

Analog/Mixed-Signal Simulation and Modeling

Course Project:

PLL Design and Verilog-A Behavioral Modeling



By:
Tarek Mohamed Sayed Hussein

Table of Contents

1. Introduction	4
2. Top-Down Design Methodology	5
3. Design of the Phase Locked Loop.....	6
4. Behavioral Simulation using Verilog-A.....	11
4.1 Tri-State Phase Frequency Detector (PFD)	11
4.2 Charge Pump (CHP).....	15
4.3 Voltage Contrroled oscillator (VCO).....	17
4.4 frequency divider (FD).....	19
4.5 system level simulation	20
5. Transistor Level Design.....	22
5.1 frequency divider (FD) Design	22
5.2 frequency divider Bottom-up verification	24
5.3 Phase frequency detector (PFD) Design.....	26
5.4 Phase frequency detector (PFD) Bottom-up verification	29
5.5 Incremental inteagation step 1.....	30
5.6 charge pump (chp) Design.....	31
5.7 charge pump (chp) Bottom-up verification	33
5.8 Incremental inteagation step 2	34
6. Future Work.....	35
7. Conclusion.....	36
8. References	36

List of Tables

Table 1: Type II PLL targeted specifications.....	8
Table 2 : Summary of the choice of the open loop parameters.....	9

List of figures

Figure 1: Block diagram of Type II charge pump PLL.....	4
Figure 2 : PLL Type II schematic using CppSim Sue2.....	10
Figure 3 : PLL Type II simulation using CppSim view.....	10
Figure 4: Tri-state PFD state diagram.	11
Figure 5: Tri-state PFD behavioral model using Verilog-A.	12
Figure 6: Simple testbench for the Tri-state PFD Verilog-A.	13
Figure 7: PFD timing when FB is leading.	13
Figure 8: PFD timing when REF is leading.	14
Figure 9: PFD timing when FB is matching REF.	14
Figure 10: charge pump (CHP) behavioral model using Verilog-A.....	15
Figure 11: Simple testbench for the charge pump (CHP) Verilog-A.....	16
Figure 12: Charge pump average output current vs. phase error (in ps) between REF and FB. The period of REF and FB is ≈ 1 ns.	16
Figure 13: VCO behavioral model using Verilog-A.	17
Figure 14: Simple testbench for the VCO Verilog-A.	18
Figure 15: VCO output frequency Vs Vc (input voltage).	18
Figure 16: frequency divider behavioral model using Verilog-A.	19
Figure 17: Simple testbench for the frequency divider Verilog-A.	19
Figure 18: frequency divider transient simulation.	20
Figure 19: PLL Behavioral simulation using Verilog-A.	20
Figure 20: VCO control voltage (behavioral simulation).	21
Figure 21: PLL in locked state (behavioral simulation). (1) VCO control (2) VCO output....	21
Figure 22: synchronous (divide by 8) frequency divider schematic.....	22

Figure 23: the basic gates used in the D flip-flop Design.	23
Figure 24: (Master-Slave) D flip-flop Design using back-to-back static latches.	23
Figure 25: synchronous (divide by 8) frequency divider cadence virtuoso schematic.....	23
Figure 26: the designed frequency divider transient simulation results.	24
Figure 27: VCO control voltage (divider Bottom-up verification).	25
Figure 28: PLL in locked state (divider Bottom-up verification).	25
Figure 29: Typical digital Tri-state PFD schematic.	26
Figure 30: 2-input Nand gate schematic.	26
Figure 31: D flip-flop Design with extra asynchronous Reset input.	27
Figure 32: PFD timing when REF is leading (transistor-level simulation).	27
Figure 33: PFD timing when FB is matching REF (transistor-level simulation).....	28
Figure 34: PFD timing when FB is leading (transistor-level simulation).....	28
Figure 35: VCO control voltage (PFD Bottom-up verification).....	29
Figure 36: PLL in locked state (PFD Bottom-up verification).....	29
Figure 37:incremental integration step 1 using hierarchy editor.	30
Figure 38: VCO control voltage (incremental integration step 1).	30
Figure 39: PLL in locked state (incremental integration step 1).	31
Figure 40: NMOS-switch current steering charge pump Topology, but Idc is replaced by 10 uA.....	31
Figure 41: NMOS-switch current steering charge pump cadence virtuoso Schematic.	32
Figure 42: VCO control voltage (CHP Bottom-up verification).....	33
Figure 43: PLL in locked state (CHP Bottom-up verification).	33
Figure 44: incremental integration step2 using hierarchy editor.	34
Figure 45: VCO control voltage (incremental integration step 2).	34
Figure 46: PLL in locked state (incremental integration step 2).	35

1. Introduction

Phase Locked Loop is a circuit which is used to generate a signal whose phase and frequency matches with the reference signal using the negative feedback Loop theory to remove the PVT Variation and the Jitter of the IC oscillators and also to generate higher frequencies which crystal oscillators can't generate. The input clock which is generated from crystal oscillator is used as a precise reference to compare with the signal generated from the IC oscillator and This loop is continued until the phase and frequency of the signal is locked. Phase Locked Loop is used in clock synthesis and generation of high clock frequency from multiple low clock frequency. If the open loop gain has one pole at origin, the PLL is considered as Type I PLL which has drawbacks as it has a limited acquisition range and also suffers from a static phase error but the biggest limitation is the tradeoff between stability and the loop bandwidth, in order to solve the above-mentioned drawbacks, a Type II PLL is addressed which has two poles at origin and, is the most widely spread PLL type. It consists of five components. The block diagram of Type II PLL circuit consisting of these blocks is shown in Fig.1. They are: -

1. Phase Frequency Detector (PFD)
2. Charge Pump (CP)
3. Low Pass Filter (LPF)
4. Voltage Controlled Oscillator (VCO)
5. Frequency Divider

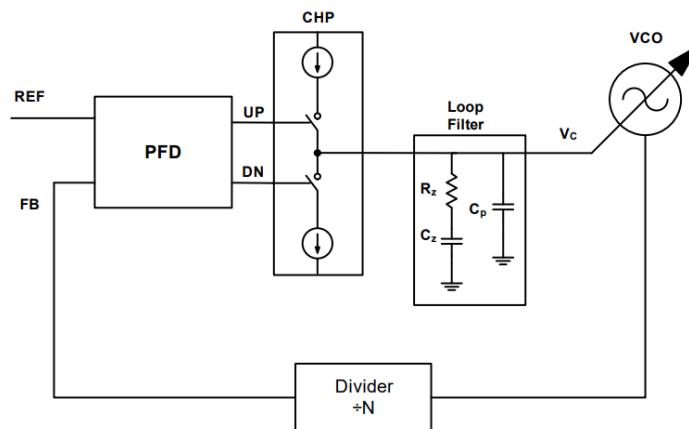


Figure 1: Block diagram of Type II charge pump PLL.

An input signal or reference signal is given to Phase Frequency Detector. Depending upon the raising edges, PFD generates two signal outputs UP and DOWN. These output signals are sent to Charge Pump which generates both positive and negative current. The output of charge pump is fed to low Pass Filter. LPF allows only the signals with low frequencies blocking the higher frequency terms that are obtained in PFD. Voltage Controlled Oscillator, is the heart of PLL circuit takes the control voltage from the output LPF. The change in output of VCO depends on the DC control voltage i.e. the output of PFD. Phase and frequency values of VCO increases due to increase in error voltage of PFD by the generation of UP at the output of PFD. In contrast, the frequency and phase of VCO output decreases due to decrease in control voltage obtained at LPF due to generation of DOWN at output of PFD.

2. Top-Down Design Methodology

Starting from the transistor level, and integrating the blocks to simulate the entire system which considered as a **Bottom-up design methodology** isn't a recommended methodology except for some cases in which the system is a little bit simple or you have designed it many times before, but generally it is not a recommended methodology for the following reasons:

- 1.it requires many numbers of design iterations, each one costs hours, and sometimes days of simulation time if the design is a much bit complex.
2. the designer will be faced by a very large number of parameters to track, and he will not be able to understand the effect of each one on the system performance.

In order to address the above-mentioned problems, A **top-down design methodology** should be used, in which the design follow described as the following procedure:

Step 1: starting by system level design in which the system architecture is chosen and the system specifications are transformed into a specification of each individual block in the system.

Step 2: verifying the system using system level behavioral simulation such as Verilog-A.

Step 3: designing and verifying each individual block in which digital blocks are designed using HDL's and synthesized using CAD tools and analog blocks cost analog designers many hours in order to meet the required specifications.

Step 4: after a block is designed, verify it using mixed signal simulation and the incremental integration as the rest of the blocks that described at a high-level act as a testbench for the designed block then after the second block in the system is designed, it is integrated with the first one and the rest of the blocks that described at a high level and verifying the system performance, then integrating three blocks and so on till we reach full integration.

From step 4, We should notice that top-down design methodology provides a formal verification plan and an incremental and methodical approach for transforming the design from an abstract block diagram to a detailed transistor-level implementation.

The rest of this document follows the above procedure from the PLL required Specification till the full system integration.

3.Design of the Phase Locked Loop

The purpose of this section is to map the required system specification to requirements on each block in the architecture shown in Fig.1.

Choice of the Loop Parameters:

The open loop transfer function of the PLL is given by Eq (1):

$$H_{OL}(S) = \frac{I_{CHP} K_{VCO} Z(S)}{SN} \quad (1)$$

Where:

I_{CHP} : the charge pump output current (A). K_{VCO} : the VCO gain (Hz/V).

N: the frequency divider ratio.

Z(S): is the loop filter impedance given by Eq (2):

$$Z(S) = \frac{1}{C_Z + C_P} \frac{SRC_z + 1}{S(SRC_S + 1)} \quad (2)$$

Where:

$$C_s = \frac{C_z C_p}{C_z + C_p} \quad (3)$$

The open loop gain can be rewritten as:

$$H_{OL}(S) = \frac{K \left(\frac{S}{\omega_z} + 1 \right)}{S^2 \left(\frac{S}{\omega_p} + 1 \right)} \quad (4)$$

Where:

$$K = \frac{I_{CHP} K_{VCO}}{(C_Z + C_P) N} \quad (5)$$

$$\omega_z = \frac{1}{R_Z C_z} = \frac{1}{\tau_z} \quad (6)$$

$$\omega_p = \frac{1}{R_Z C_s} = \frac{1}{\tau_p} \quad (7)$$

Where: C_Z generates a pole at the origin, R_Z is used to generate a zero for loop stability, and C_p is used to generate a second pole to remove high frequency ripples on the VCO control line.

The phase margin which determines the feedback loop stability can be derived from Eq (4) as:

$$\Phi_m = \tan^{-1} \left(\frac{\omega_u}{\omega_z} \right) - \tan^{-1} \left(\frac{\omega_u}{\omega_p} \right) \quad (8)$$

Where: ω_u : the unity gain frequency and it will be considered the open loop bandwidth but we should consider that the relation between the closed loop bandwidth and the unity gain frequency depends on the phase margin.

$$\omega_u = \sqrt{\omega_z \omega_p} = \alpha \omega_z = \frac{\omega_p}{\alpha} \quad (9)$$

Eq (9) is derived by differentiating Eq (8) to get the value of the loop band width that maximize the phase margin and this equation shows that it should be the geometric mean of ω_z and ω_p .

Where α^2 is the pole to zero ratio and is approximately equal to the capacitance ratio:

$$\alpha^2 = \frac{\omega_p}{\omega_z} = \frac{\tau_z}{\tau_p} = \frac{C_z}{C_s} = 1 + \frac{C_z}{C_p} \quad (10)$$

The relation between phase margin and damping factor is given by:

$$\sec(\Phi_m) - \tan(\Phi_m) = \frac{1}{4\zeta^2} \quad (11)$$

Increasing the phase margin increased the loop stability, but the damping factor increased which means slower loop response as described in Eq (11) and this is known as the trade-off between stability and the dynamic response.

Table 1: Type II PLL targeted specifications.

Spec	Value
Input clock frequency (fclk)	140 MHz
Output clock (fov)	1.12 GHz
Divider ratio (N)	8 (fixed)
VCO stages	8 (differential)
Switching speed	No requirement
Tuning resolution	No requirement

Since there is no requirement on the switching speed as shown in table 1, we can choose a high Phase margin (Let $\Phi_m = 70^\circ$).

A common rule of thumb is to choose the loop bandwidth as one tenth of the reference frequency cause all the previous continuous time analysis is valid only if the loop bandwidth is much smaller than the reference frequency (Let $\omega_u = 2\pi \times 5$ Mr/s).

The value of K_{VCO} is usually forced by the VCO architecture and ranges from 600 MHz/V to 6 GHz/V, but we will take the geometric mean of the minimum and maximum values as the design value (Let $K_{VCO} = 1.9$ GHz/V).

I_{CHP} gives the designer another degree of freedom, but we will set I_{CHP} to 10 μA , as we target a simple low-power design.

After the loop bandwidth and the phase margin have been chosen, we can calculate the time constants:

$$\tau_p = \frac{\sec(\Phi_m) - \tan(\Phi_m)}{\omega_u} \quad (12)$$

$$\tau_z = \frac{1}{\omega_u^2 \tau_p} \quad (13)$$

After substituting in Eq (12), Eq (13) and Eq (10), we get that:

$$\tau_p = 5.6126 \text{ ns} \quad \tau_z = 0.18 \text{ } \mu\text{s} \quad \alpha = 5.67$$

Using the approximation that $\alpha^2 \gg 1$, The values of the loop filter components can then be calculated as:

$$C_P = \frac{I_{CHP} K_{VCO}}{\alpha N \omega_u^2} \quad (14)$$

$$C_Z = \frac{\alpha I_{CHP} K_{VCO}}{N \omega_u^2} \quad (15)$$

$$R_Z = \frac{N \omega_u}{I_{CHP} K_{VCO}} \quad (16)$$

After substituting in Eq (14), Eq (15) and Eq (16), we get that:

$$C_P = 0.4243 \text{ pF} \quad C_Z = 13.65 \text{ pF} \quad R_Z = 13.22 \text{ k}\Omega$$

Table 2 : Summary of the choice of the open loop parameters.

Parameter	Value
ω_{ref}	$2\pi \times 140 \text{ Mr/s}$
ω_u	$2\pi \times 5 \text{ Mr/s}$
N	8
Φ_m	70°
K_{VCO}	600 MHz/V \rightarrow 6 GHz/V
I_{CHP}	10 μA

K_{VCO_mean}	1.9 GHz/V
ζ	1.2
α^2	32
C_P	0.4243 pF
C_Z	13.65 pF
R_Z	13.22 K Ω

In order to verify the correctness of the choice of the open loop parameters, we can use the CppSim system fast simulator tool as shown in Fig.2, but this step is done only as a sanity check as shown in Fig.3, and can't replace the Verilog-A simulation as it acts as a testbench environment for each block's transistor level implementation as discussed in section 2.

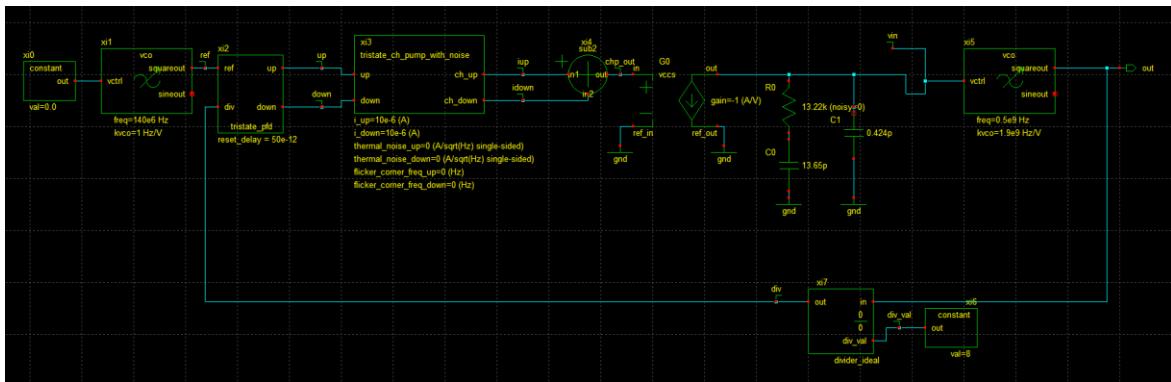


Figure 2 : PLL Type II schematic using CppSim Sue2.

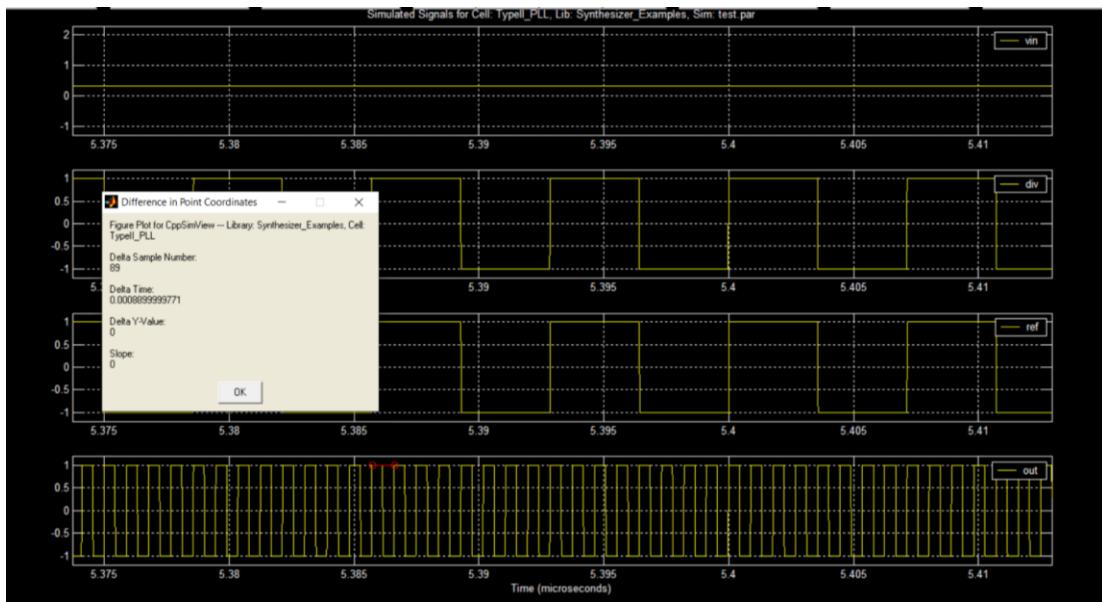


Figure 3 : PLL Type II simulation using CppSim view.

4.Behavioral Simulation using Verilog-A

The purpose of this section is to verify the system using Verilog-A for system level behavioral simulation.

4.1 TRI-STATE PHASE FREQUENCY DETECTOR (PFD)

PFD operation is clearly demystified as shown in Fig.4, and it can be summarized in the following procedure:

- 1.in the steady state (State 0) when the reference signal (A) and the feedback signal (B) have the same phase and frequency, it will assert both (Q_A and Q_B) Low.
- 2.if the reference signal (A) positive edge comes first it will assert up signal (Q_A) (State 1) High to make the VCO increases its frequency until the positive edge of the feedback signal returns it to state 0.
- 3.if the feedback (B) positive edge comes first it will assert down signal (Q_B) (State 2) High to make the VCO decreases its frequency until the positive edge of the reference signal returns it to state 0.

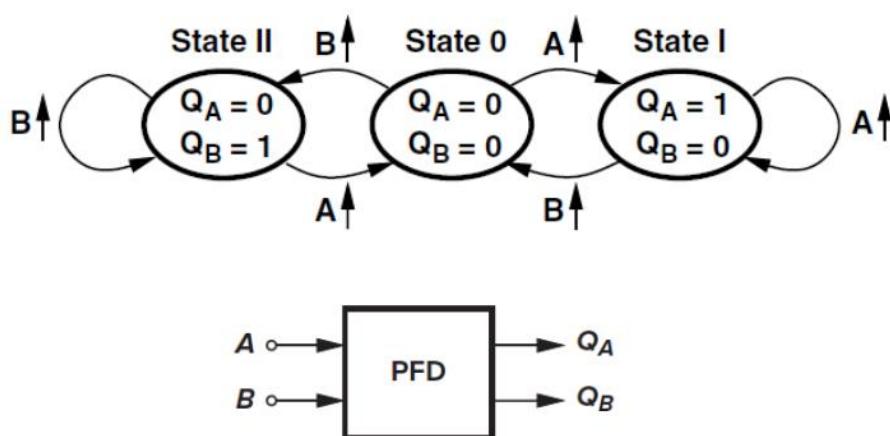


Figure 4: Tri-state PFD state diagram.

the above procedure is modeled using Verilog-A as shown in Fig.5, and have been tested and verified as shown in Fig.6, Fig.7, Fig.8 and Fig.9.

```

// AMS PLL Project: Phase Frequency Detector (PFD)

`include "constants.vams"
`include "disciplines.vams"

// REF: Reference signal
// FB: Feedback signal
// UP: Up signal (FB late)
// DN: Down signal (FB early)

module PFD(REF,FB,UP,DN);

    // VDD and threshold voltage for digital signals
    parameter real VDD = 1.2;
    parameter real thresh = 0.6;
    // rise/fall/delay times of PFD output
    parameter real trise = 10p, tfall = 10p, td = 0;

    input REF,FB;
    output UP,DN;

    electrical REF,FB,UP,DN;

    // Internal UP and DN signals
    real UP_i, DN_i;

    analog begin

        // Check DN_i state when REF arrives
        @(cross(V(REF)-thresh,1))
            if(DN_i < thresh)
                UP_i = VDD;
            else begin
                UP_i = 0;
                DN_i = 0;
            end

        // Check UP_i state when FB arrives
        @(cross(V(FB)-thresh,1))
            if(UP_i < thresh)
                DN_i = VDD;
            else begin
                UP_i = 0;
                DN_i = 0;
            end

        V(UP) <+ transition(UP_i,td,trise,tfall);
        V(DN) <+ transition(DN_i,td,trise,tfall);
    end
endmodule

```

Figure 5: Tri-state PFD behavioral model using Verilog-A.

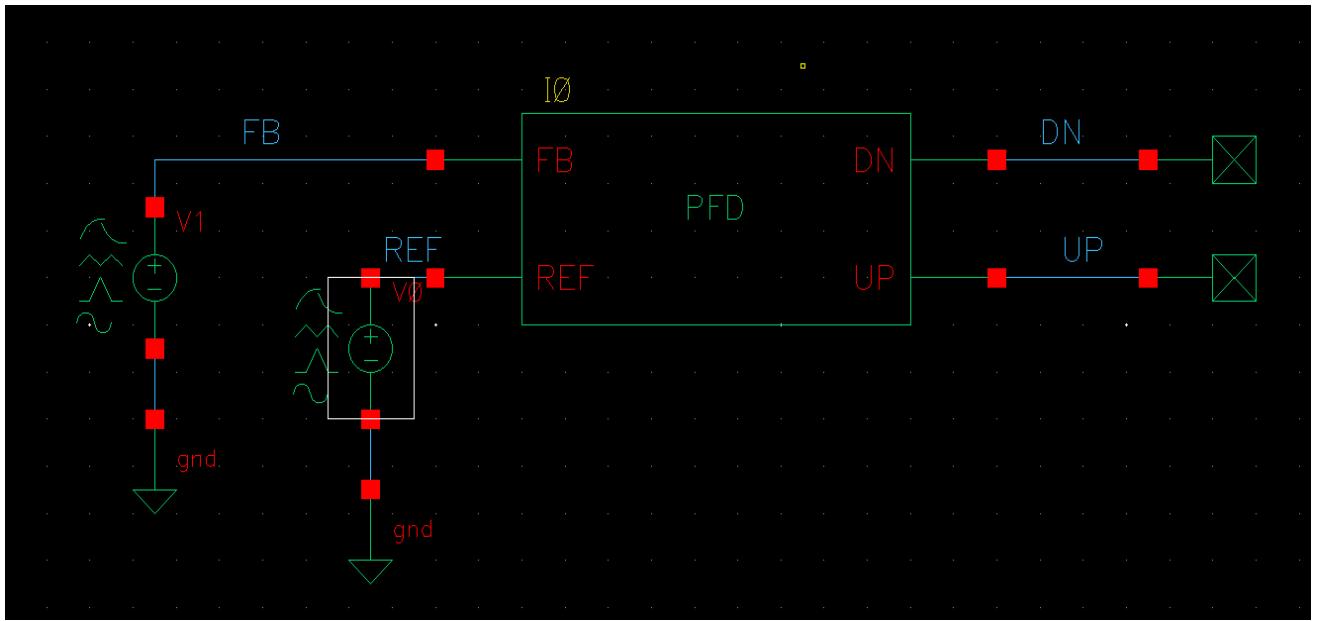


Figure 6: Simple testbench for the Tri-state PFD Verilog-A.

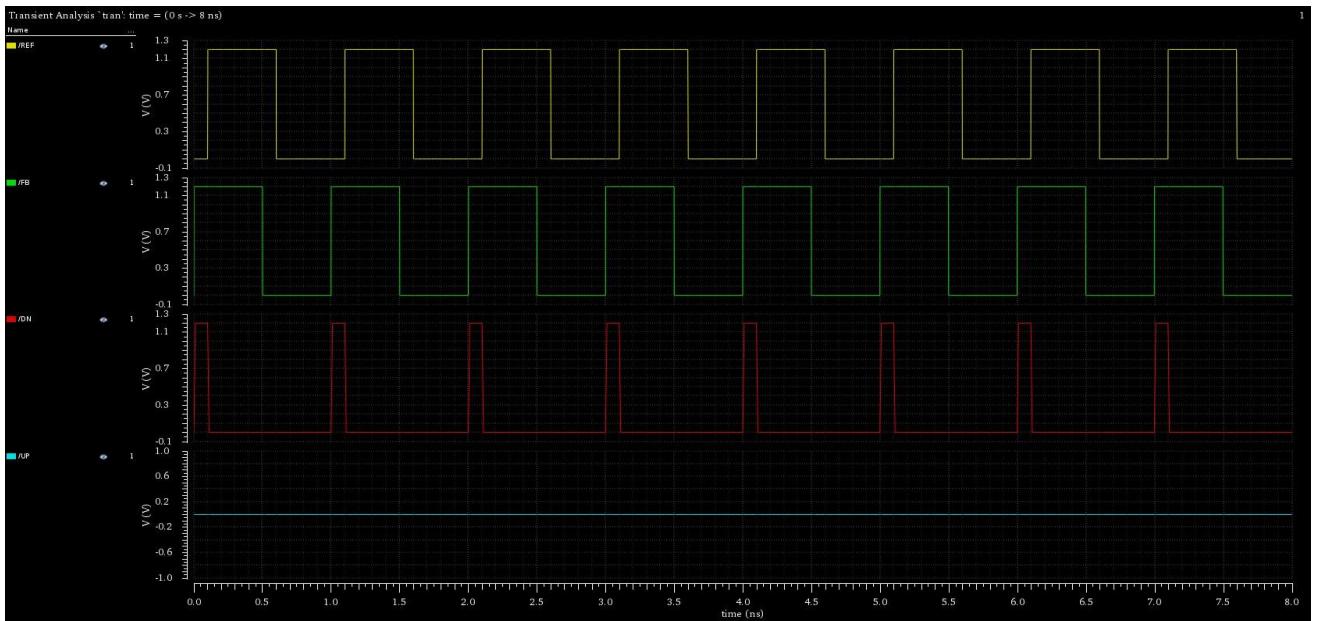


Figure 7: PFD timing when **FB** is leading.

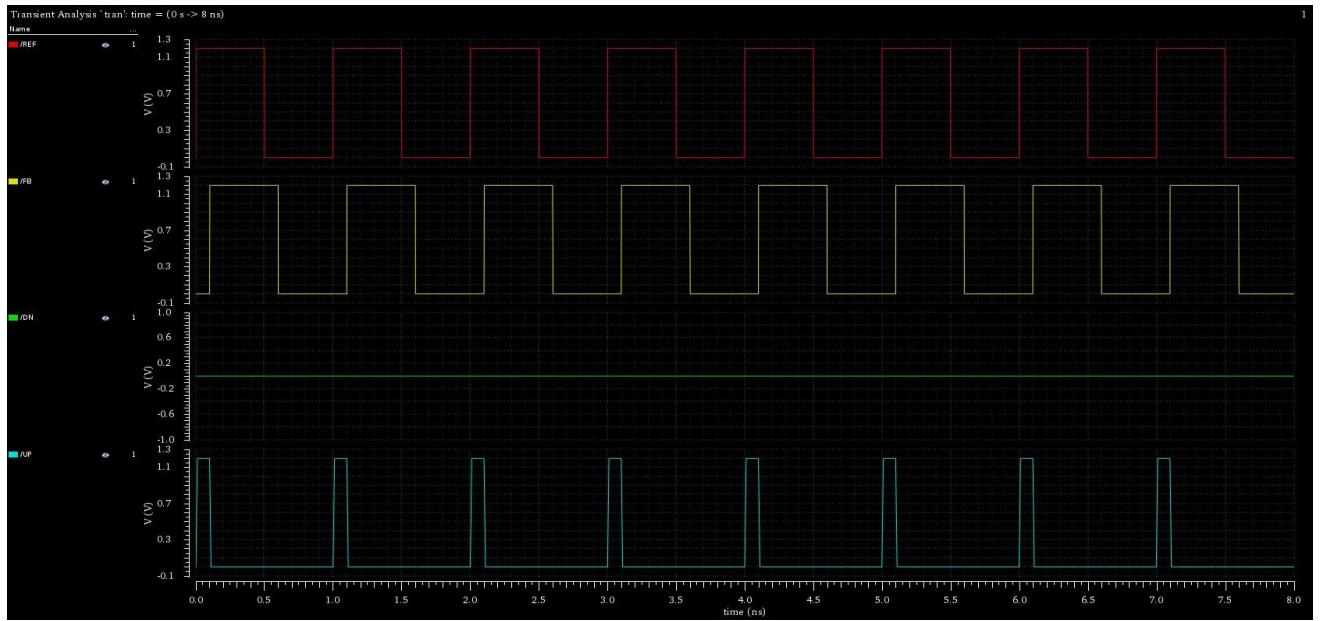


Figure 8: PFD timing when REF is leading.

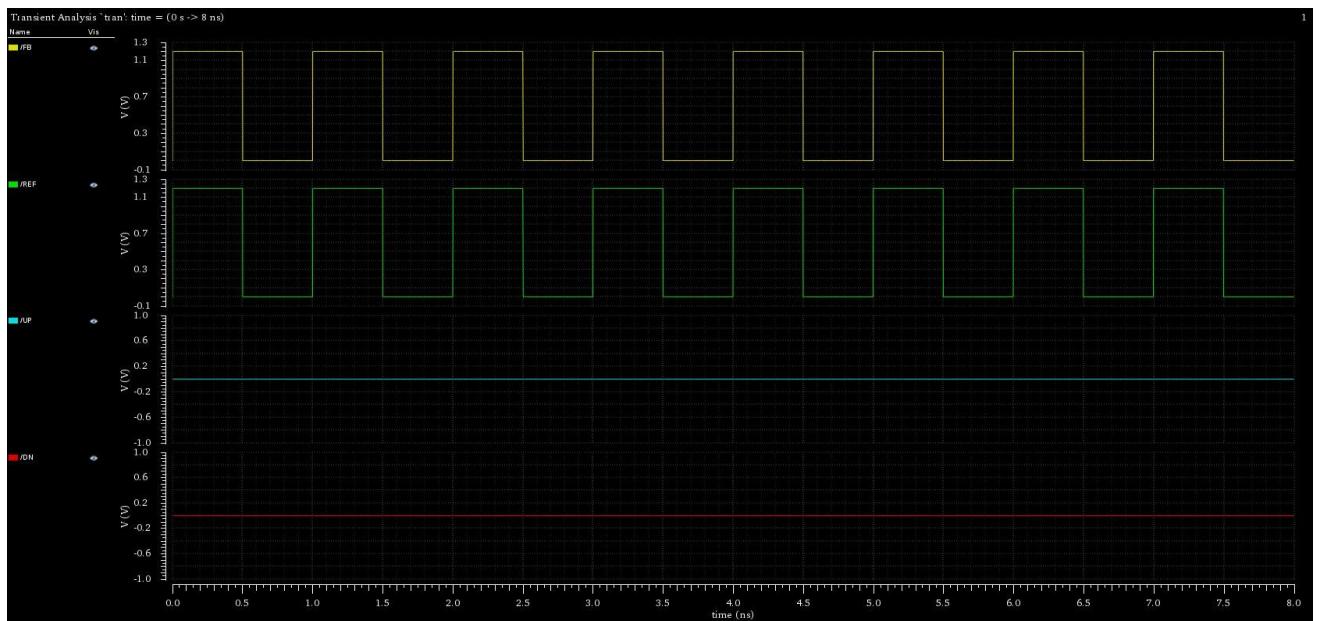


Figure 9: PFD timing when FB is matching REF.

4.2 CHARGE PUMP (CHP)

CHP operation is relatively simple and it is responsible for adding one more ideal integrator (pole at origin) by providing a current (I_{CHP}) that discharges or charges a capacitor that increases the VCO control voltage if the reference signal is faster than the feedback signal and decrease the control voltage in the opposite case.

```
// AMS PLL Project: Charge Pump (CHP)

`include "constants.vams"
`include "disciplines.vams"

// UP: Up signal
// DN: Dn signal
// IOUT: CHP current output
// Since the output is current, IOUT cannot be left unconnected (o.c.) in the testbench

module CHP(UP,DN,IOUT);
    input UP,DN;
    inout IOUT;
    electrical UP,DN,IOUT;

    // ichp: CHP current
    parameter real ichp = 10u from [0:inf];
    // Threshold voltage for digital signals
    parameter real thresh=0.6;
    // rise/fall/delay times of CHP output
    parameter real trise=10p, tfall=10p, td=0;

    // Internal variable for CHP output current
    real IOUT_i = 0;

    analog begin
        // Generate events at UP and DN transitions
        @(cross(V(UP)-thresh,0))
        ;
        @(cross(V(DN)-thresh,0))
        ;

        if ((V(UP) > thresh) && (V(DN) < thresh))
            IOUT_i = ichp;
        else if ((V(UP) < thresh) && (V(DN) > thresh))
            IOUT_i = -ichp;
        else
            IOUT_i = 0;

        I(IOUT) <+ transition(IOUT_i,td,trise,tfall);
    end
endmodule
```

Figure 10: charge pump (CHP) behavioral model using Verilog-A.

If the up signal is high and the down signal is low, the output current will be a pulse of $+I_{CHP}$ for the same width of the up-signal pulse, and $-I_{CHP}$ in case of the down signal is high and the up signal is low, in a little few word “it converts the PFD output signal to a charge and this charge is proportional to

PFD pulse widths”, this operation is modeled using Verilog-A as shown in Fig.10.

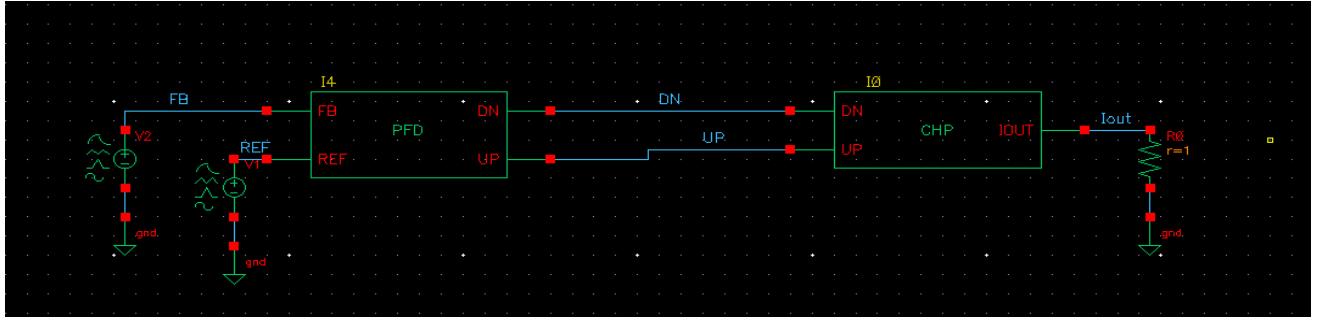


Figure 11: Simple testbench for the charge pump (CHP) Verilog-A.

The Verilog-A model have been tested as shown in Fig.11 using parametric analysis with the phase shift as a parameter and the average current is plotted vs the phase shift as shown in Fig.12, and this is a reasonable plot as it is consistent with the CHP operation as it provides average current equal to 5 uA in case of the phase shift is $\pi/2$ (delay=500 ps) and we can predict from this linear relation that the average current will be 10 uA in case of the phase shift is π as the PFD UP signal will be always high and so as the charge pump output current.

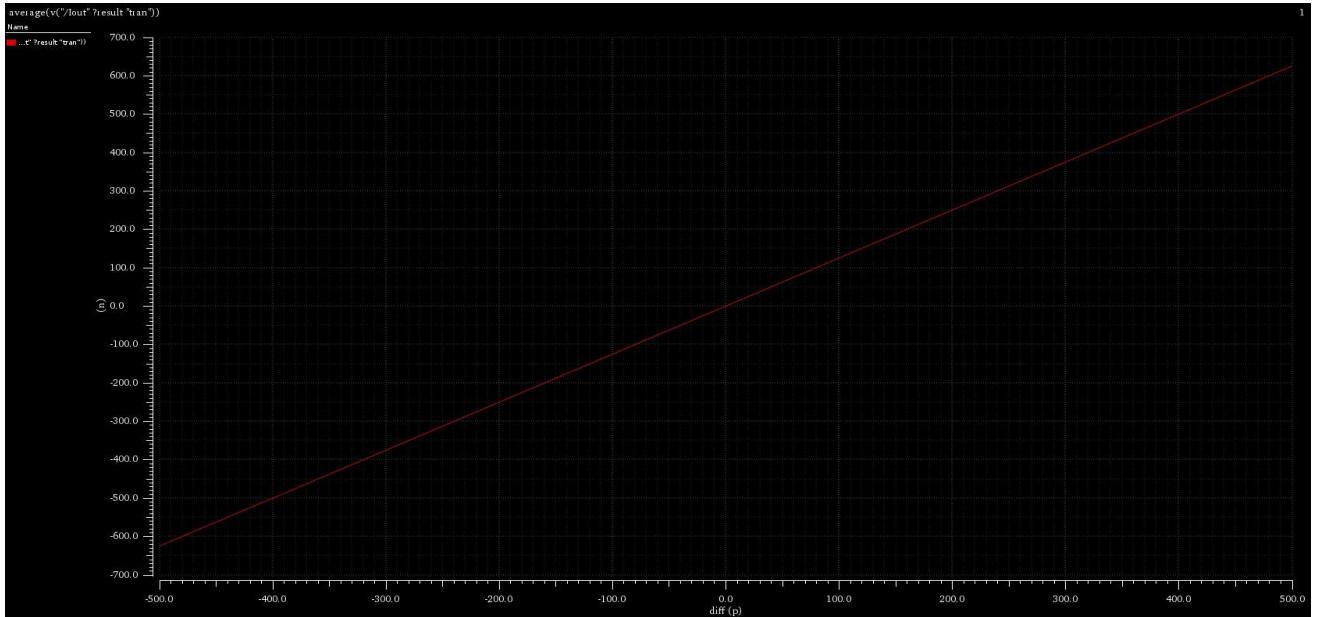


Figure 12: Charge pump average output current vs. phase error (in ps) between REF and FB.
The period of REF and FB is ≈ 1 ns.

4.3 VOLTAGE CONTROLLED OSCILLATOR (VCO)

VCO operation is to generate a clock signal whose frequency is proportional to the input voltage (control voltage), and this relation should be a linear relation, a simple Verilog-A model for the VCO is modeled as shown in Fig.13.

```
// AMS PLL Project: Voltage Controlled Oscillator (VCO)

`include "constants.vams"
`include "disciplines.vams"

module VCO(VCTRL,VOUT);

parameter real VHIGH = 1.2;
parameter real Vmin=0.2;
parameter real Vmax=1 from (Vmin:inf);
parameter real Fmin=0.5G from (0:inf);
parameter real Fmax=1.5G from (Fmin:inf);
parameter real trise=10p, tfall=10p, td=0;

input VCTRL;
output VOUT;
voltage VCTRL,VOUT;

real freq, phase, VOUT_i, sine;

analog begin

// compute the freq from the input voltage
freq =((V(VCTRL) - Vmin)*(Fmax - Fmin) / (Vmax - Vmin)) + Fmin;
// bound the frequency
if (freq > Fmax) freq = Fmax;
if (freq < Fmin) freq = Fmin;

// calculate the phase (modulo 2*pi)
phase=2 * `M_PI * idtmod(freq, 0.0, 1.0, -0.5);

// generate the output
sine = sin(phase);

@(cross(sine,0))
;

if (sine > 0)
    VOUT_i= VHIGH;
else
    VOUT_i= 0;

V(VOUT) <+ transition(VOUT_i,td,trise,tfall);

// bound the time step
$bound_step(0.1/freq);
end
endmodule
```

Figure 13: VCO behavioral model using Verilog-A.

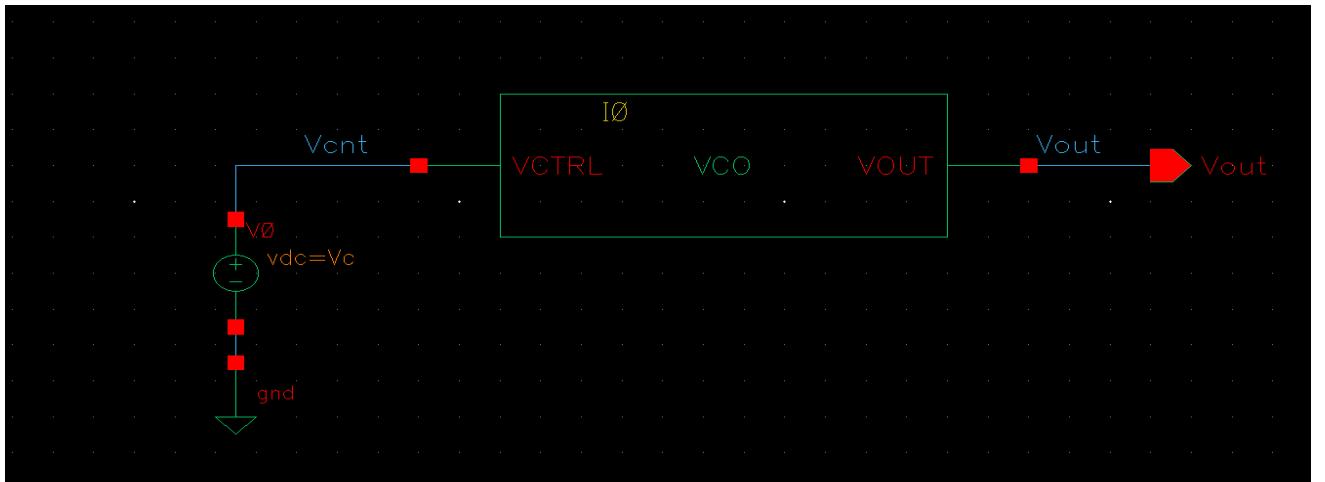


Figure 14: Simple testbench for the VCO Verilog-A.

The Verilog-A model have been tested as shown in Fig.14 using parametric analysis with the control voltage as a parameter and the frequency of the output signal is plotted as shown in Fig.15, which reflects the characteristic of the VCO.

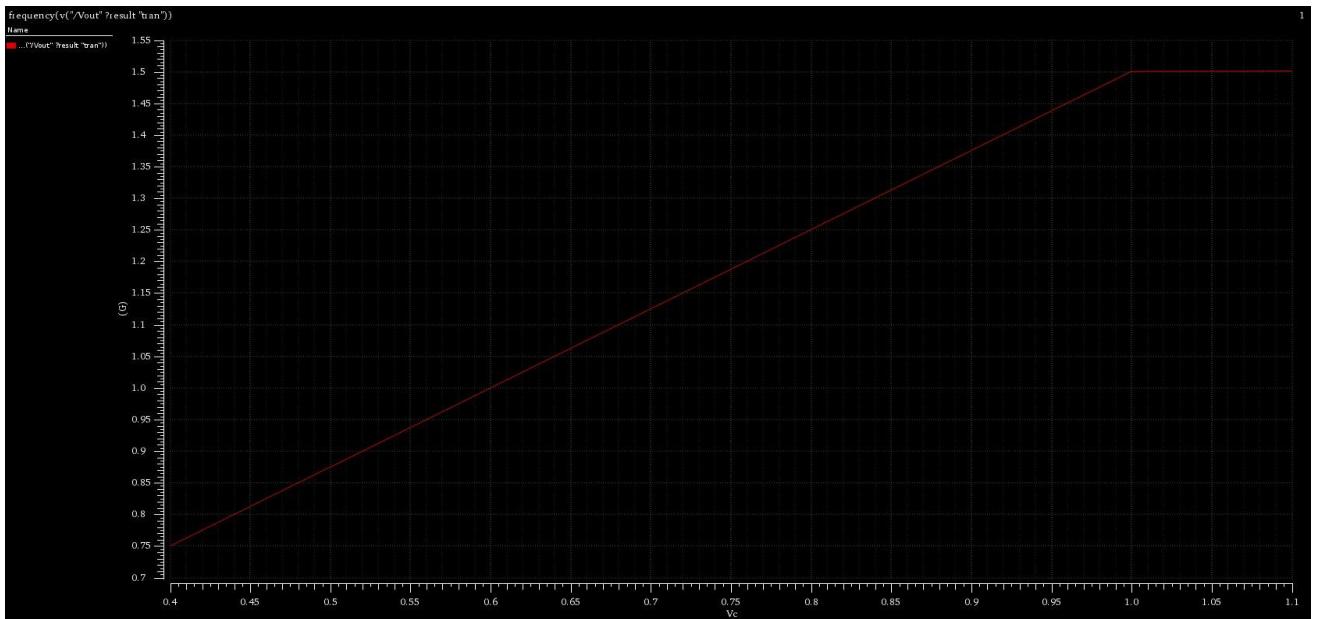


Figure 15: VCO output frequency Vs Vc (input voltage).

4.4 FREQUENCY DIVIDER (FD)

The purpose of this divider is to divide the frequency of the input signal by 8 and it is modeled clearly as shown in Fig.16, and have been tested and verified as shown in Fig.17, Fig.18.

```
// AMS PLL Project: Frequency Divider
`include "constants.vams"
`include "disciplines.vams"

module Divider(VIN,VOUT);

    output VOUT; voltage VOUT; // output
    input VIN; voltage VIN; // input (edge triggered)
    parameter real vh=1.2; // output voltage in high state
    parameter real vl=0; // output voltage in low state
    parameter real vth=0.6; // threshold voltage at input
    parameter integer ratio=8 from [2:inf]; // divider ratio
    parameter real tt=10p from (0:inf); // transition time of output signal
    parameter real td=0 from [0:inf]; // average delay from input to output

    real count=0;
    real out_value;

    analog begin

        @(cross(V(VIN) - vth, 1))
        begin
            if (count==floor(ratio/2))
                out_value = vh;
            else if (count==ratio)
                begin
                    out_value = vl;
                    count=0;
                end
            count = count + 1;
        end
        V(VOUT) <+ transition(out_value,td,tt);
    end
endmodule
```

Figure 16: frequency divider behavioral model using Verilog-A.

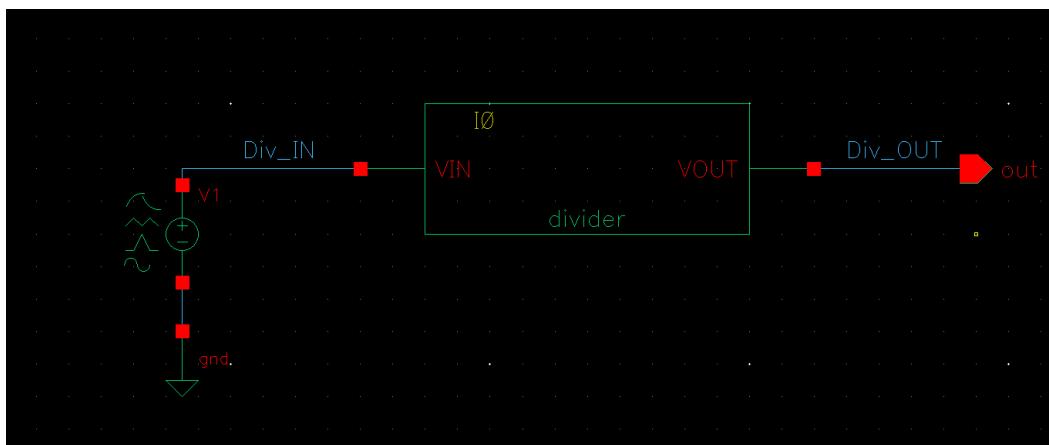


Figure 17: Simple testbench for the frequency divider Verilog-A.

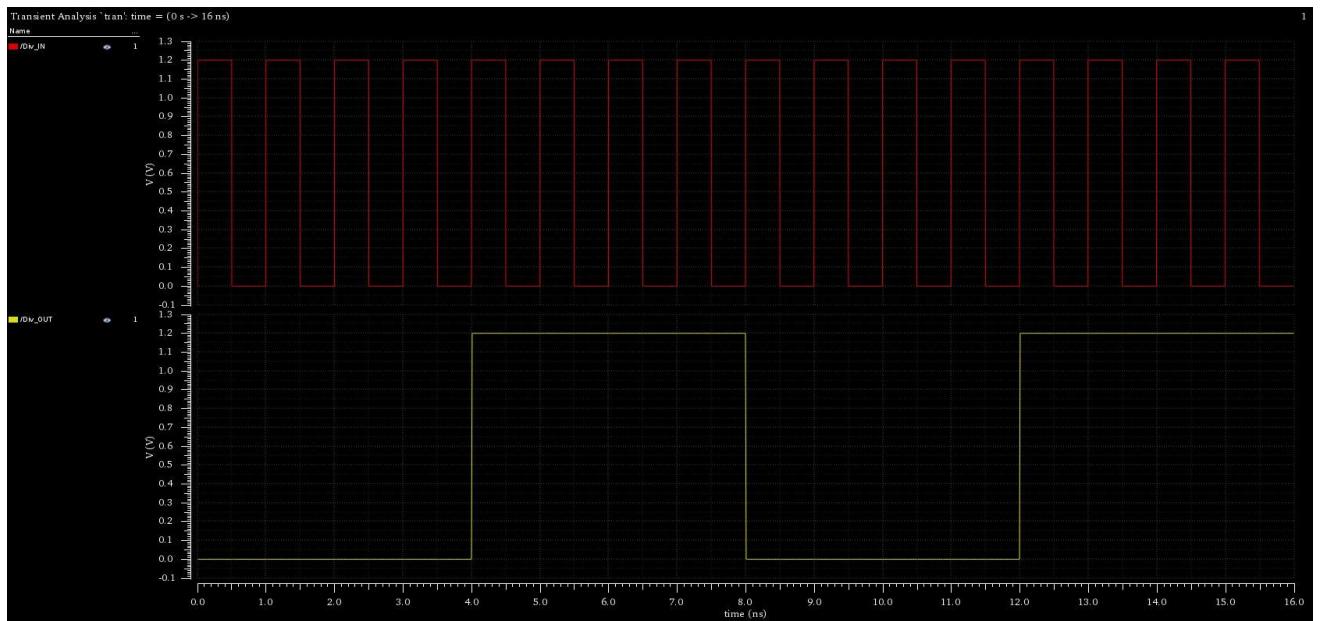


Figure 18: frequency divider transient simulation.

4.5 SYSTEM LEVEL SIMULATION

Behavioral simulation of the PLL have been done using Verilog-A, and the Transient simulation results are shown in Fig.20 and Fig.21.

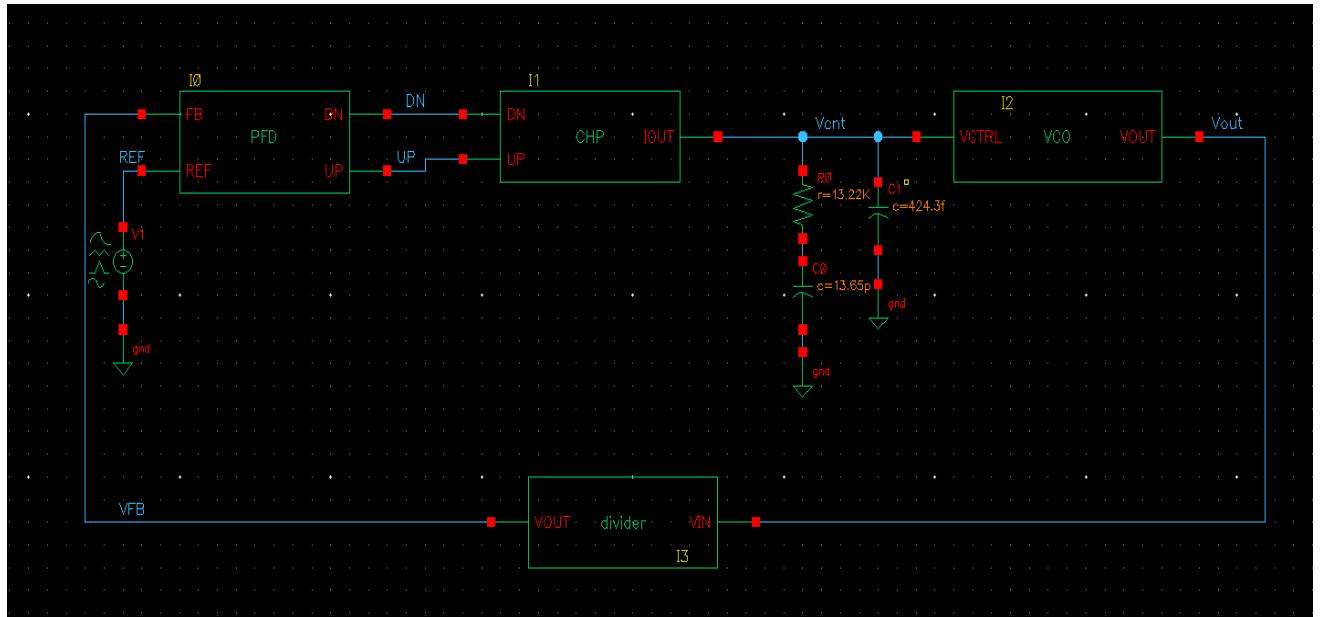


Figure 19: PLL Behavioral simulation using Verilog-A.

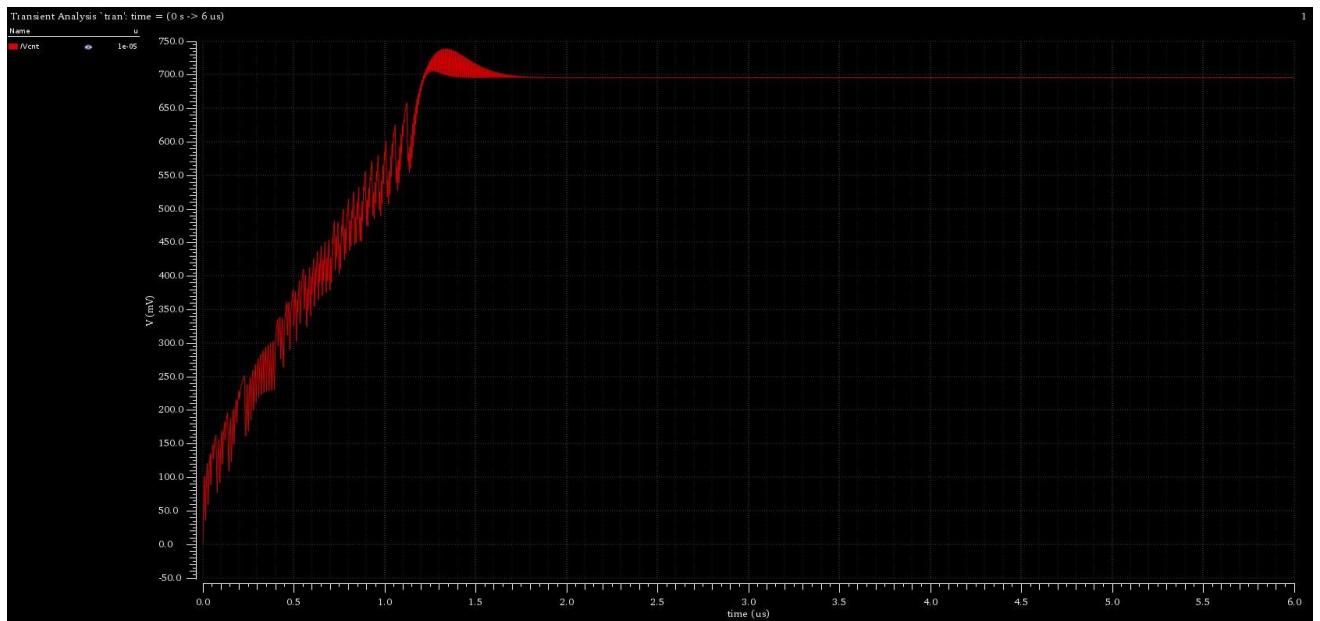


Figure 20: VCO control voltage (behavioral simulation).

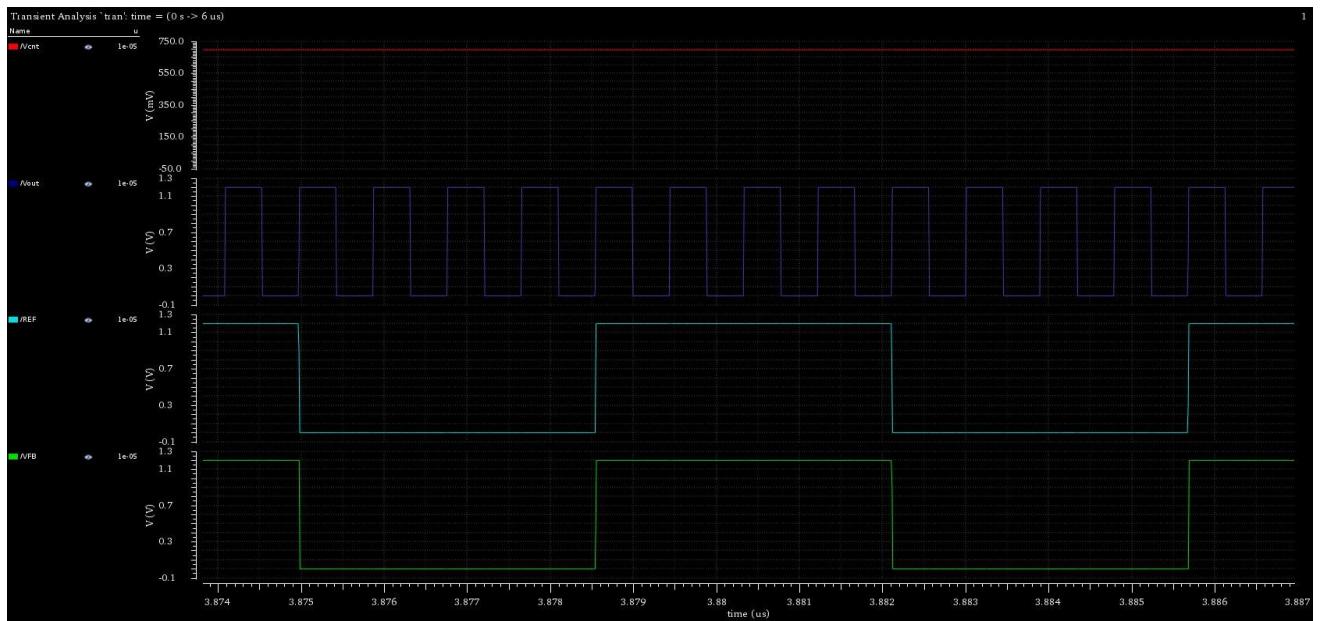


Figure 21: PLL in locked state (behavioral simulation). (1) VCO control (2) VCO output
 (3) Input reference (4) Divider output.

5. Transistor Level Design

The purpose of this section is to design each block in the system at the transistor level, and verify its operation using Bottom-up verification and incremental integration concepts.

5.1 FREQUENCY DIVIDER (FD) DESIGN

In order to design a frequency divider, there are more than one topology to choose from them including dual modulus Prescalers, asynchronous dividers, synchronous dividers, and injection-locked frequency dividers which is not a D-flip flop based. Every topology has its own advantages and disadvantages, but we need to implement a simple divide by 8 circuit, so an asynchronous divider is a sufficient one, but it suffers from an extra delay that may affect the loop stability (delays are the main source of control loop stability threats), and also suffers from jitter accumulation, so a synchronous divider is chosen as it samples the asynchronous divider output with the high frequency clock as shown in Fig.22 and this process is called “retiming” which eliminates the jitter and minimizes the delay.

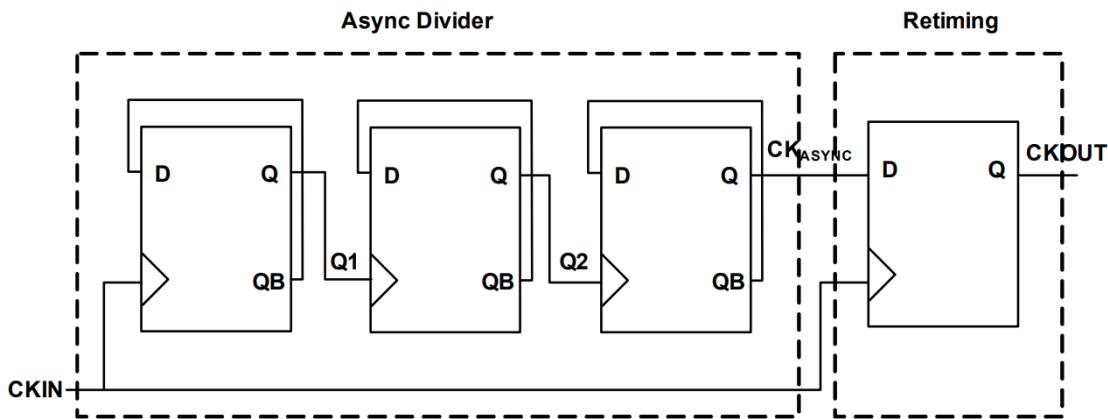


Figure 22: synchronous (divide by 8) frequency divider schematic.

The design of the used D flip-flop is shown in Fig.23 and Fig.24. finally, the designed DFF is used in the divider schematic as shown in Fig.25, and verified as shown in Fig.26 which reflects that the synchronous divider still suffers from a minimum delay.

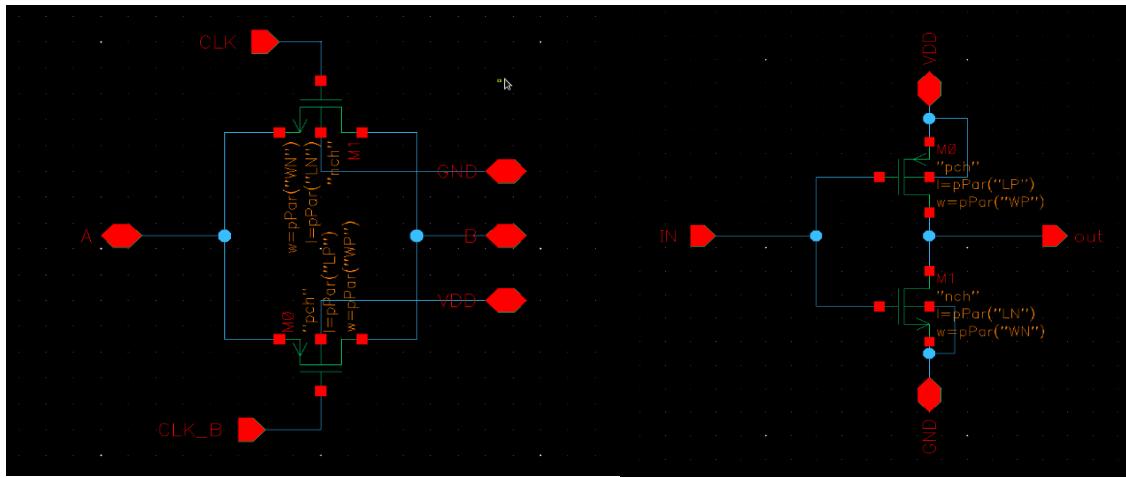


Figure 23: the basic gates used in the D flip-flop Design.

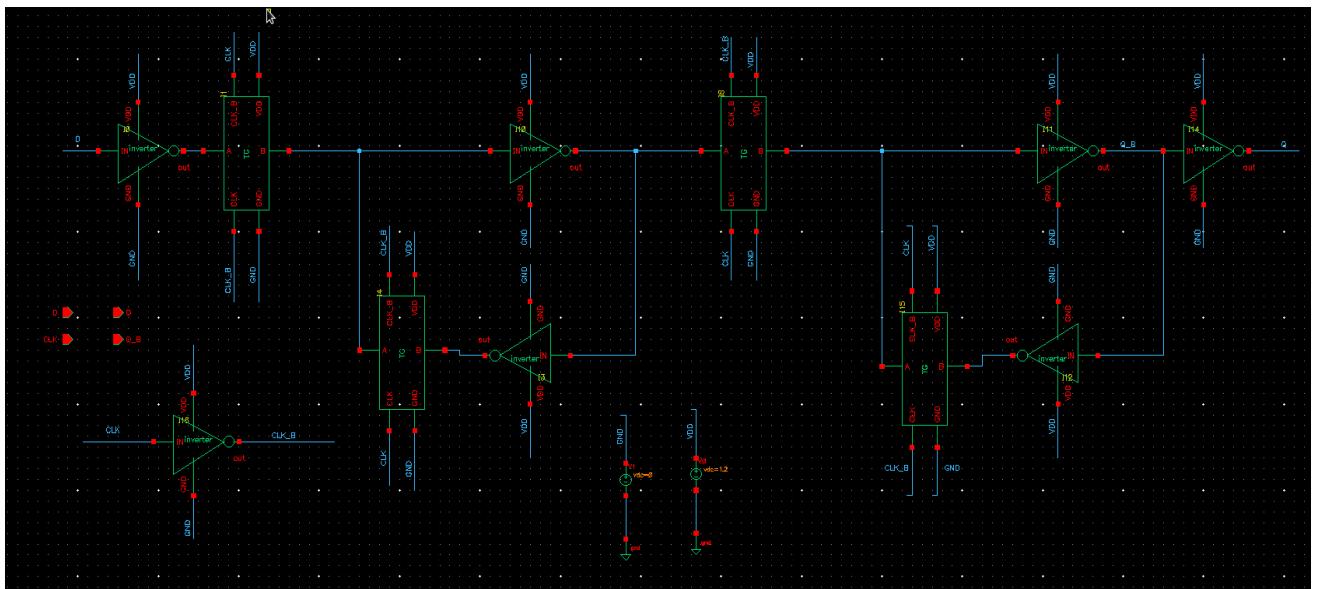


Figure 24: (Master-Slave) D flip-flop Design using back-to-back static latches.

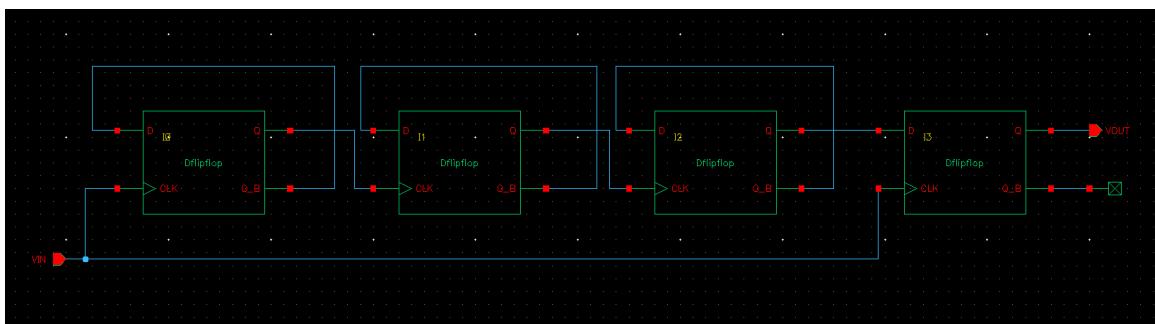


Figure 25: synchronous (divide by 8) frequency divider cadence virtuoso schematic.

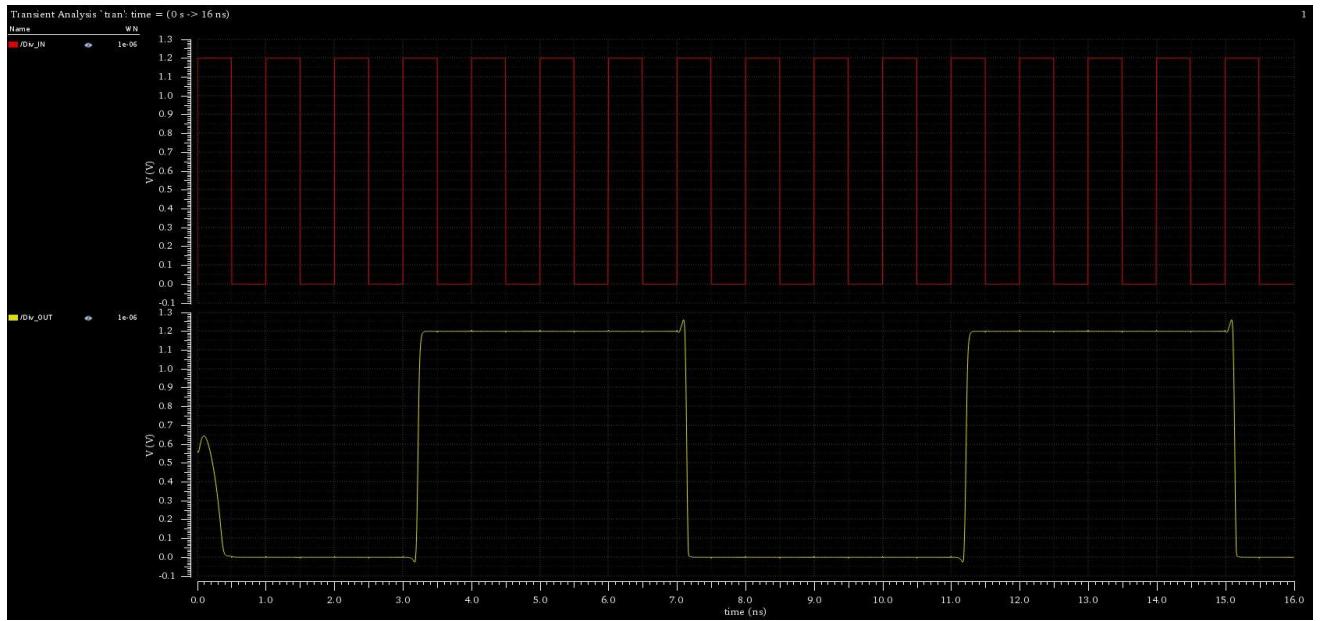


Figure 26: the designed frequency divider transient simulation results.

5.2 FREQUENCY DIVIDER BOTTOM-UP VERIFICATION

In this section, the frequency divider schematic is simulated while the rest of the system blocks are Verilog-A models, to verify the effect of imperfections in the block on the performance of the system, and this step will be repeated with each block in the system by making the designed block the only schematic block in the mixed-signal environment using the hierarchy editor in cadence virtuoso.

The system performance is still acceptable as the control voltages reaches a steady state which reflects that the PLL is locked and it's locked without phase error and the output frequency is 1.12 GHz as shown in Fig.28.

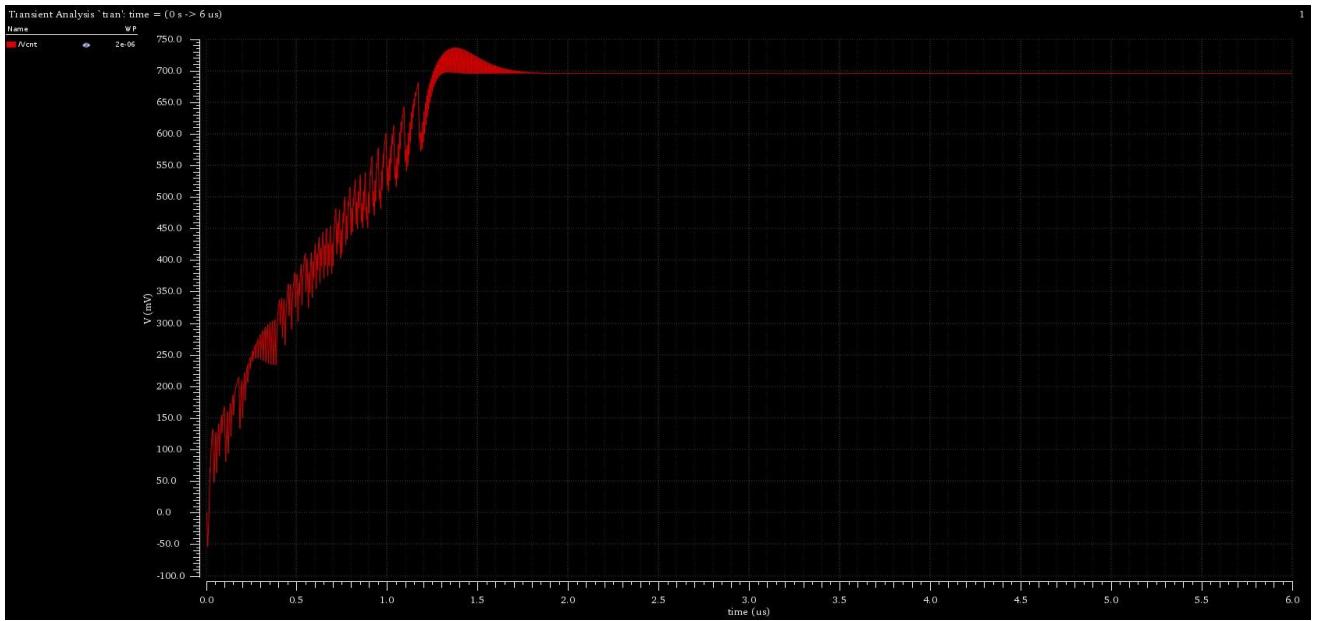


Figure 27: VCO control voltage (divider Bottom-up verification).

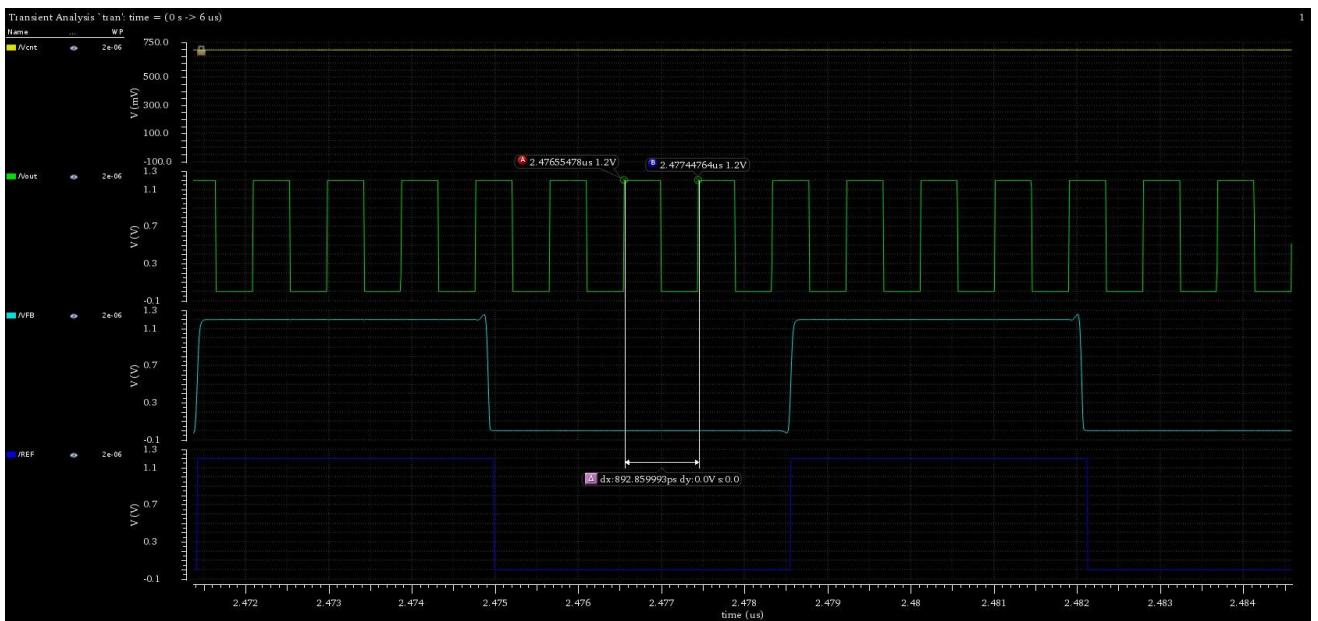


Figure 28: PLL in locked state (divider Bottom-up verification).

(1) VCO control (2) VCO output (3) Divider output (4) Input reference.

5.3 PHASE FREQUENCY DETECTOR (PFD) DESIGN

A typical digital tri-state phase frequency detector is designed, and this design requires a D flip-flop with an asynchronous reset as shown in Fig.31, the schematic of the used Nand gate is shown in Fig.30, finally these blocks are integrated in the PFD schematic as shown in Fig.29 and a simple test bench is performed as shown in Fig.32, Fig.33 and Fig.34.

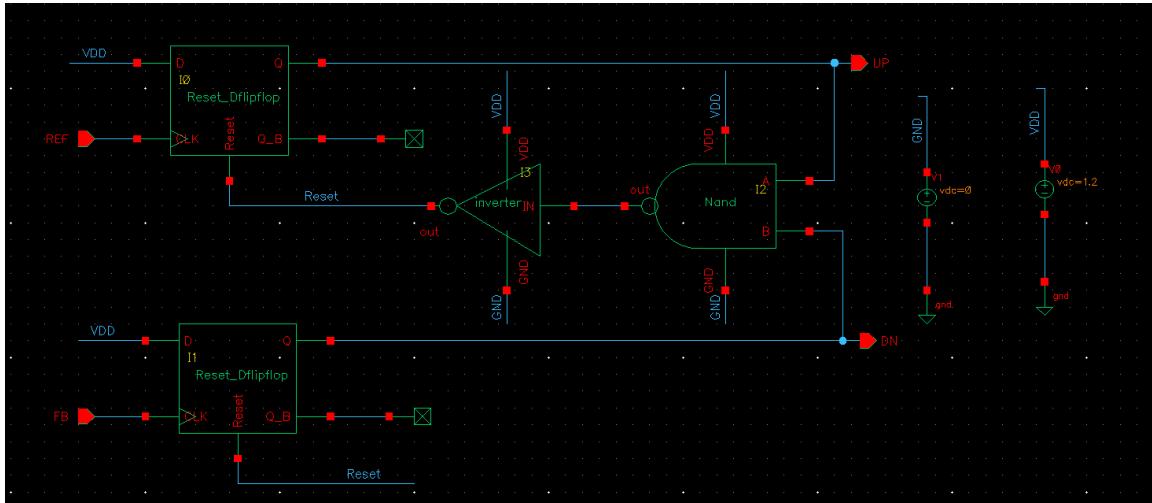


Figure 29: Typical digital Tri-state PFD schematic.

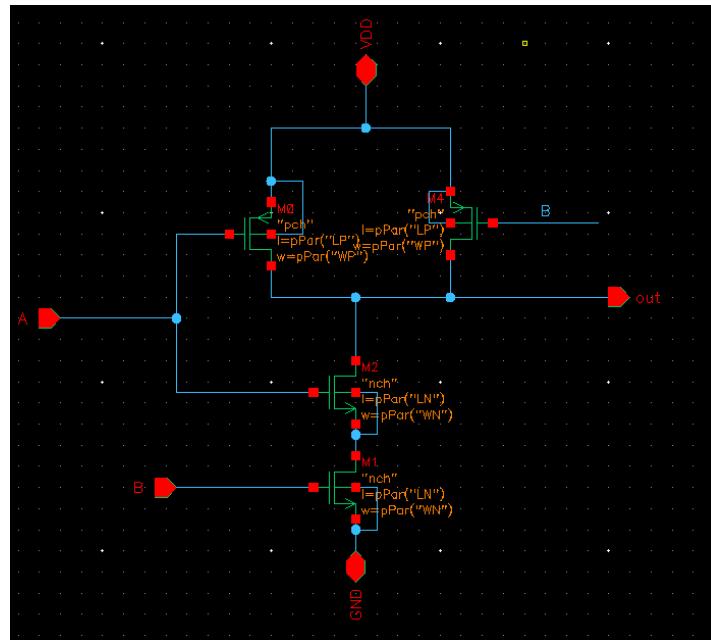


Figure 30: 2-input Nand gate schematic.

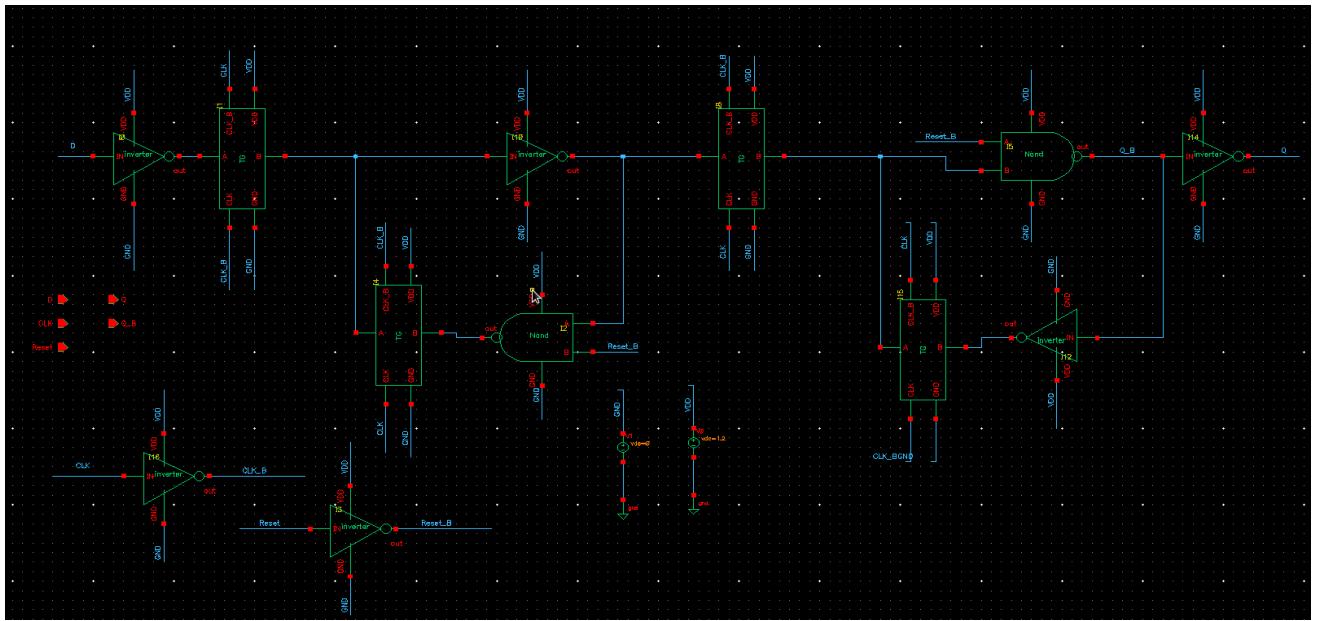


Figure 31: D flip-flop Design with extra asynchronous Reset input.

It is clearly obvious from the transient simulation results in Fig.32, that the Down- signal is high for a certain time (T_{ON}) which equals to the reset path delay.

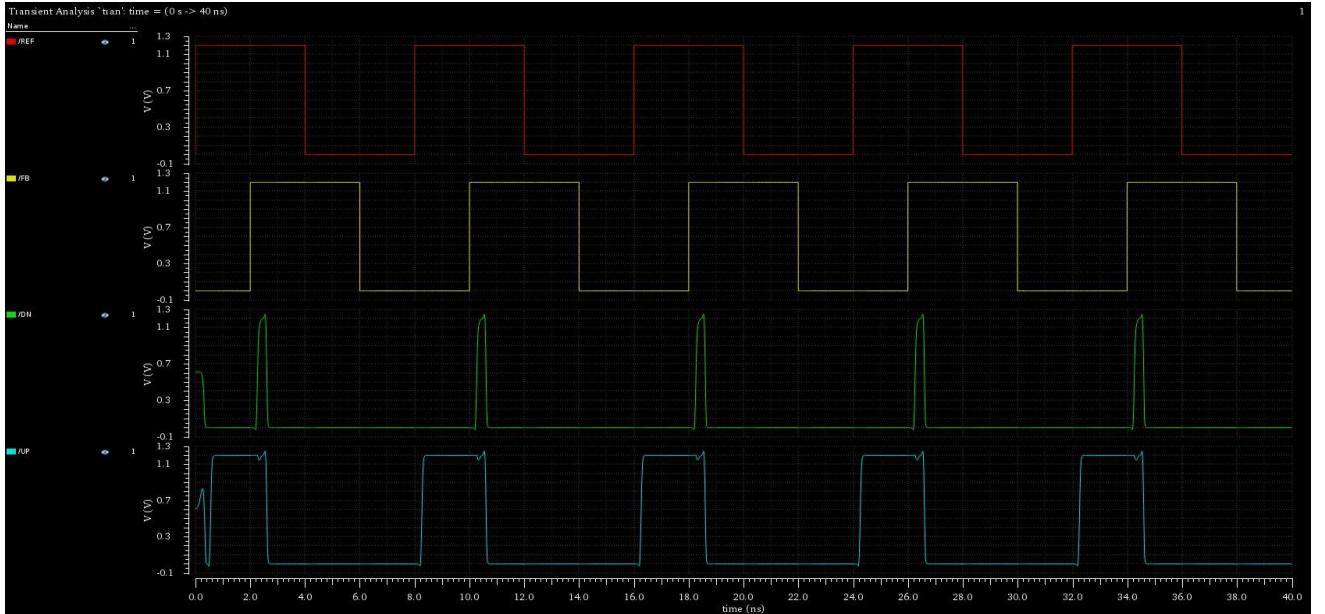


Figure 32: PFD timing when REF is leading (transistor-level simulation).

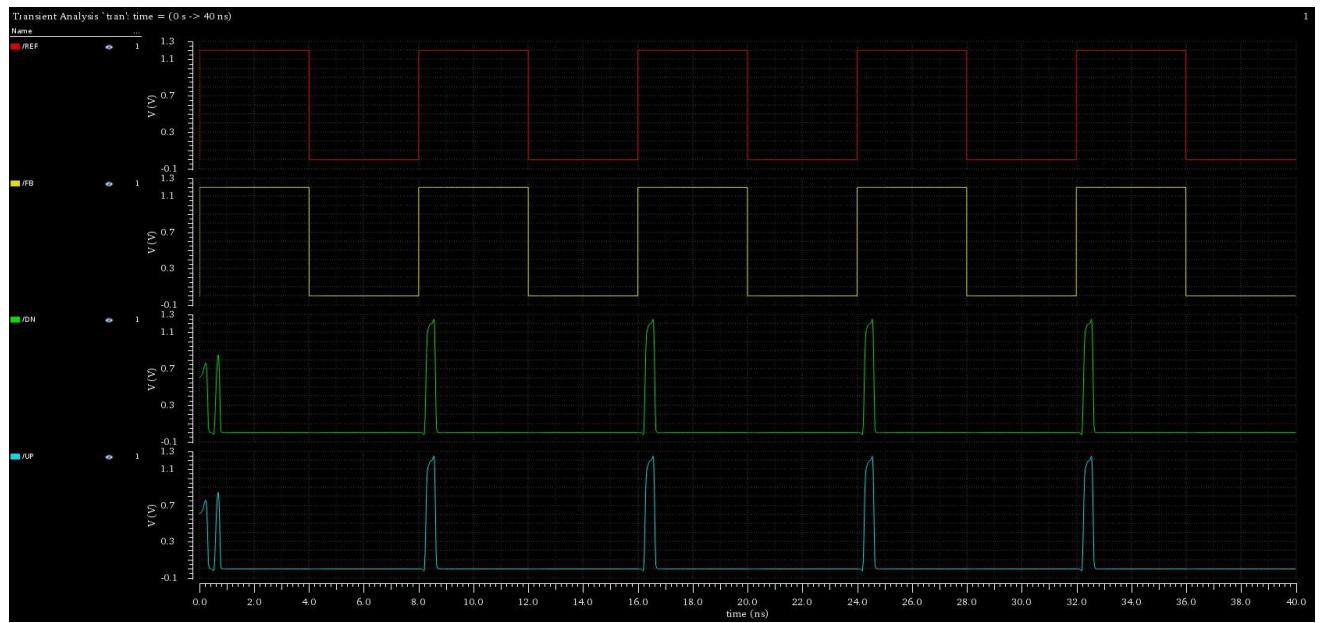


Figure 33: PFD timing when FB is matching REF (transistor-level simulation).

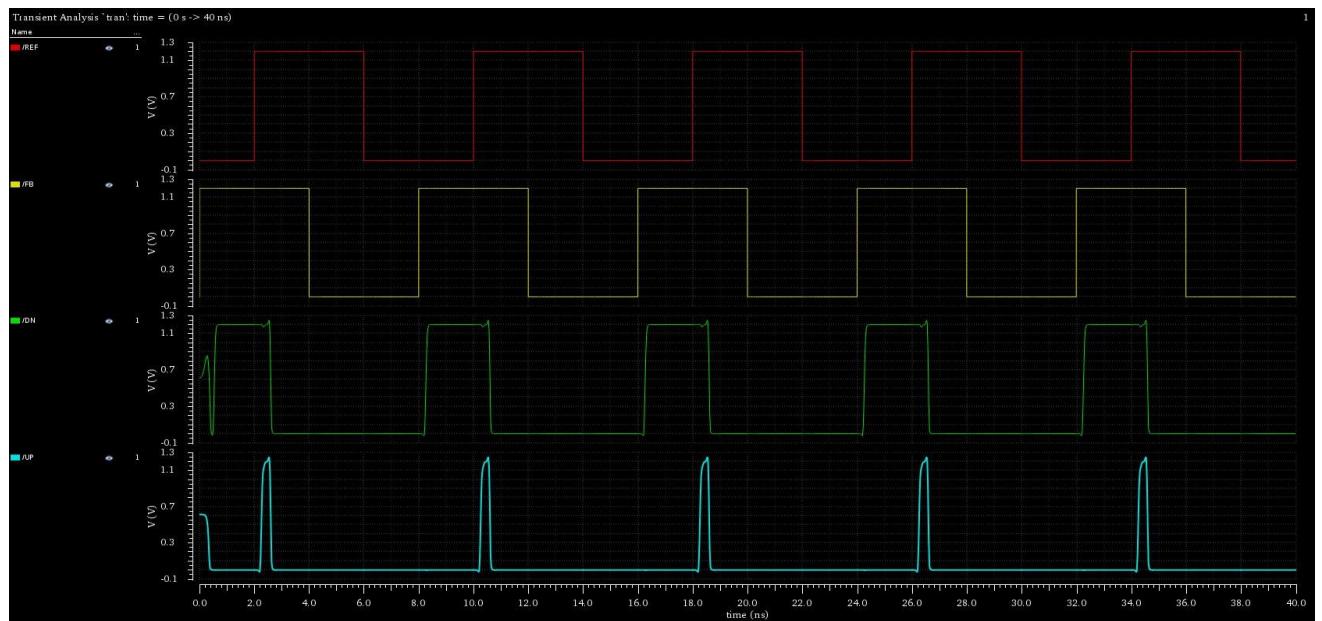


Figure 34: PFD timing when FB is leading (transistor-level simulation).

5.4 PHASE FREQUENCY DETECTOR (PFD) BOTTOM-UP VERIFICATION

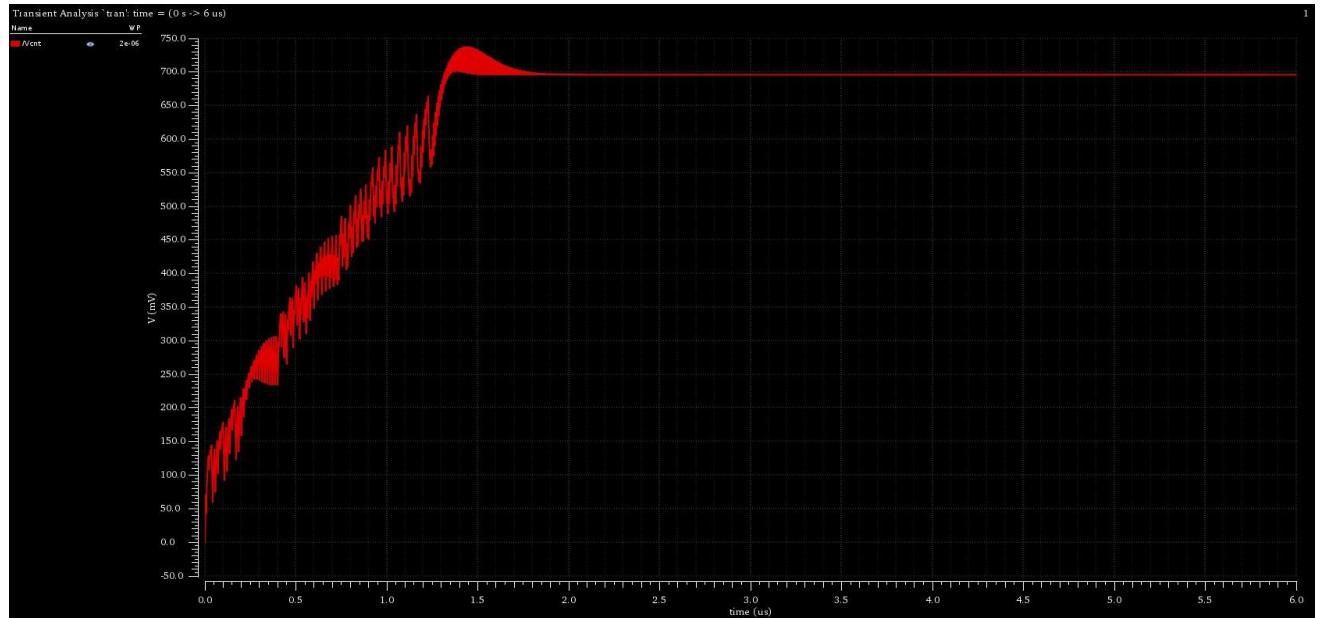


Figure 35: VCO control voltage (PFD Bottom-up verification).

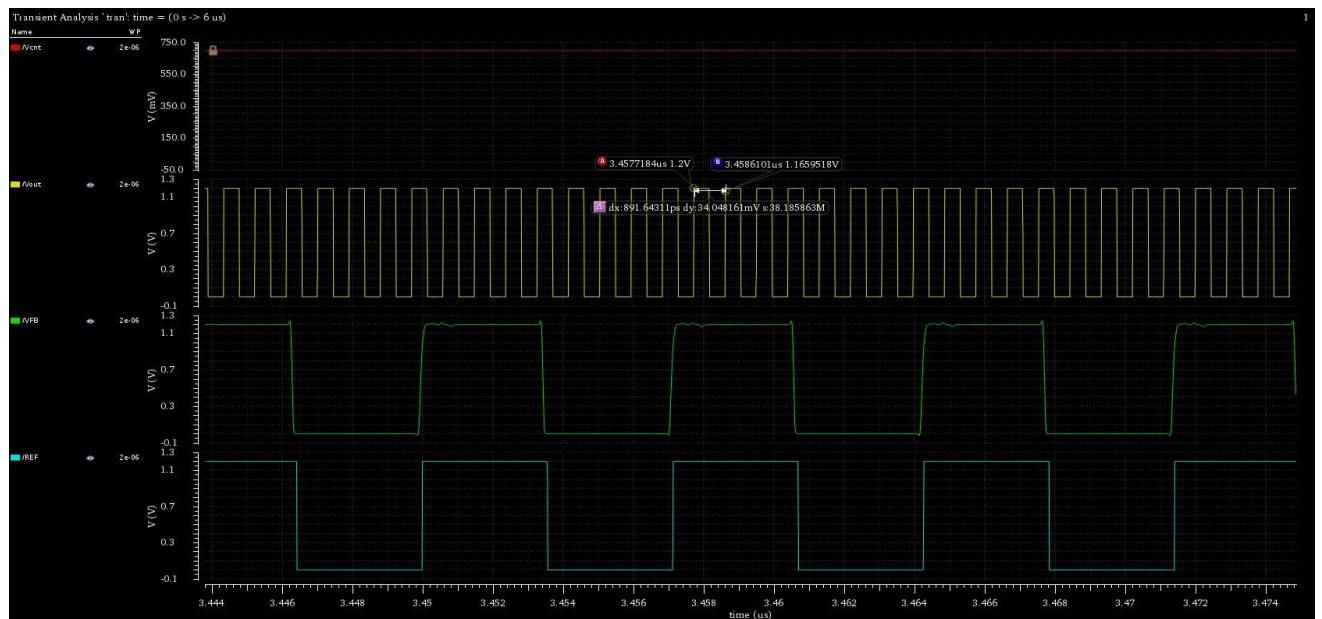


Figure 36: PLL in locked state (PFD Bottom-up verification).

(1) VCO control (2) VCO output (3) Divider output (4) Input reference.

5.5 INCREMENTAL INTEAGRATION STEP 1

After verifying the PFD as a stand-alone schematic as shown in Fig.35 and Fig.36, we moved up one step into the integration process, and simulated the PFD and the divider as schematic blocks as shown in Fig.37, and the system performance have been verified as shown in Fig.38 and Fig.39.

Library	Cell	View Found
AMS	TG	schematic
PLL_Design	CHP	verilog
PLL_Design	Nand	schematic
PLL_Design	PFD	schematic
PLL_Design	PLL	schematic
PLL_Design	Reset_Dflipflop	schematic
PLL_Design	VCO	verilog
PLL_Design	divider	schematic
PLL_Design	inverter	schematic
analogLib	cap	spectre
analogLib	nmos4	spectre
analogLib	pmos4	spectre
analogLib	res	spectre
analogLib	vdc	spectre
analogLib	vsource	spectre

Figure 37:incremental integration step 1 using hierarchy editor.

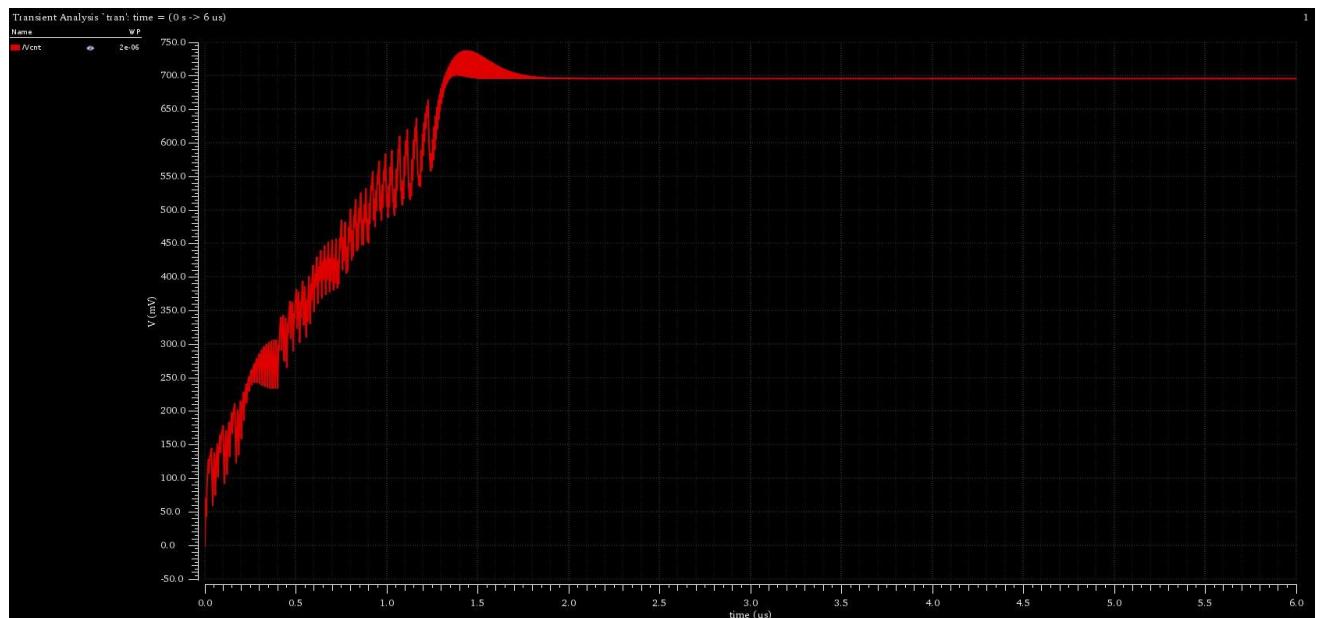


Figure 38: VCO control voltage (incremental integration step 1).

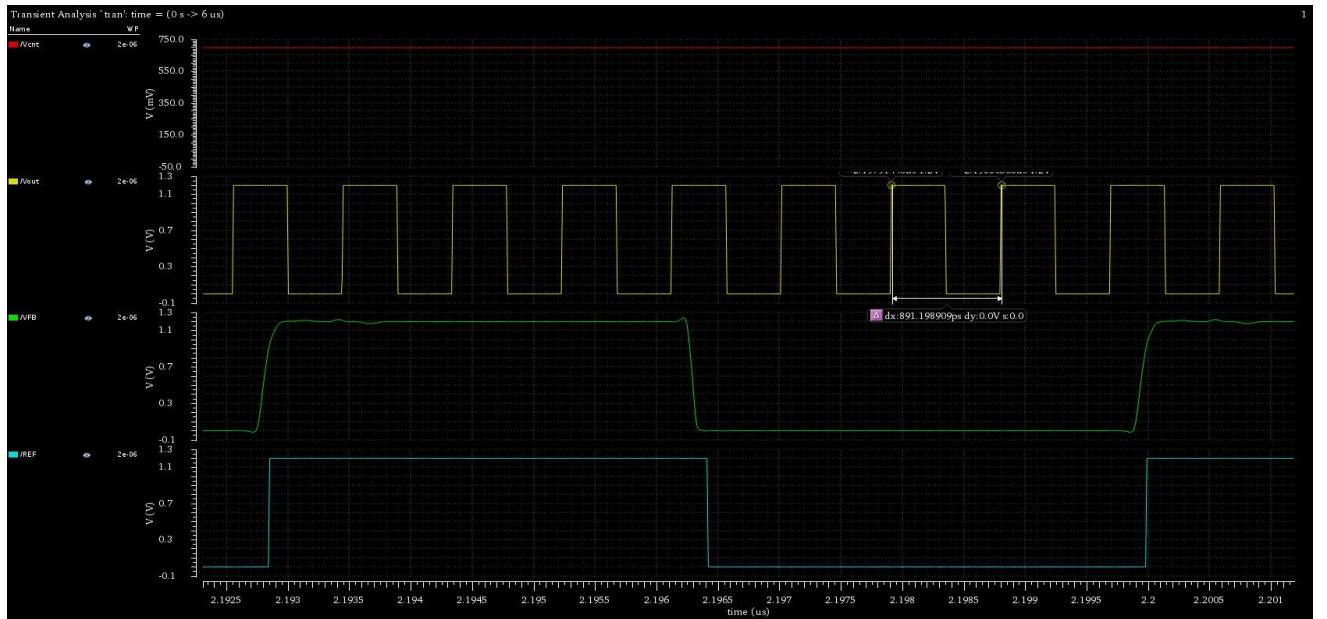


Figure 39: PLL in locked state (incremental integration step 1).

(1) VCO control (2) VCO output (3) Divider output (4) Input reference.

5.6 CHARGE PUMP (CHP) DESIGN

An NMOS-switch current-steering charge pump topology as shown in Fig.40 is designed, and it operates as described in the following cases:

Case 1: When the UP signal is high and DN signal is low: MN4 is turned ON, which turns on the PMOS current mirror (MP1-MP2). Hence, the charge pump current I_{UP} will flow in the MP2. Meanwhile, the MN7 is turned OFF since the DN signal is low. The current of the MN3 current sink is steered into MN6.

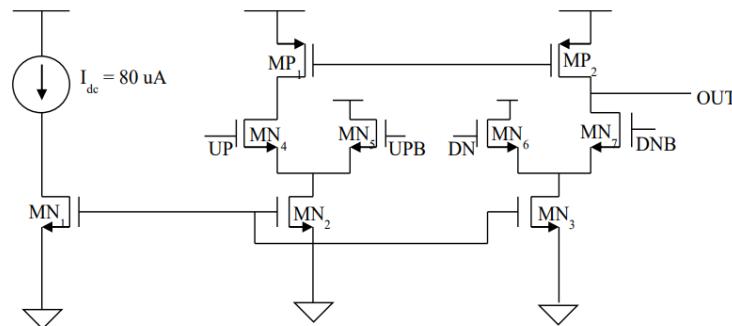


Figure 40: NMOS-switch current steering charge pump Topology, but I_{dc} is replaced by $10 \mu A$.

Case 2: When the UP signal is low and DN signal is high: similarly, the current I_{DN} is drawn through MN7.

Case 3: When the UP and DN signals are driven high simultaneously: the current I_{UP} becomes equal to I_{DN} , resulting in the absence of charging or discharging of the output, thus keeping the output V_{ctrl} signal constant.

Case 4: when both UP and DN signals go low: both MN4 and MN7 will be turned OFF. All current mirrored from I_{cp} will be steered into MN5 and MN6. Therefore, no current will be flowing in or out of the output node to change the output voltage.

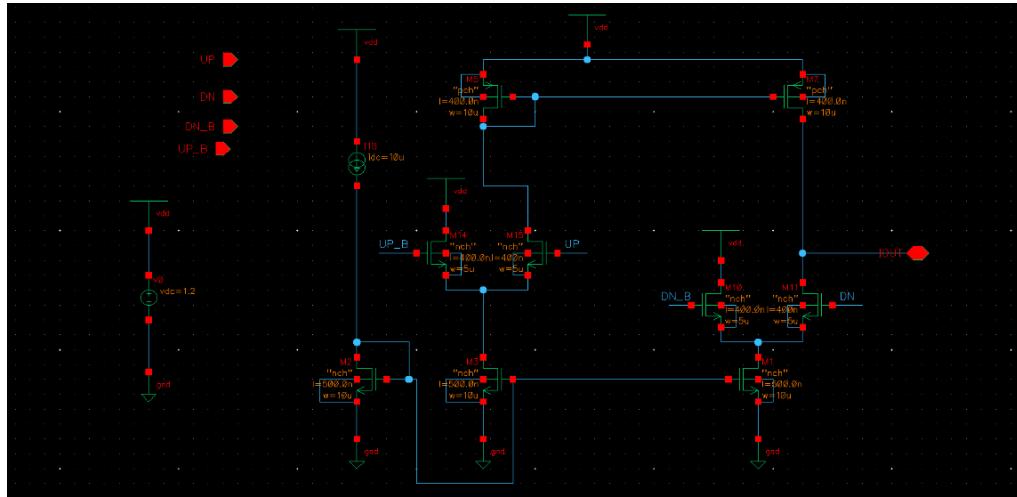


Figure 41: NMOS-switch current steering charge pump cadence virtuoso Schematic.

After designing the CHP as shown in Fig.41, we have verified the CHP as a stand-alone schematic as shown in Fig.42 and Fig.43, then we moved up one more step into the integration process, and simulated the PFD, Divider and the CHP as schematic blocks as shown in Fig.44, and the system performance have been verified as shown in Fig.45 and Fig.46.

5.7 CHARGE PUMP (CHP) BOTTOM-UP VERIFICATION

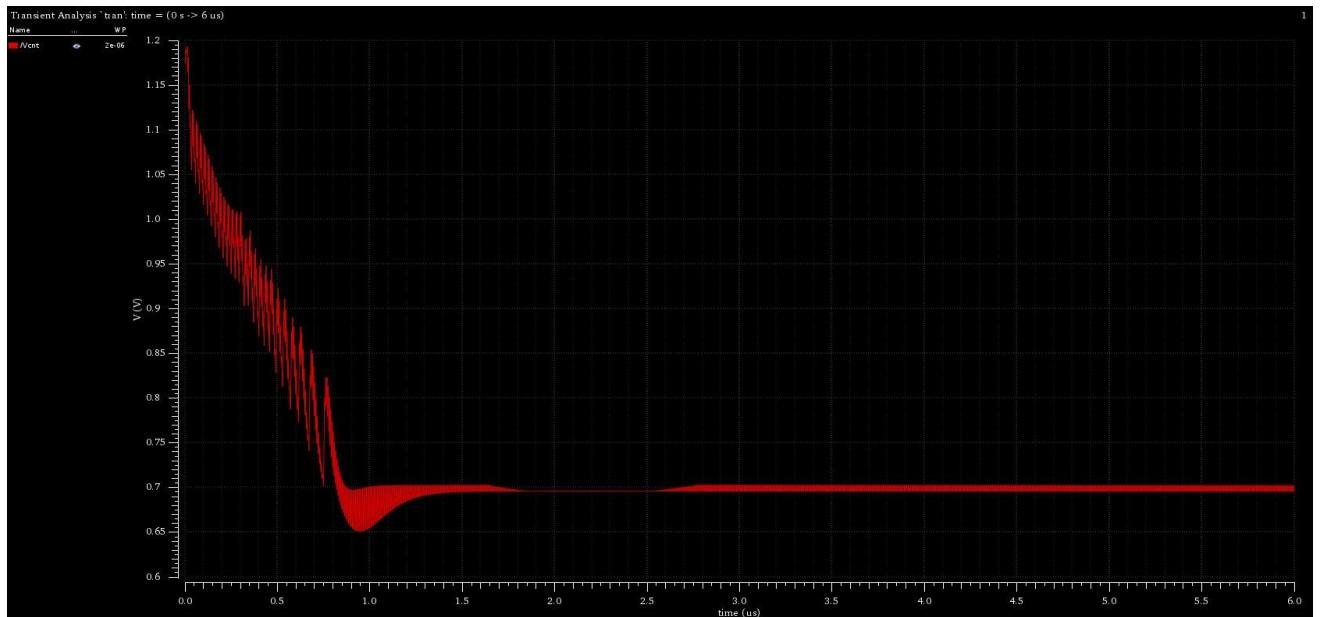


Figure 42: VCO control voltage (CHP Bottom-up verification).

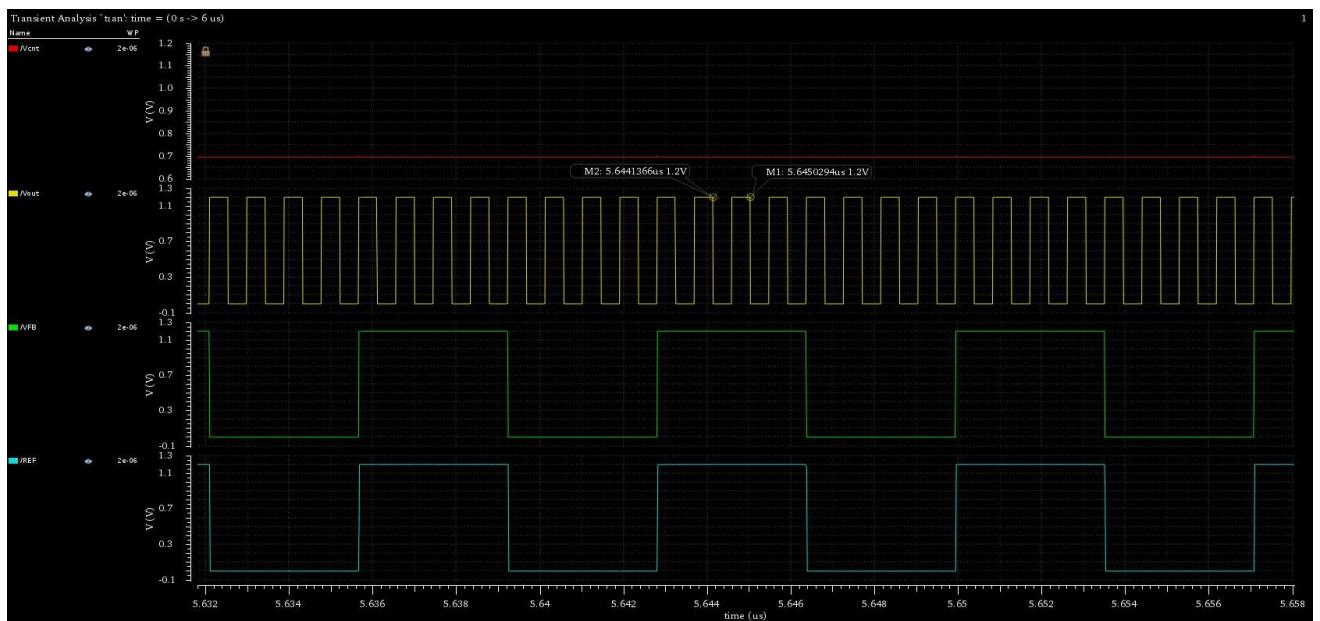


Figure 43: PLL in locked state (CHP Bottom-up verification).

- (1) VCO control (2) VCO output (3) Divider output (4) Input reference.

5.8 INCREMENTAL INTEAGRATION STEP 2

Cell Bindings				
Library	Cell	View Found	View To Use	Inherited View List
AMS	TG	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	CHP	schematic	schematic	spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	Dflipflop	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	Nand	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	PFD	schematic	schematic	spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	PLL	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	Reset_Dflipflop	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	VCO	veriloga		spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	divider	schematic	schematic	spectre cmos_sch cmos.sch schematic veriloga a...
PLL_Design	inverter	schematic		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	cap	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	idc	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	nmos4	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	pmos4	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	res	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	vdc	spectre		spectre cmos_sch cmos.sch schematic veriloga a...
analogLib	voltage	spectre		spectre cmos_sch cmos.sch schematic veriloga a...

Figure 44: incremental integration step2 using hierarchy editor.

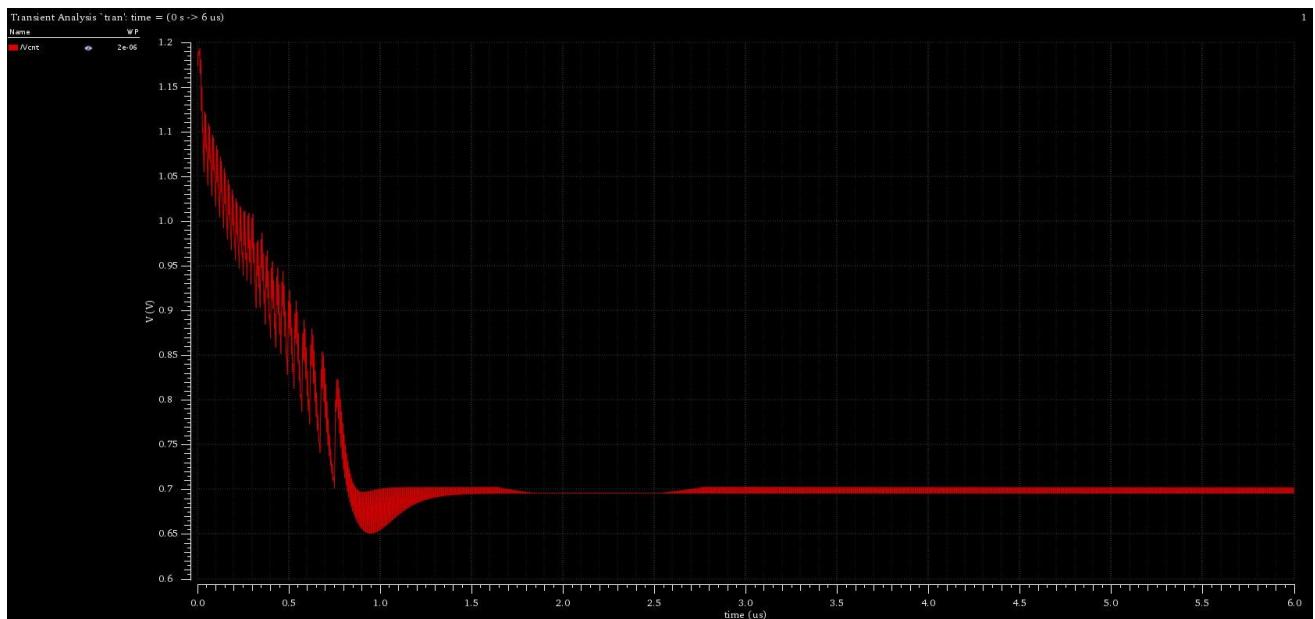


Figure 45: VCO control voltage (incremental integration step 2).

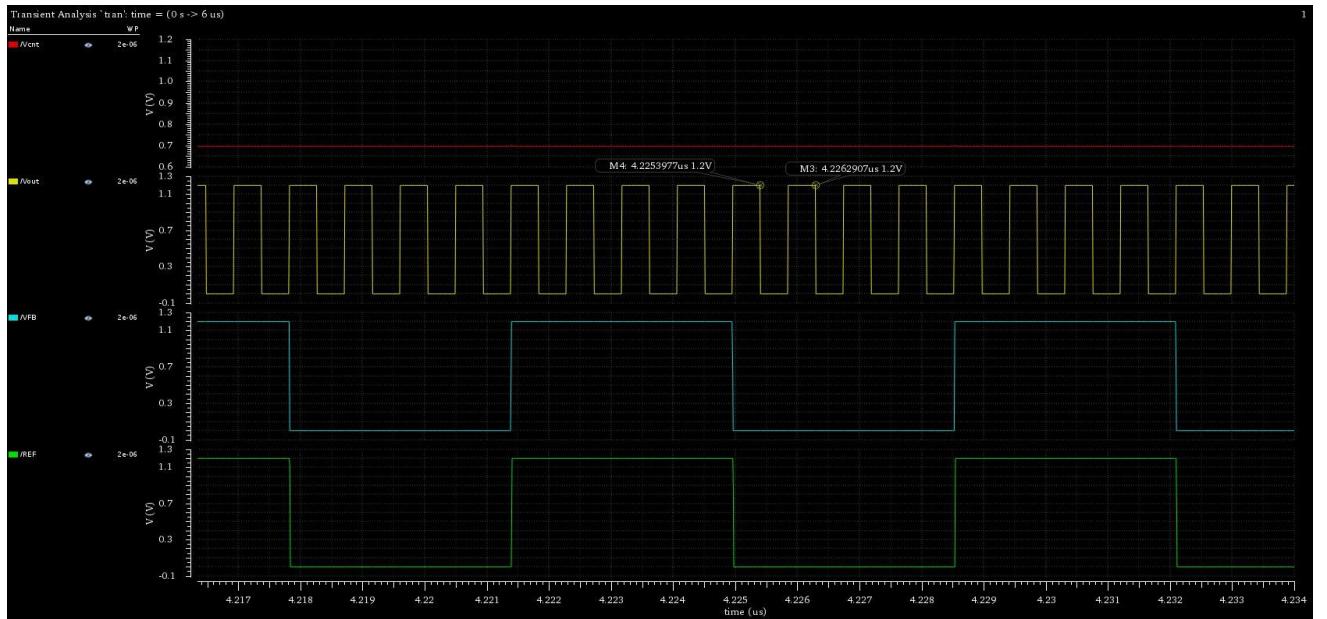


Figure 46: PLL in locked state (incremental integration step 2).

- (1) VCO control (2) VCO output (3) Divider output (4) Input reference.

6. Future Work

- 1.Finishing the transistor level implementation of the Voltage controlled Oscillator Block and Performing a simple testbench as in sec 4.3.
- 2.Verifying the VCO as a stand-alone schematic, to verify its impacts on the performance of the entire PLL system.
- 3.Full integration of the schematic blocks, and verifying the performance of the entire PLL system.

7. Conclusion

In this document, a Top-Down design methodology of a PLL system is proposed starting from system level design and system level simulation and modeling using Verilog-A, then verifying every block's transistor level implementation using bottom-up verification and incremental integration concepts.

8. References

- [1] Kundert, Ken. "Top-Down Design of Mixed-Signal Circuits." *Analog Circuit Design: High-Speed Analog-to-Digital Converters; Mixed Signal Design; PLL's and Synthesizers*. Boston, MA: Springer US, 2000. 185-202..
- [2] Omran, Hesham "Design of Low Power CMOS Direct Digital Frequency Synthesizer".
- [3] C. Zhang, T. Au and M. Syrzycki, "A high performance NMOS-switch high swing cascode charge pump for phase-locked loops," 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID, USA, 2012, pp. 554-557, doi: 10.1109/MWSCAS.2012.6292080.
- [4] Analog/Mixed Signal Simulation and Modeling Master Micro Course.