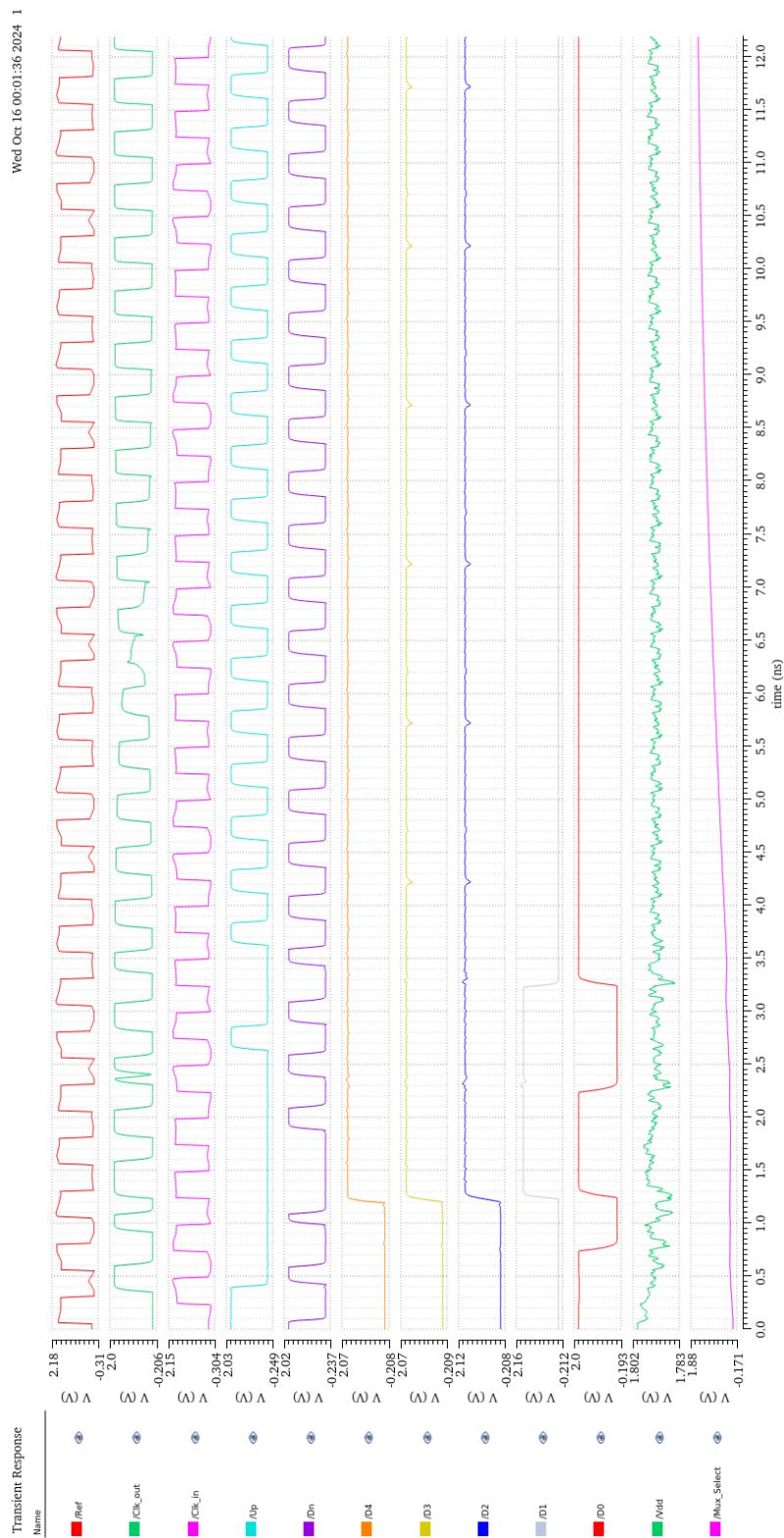


Figure 17: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

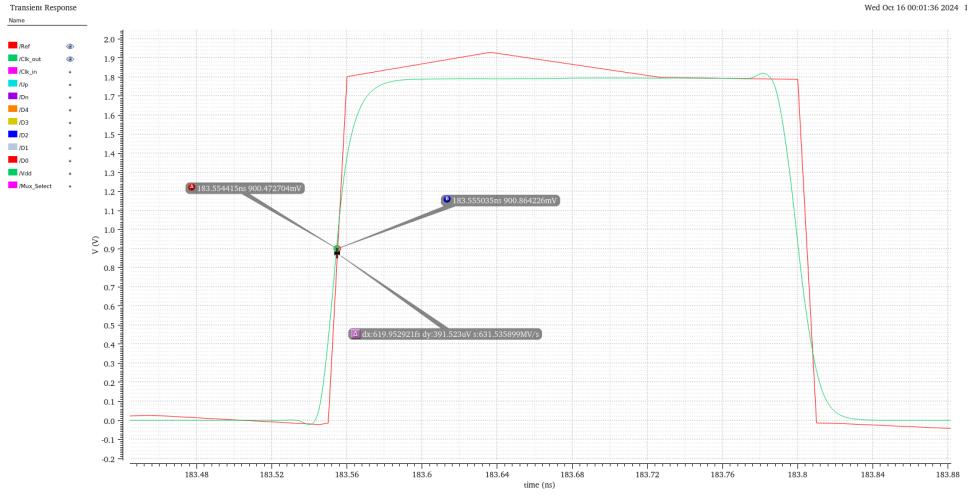
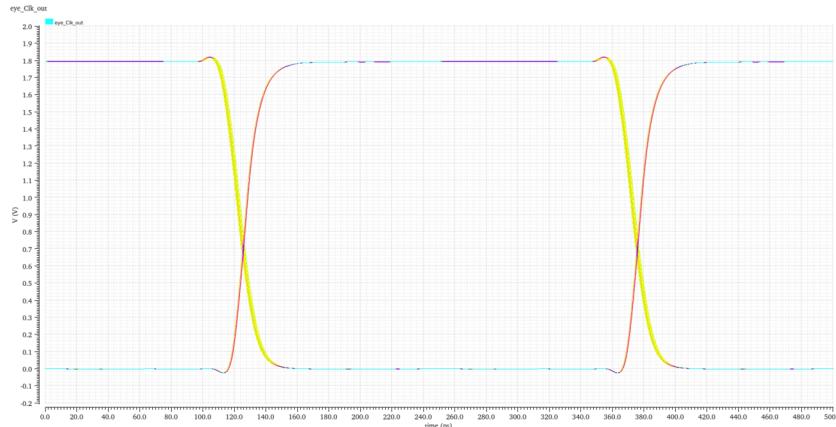
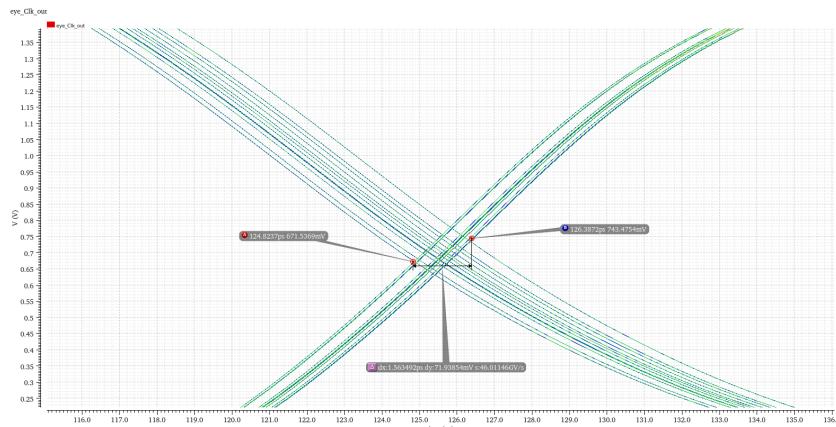


Figure 18: Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at $f = 1$ GHz to analyze effects of noise on circuit



(b) Cross-OVER Point with maximum jitter of 9.060 fs at $f = 1$ GHz with Delay time of 350ps

Figure 19: Eye Diagram at 1GHz and associated measurements

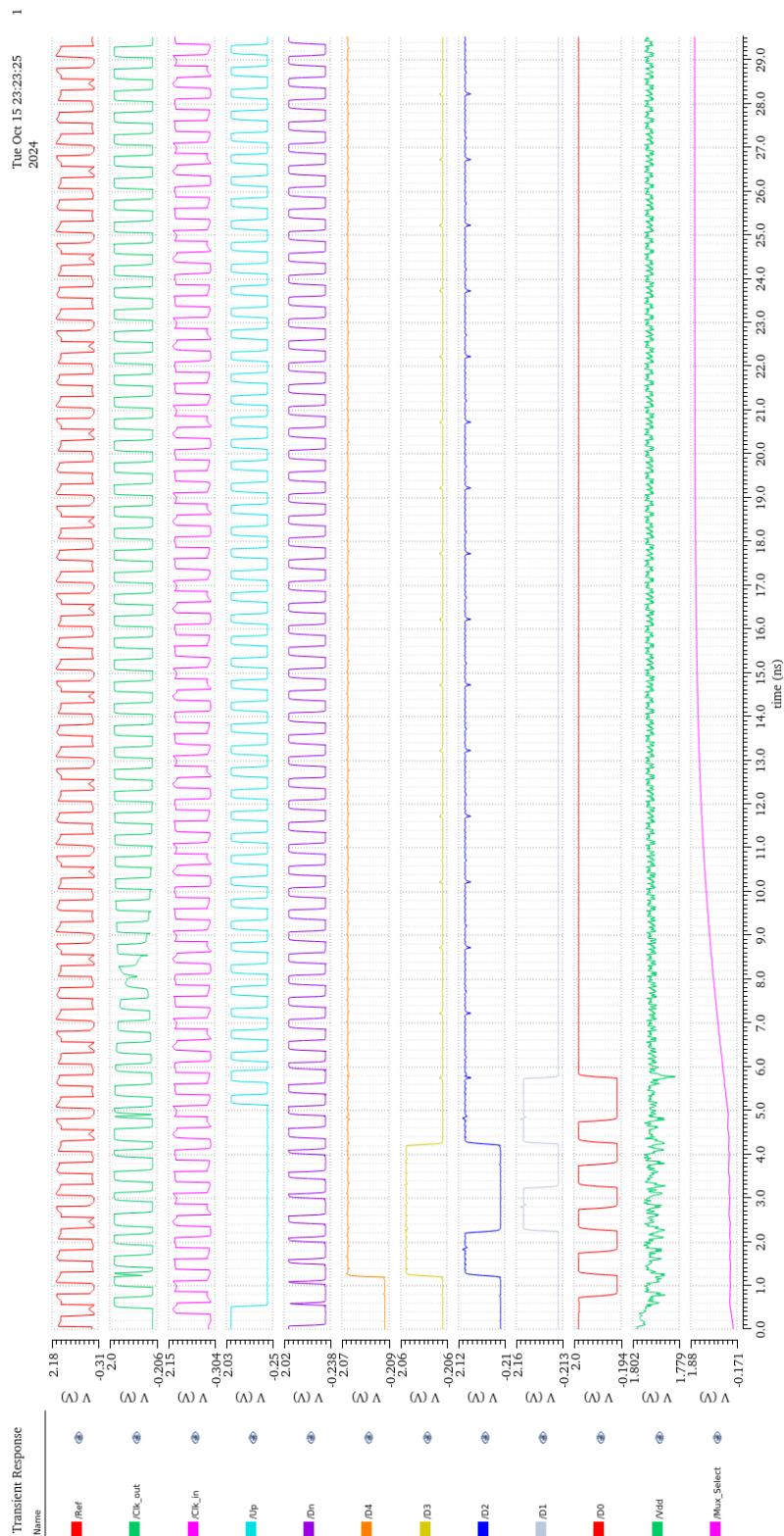


Figure 20: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

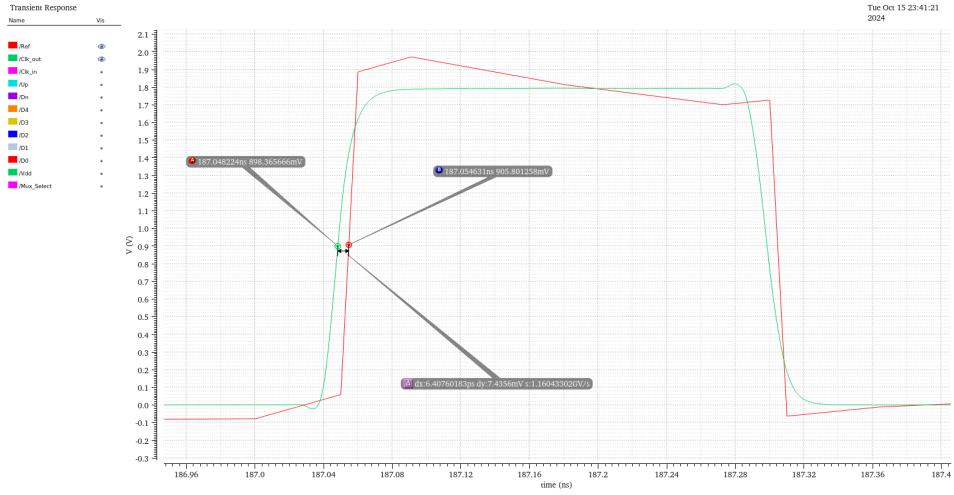
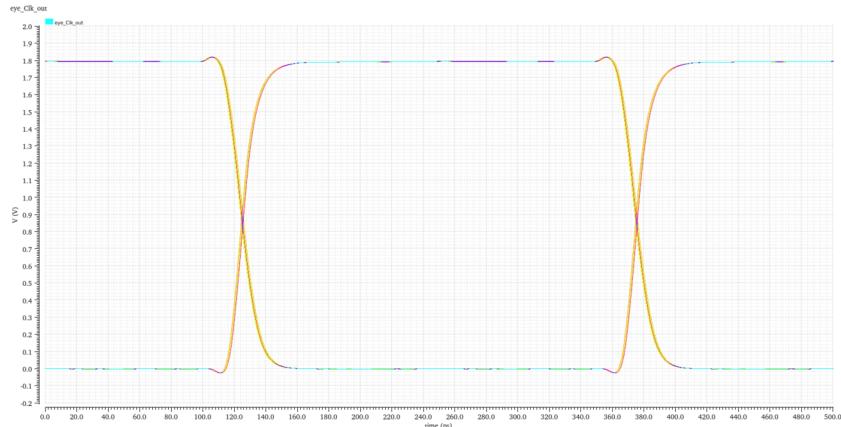
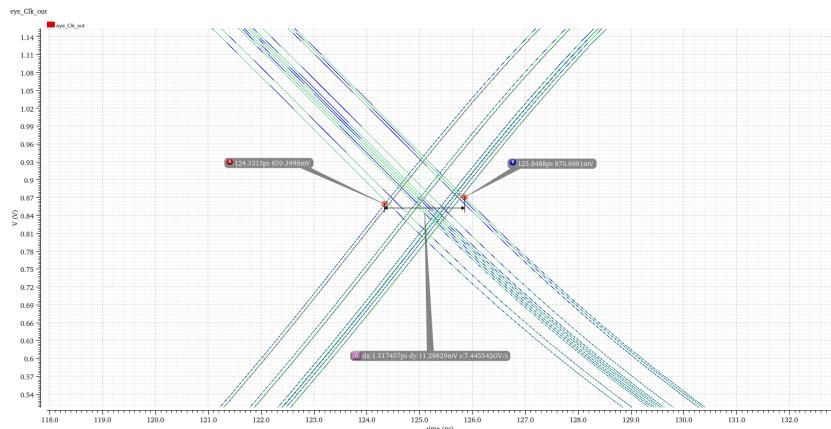


Figure 21: Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at $f = 1$ GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at $f = 1$ GHz with Delay time of 350ps

Figure 22: Eye Diagram at 1GHz and associated measurements

Figures of the outputs of the SAR type DLL

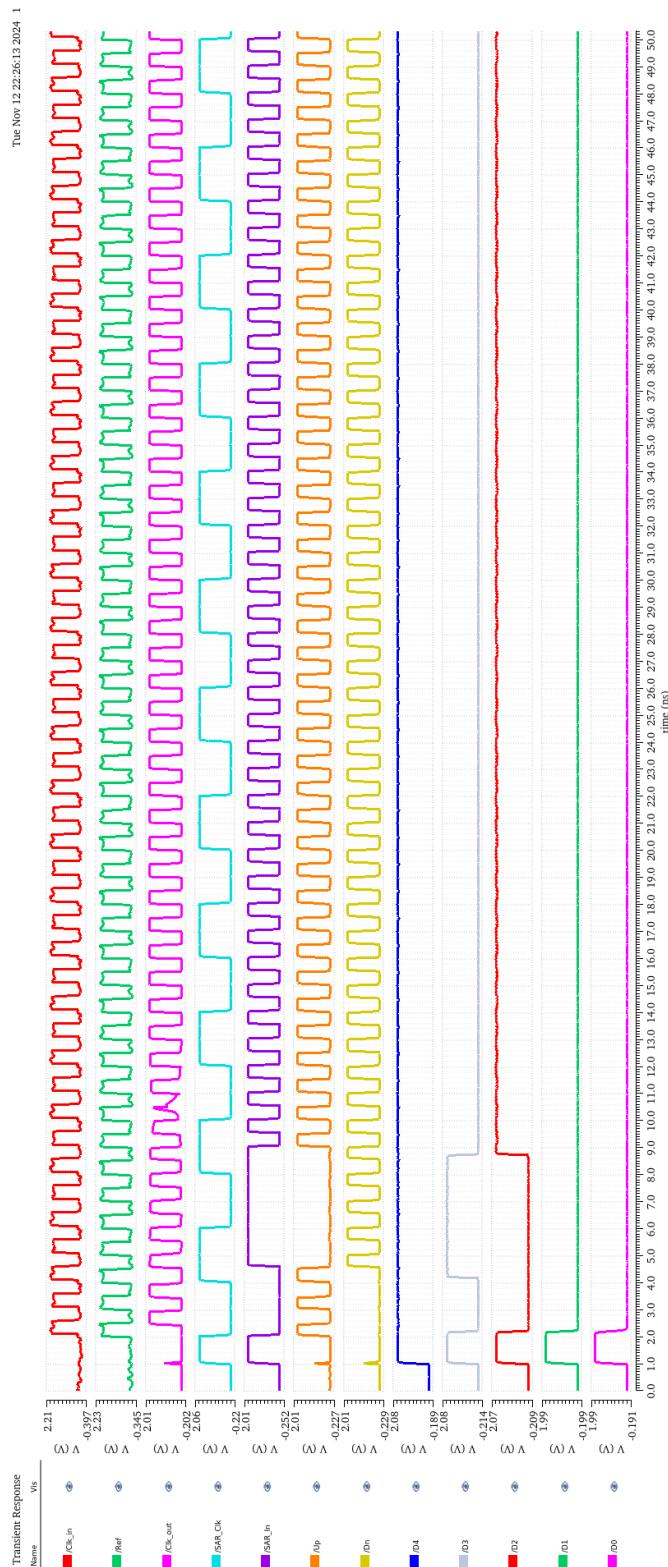


Figure 23: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

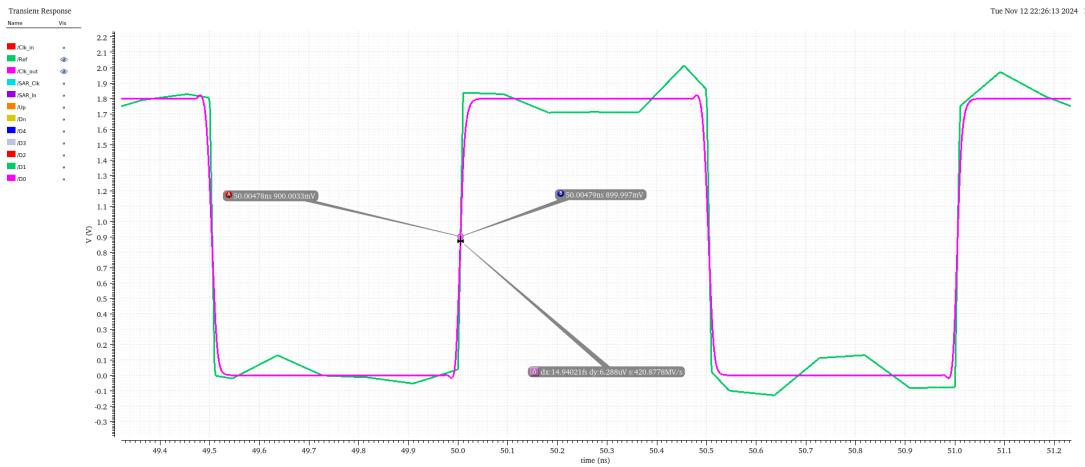
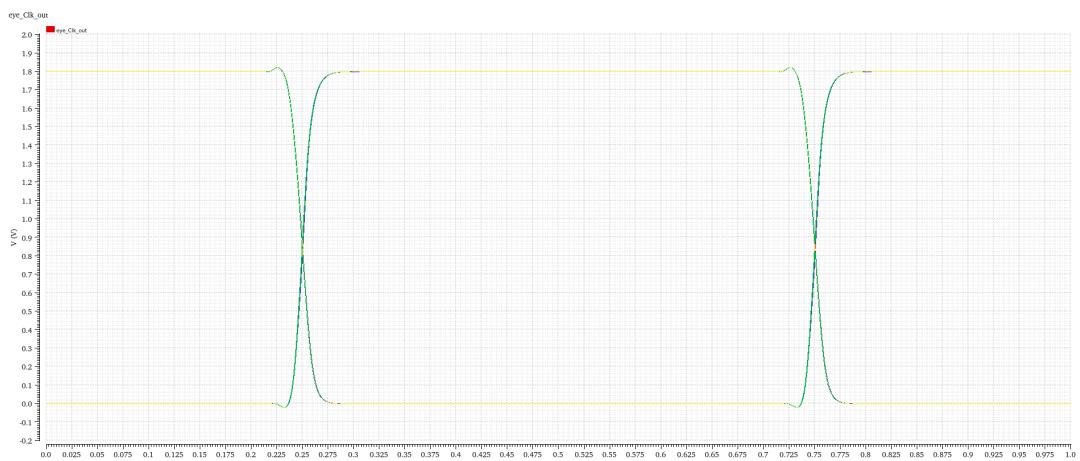
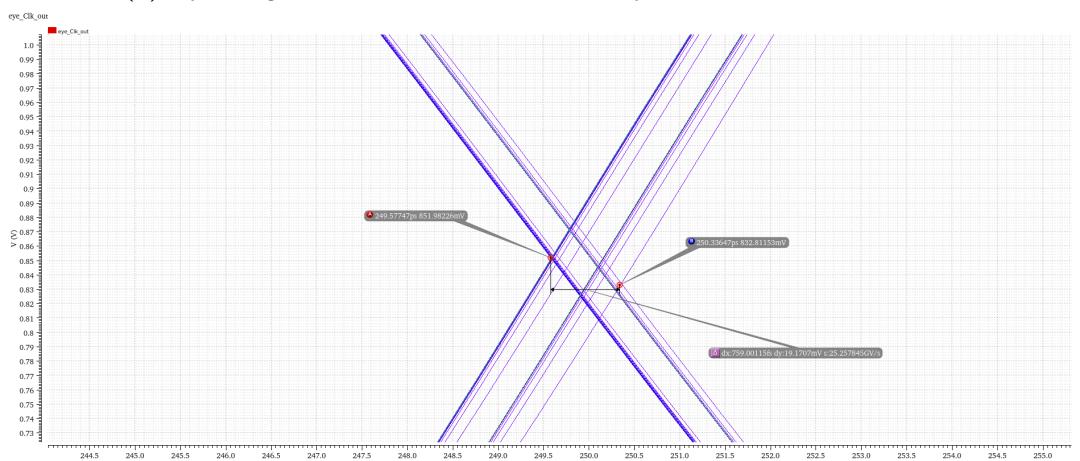


Figure 24: Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at $f = 1$ GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at $f = 1$ GHz with Delay time of 350ps

Figure 25: Eye Diagram at 1GHz and associated measurements

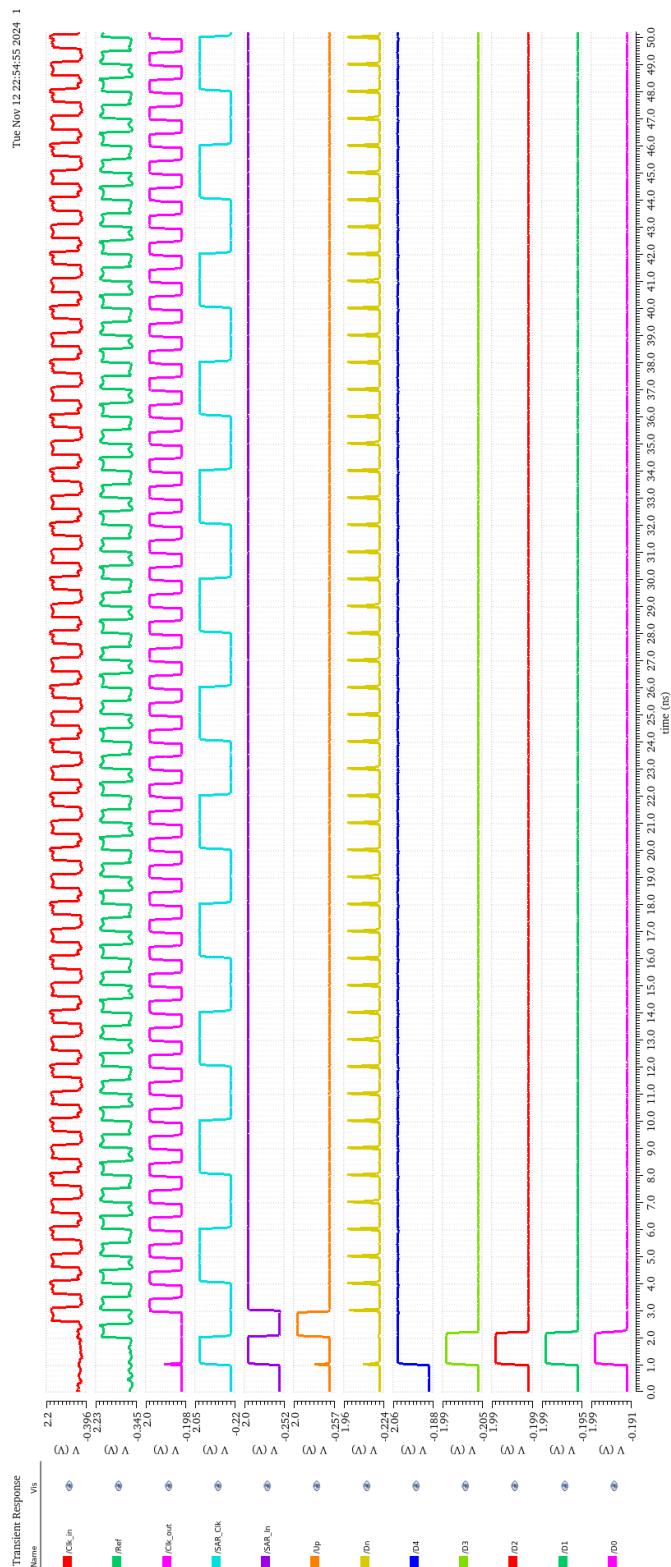


Figure 26: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

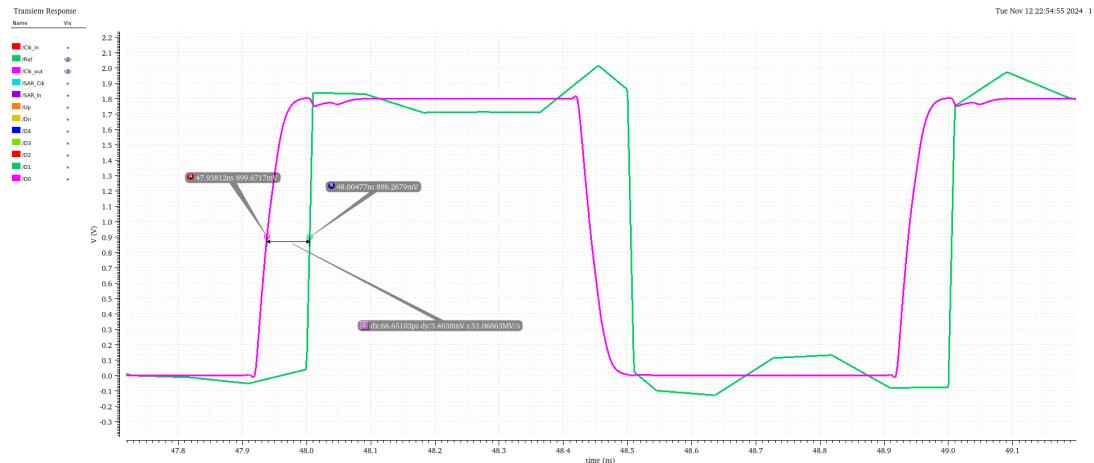
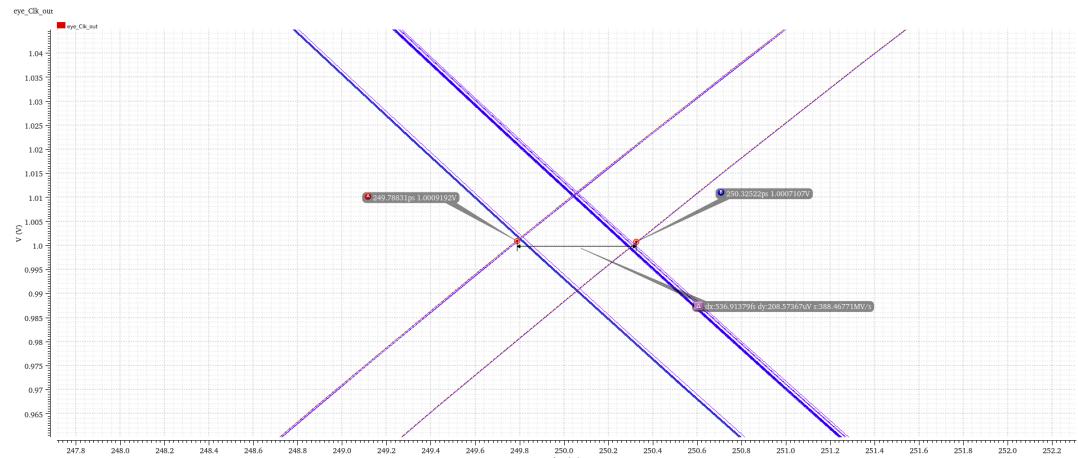


Figure 27: Steady State Phase Error of 18.77 ps at f = 1 GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at f = 1 GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at f = 1 GHz with Delay time of 350ps

Figure 28: Eye Diagram at 1GHz and associated measurements

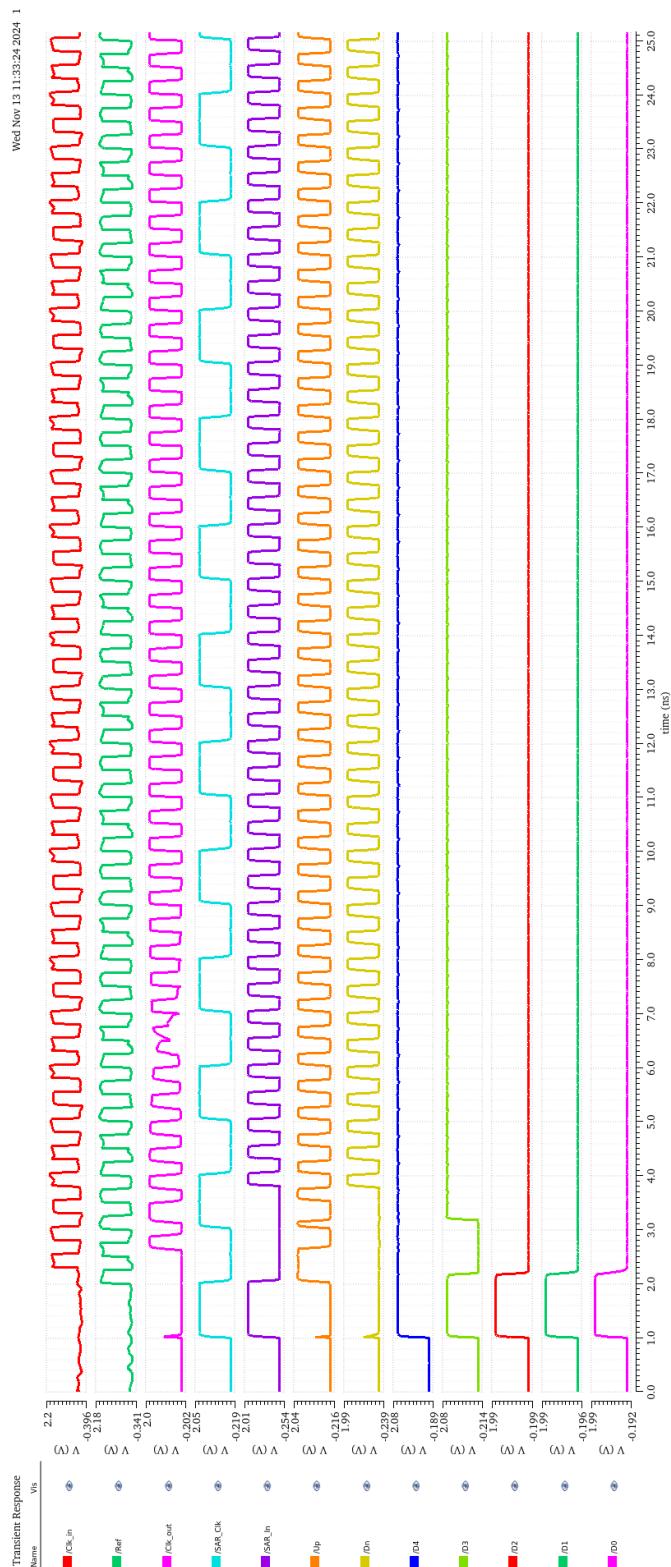


Figure 29: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

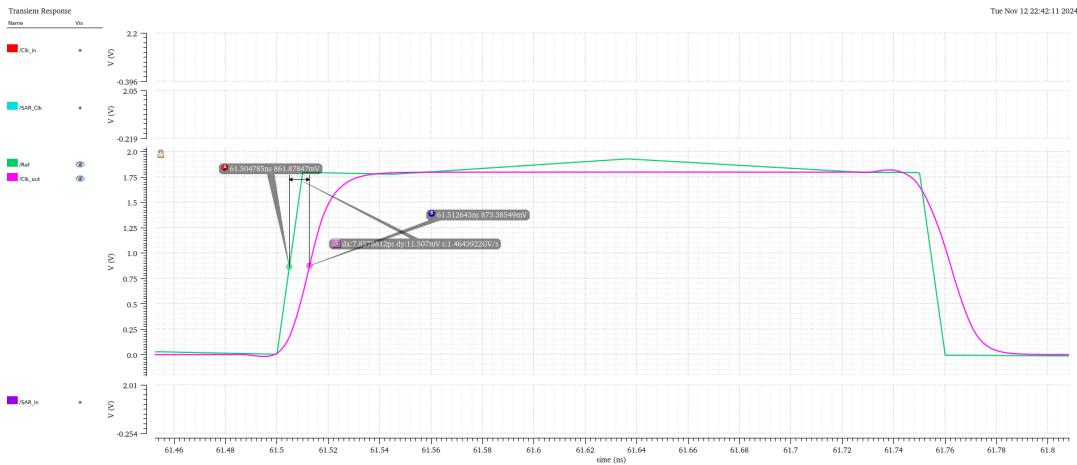
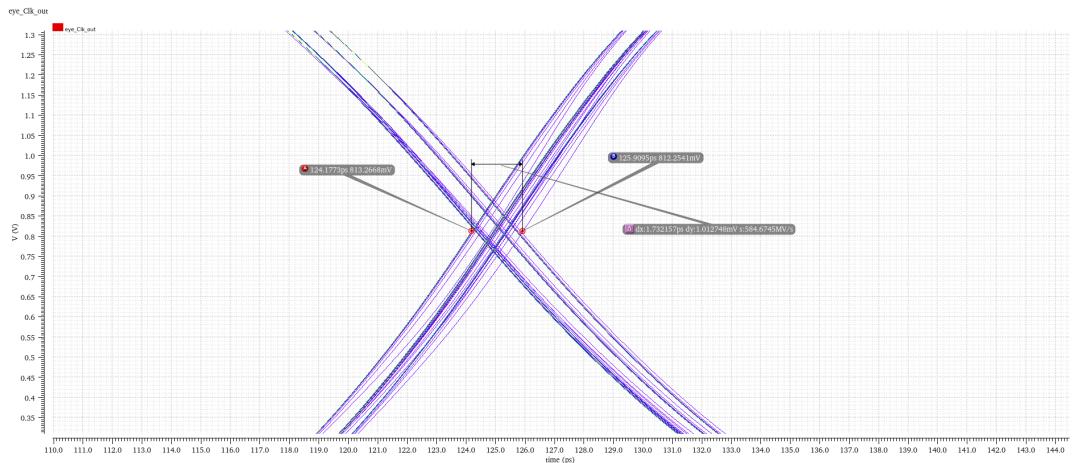


Figure 30: Steady State Phase Error of 18.77 ps at f = 1 GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at f = 1 GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at f = 1 GHz with Delay time of 350ps

Figure 31: Eye Diagram at 1GHz and associated measurements

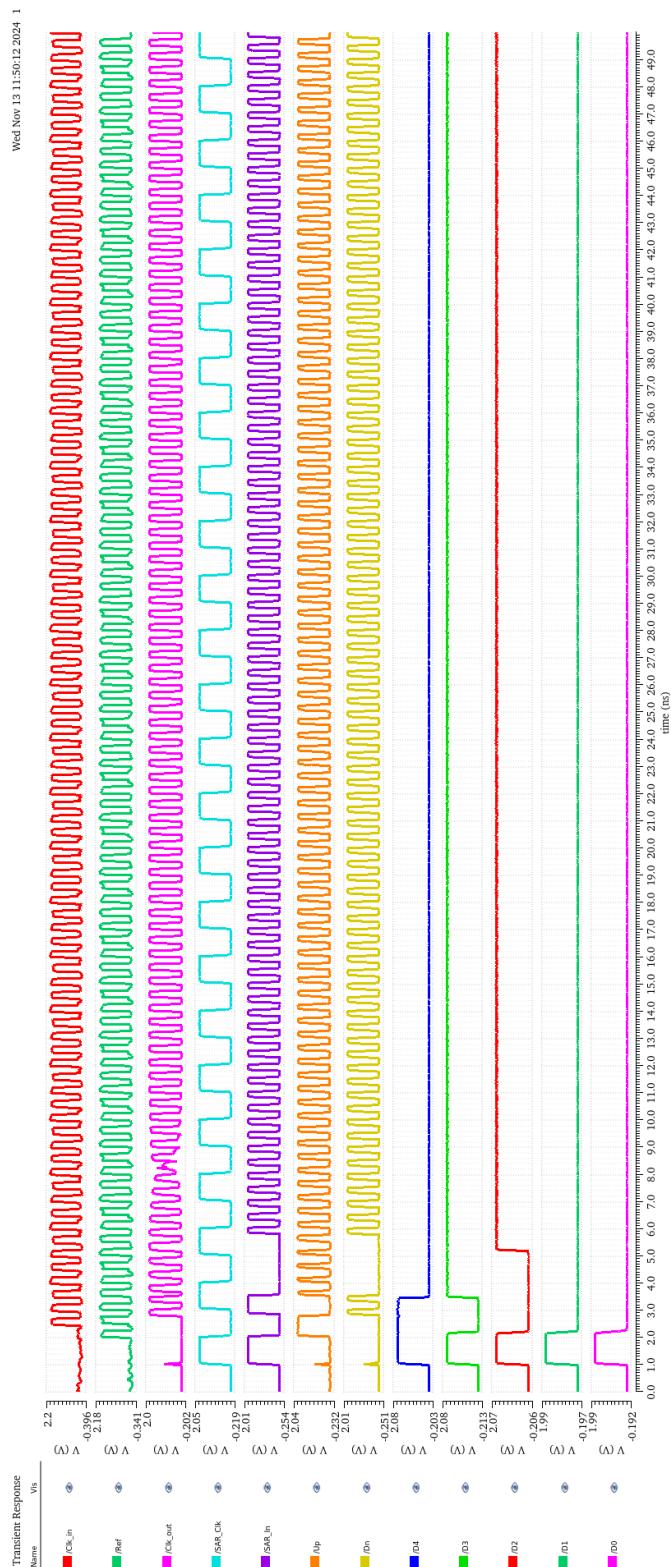


Figure 32: DLL Timing Diagram: Locking Stage at $f = 1$ GHz with Delay time of 200 ps

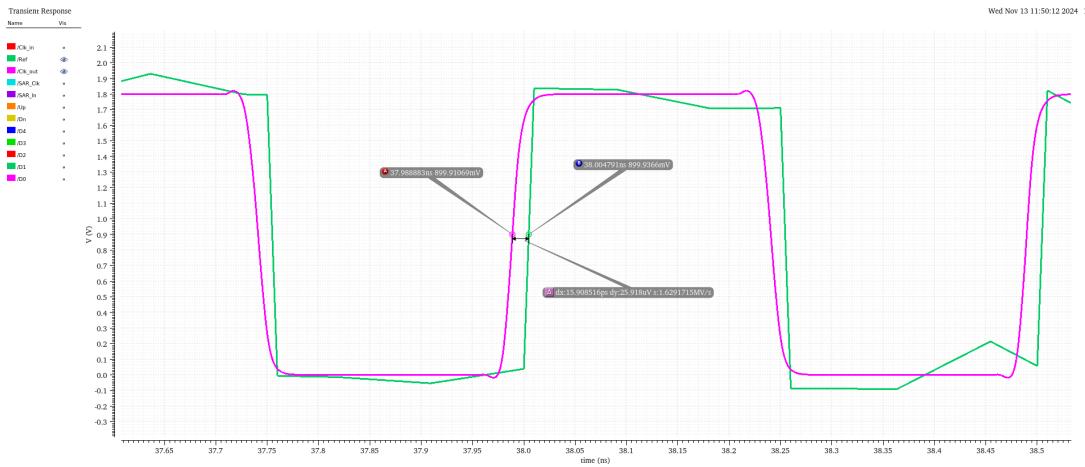
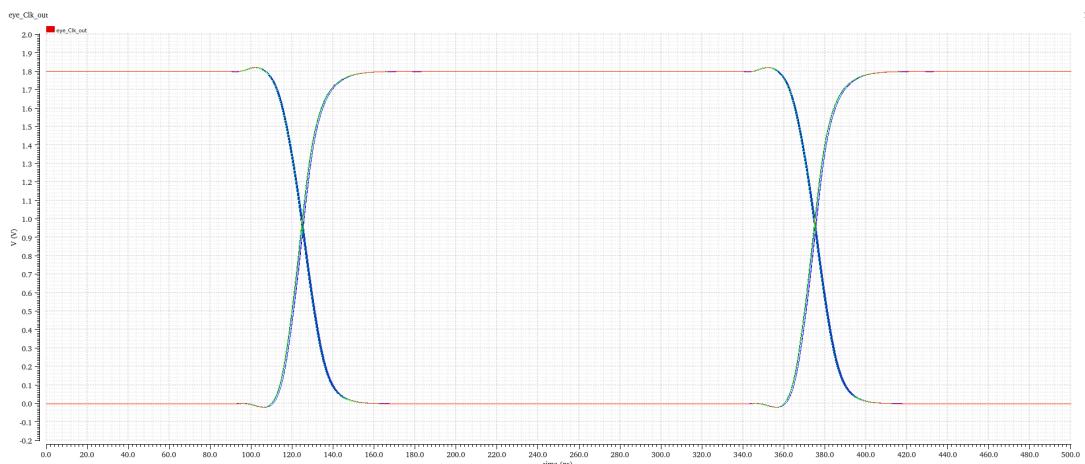
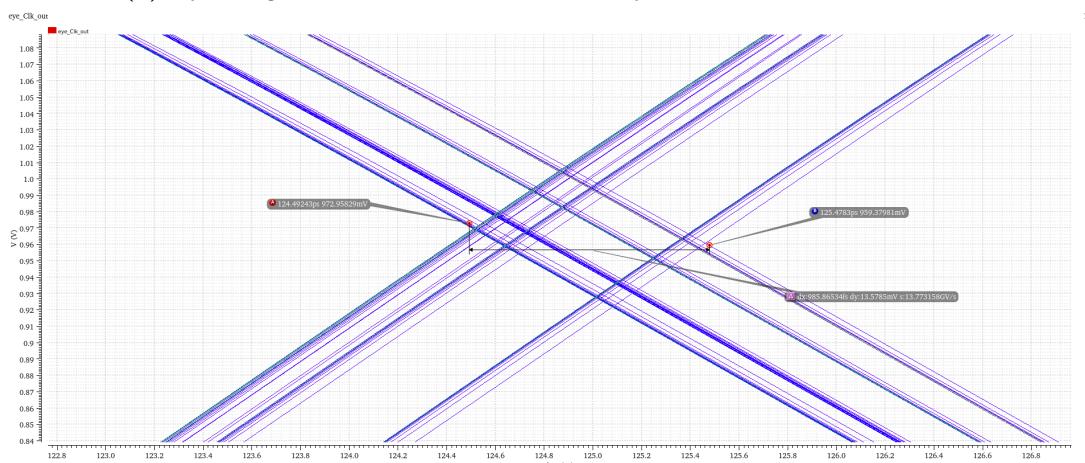


Figure 33: Steady State Phase Error of 18.77 ps at $f = 1$ GHz between Reference and Output signals with Delay time of 200ps between Reference and Input Clock



(a) Eye Diagram Plot at $f = 1$ GHz to analyze effects of noise on circuit



(b) Cross-Over Point with maximum jitter of 9.060 fs at $f = 1$ GHz with Delay time of 350ps

Figure 34: Eye Diagram at 1GHz and associated measurements

9 Conclusion and Future Scope

In this project, we successfully designed and implemented a Delay Locked Loop (DLL) at the transistor level, utilizing key components such as the Phase-Frequency Detector, Delay Line and a Successive Approximation Register that replaced the requirement of the UP-DOWN Counter, which was initially used to provide the necessary delay. Further optimizations were made by critical observations and rigorous analysis of all the circuit behaviours by carefully inserting strong buffers and other required logic circuits to reinforce smooth conduction of the DLL and obtain the expected results.

Simulations were performed at 1 GHz and 2 GHz frequencies, wherein results based on complete Timing Diagram, Steady State Phase Errors, Eye Diagrams and Cross-OVER points were plotted and verified. Further results were obtained by continually varying noise parameters and observing its effects from the Eye Diagrams and shifts in Cross-OVER points. We first implemented the DLL with a Counter. A disadvantage of that circuit was the increased power consumption as well as longer locking time. We then designed a SAR Register based on its operation and incorporated it into our design. The resulting DLL had the virtues of lower power consumption as well as faster locking time, both are which are key parameters that enhances the overall performance metrics of the DLL.

For the future scope, certain modifications can be undertaken to enhance the performance of the DLL:

1. To upgrade this design to a edge combining, frequency multiplying DLL to generate multiple harmonics of the reference signal, which are phase locked to it.
2. To make the DLL circuit to reliably work at frequencies higher than 2GHz by introducing some high bandwidth logic such that its industrial utility increases.
3. To design the layout and design it as an IP Block so that it can be used as a drop-in part in larger ASICs, FPGAs or Processors.
4. To design the circuit in smaller node such as 45nm or 32nm to achieve faster operation and lower power consumption.

References

- [1] Laxminidhi T., Rekha S., and Kirankumar H. Lad. “A Dead-Zone-Free Zero Blind-Zone High-Speed Phase Frequency Detector for Charge-Pump PLL”. In: *Circuits, Systems, and Signal Processing* 39.8 (2020).
- [2] Yang Rong-Jyi and Liu Shen-Iuan. “A 2.5 GHz All-Digital Delay-Locked Loop in 0.13 μm CMOS Technology”. In: *IEEE JOURNAL OF SOLID-STATE CIRCUITS* 42.11 (2007).
- [3] Dehng Guang-Kaai et al. “Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop”. In: *IEEE JOURNAL OF SOLID-STATE CIRCUITS* 35.8 (2000).
- [4] Bayram Erkan et al. “1.5–3.3 GHz, 0.0077 mm², 7 mW All-Digital Delay-Locked Loop With Dead-Zone Free Phase Detector in 0.13 μm CMOS”. In: *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I* 65.1 (2018).
- [5] Behzad Razavi. “The Delay-Locked Loop [A Circuit for All Seasons]”. In: *IEEE Solid-State Circuits Magazine* 10.3 (2018), pp. 9–15. DOI: [10.1109/MSSC.2018.2844615](https://doi.org/10.1109/MSSC.2018.2844615).
- [6] M. Bazes. “A novel precision MOS synchronous delay line”. In: *IEEE Journal of Solid-State Circuits* 20.6 (1985), pp. 1265–1271. DOI: [10.1109/JSSC.1985.1052467](https://doi.org/10.1109/JSSC.1985.1052467).
- [7] M.G. Johnson and E.L. Hudson. “A variable delay line PLL for CPU-coprocessor synchronization”. In: *IEEE Journal of Solid-State Circuits* 23.5 (1988), pp. 1218–1223. DOI: [10.1109/4.5947](https://doi.org/10.1109/4.5947).
- [8] J. J. Spilker and D. T. Magill. “The Delay-Lock Discriminator-An Optimum Tracking Device”. In: *Proceedings of the IRE* 49.9 (1961), pp. 1403–1416. DOI: [10.1109/JRPROC.1961.287899](https://doi.org/10.1109/JRPROC.1961.287899).