PFD-CP Phase Locked Loop Design

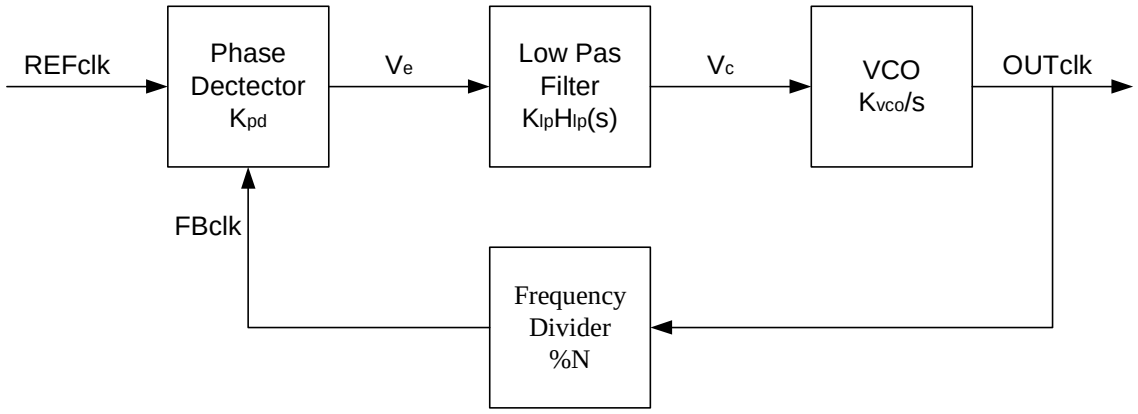
Fuding Ge

Contents

Chapter 1

Introduction

1.1 Closed loop transfer function between the output phase Φ_o and the input phase Φ_i



A general PLL model

Assume the transfer function of the loop filter is:

$$H_{lp}(s) = K_{lp}F_{lp}(s)$$

Where for passive filter K_{LP}=1, and for active filter, K_{LP} can be much larger than 1.

From the feedback control system theory, we know that the closed loop transfer function of the PLL can be generally written as:

$$H(s) = \frac{K_d H_{lp}(s) K_o}{s + K_d H_{lp}(s) K_o / N}$$

where N is the divide down ratio for the output of VCO to the feed back input of the phase detector.

For the system where phase detector is implemented with a PFD followed by a charge pump with a current of I_P, then followed with a passive filter, which is composed of a resister R in series with a capacitor C, we have:

$$K_d = \frac{I_P}{2\pi}$$

and:

$$K_d H_{lp}(s) = \frac{I_P}{2\pi}(R + \frac{1}{sC})$$

The open-loop forward transfer function is:

$$\left.\frac{\Phi_{out}}{\Phi_{in}}\right|_{openloop} = \frac{I_p}{2\pi}\left(R + \frac{1}{sC}\right)\frac{K_o}{s} = \frac{I_p(1+sRC)K_o}{2\pi s^2 C}$$

Since the open loop gain has two poles at the origin, this topology is called “type II” PLL. The system has a zero at

ω_z=1/(RC).

The close loop transfer function is:

$$H(s) = \left.\frac{\Phi_{out}}{\Phi_{in}}\right|_{closeloop} = \frac{\frac{I_P}{2\pi}(R + \frac{1}{sC})K_o}{s + \frac{I_P}{2\pi}(R + \frac{1}{sC})K_o / N}$$

After some mathematical manipulations, we have:

$$H(s) = \frac{\frac{I_P}{2\pi} R (s + \frac{1}{RC}) K_o}{s^2 + s\frac{I_P K_o R}{2\pi N} + \frac{I_P K_o}{2\pi NC}}$$

This equation is a typical second order system with:

$$\omega_n = \sqrt{\frac{I_P K_o}{2\pi NC}}$$

and:

$$\xi = \frac{\omega_n RC}{2} = \frac{R}{2}\sqrt{\frac{I_P K_o C}{2\pi N}}$$

It is interesting to find out that the natural frequency of the closed loop PLL I is independent of the value of R. It is also helpful to note the damping factor ζ is proportional to the value of R. It also should be noted that the both the natural frequency and the damping factor are dependent on the value of N.

Its decay time constant

$$\tau_d = \frac{1}{\zeta\omega_n} = \frac{4\pi N}{I_P K_o R}$$

It can be seen that the decay time constant is independent of capacitor C, only on the resistor R. It makes sense, since an ideal capacitor only charge and discharges but does not dissipate any power, therefore does not contribute to the decay time constant. If the damping factor is smaller than 1, $\zeta<1$, for a step change in the reference input frequency (or phase), the output will response with a sinusoidal component at a frequency of $\omega_n(1-\zeta^2)^{0.5}$ and approach its final value with the decay time constant $1/(\zeta\omega_n)$. Note that if $\zeta=0.707$ which is usually the case, $\omega_n(1-\zeta^2)^{0.5} =\zeta\omega_n$. It should be noted that the decay time constant is proportional to the frequency divide ratio N.
The closed loop transfer function between the output phase Φ_o and the input phase Φ_i can also be written as:

$$H(s) = \frac{N(2\xi\omega_n s + \omega_n{}^2)}{s^2 + 2\xi\omega_n s + \omega_n{}^2}$$

Define:

$$K = \frac{K_o I_P R}{2\pi N}$$
$$\tau_z = RC$$
$$\omega_z = 1/\tau_z$$

H(s) can also be written as:

$$H(s) = \frac{N(Ks + K\omega_z)}{s^2 + Ks + K\omega_z}$$

$$\zeta = \frac{1}{2}\sqrt{\frac{K}{\omega_z}}$$

Later we will find that K is approximately equal to the unit gain bandwidth (crossover frequency) ω_c of the PLL

1.2 Values for loop filter components R and C: an example.

Let's assume the frequency divider ration N is 30. If we use a charge pump current of I_p=10μA, then the gain of the PFD is:

K_d=10μA/(2π)=1.59μA/rad.

We assume the characteristic of VCO is F_{osc} = (-120×Vc + 342) MHz, so the gain of VCO is:

K_o = -120 MHz/V=754Mrad/V

The design requires that the instantaneous jumps of VCO control voltage (Vc) not exceeding 10mV. For a passive loop filter composed of a resistor R in series with a capacitor C, the instantaneous jumps of Vc is I_pR. Ip=10μA, then R ≤ 1KΩ (for Vc jumps not exceed 100mV R ≤ 10KΩ).

We already know the natural frequency ω_n=251.3 Krad/s. From $\omega_n = \sqrt{\frac{I_P K_o}{2\pi NC}}$ (note that in this expression the unit of Ko is rad/s and the unit of ω_n is rad/s too) we can find:

$$C = \frac{I_P K_o}{2\pi N \omega_n^{\ 2}}$$

and we get C = 633pF.

Normally the capacitor C of the loop filter occupies a large silicon area. For a design, the input reference clock signal is given and the natural frequency is set for stability reason. From the equation of C we can find that a small VCO gain K_o will help to reduce the area if the frequency range is large enough, also reduce K_O will improve the noise performance of the loop. Therefore only deign K_O for enough frequency tuning range, never too large. Also note that reduce the charge pump current I_P helps to reduce C too. But reduction of C will increase the noise of the loop filter which is proportional to $\sqrt{KT/C}$. So in the design there is a tradeoff between noise and area.

From $\xi = \frac{\omega_n RC}{2}$, we get: $R = \frac{2\zeta}{\omega_n C}$. ζ=0.707, then R=8.87KΩ which is smaller than 10KΩ, which means that the maximum control voltage Vc jump is about 88.7mV.

1.3. Closed loop transfer function between the error phase Φ_e and the input phase Φ_i

If we define the loop gain as following:

$$L_p(s) = \frac{K_d H_{lp}(s) K_o}{sN}$$

the phase error transfer function, in a general form, is:

$$H_e(s) = \frac{\Phi_e}{\Phi_i} = \frac{1}{1 + L_p(s)}$$

Which can be written as:

$$H_e(s) = \frac{s^2}{s^2 + s\frac{I_P K_o R}{2\pi N} + \frac{I_P K_o}{2\pi NC}}$$

It can be written in a general form as following:

$$H_e(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^{\ 2}}$$

Which shows a high-pass characteristic.

We can derive this equation in another way:

$$H(s) = \frac{\Phi_o}{\Phi_i}$$

$$H_e(s) = \frac{\Phi_e}{\Phi_i} = \frac{\Phi_i - \Phi_o / N}{\Phi_i} = 1 - \frac{H(s)}{N}$$

If N=1, we have:

$$H_e(s) = 1 - H(s)$$

1.4. Transfer function between ω_i and ω_o, ω_e and ω_i

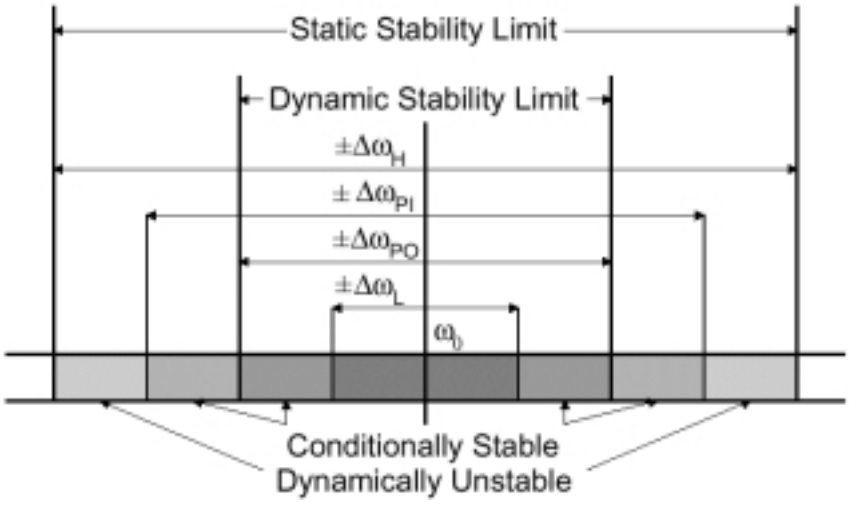
$$\frac{\omega_e}{\omega_i} = \frac{\Phi_e}{\Phi_i} = H_e(s)$$

$$\frac{\omega_O}{\omega_i} = \frac{\Phi_O}{\Phi_i} = H(s)$$

$$H(s) = \frac{\omega_o}{\omega_i}\bigg|_{closeloop} = \frac{\frac{I_P}{2\pi}(R + \frac{1}{sC})K_o}{s + \frac{I_P}{2\pi}(R + \frac{1}{sC})K_o / N}$$

$$H_e(s) = \frac{\omega_e}{\omega_i} = \frac{\omega_i - \omega_o / N}{\omega_i} = 1 - \frac{H(s)}{N}$$

1.5. PLL operation ranges



Hold-in range (Hold range, Lock range): It is the static stability limit. When the PLL is in the phase-locked state, the maximum frequency range in which the frequency of the input reference signal can slowly be pulled away from the free running frequency of the VCO but the PLL still maintain the phase-locked condition is called the hold-in range.
For a PLL using PFD and charge pump, the hold range is only limited by the VCO output frequency range.

Pull-in range and pull-in time: (Pull-in range is also referred as **capture range** in some literatures): It refers to the condition that initially the PLL is not in the phase-locked state, then the reference input signal frequency slowly approaches the free running frequency of the VCO, the maximum frequency range in which the input signal eventually becomes phase-locked is called the lock-in range or capture range.
For a PLL using PFD and charge pump, the hold range is also only limited by the VCO output frequency range.

Both he hold and capture ranges of PFD followed by charge pump type PLL are only limited by the VCO output frequency range. This is one important reason that this type PLL is so popular.

Let ω_{eo} to be the initial frequency error, the pull-in (capture) time is

$$T_P = \frac{\omega_{eo}/K_{Forward} - 2N\pi}{N\pi\omega_z}$$

Where:

$$K_{Forward} = \frac{I_P K_O}{2\pi} R = NK$$

So the pull-in time can be written as:

$$T_P = \frac{\omega_{eo}/K - 2\pi}{\pi\omega_z}$$

It can be seen that the pull-in time is proportional to the initial frequency error ω_{eo}.
We know

$$\omega_z = \frac{1}{RC}$$

So:

$$T_P = \frac{RC(\omega_{eo}/K - 2\pi)}{\pi} = C\left(\frac{2N\omega_{eo}}{I_P K_O} - 2R\right) = \left(\frac{\omega_{eo}}{\omega_n^{\,2}} - \frac{2}{\omega_z}\right)$$

It can be seen that as C increases, pull-in time T_P increases too. Note that the natural frequency $\omega_n \propto \frac{1}{\sqrt{C}}$, which means the larger the value of C, the smaller the bandwidth, and the longer the pull-in time. Also we can see that increase the value of the resistor R can reduce the pull-in time too.

Pull-out range: It is the dynamic stability limit of the PLL. It is the maximum value of a frequency step that applies to the input reference signal of a phase locked PLL and the PLL still maintains the lock state. If the frequency step exceeds the pull-out range, the PLL will not be able to track the input signal and fall out lock. The PLL may acquire lock again through a slow pull-in process.

We already know:

$$\frac{\Phi_e(s)}{\Phi_i(s)} = H_e(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2}$$

$$\Phi_i(s) = \frac{\Delta\omega_i(s)}{s}$$

So we have:

$$\frac{\Phi_e(s)}{\Delta\omega_i(s)} = \frac{1}{s} H_e(s) = \frac{s}{s^2 + 2\zeta\omega_n s + \omega_n{}^2}$$

For a step frequency change with a size of $\Delta\omega$, we have

$$\omega_i(s) = \frac{\Delta\omega_i(s)}{s}$$

Then:

$$\Phi_e(s) = \frac{1}{s} H_e(s) \cdot \Delta\omega_i(s) = \frac{\Delta\omega}{s^2 + 2\zeta\omega_n s + \omega_n{}^2}$$

For the PLL to keep in lock state, $\Phi_e < 2\pi$. Apply the inverse Laplace transform to the above equation, and let $\Phi_e(t) < 2\pi$, we can get the pull out range [BEST97]:

$$\Delta\omega_{Po} = 2\pi\omega_n \exp\left(\frac{\zeta}{\sqrt{1-\zeta^2}} \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}\right), \text{ when } \zeta < 1$$

$$\Delta\omega_{Po} = 2\pi\omega_n e, \text{ when } \zeta = 1$$

$$\Delta\omega_{Po} = 2\pi\omega_n \exp\left(\frac{\zeta}{\sqrt{\zeta^2-1}} \tan^{-1}\frac{\sqrt{\zeta^2-1}}{\zeta}\right), \text{ when } \zeta > 1$$

The least-squares fit gave the linear approximation:

$$\Delta\omega_{Po} = 11.55\omega_n(\zeta + 0.5)$$

Here we gave a few typical values:

$$\Delta\omega_{Po} = 4.38\pi\omega_n, \text{ when } \zeta = 0.707$$

$$\Delta\omega_{Po} = 5.44\pi\omega_n, \text{ when } \zeta = 1$$

Now let's have a look at the physical meaning of the pullout range.
For the case of ζ=1, when there is a $\Delta\omega$ step in the reference clock signal, we have:

$$\Phi_e(t) = \Delta\omega t \exp(-\zeta\omega_n t)$$

It has a peak value of $0.74\Delta\omega/K$. In any case if:

$$\frac{\Delta\omega}{K} \le \Phi_{em} = 2\pi$$

The PLL will keep in lock state [WOLA91].
Where

$$K = \frac{K_o I_P R}{2\pi N}$$

which in our configuration transformed into:

$$\Delta\omega \le \frac{K_o I_P R}{N}$$

We already know that the decay time constant of the PLL is

$$\tau_d = \frac{1}{\zeta\omega_n} = \frac{4\pi N}{I_P K_o R}$$

So:

$$\Delta\omega_{PO} \le 4\pi\zeta\omega_n = \frac{K_o I_P R}{N}$$

It is only depends on the value of R.

For a ramp frequency change, PLL keep in locking requires:

$$\Delta\dot{\omega} = \frac{d}{dt}(\Delta\omega_i) < \frac{K_o I_P}{C}$$

It can be seen that the capture frequency change ratio range is independent of the value of R.

Lock range and lock time: This range is a subset of pull-in range, It is the offset range between the reference and the scaled-down VCO frequency that the PLL will acquire lock within a single beat between the input reference signal and the feedback signal. The initial state of the PLL is unlock state.

The lock range, according to [BEST97] is:

$$\Delta\omega_L \le 4\pi\zeta\omega_n$$

It is only depends on the value of R, not dependent on C.
The lock time (settling time), according to [BEST97] is:

$$T_L \approx \frac{2\pi}{\omega_n}$$

We need a detailed inspection of the settling time. In fact PLL is a negative feedback system, theoretically, the output frequency can never be exactly equal to the input reference frequency, but always approaches it asymptotically. Thus **lock time** (**settling time, switching time**) is usually practically defined to be the time when the output approaches the input reference within a predefined margin. This margin is based on the overall system performance requirement. We know the system asymptotically converge to zero phase (frequency) error in the form of $\exp(-t\zeta\omega_n)$, let δ to be the frequency tolerance that we can say the PLL is locked, then

$$T_L = -\frac{\ln(\delta)}{\zeta\omega_n} = -\ln(\delta)\cdot\tau_d = -\ln(\delta)\cdot\frac{4\pi N}{I_P K_O R}$$

For δ=1/20000=0.005% and ζ=1, $T_L=10/\omega_n$.

The lock time (switching time) is a very important parameter, especially in wireless communication. There are several methods to improve the switch time [TI99]. One is to "pre-tune" the VCO to the desired frequency so that to bring the PLL to lock proximity, then the loop can start settle. Another is to increase the natural frequency ω_n for a short time by charging the largest capacitor in the loop filter directly or even increase the charge current in this time. One popular technique is to create a separate port that can charge the capacitor in the transition, another is using a switch that bypass the resistor R and allow charge the capacitor directly [TI99].

1.6. Stead state phase error

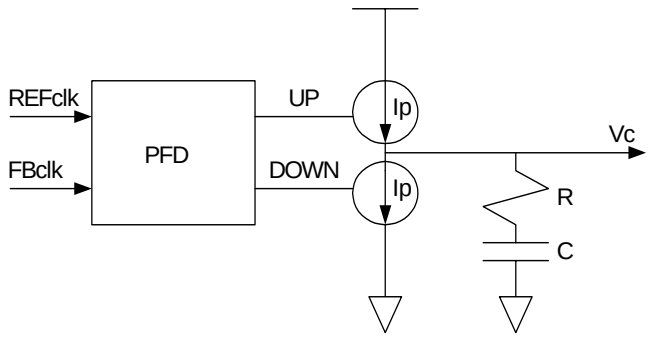
Stead state phase error (static phase error, or loop stress) is the average value of phase error when the PLL is in lock. Let Δw to be the frequency offset between the input signal and the free-running frequency of the VCO. We have [1]:

$$\Phi_{eo} = \frac{2\pi\Delta\omega}{K_o I_p Z_F(0)}$$

It is usually desirable to have static phase error, Φ_{eo}, near zero. For the PLL configuration we will use in the projects, i.e., PFD followed by charge pump, followed by passive filter and VCO, $Z_F(0)=\infty$, so that the static phase error is zero.
Therefore we can say that from the point view of static phase error, this configuration has the best tracking performance. This is another reason that this type of PLL is popular in practical application.

1.7 PLL bandwidth and stability analysis

We study the PLL with the following loop filter structure:



We already know that the open-loop gain is:

$$H(s)|_{openloop} = \frac{I_p}{2\pi}\left(R + \frac{1}{sC}\right)\frac{K_o}{Ns} = \frac{I_p(1+sRC)K_o}{2\pi s^2 NC}$$

The unit gain bandwidth (crossover frequency) is the value of the frequency when the magnitude of the open loop gain is 1.

$$\left|\frac{I_p(1+sRC)K_o}{2\pi s^2 NC}\right| = 1$$

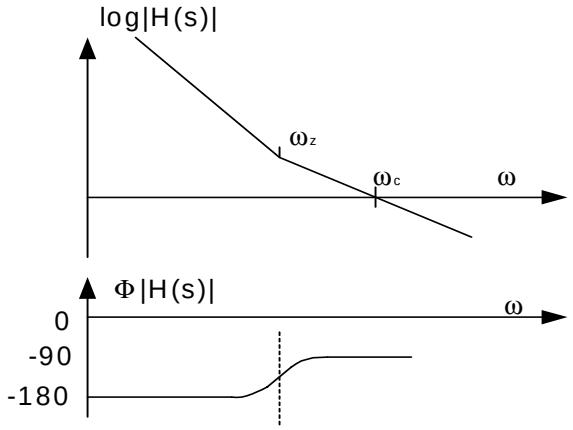
Assume s*RC* >> 1 we have the unit gain bandwidth ω_c as:

$$\omega_c = \frac{I_p}{2\pi}\cdot\frac{K_O}{N}R$$

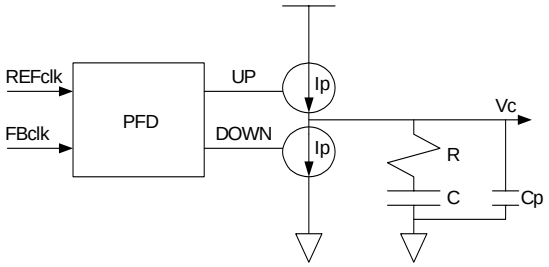
Note that:

$$\omega_c = 2\zeta\omega_n$$

Its Bode plot is shown in the following figure. It can be see it is unconditionally stable as long as ω_z is smaller than ω_c.



For a PLL has the following loop filter structure:



Its open loop gain is:

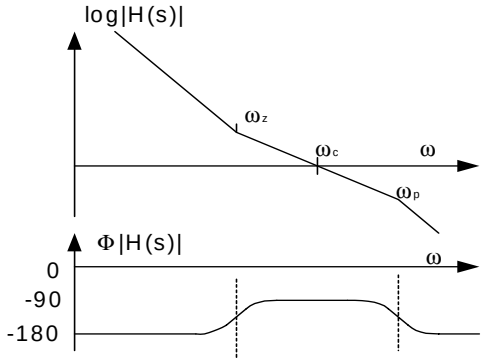
$$H(s)|_{openloop} = \frac{I_p}{2\pi}\frac{\left(R+\frac{1}{sC}\right)\frac{1}{sC_P}}{R+\frac{1}{sC}+\frac{1}{sC_P}}\frac{K_O}{Ns}$$

$$= \frac{I_p K_O}{2\pi s N}\cdot\frac{\left(R+\frac{1}{sC}\right)}{1+\frac{C_P}{C}+sRC_P} = \frac{I_p K_O}{2\pi s N}\cdot\frac{(1+sRC)}{sC+sC_P+s^2RC_PC}$$

$$= \frac{I_p K_O}{2\pi s^2 N}\cdot\frac{(1+sRC)}{(C+C_P)(1+sR\frac{C_PC}{C_P+C})} = \frac{I_p K_O}{2\pi s^2 N}\cdot\frac{(1+sRC)}{(C+C_P)(1+sRC_P \leftrightarrow C)}$$

Where $C_P \leftrightarrow C$ means C_P and C are in series. The value of $C_P \leftrightarrow C$ is determined by the smaller one.

Its Bode plot is shown in the following figure:



Its unit gain bandwidth is still about:

$$\omega_c = \frac{I_p}{2\pi}\cdot\frac{K_O}{N}R$$

It has a zero, which is:

$$\omega_c = \frac{1}{RC}$$

and a pole which is at:

$$\omega_P = \frac{1}{RC \leftrightarrow C_P}$$

It can be seen that if C_P is much smaller that C, then the pole is almost totally determined by C_P. Also if C_P is much smaller than C, the phase margin is quite large. In practical design, C_P is usually chosen to be about $C/10$. Generally speaking, we can place the loop gain zero ω_z a factor α below the crossover frequency ω_c and the 3rd pole ω_p a factor β above ω_c. To guarantee enough phase margin for the PLL to be sable, α and β are typically at least to 4 or large, which give a phase margin of 60° or larger.

In the design of a PLL, we may also need to consider the –3dB bandwidth ω_B. For the case of damping factor smaller than 1, we have:

$$\omega_B = \omega_n \sqrt{2\zeta^2 + 1 + \sqrt{2\zeta^2 + 1)^2 + 1}}$$

When $\zeta = 0.707$, $\omega_B = 2.06\omega_n$.

Example: We have known the lock-in time is 0.2ms, and is given $Ts = 16\pi/\omega_n$, then $\omega_n = 251.3$Krad/s.
We chose damping factor ζ to be 0.707, which will give the best settling time.

$\omega_B = 517.7$Krad/s and $f_B = 80.0$KHz.

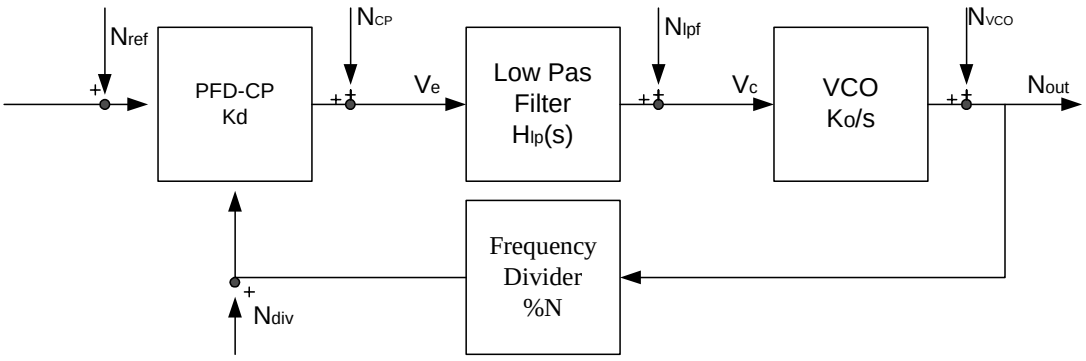
The input reference is $f_{in} = 8$MHz, $f_B/f_{in} = 1.0\%$, a quite narrow bandwidth PLL.
The loop gain Lp is defined as

$$L_P = K_d K_F K_O / N$$

$$L_P = \frac{I_P}{2\pi}(R + \frac{1}{sC})\frac{K_O}{sN}$$

1.8 Noise transfer functions

Every building block of the PLL will contribute to the total output noise. The noise sources of the PLL are shown in the following figure.



N_{ref}: noise from the reference signal;
N_{CP}: noise from the charge pump;
N_{lpf}: noise from the loop filter;
N_{VCO}: noise from VCO;

N_{div}: noise from the frequency divider.

Assume these noise are not correlated, the overall phase noise performance at the output of the PLL depends on the terms described above.

$$N_{total}{}^2 = N_{ref,out}{}^2 + N_{CP,out}{}^2 + N_{lpf,out}{}^2 + N_{VCO,out}{}^2 + N_{div,out}{}^2$$

Where $N_{xxx,out}{}^2$ is the noise power at the output due to noise source Nxxx.

The transfer function of N_{ref}, N_{div} to the output is same as the closed-loop transfer function of the PLL, which is:

$$H(s) = \frac{N_{out}}{N_{ref}} = \frac{N(2\xi\omega_n s + \omega_n{}^2)}{s^2 + 2\xi\omega_n s + \omega_n{}^2}$$

and has a low pass characteristic.

The transfer function of the noise from charge pump N_{CP} to the output has the same form as N_{ref} and N_{div} if we reference NCP to the input of PFD by

$$N_{cp,in} = N_{cp} / K_d$$

Where K_d is the gain of the PFD, and can be written as

$$K_d = \frac{I_P}{2\pi}$$

So it shows low pass characteristics too.

In order to reduce the output noise due to input noise sources from frequency divider, the reference signal and PFD, the bandwidth of the PLL should be as small as possible. Note that these noises are amplified by a factor of the divider ratio N, so if these noises are concerns, N should be small.

Now let's have a look at the transfer function of the noise due to the loop filter. Assume the noise can be written as a voltage source V_{nlpf} , we have:

$$H(s) = \frac{\Phi_{out}}{V_{nlpf}} = \frac{K_o / s}{1 + \frac{I_P}{2\pi}(R + \frac{1}{sC})\frac{K_o}{sN}} = \frac{K_o}{s + \frac{I_P}{2\pi}(R + \frac{1}{sC})\frac{K_o}{N}} = \frac{sK_o}{s^2 + \frac{I_P}{2\pi}(sR + \frac{1}{C})\frac{K_o}{N}}$$

This is of a bandpass filter characteristic. We can simply verify by let ω=0 and ω=infinity, at both cases, the output is zero. Otherwise the output is large than zero.

The transfer function from VCO noise to output can be written as:

$$H(s) = \frac{\Phi_{out}}{V_{nVCO}} = \frac{1}{1 + \frac{I_P}{2\pi}(R + \frac{1}{sC})\frac{K_o}{sN}} = \frac{s^2}{s^2 + \frac{I_P}{2\pi}(sR + \frac{1}{C})\frac{K_o}{N}}$$

which has a high-pass characteristic.

From above results, we can see that if the noise of VCO is the dominant noise source, a large PLL bandwidth should be used, which is in conflict with the requirement of the reference noise. So the bandwidth of the PLL is a tradeoff between noise from reference and noise from VCO, and the speed requirement.

Chapter 2. PLL function blocks design

2.1 Project specifications

The purpose of this project is to design a phase-locked loop clock synthesizer that will be able to generate a 200MHz clock from an reference clock of 8MHz.

The settling time or lock-in time Ts for the PLL is less than 0.15ms. Also, the settling time for the PLL is given by $Ts=16\pi/\omega_n$.

The maximum value of the instantaneous jumps of the VCO control voltage should be less 100mV.

The charge pump current is Ip=10uA.

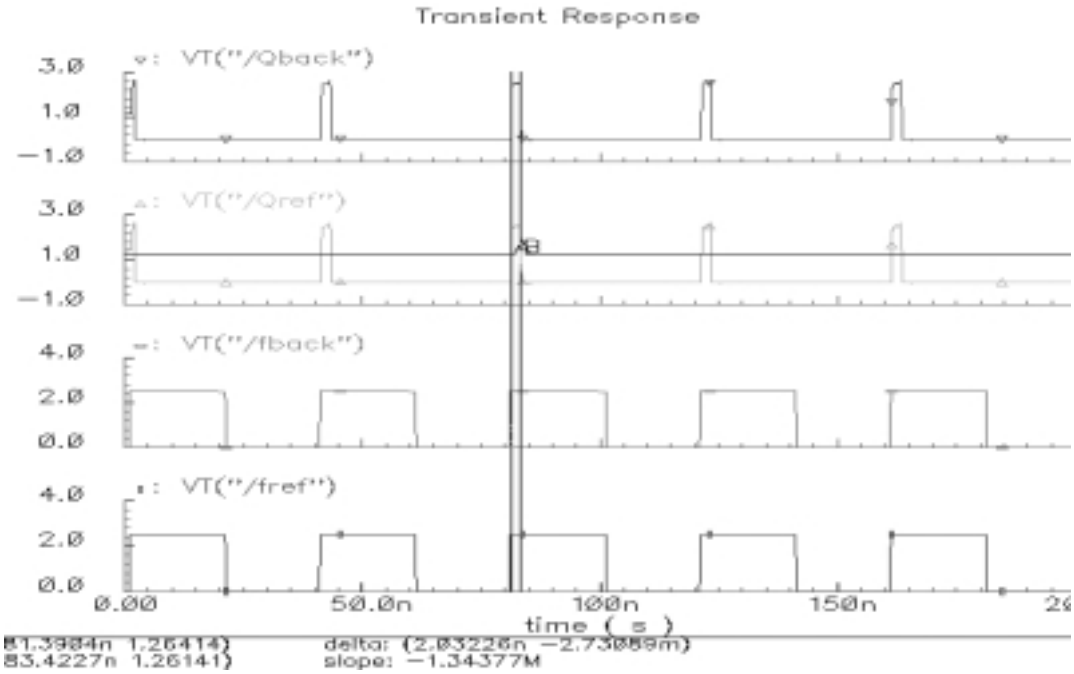
This PLL clock synthesizer includes a divide-by-N circuit, which will determine the multiplicity of the oscillator clock.

In project#1 we will design the phase-frequency detector (PFD) accompanied by the charge-pump, and the passive loop filter.

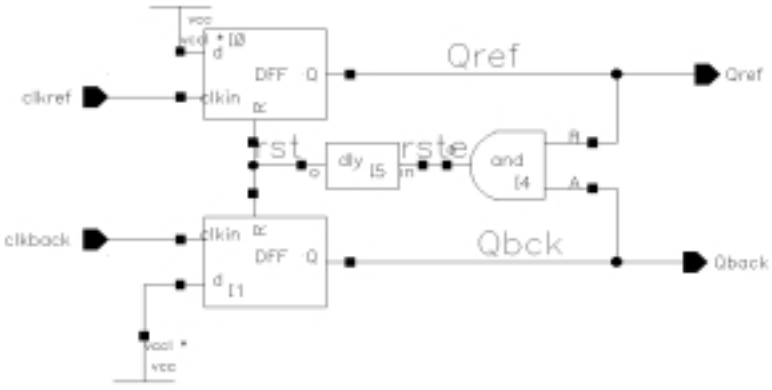
1. Derive the values for the natural frequency ω_n, the damping factor ζ, the loop bandwidth ω_B and the loop gain Lp.
2. Design the PFD so that during the lock condition, the narrow pulse at the rising edges of the input and output clocks to be around 2ns at room temperature (27°C) and typical processing conditions.
3. Design the charge pump as explained in class to charge and discharge Ip as calculated in part 5.
4. Design the voltage-to-current and the VCO to generate 200mHz at a value of the control voltage Vc close to Vdd/2.
5. Design a programmable divider, which accepts a register value in order to change the divide down ration N.
6. From your simulation plot the VCO generated frequency Fosc versus the control voltage Vc. Determine the VCO gain at 200MHz.
7. Build the schematic and run simulation for the whole PLL and plot the VCO control voltage for N=25, N=15 and N=35. In this part you need to observe the PLL lock condition by observing and plotting the VCO control voltage.

2.2 PFD design

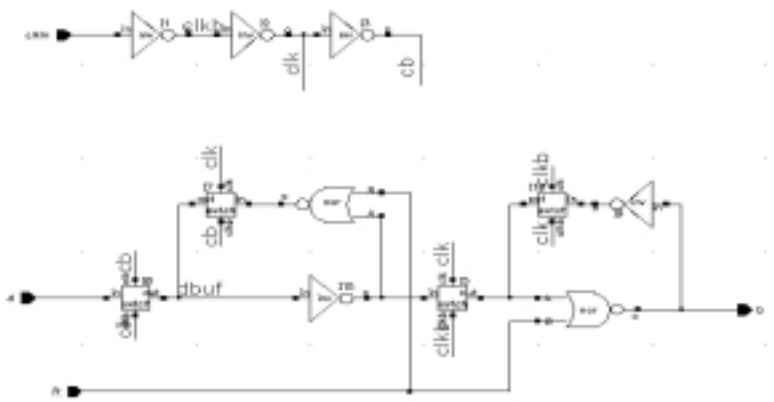
In order to eliminate the PFD dead zone, we require the pulse width of the DFF output to be about 2ns when the PLL is in lock state. In this way the up and down signal are long enough to reach a valid logic level and turn on the switches in the charge pump. This pulse width should not be very large, since a large pulse width will introduce jitter due to the unmatched UP and DOWN current. It should also be long enough to fully turn the UP and DOWN switches. The simulation result is shown in the following figure. Note that right now we only test the function of the blocks, in order to save simulate time, the input clock signal is not 8Mhz, but much larger.



This pulse width is tuned by the delay block, which is showed in the following figure as “dly”.

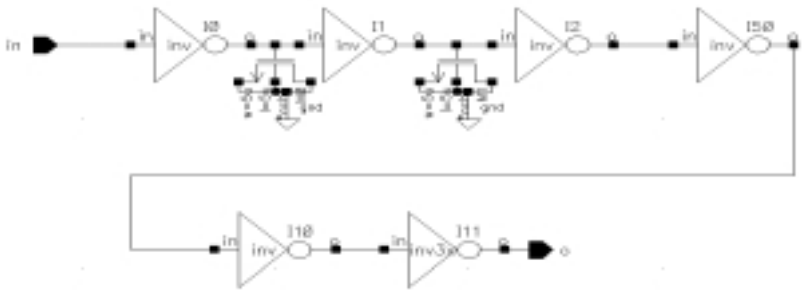


PFD schematics

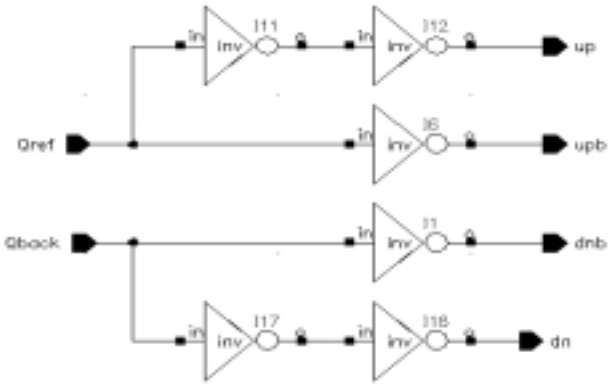


Schematics of DFF used in the PFD block

The “dly” block is shown in the following figure. In this design some transistors (5uX5u) used as capacitors to increase the delay is used. The size of the inverter is: P: 1.25u/0.25u; N:0.5u/0.25u. The last one is 3x larger to drive large load.

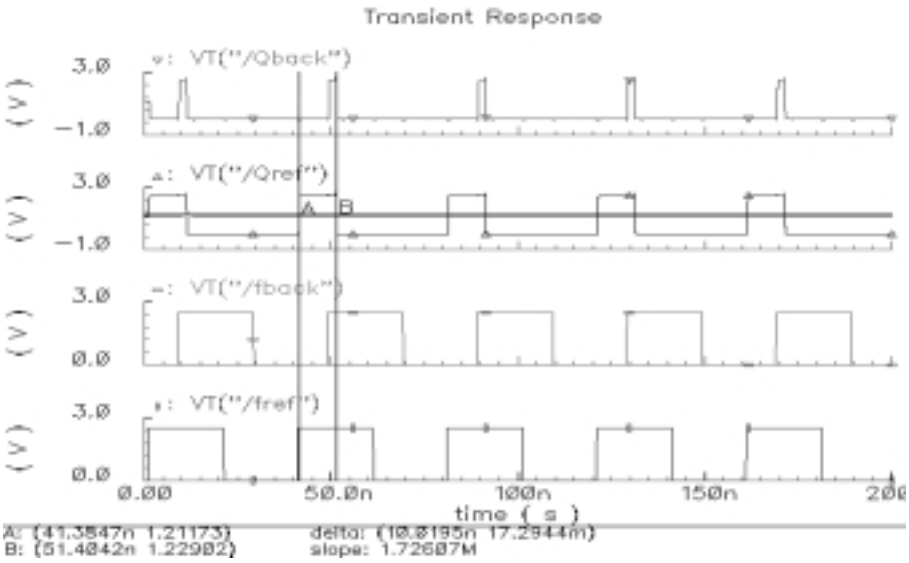


In order to get the “UP”, “DOWN” signals and their complementary, we used a block named “PFDbuf” which consists some inverters as shown below.

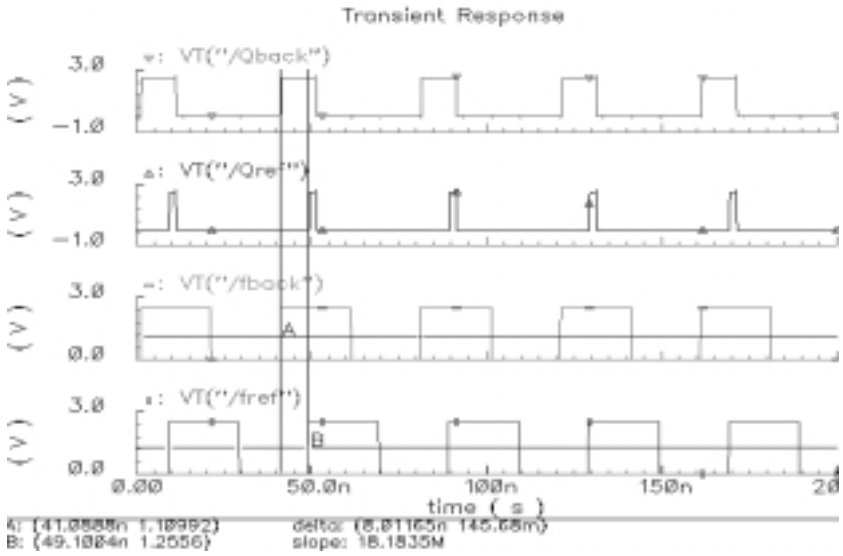


Other gates **transistor sizes**: NAND gate: NMOS: 0.5u/0.25u; PMOS: 0.5u/0.25u; NOR gate: NMOS: 1u/0.25u; PMOS: 4u/0.25u.

The simulation results of the PFD are shown in the following figures.



The f_{ref} signal leads f_{back} signal

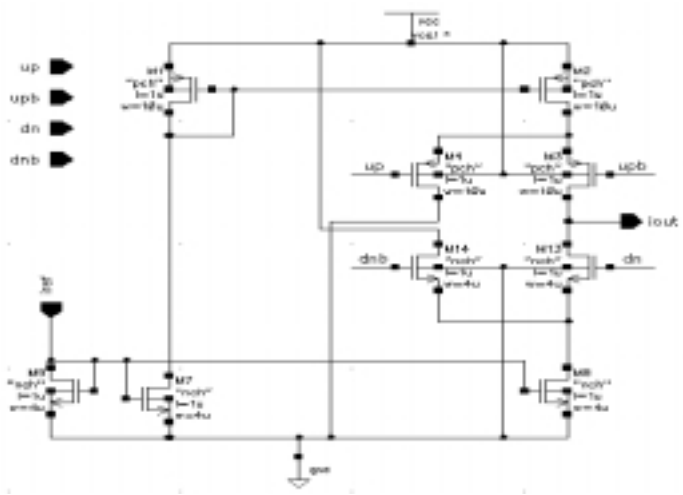


The fref signal lags fback signal

From these simulation results we find the PFD functions correctly.

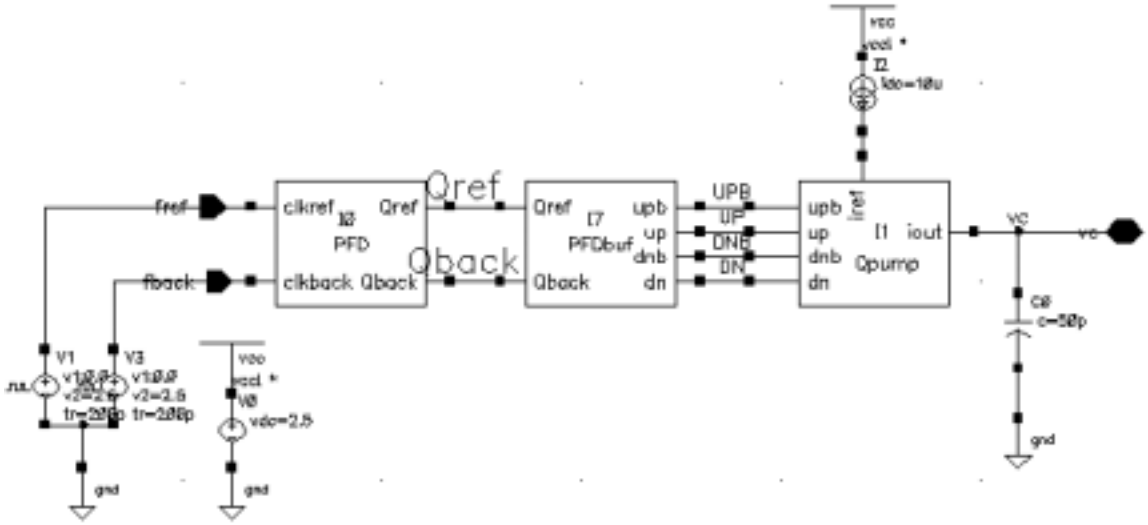
2.3 Charge pump design

In this design, we use a current steering configuration for the charge pump. The schematics and the transistor sizes are shown in the following figure.



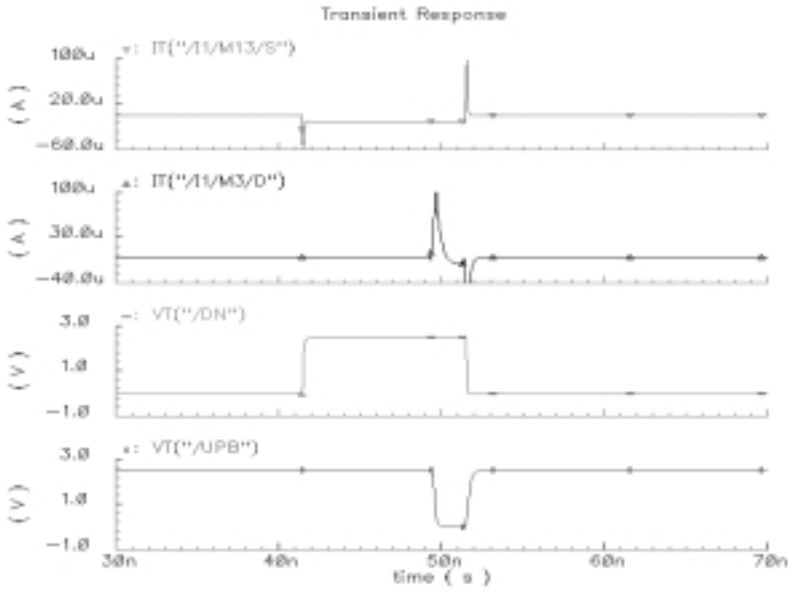
Current steering charge pump

The following circuit is the test bench for the charge pump, this test bench is used to verify the function of charge pump. Note that in order to test the function, only a capacitor is used in place of the filter.

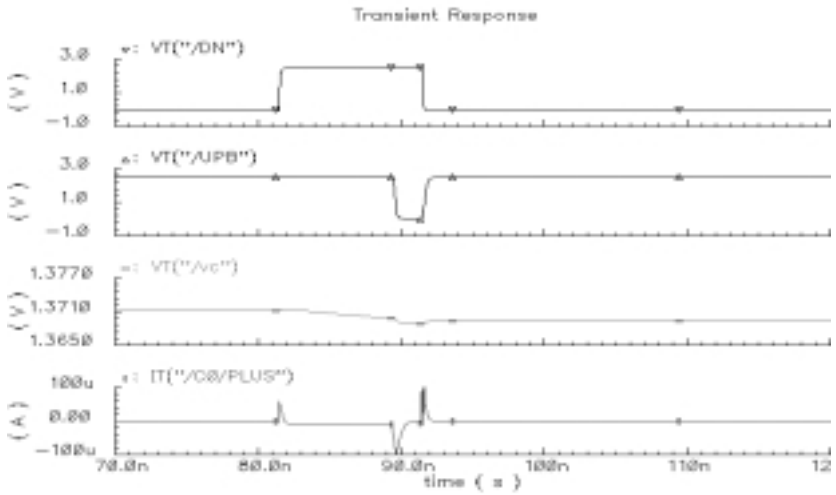


Test bench for the charge pump.

The following figure shows the simulation result for the charge pump. The transistor M3 and M13 are the switches control the charge current (see above figure).

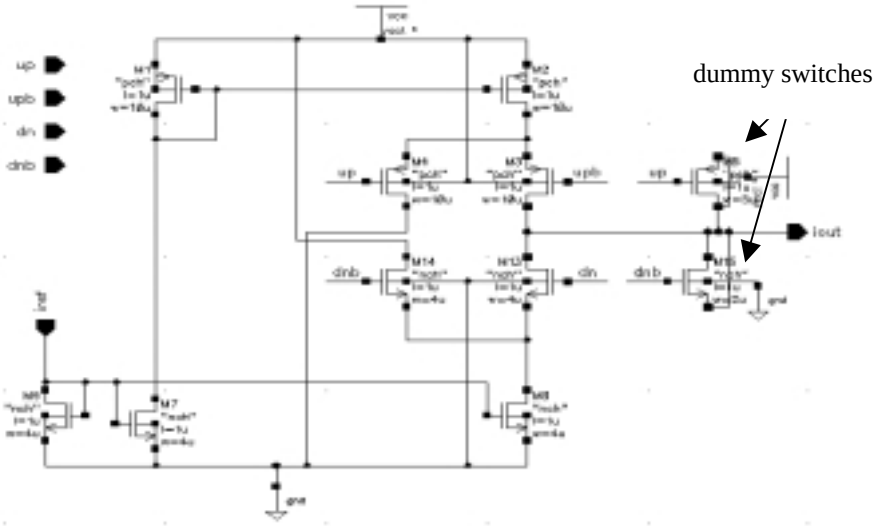


Transistor current

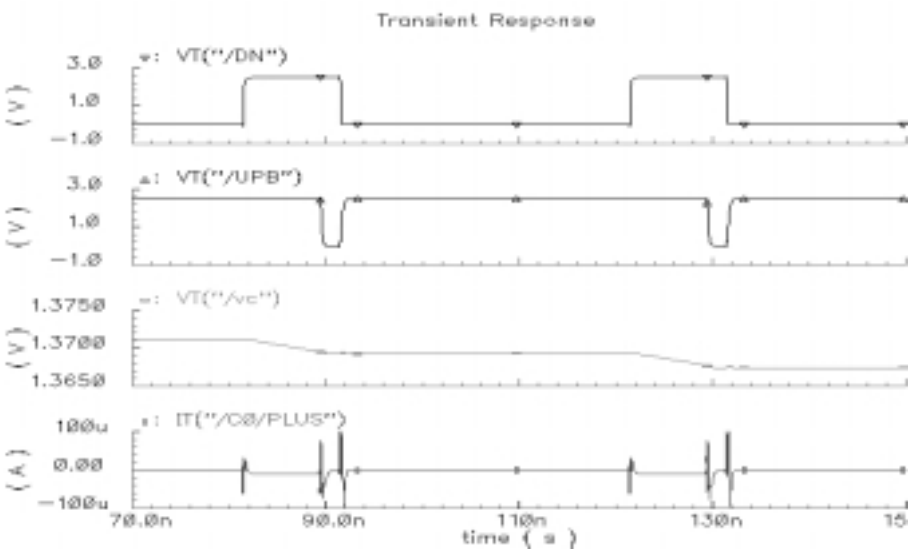


Current and voltage at the loop filter capacitor

From the above two figures we find that the function of charge pump is correct. We also can see large charge injection when the switches change state. In order reduce the charge injection, dummy switches are used in this design as shown in the following figures.



Charge pump with dummy switches.



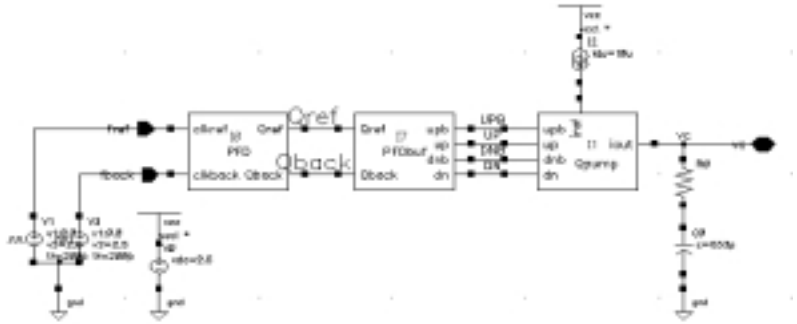
Simulation results for the charge pump with dummy switches.

Though we used the dummy transistor to reduce the charge

Relationship between phase error and VC

The input clock is fixed at 8MHz. Force the feedback output clock to be 8MHz. Select a reference voltage Vc at which the input and output clocks are aligned, then run a series of simulations in order to plot VC verse Φe=Φi-Φo/N.

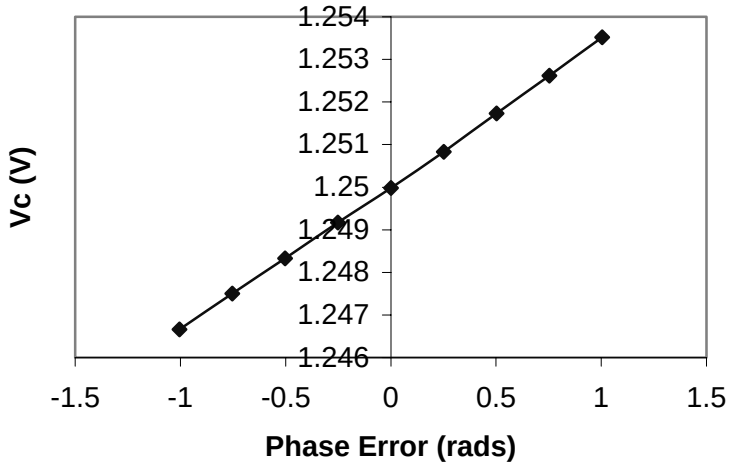
The test bench we used is shown following:



Note that the filter has been implemented using the values calculated above.

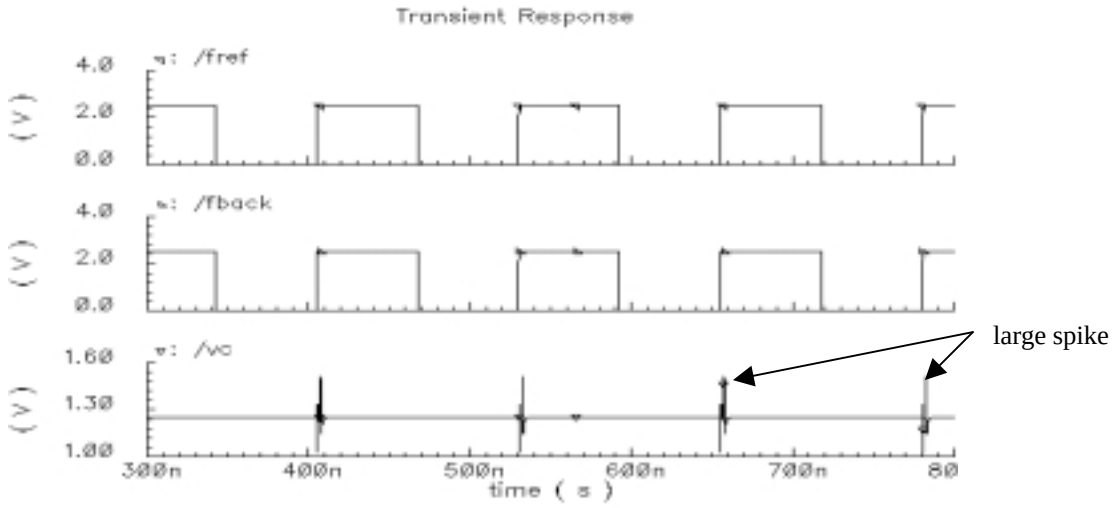
We force the two clock signals into PFD as 8MHz but let the feedback one to be lag or lead the reference. Then run the simulation for 10 clcles, then measure the Vc value. The following table and figure shows the relationship between phase error and Vc. From the figure we can find that the linearity of this design is quite good.

Output clock lags the input clock (ns)	Corresponding phase error Φ_e (rads)	After 10 clycles measured Vc
20	0.16	1.25352
15	0.12	1.25262
10	0.08	1.25173
5	0.04	1.25083
0	0	1.24998
-5	-0.04	1.24917
-10	-0.08	1.24833
-15	-0.12	1.2475
-20	-0.16	1.24666

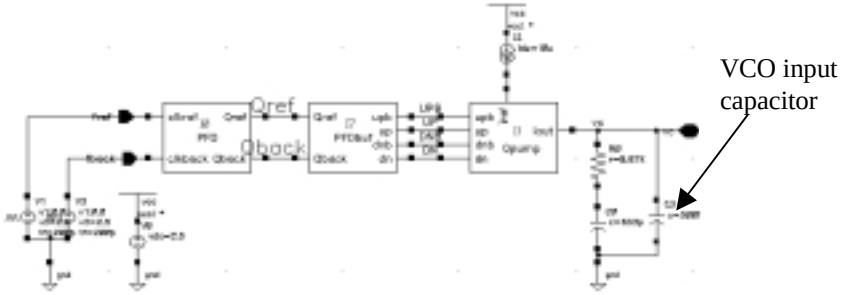


Discussion and conclusions

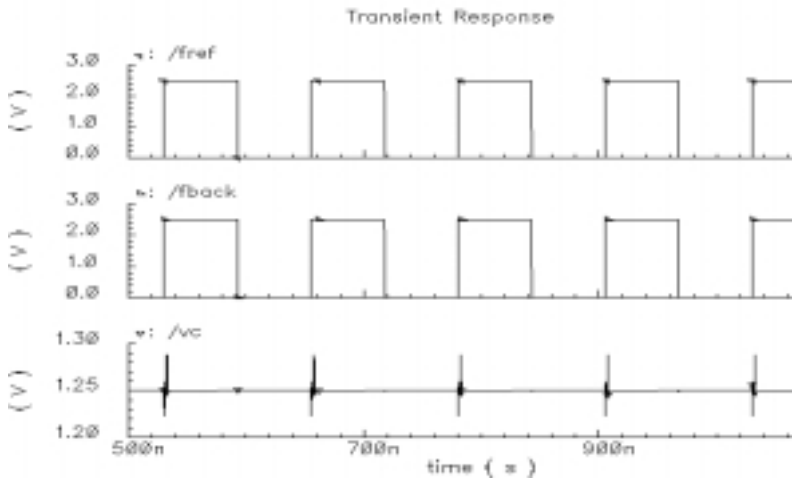
Due to the nonideality of the charge pump, PFD (charge injection, the delay between UP and its complementary signal, DOWN and its complementary, etc), there are large Vc spike even in the lock state, shown in the following figure.



The spike is quite large (about 480mV peak to peak in the figure). Fortunately, the output of the filter drives the VCO, which always shows some load capacitor. BUT this load capacitor has a big effect on the voltage jump. We now model this load capacitor to be about 0.3pF, as show in the following test bench:



Then the Vc spikes become much smaller, as shown in the following, the peak-to-peak value is about 65mV.



In other words, this design should meet the maximum 100mV Vc jump requirement. If the VCO input load capacitance is not large enough to solve the Vc jump problem, a capacitor can be added intentionally. Actually the nonideality of the charge will also introduce phase noise, it should be reduced through both good design (a better charge pump, for example better match between the UP current and DOWN current).

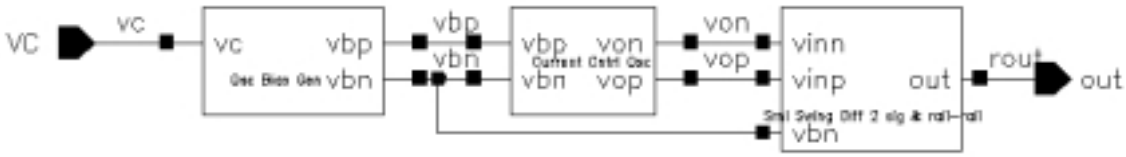
Conclusion: the function of the PFD and charge pump are tested and the results show that their function are correct.

2.4 Self-biased VCO design [MANE96]

2.4.1 VCO circuit design

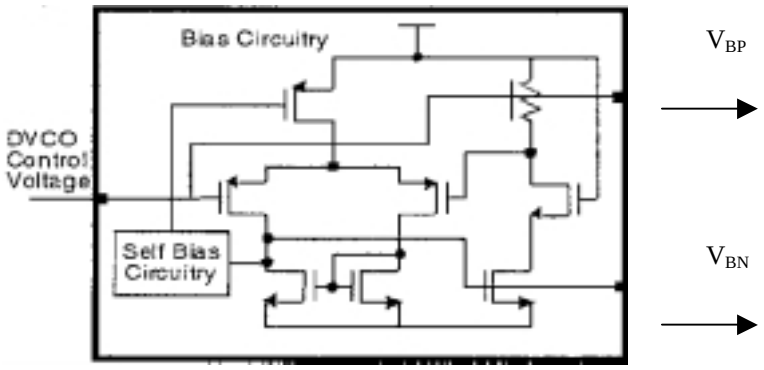
If a precise 50% duty cycle clock is required, a VCO runs at twice the desired frequency is used in the PLL then this twice-frequency signal is divided by 2 using D filp-flop. In this design we did not do this.

The schematic of the VCO is shown in the following figure. It consists of three parts: bias generation circuit, current controlled oscillator and a circuit transfer the small swing differential output from oscillator to a single ended CMOS rail-to-rail clock signal.

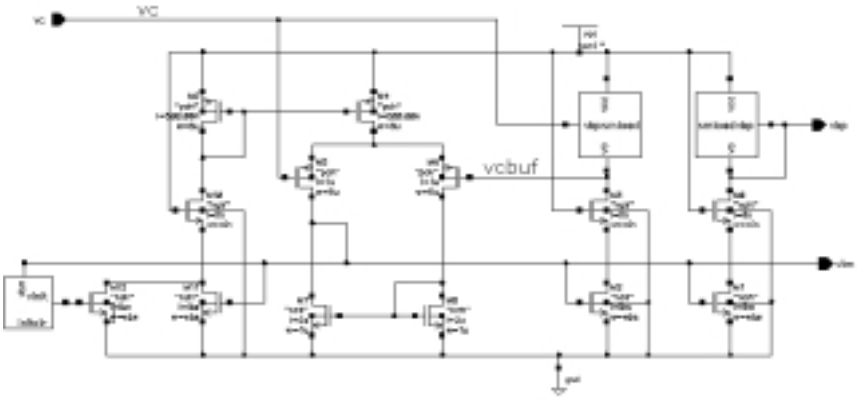


Bias generation circuit

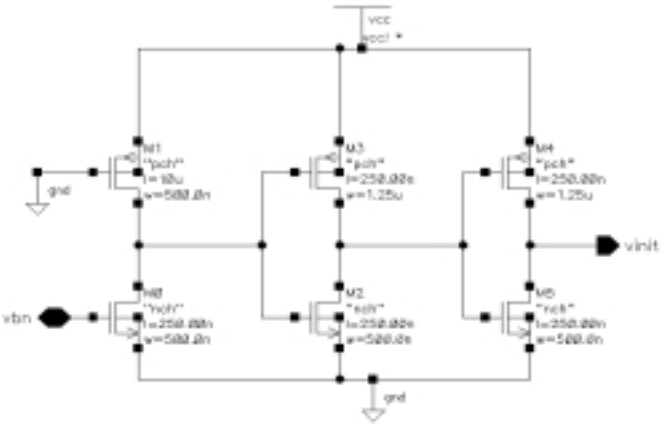
Because the delay cell is current controlled and the VCO frequency is proportional to the bias current, we need a V-I converter to convert the control voltage Vc from the low-pass filter to a current whose value is proportional to Vc. The function block can be shown as following:



It can be implemented by the following schematic.



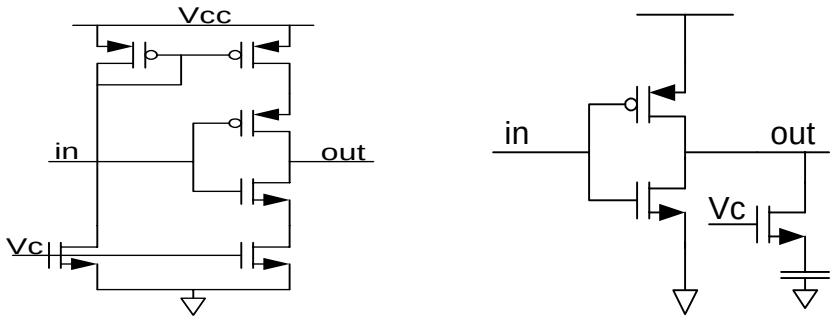
Because this bias generation circuit is self-biased, it needs an initial circuit which can be implemented as shown below:



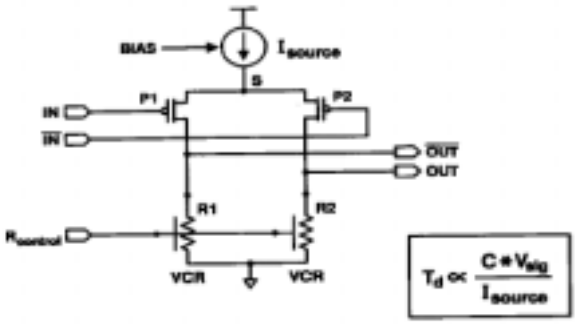
Note that the channel length of M1 is very large which means it is a weak transisitor.

Oscillator circuit

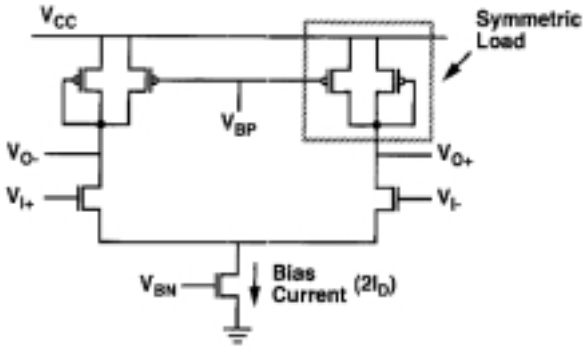
The conventional current delay elements (show following) are power supply sensitive. In this design we did not use this sort of delay elements.



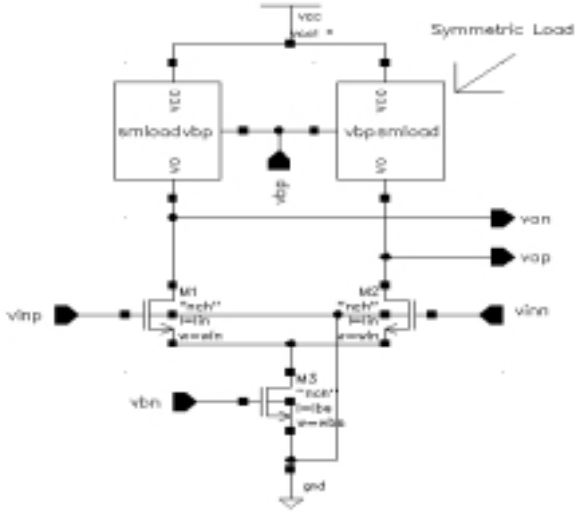
In this design we use differential current-controlled delay element.
The following figure shows the delay cell. It is based on a P-MOSFET source-coupled pair (N-MOSFET source coupled pair can also be used in the design) with voltage-controlled resistor (VCR) load element. By controlling signal $R_{control}$ to VCR, the voltage swing is held constant and independent on power supply voltage. Then the delay is only dependent on I_{source} so that VCO frequency is directly proportional to I_{source}.



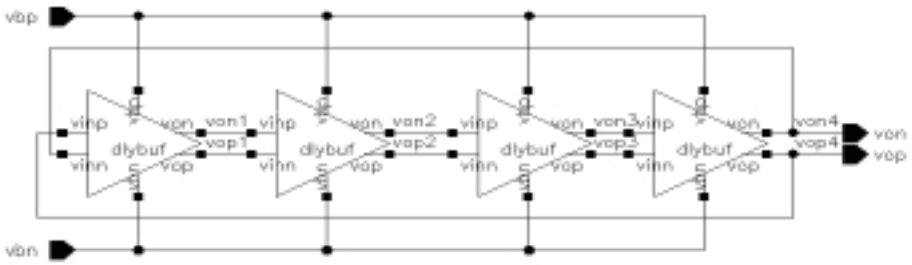
The following figure shows a N-MOSFET source coupled pair delay cell. The symmetric load acts as the VCR and V_{BP} is the control voltage $R_{control}$. Signal VBN is the bias voltage controls the I_{source}.



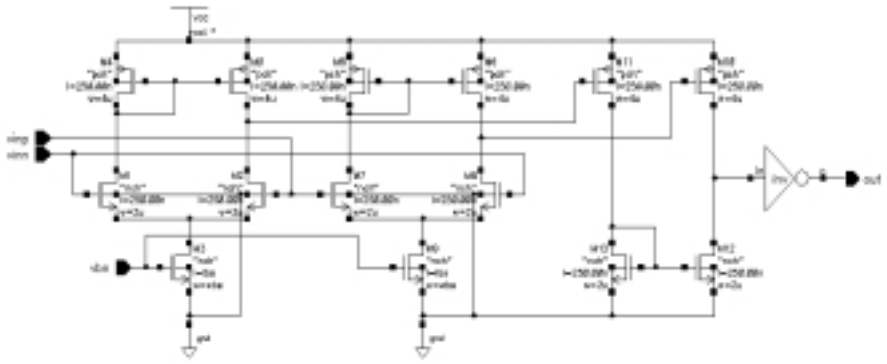
In our design it is implemented as shown in the following figure.



In our design there are 4 delay stages in the oscillator. The four stages are connected as shown in the following figure.



The circuit that transfers the small swing differential output from oscillator to a single ended CMOS rail-to-rail clock signal is shown in the following figure.

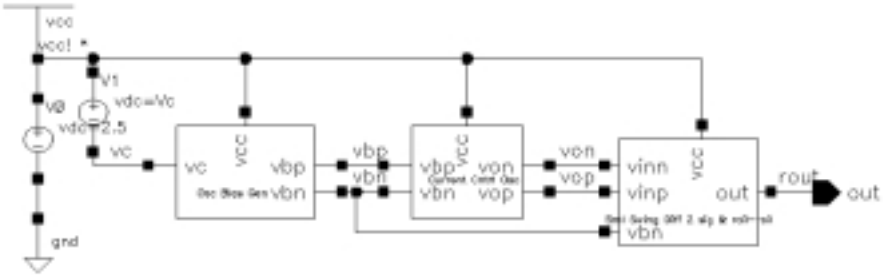


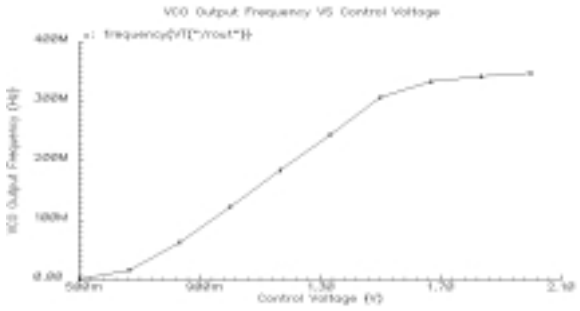
$$F = \frac{1}{2nt} = \frac{\sqrt{\mu C_{ox}(W/L)I_D}}{C_{Load}}$$

$$K_{vco} = \frac{\mu C_{ox}}{2}\left(\frac{W}{L}\right)_{vcr}\frac{1}{C_{load}}$$

Here I_D is the bias current of the delay stages and C_{load} is the total buffer output capacitance for all stages. The following figure shows the frequency of the VCO versus the control voltage.

The following figure shows the VCO test circuit.





From the figure we can find that its gain is as:

Kv=366MHz/V (From Vc=0.7 V to 1.5 V)

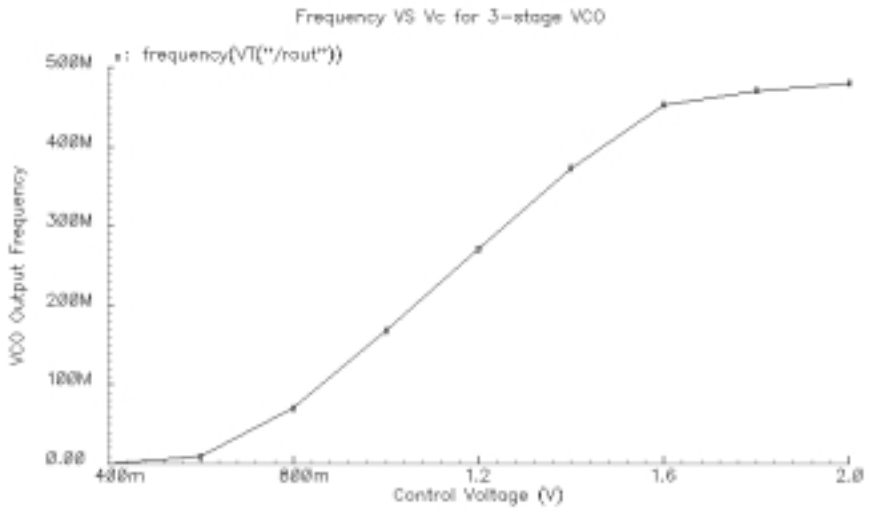
Some of its output frequency and the corresponding control voltage are as follows:

Fo=1.33 MHz @ Vc=0.5V
Fo=16.6 MHz @ Vc=0.667V
Fo=62.7 MHz @ Vc=0.833V
Fo=122.5 MHz @ Vc=1.0V
Fo=184.0 MHz @ Vc=1.167V
Fo=214.0 MHz @ Vc=1.25V
Fo=243.8 MHz @ Vc=1.33V
Fo=304.4 MHz @ Vc=1.495V
Fo=332.8 MHz @ Vc=1.664V
Fo=341.8 MHz @ Vc=1.833V
Fo=347.2 MHz @ Vc=2.0V

We use some variables to optimize the VCO. The following table shows the final results we used in the design.

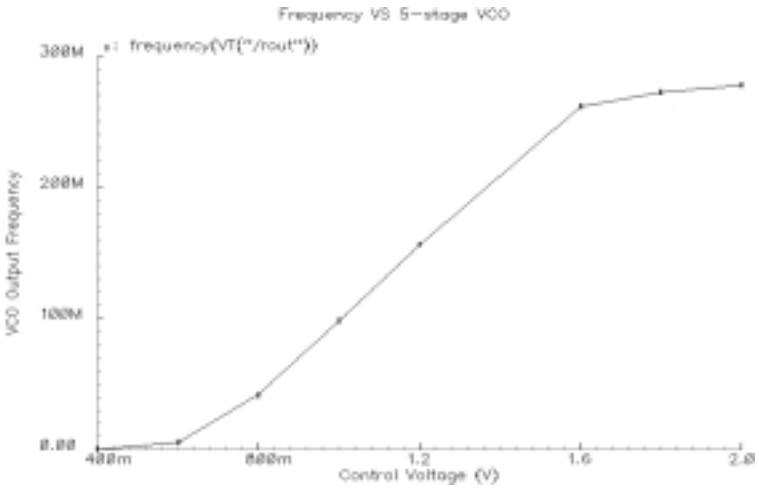
Wl/Ll	4.5/1.5
Win/Lin	1.0/0.8
Wbs/Lbs	3.0/3.0

With the same size of delay cell and bias generation circuits, we test the VCO with different delay stages. The following figure shows the results of three-stage VCO:



From the figure we can find that gain of this 3-stage VCO is:

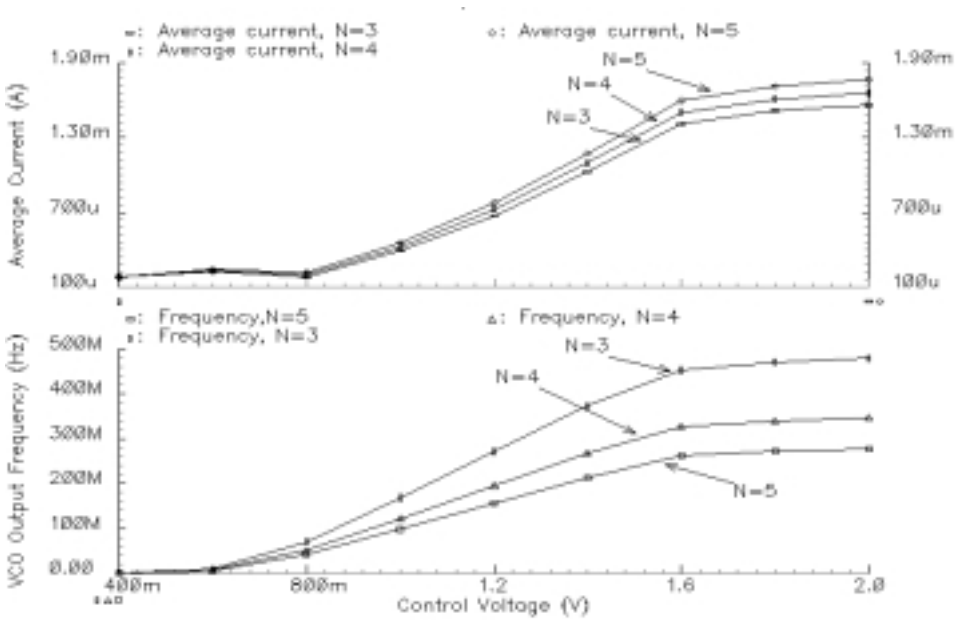
Kv=478MHz/V (From Vc=0.8 V to 1.6 V)

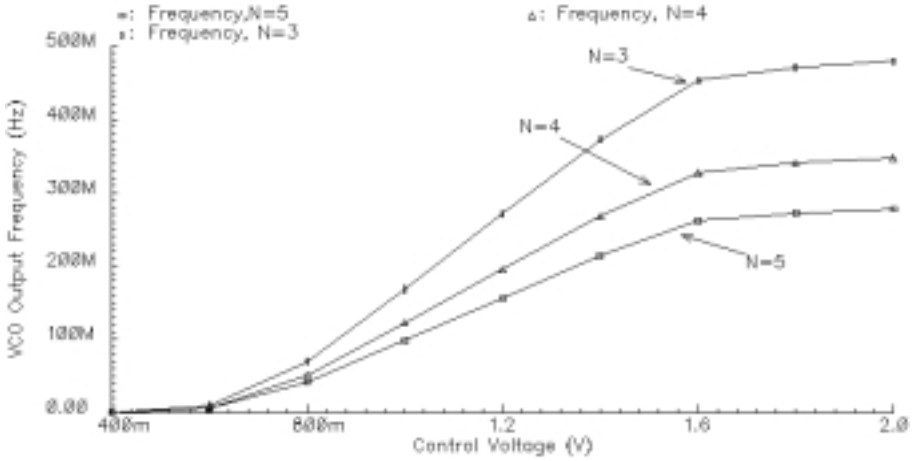


Kv=275MHz/V (From Vc=0.8 V to 1.6 V)

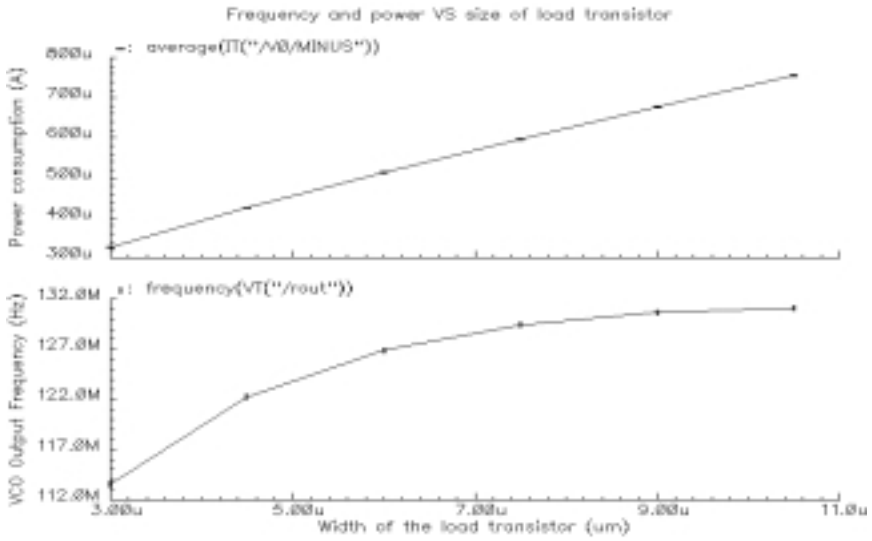
VCO output frequency, power VS number of delay stages

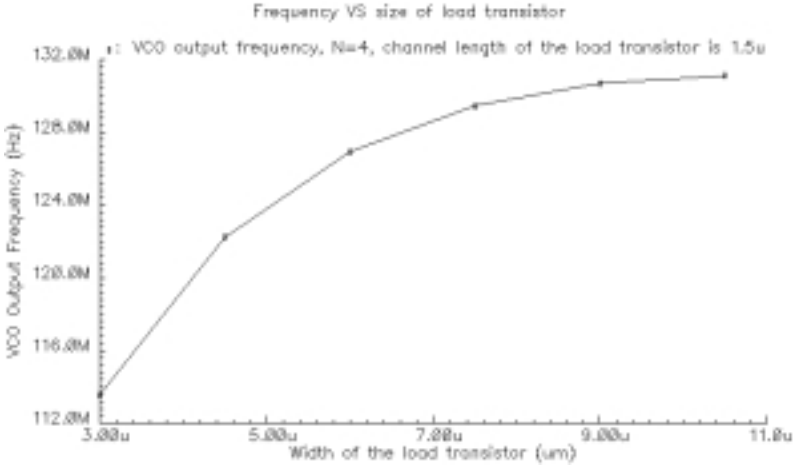
In the following simulations, we keep the bias generation circuit and the circuit transfer the small swing differential output from oscillator to a single ended CMOS rail-to-rail clock signal to be identical. The delay stage itself also keeps the same. We only change the number of delay stages in the current control oscillator block.



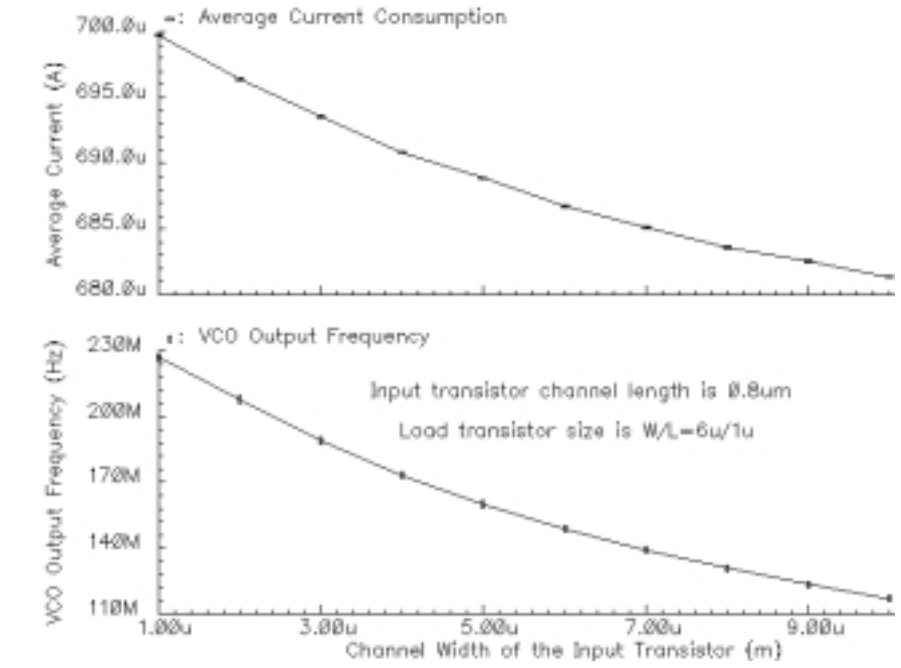


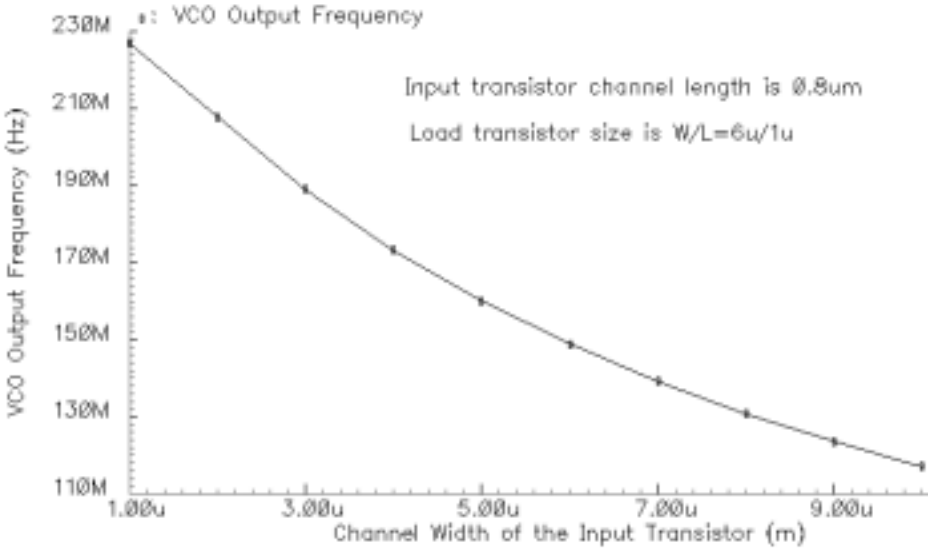
VCO output frequency, power VS size of load transistor





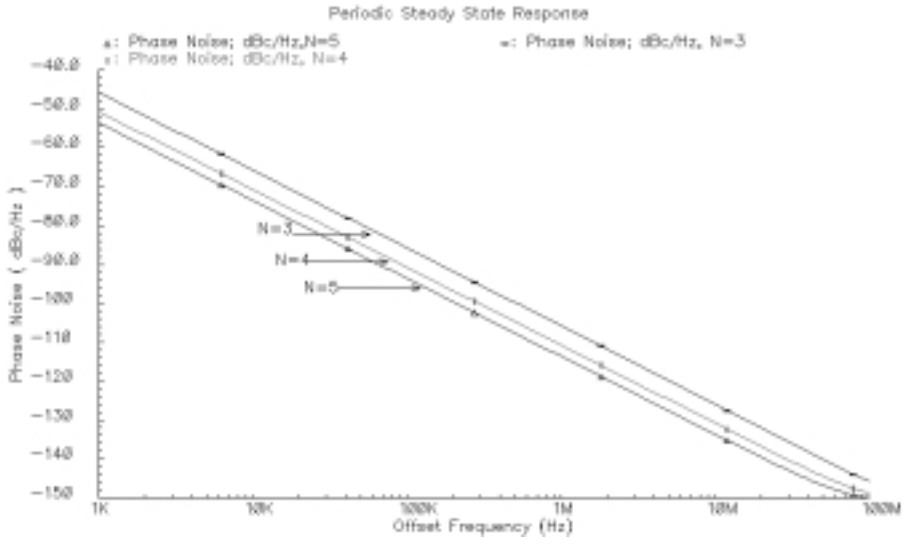
VCO output frequency, power VS size of the input transistor



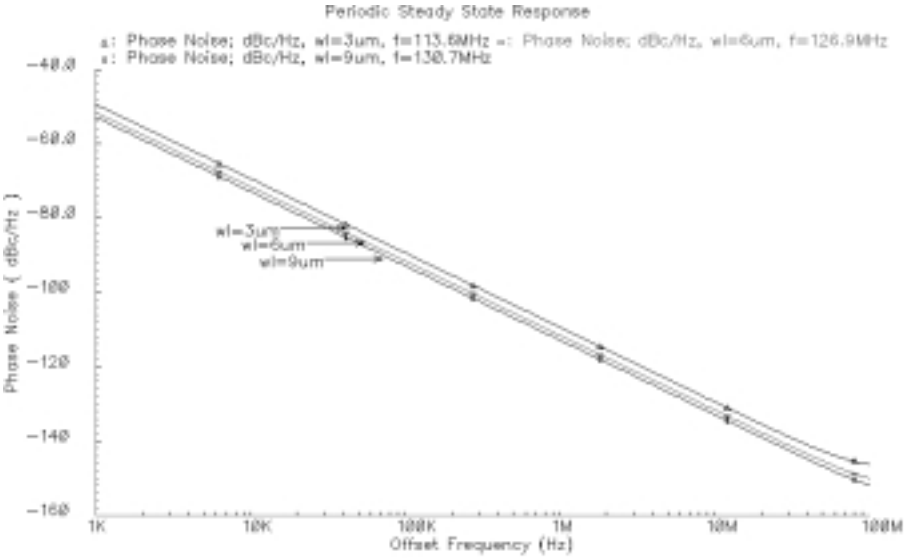


2.4.2 VCO Phase noise simulation results

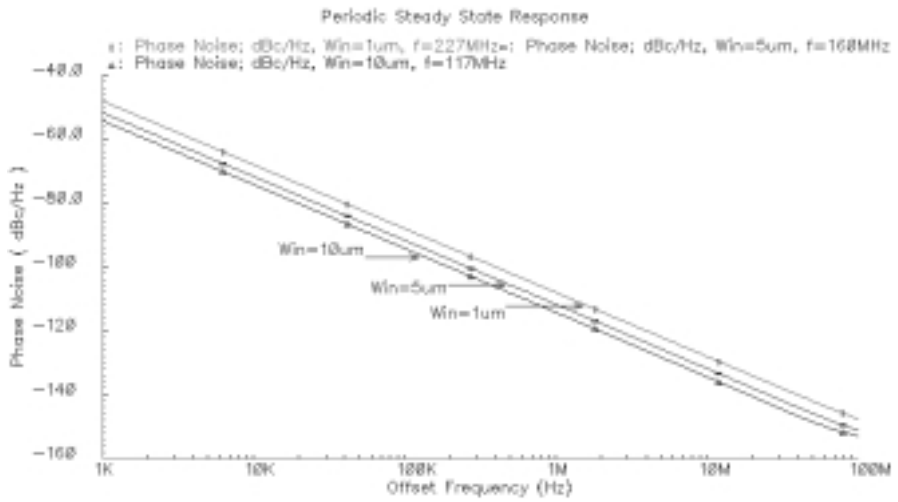
The following figures show the simulation results using SpectreRF Periodic Steady-State (PSS) analysis and Periodic Noise (PNoise) analysis tools.



Phase noise VS delay stage number



Phase noise VS load transistor size. Number of delay stages keeps the same.



Phase noise VS input transistor size. Number of delay stages keeps the same, in this case it is 4.
From Win=1um to Win=5um, there is a 3.67dB noise reduction.
From Win=5um to Win=10um, there is a 2.5dB noise reduction.

Some conclusions about the noise of ring oscillator: The jitter resulting from the device intrinsic noise generally decreases as the power consumption increases. The effect of power supply and substrate noise on the total noise of a given oscillator topology is relatively independent of the power consumption [HERZ98].

2.5 Loop filter design

We have known the lock-in time is 0.15ms, and is given $Ts=16\pi/\omega_n$, then

ω_n=335Krad/s or 53KHz

We chose damping factor ζ to be 0.707, which will give the best settling time. Here loop bandwidth ω_B is the -3dB bandwidth, and which is:

$$\omega_B = \omega_n \sqrt{2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2 + 1}}$$

When $\zeta = 0.707$, $\omega_B = 2.06\omega_n$. So

ω_B = 670Krads/s and f_B=106KHz.

The input reference is f_{in}=8MHz, f_B/f_{in}=1.25%, is still a quite narrow bandwidth PLL.

The divide down ratio should be the ratio between the desired output frequency and the input reference frequency, in this project, it is:

N=200MHz/8MHz=25.

If we use a charge pump current of I_p=10μA, then the gain of the PFD is:

K_d=10μA/(2π)=1.59μA/rad.

We have design a VCO with a gain of:

K_o = 365 MHz/V = 2,293 Mrad/V

The design requires that the instantaneous jumps of VCO control voltage (Vc) not exceeding 10mV (later the instructor changed this value to 100mV). For a passive loop filter composed of a resistor R in series with a capacitor C, the instantaneous jumps of Vc is I_pR. Ip=10μA, then R ≤ 1KΩ (for Vc jumps not exceed 100mV R ≤ 10KΩ).

We already know the natural frequency ω_n=335Krad/s. From $\omega_n = \sqrt{\frac{I_P K_o}{2\pi N C}}$ we can find:

$$C = \frac{I_P K_o}{2\pi N \omega_n^2}$$

$$C = \frac{10\times10^{-6}\times2\pi\times365\times10^6}{2\pi\times25\times(335\times10^3)^2} = 1.3\text{nF}$$

From $\xi = \frac{\omega_n RC}{2}$, we get: $R = \frac{2\zeta}{\omega_n C}$. ζ=0.707, then R=3.25KΩ. The value of R is smaller than 10KΩ, which means that the maximum Vc jump is about 33 mV.

Summary:

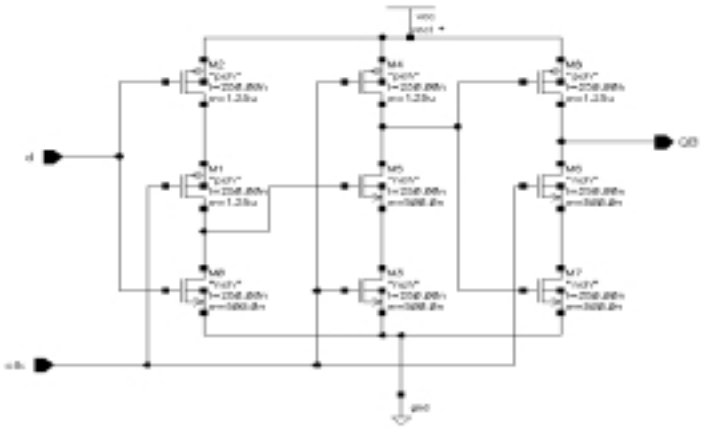
Ip=10uA;
Ko=365KHz/V
R=3.25Kohms;

C=1.3Nf
f_n=53KHz.

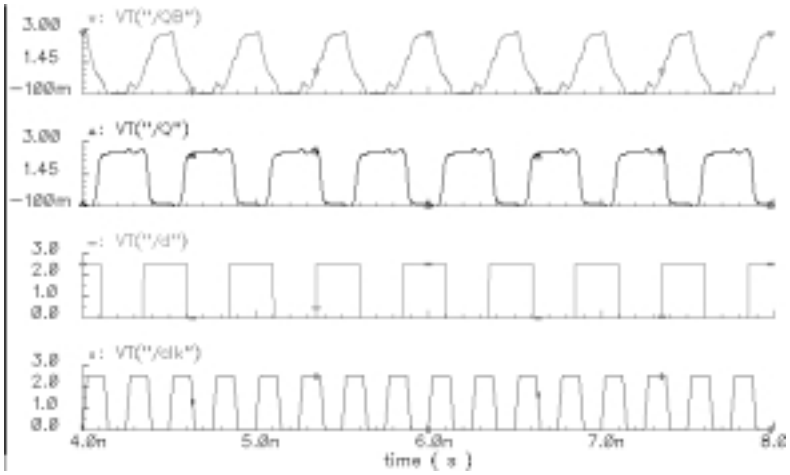
2.6 High-speed programmable divider design [LARS96]

From the point view of deskew, it simply masks out the clock cycles, dose not introduce any additional delay in the feedback clock. One important requirement for the divider is that the divider should be able work at the **maximum** VCO output frequency. This design requirement is not trivial for modern high frequency system. For our VCO designed in this project the maximum output frequency is about 400MHz. For a 0.25um CMOS process, it is not difficulty to meet this requirement.

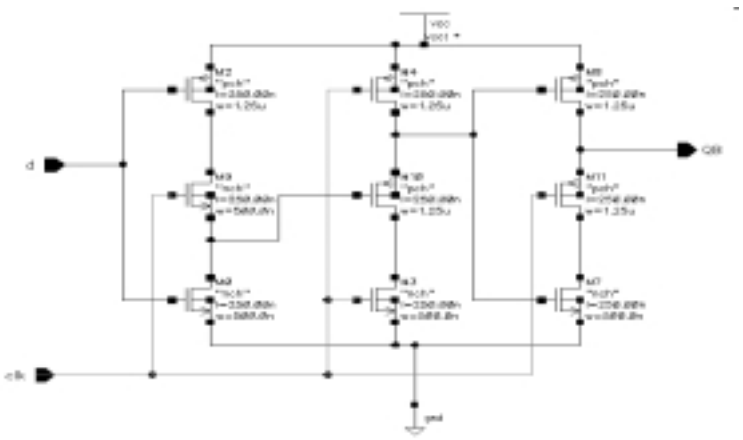
But when we design the divider, our goal is a high speed (above GHz), general purpose programmable and high portable divider. TSPC DFF is a very popular high speed dynamic circuit. The following figure shows a rising edge triggered TSPC DFF and its simulation results. It can be seen that this DFF can run at a input clock frequency of about 4GHz .



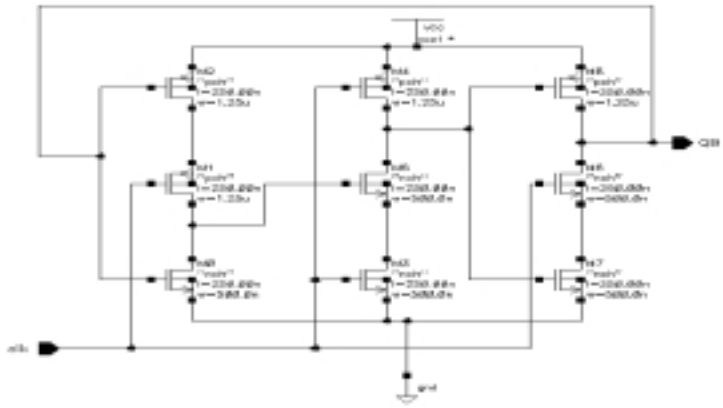
A rising edge triggered TSPC DFF.



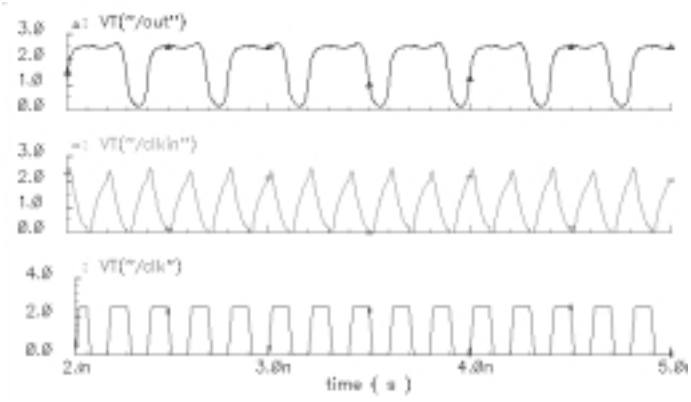
TSPC DFF running at 4GHz input clock.



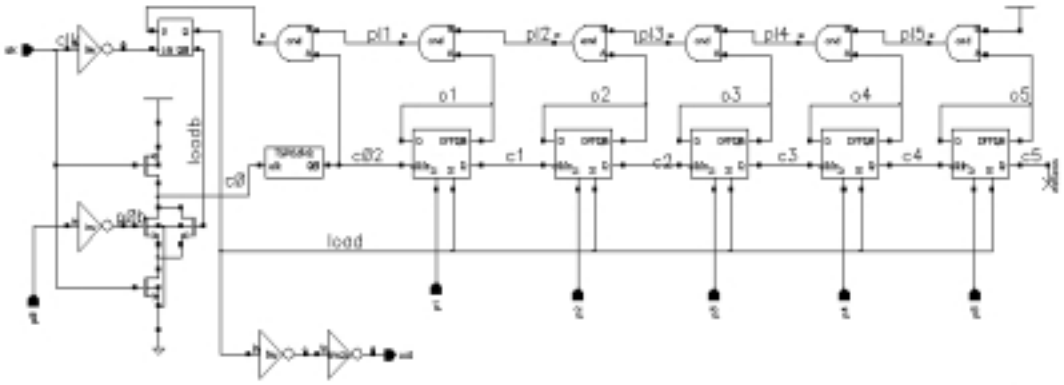
A falling edge triggered TSPC DFF



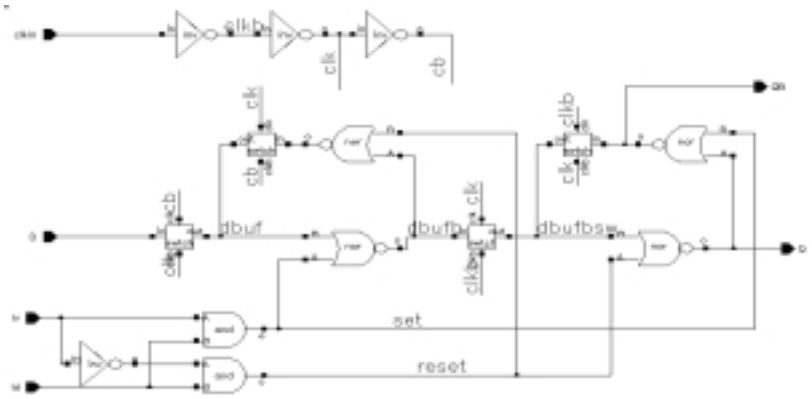
Divide by 2 circuit using TSPC DFF



Simulation results for the divide-by-2 TSPC DFF.

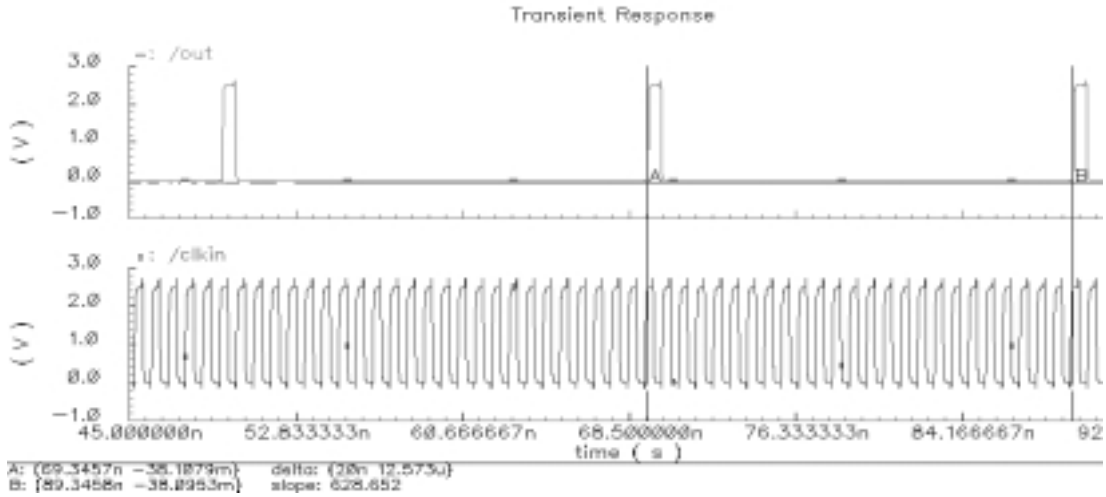


Schematic of the 6-bit programmable divider



Static loadable DFF used for the divider

This divider can work with a input frequency of 1GHz, as shown in the following figure.



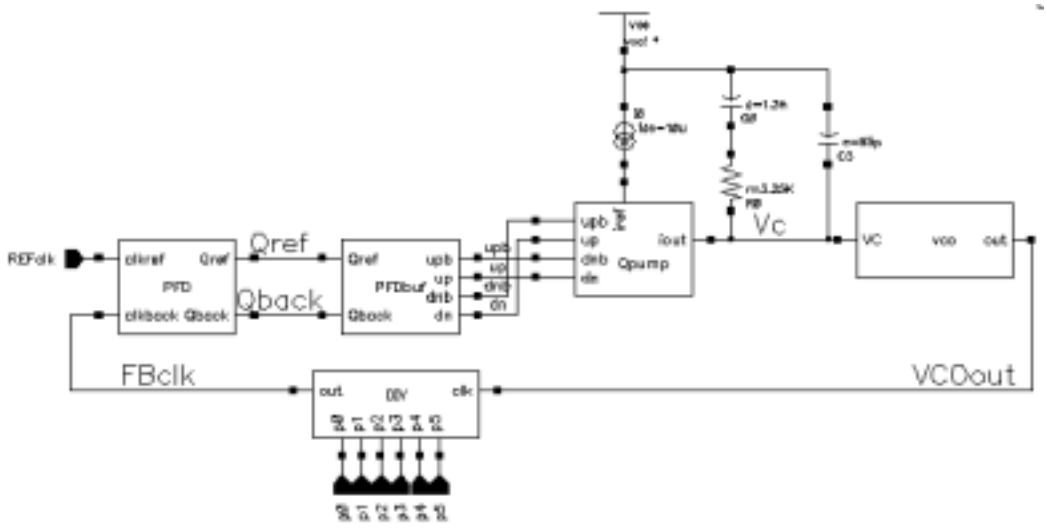
VerilogA modeling of the divider

During the simulations of the whole PLL, the divider consumes a lot of computing time. In order to save the simulation time when checking the PLL in a system level, we can use the build-in verilogA function of Cadence Analog Artist Mixed signal simulation environment. The following is a verilogA code for a divide-25 divider:

```
// VerilogA for pll, DIV_va, veriloga

`include "constants.h"
`include "discipline.h"

module DIV_va (in,out);
input in; output out; electrical in, out;

parameter real vlo=0, vhi=2.5;
parameter integer ratio=25 from [2:inf);

parameter integer dir=1 from [-1:1] exclude 0;
             //dir=1 for positive edge trigger
             //dir=-1 for negative edge trigger

parameter real tt=100p from (0:inf);
parameter real td=10p from (0:inf);
parameter real ttol=1p from (0:td/5);

integer count,n;
real dt;

analog begin

             //Phase /frequency detector state machine
       @(cross (V(in)-(vhi+vlo)/2, dir, ttol)) begin
             count = count+1;
             if (count >= ratio)
                    count=0;
             n=(2*count >= ratio);

       end

V(out) <+ transition(n? vhi: vlo, td, tt);

end
endmodule
```

Chapter 3. PLL simulation results

The following figure shows the whole PLL.

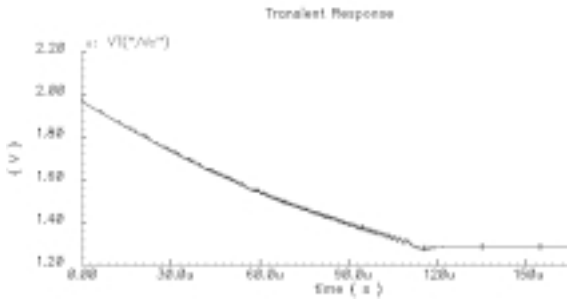


Schematic of the whole PLL

Results for divide ratio N=25

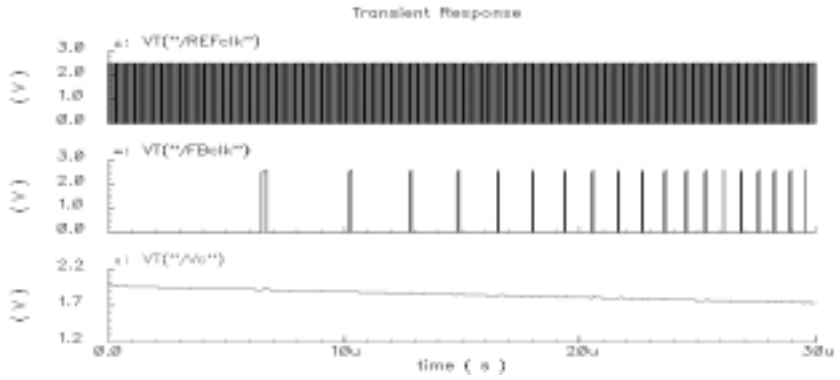
Pull-in process

In the simulation, we initialize the voltage of Vc to be 2V which means the VCO is in a low frequency. The frequency of the reference clock is 8MHz.

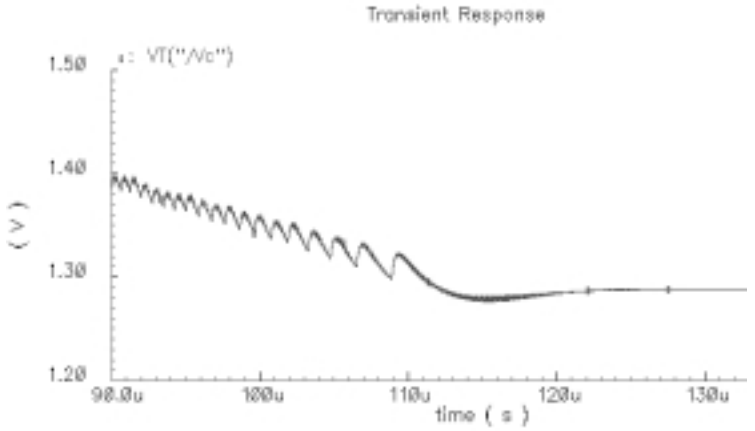


Vc versus time, initial value is 2 V

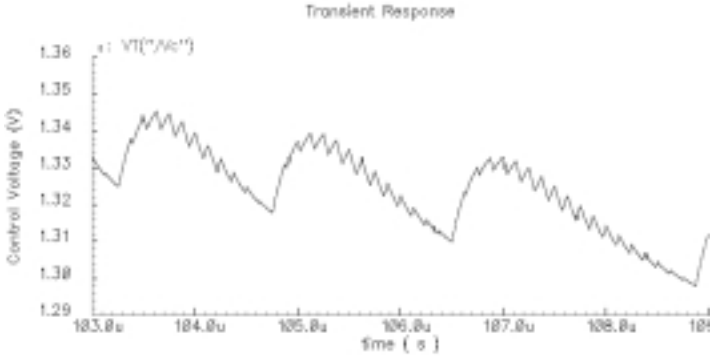
It can be see that the control voltage stabilized after 120 μS, which means the PLL is in lock state. This is a typical pull-in process.



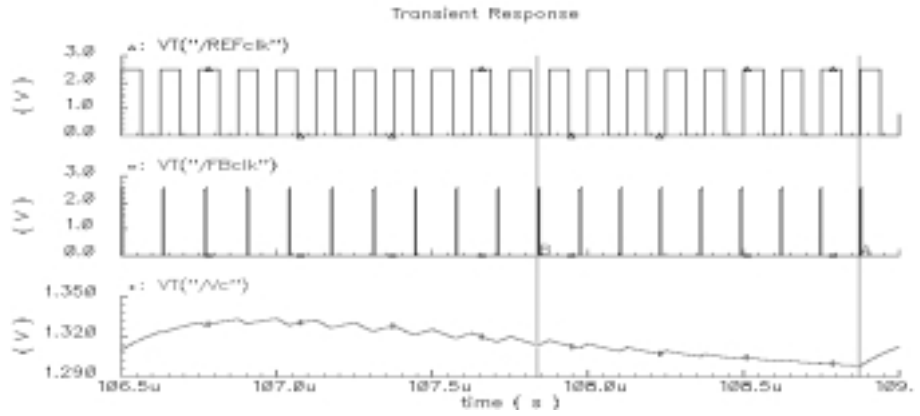
The PLL is not in lock state at first



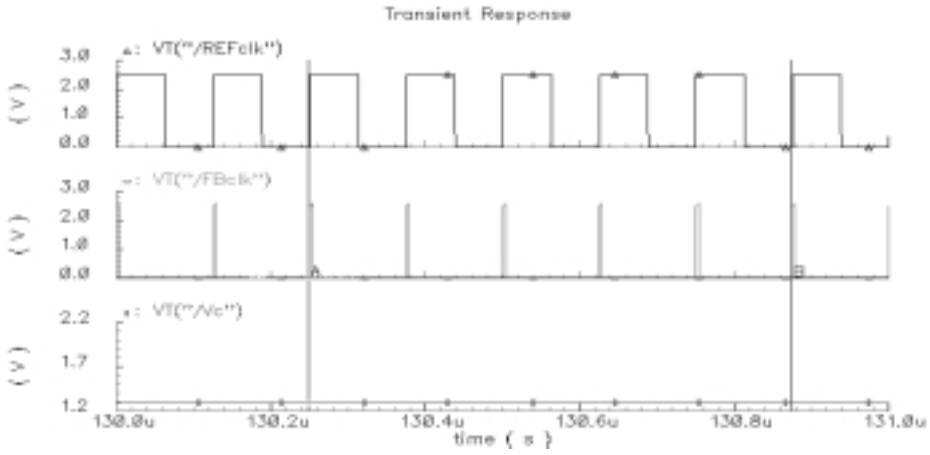
There are some ripples in Vc when it approaches lock state



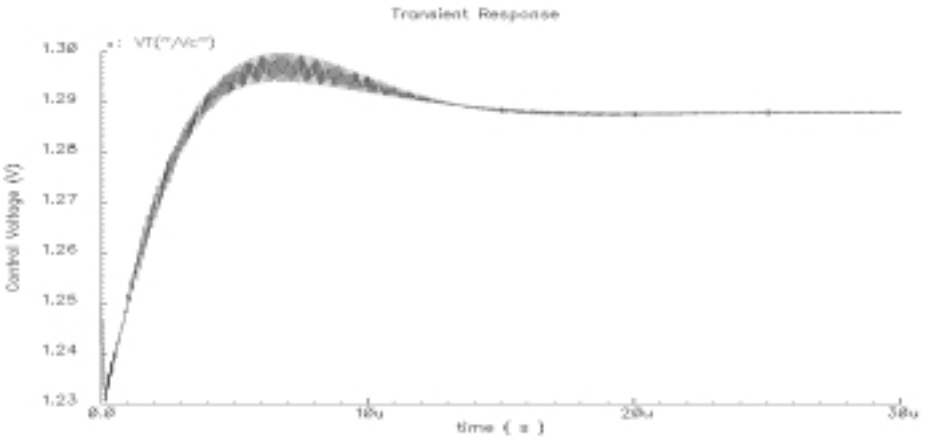
Much detailed Vc. The smallest ripple is due to the time-discrete nature of the PFD. The longer ripple is due to the finite bandwidth of the PLL



It is interesting to notice that lower point of the Vc ripple corresponding to the rising edge of the feedback clock signal and the upper point corresponding to the rising edge of the reference clock signal.



This figure just shows that the two clocks are aligned.



This figure shows the control voltage versus time with an initial value of 1.25 V.

Frequency step response

The hold range of frequency step is:

$$\omega_H = 2\pi NK_F = K_o I_P R$$

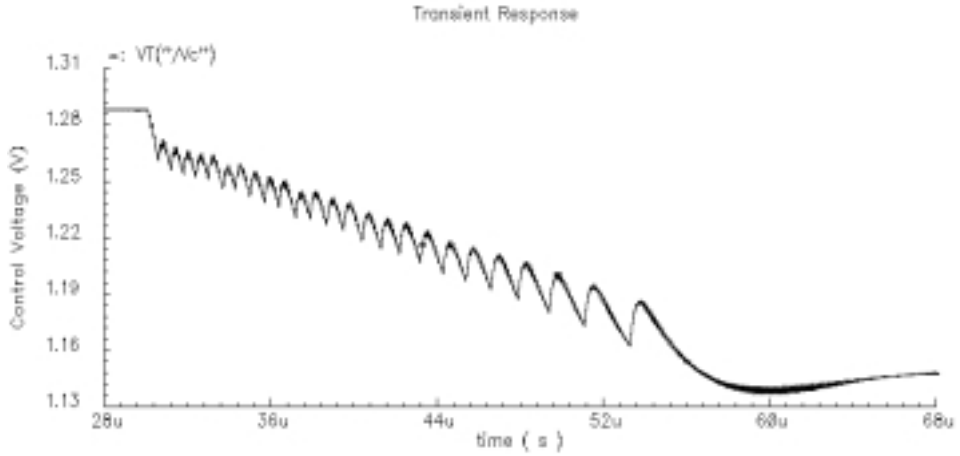
For this design, Ko=366MHz, I_P=10 μA, R=3.25Kohms, f_H=11.9MHz.

The lock range is:

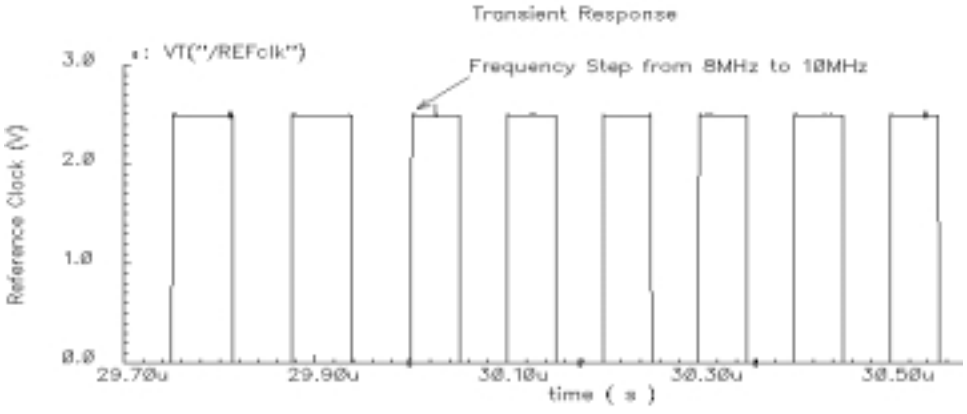
$$\omega_L = 4\pi\zeta\omega_n$$

For this design, ζ=0.707, ω_n=53KHz, f_L=0.47MHz.

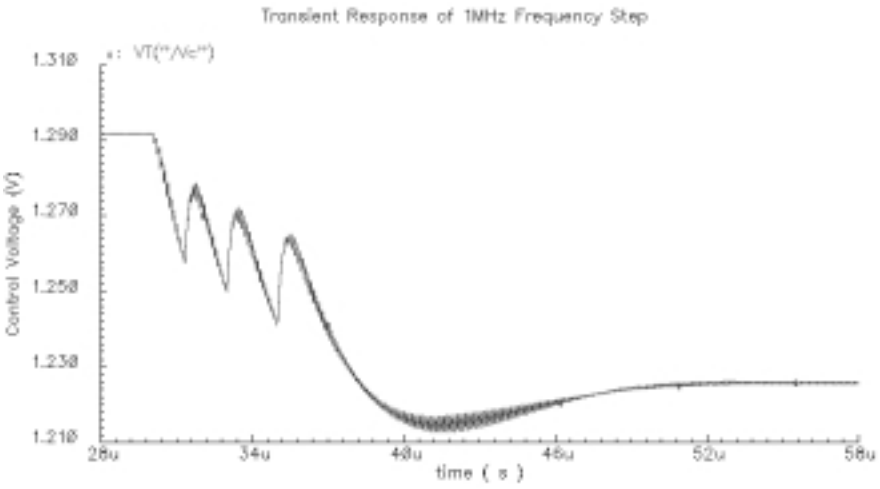
Case I: large frequency step



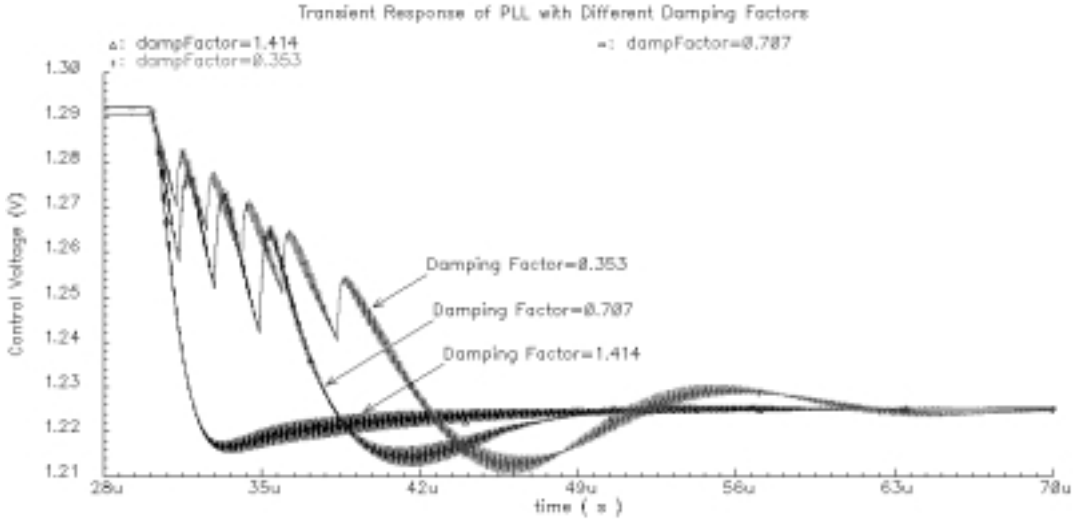
Control voltage Vc VS time: The PLL is initially in lock state with a 8MHz reference clock. At time=30μS, the reference increase to 10MHz with a step frequency change of 2MHz.



The PLL is initially in lock state with a 8MHz reference clock. At time=30μS, the reference clock frequency increase from 8MHz to 10MHz with a step frequency change of 2MHz

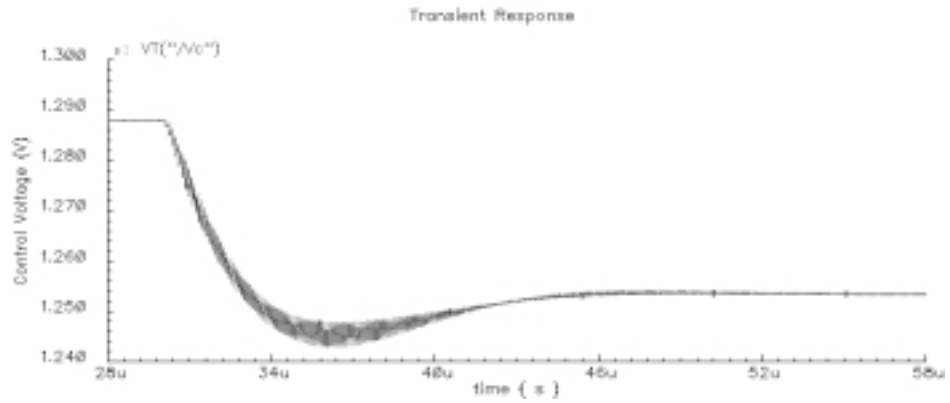


Control voltage Vc VS time: The PLL is initially in lock state with a 8MHz reference clock. At time=30μS, the reference increase to 9MHz with a step frequency change of 1MHz.

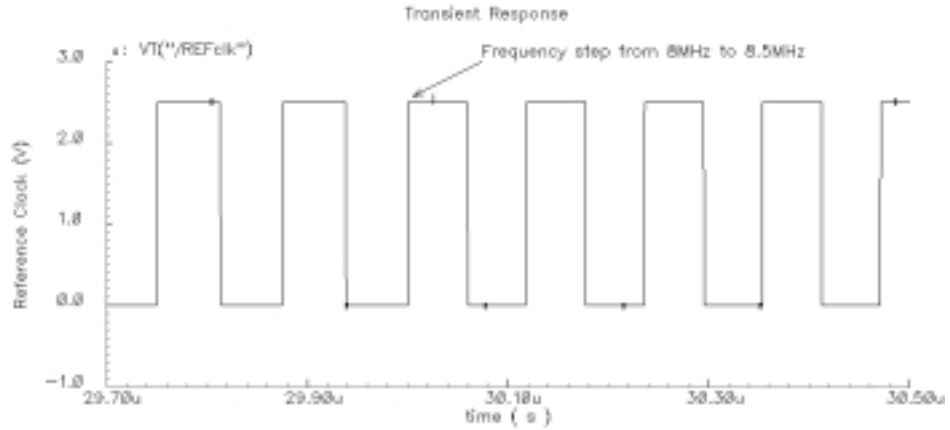


Control voltage Vc VS time for PLL with different damping factors but the same natural frequency: The PLL is initially in lock state with an 8MHz reference clock. At time=30μS, the reference increase to 9MHz with a step frequency change of 1MHz.

Case II: small frequency step

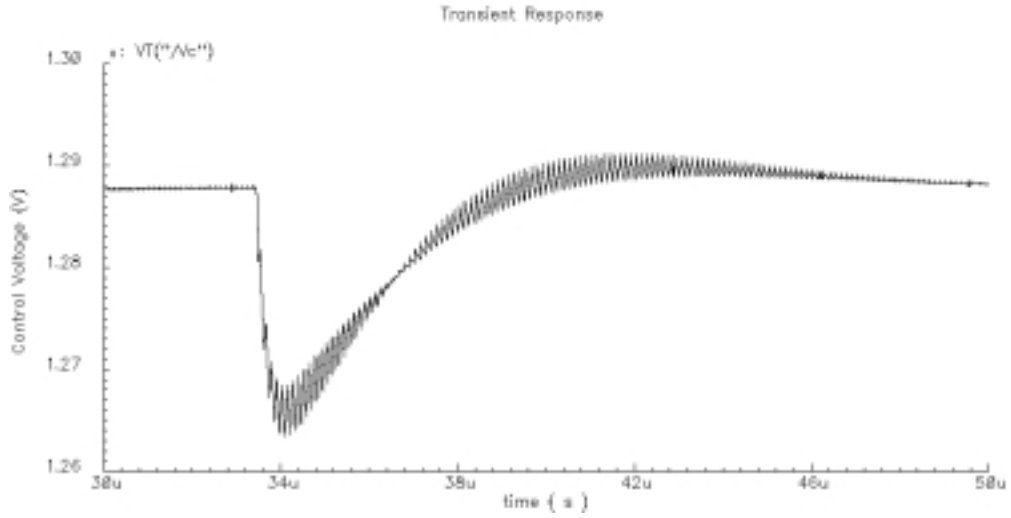


Control voltage Vc VS time: The PLL is initially in lock state with a 8MHz reference clock. At time=30μS, the reference increase to 8.5MHz with a step frequency change of 0.5MHz.

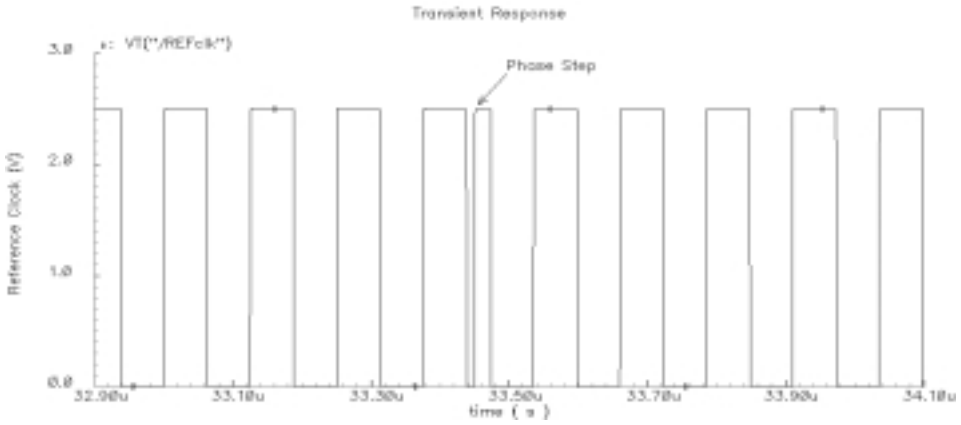


The PLL is initially in lock state with a 8MHz reference clock. At time=30μS, the reference increase to 8.5MHz with a step frequency change of 0.5MHz.

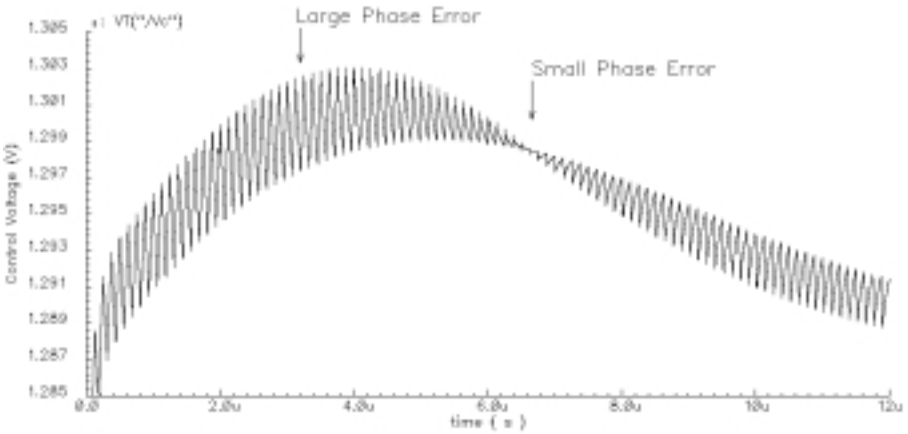
Phase step response



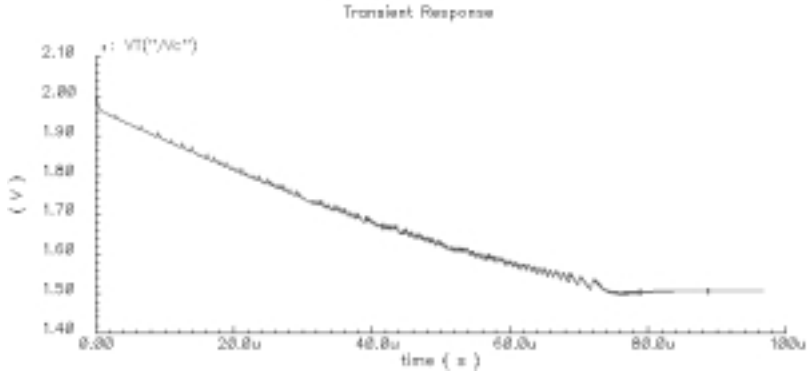
Control voltage VS time in the case of a 50ns phase step 90ns.



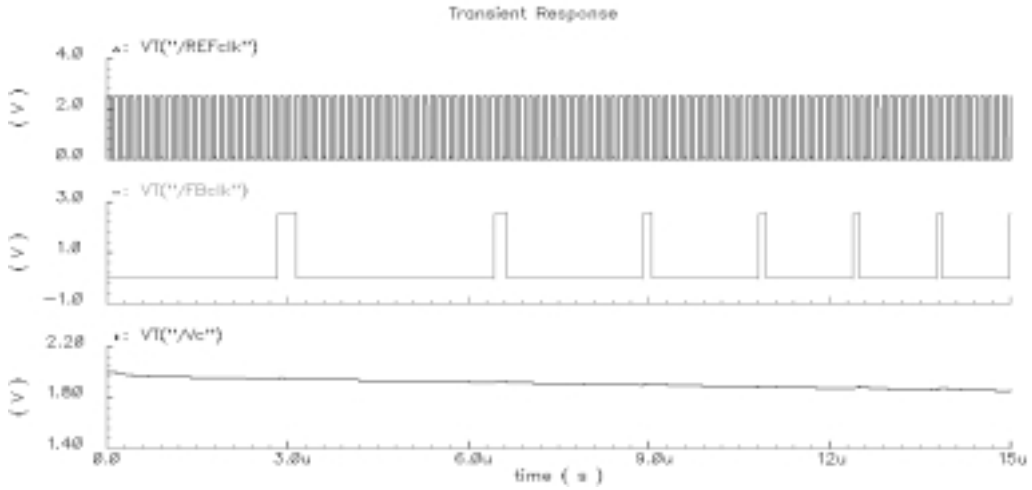
Input reference clock with a phase step of 90ns.

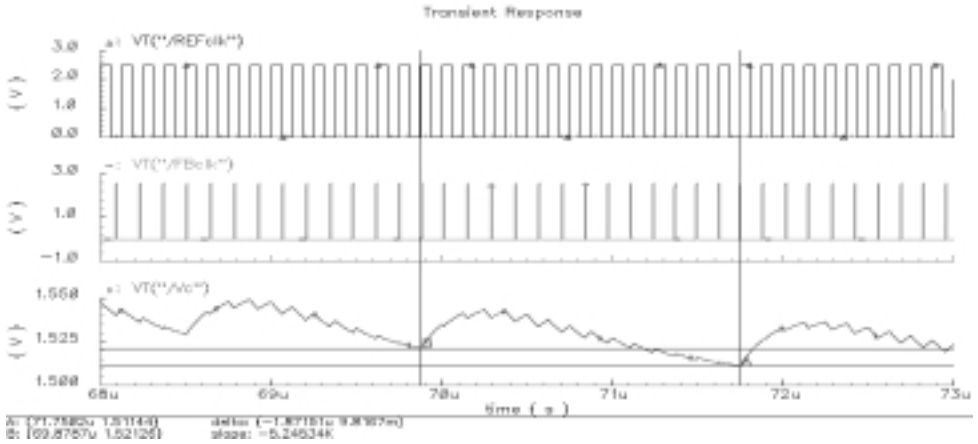


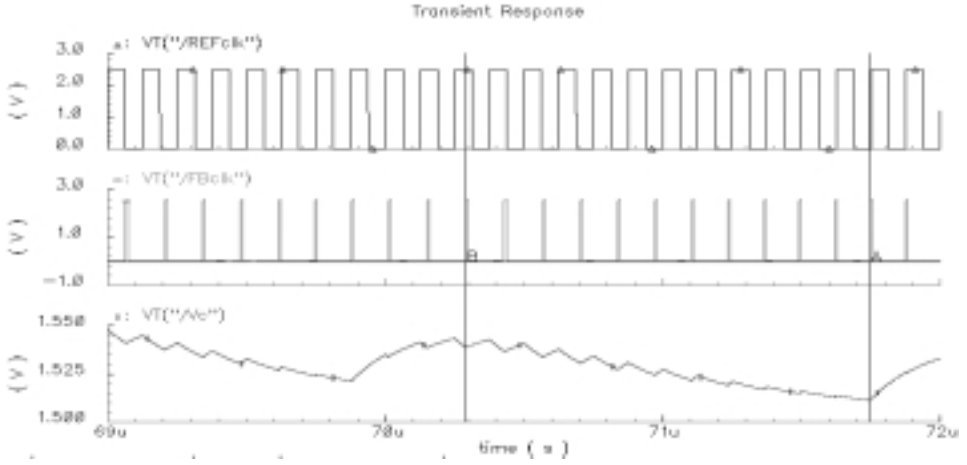
Results for divide ratio N=15

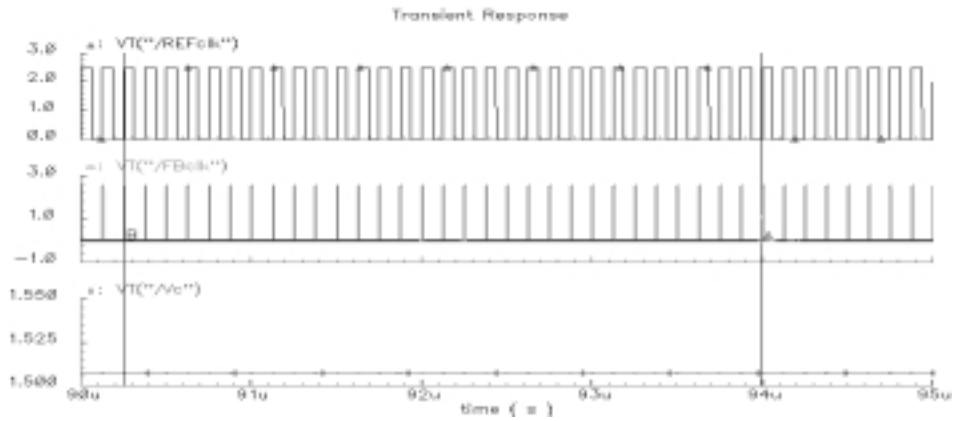


Compared to the case of N=25, the case of N=15 need less time to lock. This is due to different initial condition.


Transient Response
VT("/REFclk")
VT("/FBclk")
VT("/Vc")
4.0
2.0
0.0
3.0
1.0
-1.0
2.20
1.80
1.40
(V)
0.0
3.0u
6.0u
9.0u
12u
15u
time (s)


Transient Response
VT("/REFclk")
VT("/FBclk")
VT("/Vc")
3.0
2.0
1.0
0.0
-1.0
1.550
1.525
1.500
(V)
68u
69u
70u
71u
72u
73u
time (s)


Transient Response
VT("/REFclk")
VT("/FBclk")
VT("/Vc")
3.0
2.0
1.0
0.0
-1.0
1.550
1.525
1.500
(V)
69u
70u
71u
72u
time (s)

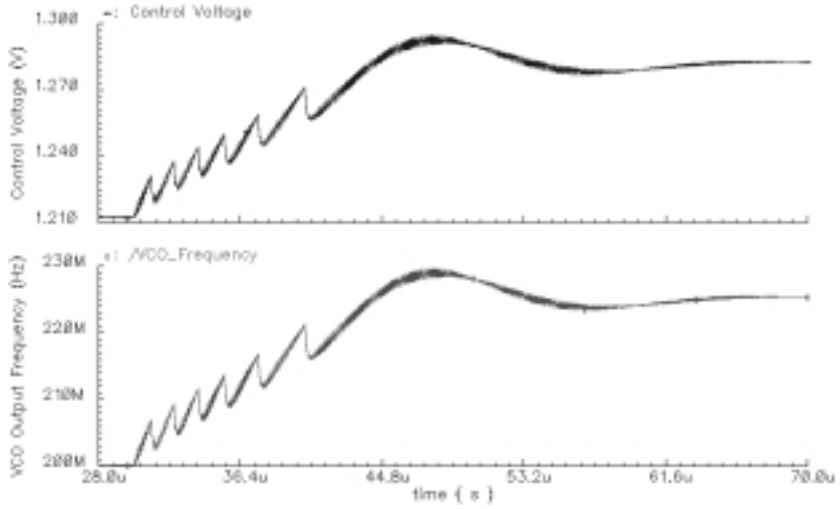


Check the instantaneous frequency of VCO output

In order to observe the instantaneous frequency of VCO output, I built a VerilogA model which transform the instantaneous period of the VCO output to frequency using frequency=1/period. The following are the code I used:

```
////////////////////////////////////
// Instantaneous Frequency Meter ///
////////////////////////////////////

// This verilogA model will output the frequency of the input
// based on the instantaneous period of input clock siganl

// You need to change the value of hlfvcc to match the power supply
// of the clock signal. In this version vcc=2.5 and hlfvcc=1.25
// The time of the first transition is unknown, so it is discarded
// and the frequency always begin at 0 Hz

`include "constants.h"
`include "discipline.h"

nature Frequency
      abstol = 1m;
      access = FF;
      units = "Hz";
       blowup = 1.0e10;
endnature

discipline freq_current
      potential Frequency;
      flow Current;
enddiscipline

module Fremeter(vin,fout);

input vin;
output fout;

freq_current fout;
electrical vin;

  real tlatest;
  real tearly;
  real fout_val;
```

```
real hlfvcc;
integer counter;

analog begin

  @ ( initial_step ) begin
    tearly=0;
    tlatest=1.0;
    hlfvcc=1.25;          //Change your hlfvcc here !
    counter =0;
  end

  @ ( cross ((V(vin)-hlfvcc),+1 )) begin
    tlatest = $realtime;
    fout_val =  1/(tlatest-tearly);
    tearly = tlatest;
    counter = counter +1;
  end
    if (counter == 1)
    FF(fout) <+ 0.0;
    else
    FF(fout) <+ fout_val;
 end

endmodule
```



From this figure we can see that by watching Vc, we do observe the change of VCO output frequency. Damping factor is 0.303.

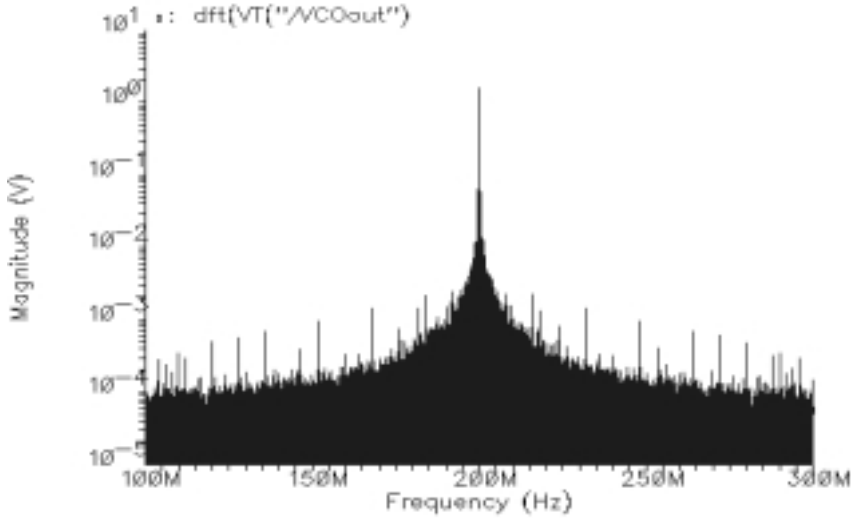
3.4 Noise simulation

3.4.1 Fourier analysis

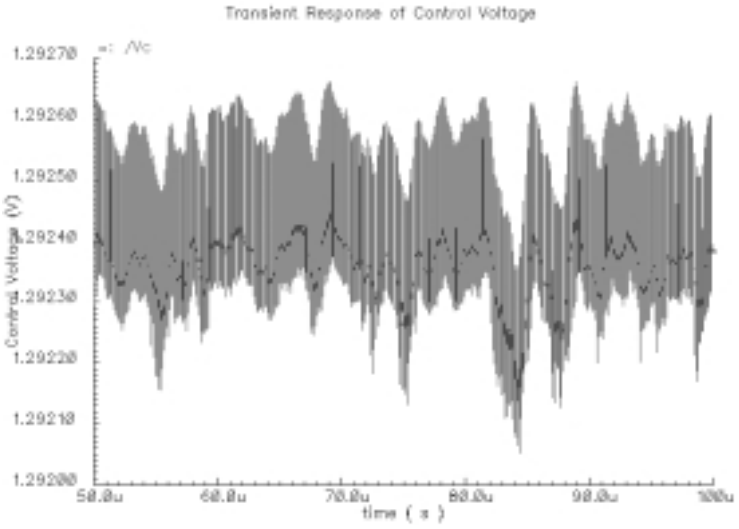
Phase noise is defined as in the following equation [HAJI99]:

$$L\{\Delta\omega\} = 10 \cdot \log\left[\frac{P_{\text{sideband}}(\omega_0 + \Delta\omega, 1\text{Hz})}{P_{\text{carrier}}}\right]$$

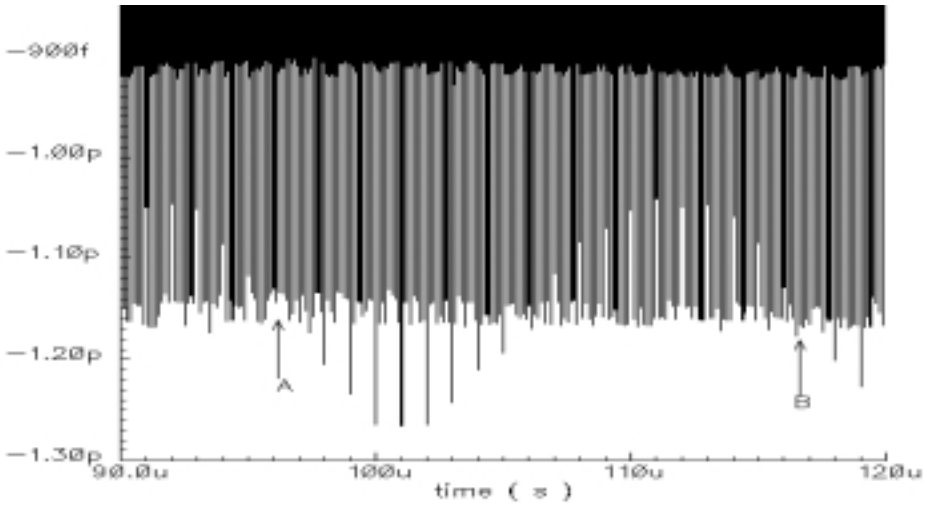
Where $P_{sideband}(\omega_0 + \Delta\omega, 1Hz)$ is the single sideband power at a frequency offset, $\Delta\omega$, from the carrier in a measurement bandwidth of 1 Hz, and $P_{carrier}$ is the total power under the power spectrum. It includes both amplitude and phase fluctuations.



This shows the VCO output spectrum. The input reference is a ideal clock with a frequency of 8MHz. Power supply is 2.5V. The magnitude of the 200MHz signal is 1.57595V and the 192MHz spurious signal is 2.499mV. 20log(2.499mV/1.57595V)= –56dB .



This figure shows the control voltage when the PLL is in lock stage. This figure shows a largest change in Vc is about 0.6mV, the VCO gain is about 360MHz/V, then the frequency is about 200Khz.



This figure shows the time distribution of jitter, which is defined as instantaneous period minus the average of period when the PLL is in lock state. The time from A to B is about 20uS, roughly about the natural frequency.

3.4.2 Cycle-to-cycle jitter simulation

The cycle-to-cycle jitter of a clock signal is defined as the r.m.s. variation (standard deviation) in the period, which can be described by the following equation:

$$\sigma_{CTC} = \sqrt{\frac{1}{M}\sum_{i=1}^{M,M\to\infty}(t_i - t_{avg})^2}$$

Where t_i is the instantaneous period of the signal and t_{avg} is the average period value of the clock signal[1]. Using the so-called short-cut formula we can rewrite the above equation as:

$$\sigma_{CTC} = \sqrt{\frac{1}{M}\sum_{i=1}^{M,M\to\infty} t_i^{\,2} - (t_{avg})^2} = \sqrt{\frac{1}{M}\sum_{i=1}^{M,M\to\infty} t_i^{\,2} - (\frac{1}{M}\sum_{i=1}^{M,M\to\infty} t_i)^2}$$

We use this equation and transformed it into a VerilogA module to calculate the cycle-to-cycle jitter. The VerilogA code is shown in the following:

```
// ////////////////////////////////////////////
// Cycle-to-cycle jitter meter             /////
// Copyright by Fuding Ge, All right reserved //
////////////////////////////////////////////////

// This verilogA model output the rms cycle-to-cycle jitter of the input
// clock signal. The definition of the cycle-to-cycle jitter is the rms
// variation in itd period.
```

[1] Here we divide the sum by M instead of more formal mathematical definition:

$$\sigma_{CTC} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M,M\to\infty}(t_i - \bar{t})^2}$$

```
// JitCTC=sqrt((sum(ti-tavg)**2)/M) where M is the number of periods of test.
// M should be as large as possible. ti is the instantaneous and tavg
// is the average of the periods: tavg=(t1+t2+...+tM)/M

// In this model we use the following equation:
////////////////////////////////////////////
// JitCTC=sqrt((sum(ti)**2)/M-(tavg)**2)     ///
////////////////////////////////////////////

// You need to change the value of hlfvcc to match the power supply
// of the clock signal. In this version vcc=2.5 and hlfvcc=1.25

`include "constants.h"
`include "discipline.h"

nature  Time
        abstol = 1e-25;
        access = TT;
        units = "ps";
        blowup = 1.0e10;
endnature

nature  Number
        abstol = 1e-3;
        access = NUM;
        units = "";
        blowup = 1.0e200;
endnature

discipline time_current
        potential Time;
        flow Current;
enddiscipline

discipline number_current
        potential Number;
        flow Current;
enddiscipline


////////////////////////////////////////////////////////

module Jittermeter(vin,jitter,num_period);
input vin;
output jitter,num_period;

electrical vin;
time_current jitter;
number_current num_period;


   real tlatest;
   real tearly;
   real period_early, period_latest;

   real period_square;
   real period_square_sum;
   real period_square_avg;

   real period_total;
   real period_avg;
   real period_avg_square;

   real jitter_val;
   real hlfvcc;
   integer counter;
   integer counter_begin;

////////////////////////////////////
   analog begin
```

```
      ///////   initialize the parameters///////////////
      @ ( initial_step ) begin
        tearly=0.0;
        tlatest=0.0;
        hlfvcc=1.25;          //Change your hlfvcc here !
        counter =0;
        counter_begin=20000;
        period_early=0.0;
        period_latest=0.0;

        period_square=0.0;
        period_square_sum=0.0;
        period_square_avg=0.0;

        period_total=0.0;
        period_avg=0.0;
        period_avg_square=0.0;
      end


////////////////////////////////////////////////////////////////
      @ ( cross ((V(vin)-hlfvcc),+1 )) begin
         tlatest = $realtime*1e12;
         period_latest  = tlatest-tearly; //Current period value
         tearly = tlatest;
         period_early = period_latest;
         counter = counter +1;

if (counter >= counter_begin+2) begin

         period_square = period_latest*period_latest;
         period_square_sum = period_square_sum + period_square;
         period_square_avg =  period_square_sum/(counter-counter_begin-1);

         period_total = period_total + period_latest;
         period_avg = period_total/(counter-counter_begin-1);
         period_avg_square =  period_avg*period_avg;

         jitter_val = period_square_avg - period_avg_square;
end
end
         ////////////////////////////////////////

      //// output jitter value and number of measure periods /////
         if  (counter-counter_begin < 10000)
            begin
                 TT(jitter) <+ 0;
                 NUM(num_period) <+ 0;
            end
         else
            begin
                TT(jitter) <+ sqrt(jitter_val);
                NUM(num_period) <+ counter-counter_begin;
            end

    end
endmodule
--------------------------------------------------------------------------
```

Herzel [HERZ98] defined the cycle-to-cycle jitter as the variance between successive periods:

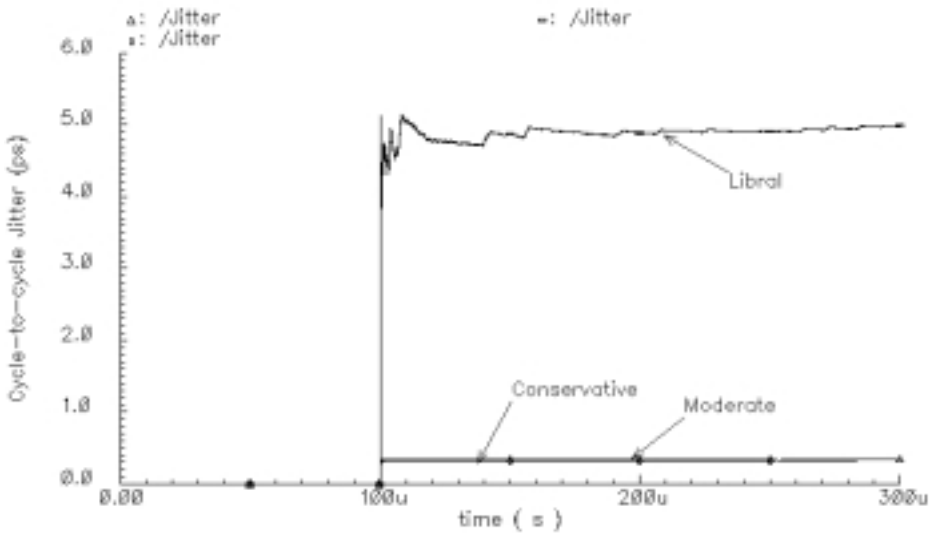
$$\sigma_{\mathrm{CTC},CC} = \sqrt{\frac{1}{M}\sum_{i=1}^{M,M\to\infty}(t_{i+1}-t_i)^2}$$

For white noise source, two successive periods are uncorrelated, and since adjacent cycle jitter represents the difference between two periods, it is twice as large as the variance of one period, so:

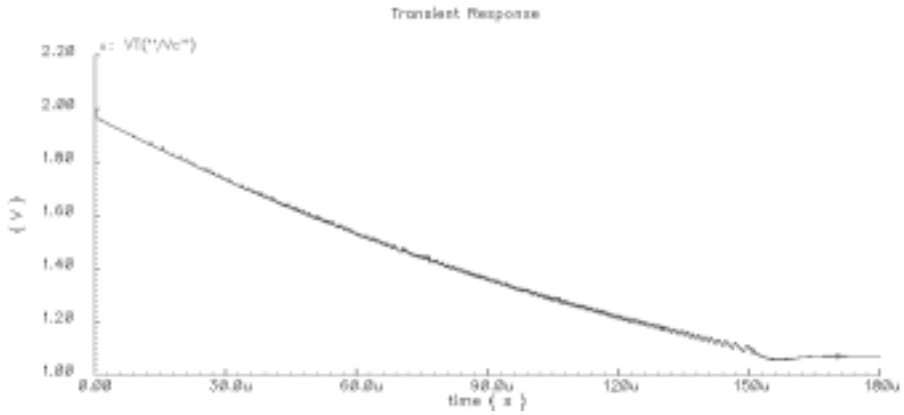
$$\sigma_{CTC,CC} = \sqrt{2}\sigma_{CTC}$$

We can also use this equation to implement jitter simulation.

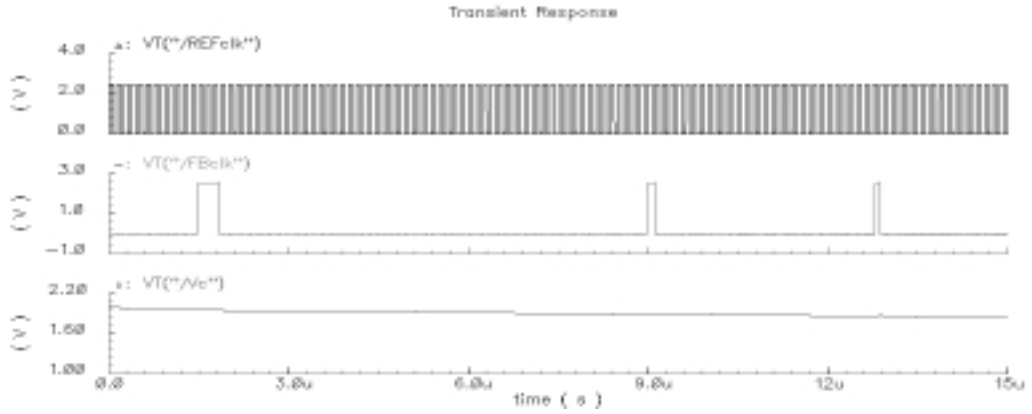
The following figure shows the simulation results of the cycle-to-cycle jitter using Spectre. It can be seen that the accuracy of the transient analysis is very important. If the accuracy is set to be libral, the jitter is about 5 ps, but if the accuracy is set to be conservative or moderate, the jitter is only about 0.3 ps. It can also be seen that there are not much difference between moderate and conservative accuracy.

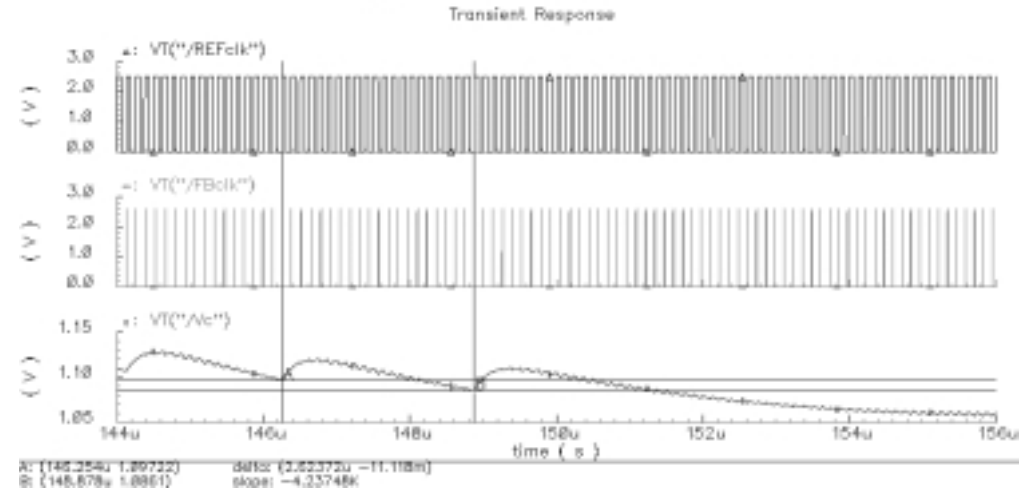


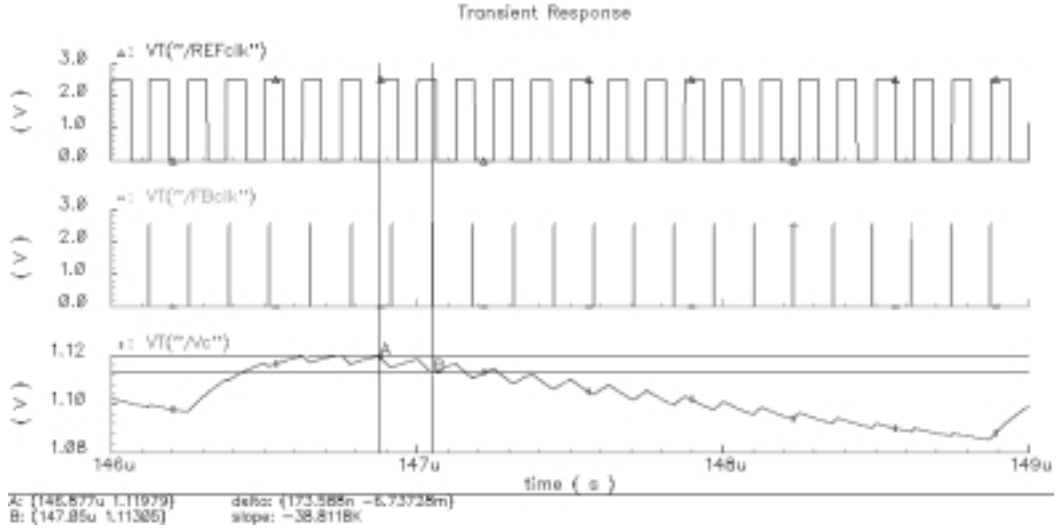
Results for N=35



Vc versus time, Vc stabilized at 1.0698


Transient Response
A: VT("/REFclk")
B: VT("/FBclk")
C: VT("/Vc")
4.0
2.0
0.0
3.0
1.0
-1.0
2.20
1.60
1.00
(V)
0.0
3.0u
6.0u
9.0u
12u
15u
time (s)


Transient Response
A: VT("/REFclk")
B: VT("/FBclk")
C: VT("/Vc")
3.0
2.0
1.0
0.0
1.15
1.10
1.05
(V)
144u
146u
148u
150u
152u
154u
156u
time (s)


Transient Response
A: VT("/REFclk")
B: VT("/FBclk")
C: VT("/Vc")
3.0
2.0
1.0
0.0
1.12
1.10
1.08
(V)
146u
147u
148u
149u
time (s)

Notes about Laplace Transform

Laplace transform:

$$F(s) = \int_0^{\infty} f(t)e^{-st}dt$$

Inverse Laplace transform:

$$f(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}dt$$

Delay in time domain:

$$L\{f(t-\tau)\} = F(s)e^{-s\tau}$$

Differentiation in time domain:

$$L\{\frac{df(t)}{dt}\} = sF(s)$$

Integration in time domain:

$$L\{\int_0^t f(t)dt\} = \frac{F(s)}{s}$$

Initial value theorem:

$$f(t=0) = \lim_{s\to\infty} sF(s)$$

Final value theorem:

$$f(t=\infty) = \lim_{s\to 0} sF(s)$$

MOSFET model used in this project

In this and next projects, we will use the TSMC 0.25μm process. The power supply is about 2.5V. We download the model files from MOSIS web page then modified to support the Cadence Spectre mixed signal simulation tools.
The typical transistor parameters for BSIMV3V are shown in the following. The gate oxide thickness is 57A. The threshold voltage of NMOS is about 0.412V and about –0.58V for PMOS.

```
model nch bsim3v3 type=n
+ tnom        =27                version = 3.1      tox     = 5.7E-9
+ xj      = 1E-7           nch     = 2.3549E17      vth0    = 0.4122189
+ k1      = 0.4555302      k2      = 5.728531E-3    k3      = 1E-3
+ k3b     = 2.9233592      w0      = 1.816641E-7    nlx     = 2.10175E-7
+ dvt0w   = 0              dvt1w   = 0              dvt2w   = 0
+ dvt0    = 0.4608679      dvt1    = 0.5163149      dvt2    = -0.5
+ u0      = 318.8665069    ua      = -9.32302E-10   ub      = 2.162432E-18
+ uc      = 3.868128E-11   vsat    = 1.458598E5     a0      = 1.626434
+ ags     = 0.3003966      b0      = -4.103132E-7   b1      = 5E-6
+ keta    = -4.297464E-4   a1      = 0              a2      = 0.4274964
+ rdsw    = 116            prwg    = 0.5            prwb    = -0.2
+ wr      = 1              wint    = 0              lint    = 1.056485E-8
+ xl      = 3E-8           xw      = -4E-8          dwg     = -1.833849E-8
+ dwb     = 2.423085E-9    voff    = -0.1085255     nfactor = 1.6188811
+ cit     = 0              cdsc    = 2.4E-4         cdscd   = 0
+ cdscb   = 0              eta0    = 4.244913E-3    etab    = 5.001973E-4
+ dsub    = 0.0551518      pclm    = 1.9563343      pdiblc1 = 1
+ pdiblc2 = 7.485749E-3    pdiblcb = -0.0349745     drout   = 0.8869937
+ pscbe1  = 7.999904E10    pscbe2   = 5E-10         pvag    = 0
+ delta   = 0.01           rsh     = 4.4            mobmod  = 1
+ prt     = 0              ute     = -1.5           kt1     = -0.11
+ kt1l    = 0              kt2     = 0.022          ua1     = 4.31E-9
+ ub1     = -7.61E-18      uc1     = -5.6E-11       at      = 3.3E4
+ wl      = 0              wln     = 1              ww      = 0
+ wwn     = 1              wwl     = 0              ll      = 0
+ lln     = 1              lw     = 0               lwn     = 1
+ lwl     = 0              capmod  = 2              xpart   = 0.5
+ cgdo    = 6.08E-10       cgso    = 6.08E-10       cgbo    = 1E-12
+ cj      = 1.758361E-3    pb      = 0.99           mj      = 0.4595413
+ cjsw    = 3.984847E-10   pbsw    = 0.99           mjsw    = 0.3488353
+ cjswg   = 3.29E-10       pbswg   = 0.99           mjswg   = 0.3488353
+ cf      = 0              pvth0   = -0.01          prdsw   = -10
+ pk2     = 2.330213E-3    wketa   = 7.327459E-3    lketa   = -6.323718E-3

model pch bsim3v3 type=p
+ tnom    = 27             version = 3.1           tox     = 5.7E-9
+ xj      = 1E-7           nch     = 4.1589E17      vth0    = -0.5806205
+ k1      = 0.6160845      k2      = 6.878321E-3    k3      = 0
+ k3b     = 12.103096      w0      = 1E-6           nlx     = 1E-9
+ dvt0w   = 0              dvt1w   = 0              dvt2w   = 0
+ dvt0    = 3.0107968      dvt1    = 0.7196372      dvt2    = -0.1087959
+ u0      = 113.5280395    ua      = 1.428339E-9    ub      = 1E-21
+ uc      = -1E-10         vsat    = 2E5            a0      = 0.8190934
+ ags     = 0.1198761      b0      = 1.300549E-6    b1      = 5E-6
+ keta    = 0.0159086      a1      = 7.587589E-4    a2      = 0.5038746
+ rdsw    = 826.8966468    prwg    = 0.3121607      prwb    = -0.335727
+ wr      = 1              wint    = 0              lint    = 3.708011E-8
+ xl      = 3E-8           xw      = -4E-8          dwg     = -4.149242E-8
+ dwb     = 5.696148E-9    voff    = -0.127106      nfactor = 1.154513
+ cit     = 0              cdsc    = 2.4E-4         cdscd   = 0
+ cdscb   = 0              eta0    = 0.8298199      etab    = -0.3400479
+ dsub   = 1.2581086       pclm    = 1.175511       pdiblc1 = 6.170092E-3
+ pdiblc2 = -3.886687E-9   pdiblcb = -1E-3          drout   = 0.0670916
```

```
+ pscbe1  = 7.623894E9     pscbe2  = 1.944679E-9    pvag    = 0.6564436
+ delta   = 0.01           rsh     = 3.4            mobmod  = 1
+ prt     = 0              ute     = -1.5           kt1     = -0.11
+ kt1l    = 0              kt2     = 0.022          ua1     = 4.31E-9
+ ub1     = -7.61E-18      uc1     = -5.6E-11       at      = 3.3E4
+ wl      = 0              wln     = 1              ww      = 0
+ wwn     = 1              wwl     = 0              ll      = 0
+ lln     = 1              lw      = 0              lwn     = 1
+ lwl     = 0              capmod  = 2              xpart   = 0.5
+ cgdo    = 6.51E-10       cgso    = 6.51E-10       cgbo    = 1E-12
+ cj     = 1.893794E-3     pb      = 0.99           mj      = 0.4672189
+ cjsw    = 3.253221E-10   pbsw   = 0.5596564      mjsw    = 0.2788539
+ cjswg   = 2.5E-10        pbswg   = 0.5596564      mjswg   = 0.2788539
+ cf      = 0              pvth0   = 5.588398E-3    prdsw   = 3.2881861
+ pk2     = 3.613465E-3    wketa   = 0.0252565      lketa   = -0.0108543
```

Reference

[GARD80] F.M. Gardner, “Charge-pump phase-lock loops”, IEEE Trans. Comm., Vol.COM-28, pp.1849-1858, Nov 1980.

[HERZ99] Frank Herzl and Behzad Razavi, “A study of oscillator jitter due to supply and substrate noise”, IEEE Trans. Circuit and system – II, Vol46 (1), PP56-62, 1999.

[WOLA91] D.H. Wolaver, “Phase-locked loop circuit design”, Prentice Hall, Englewood Cliffs, NJ, 1991.

[MANE96] J. G. Maneatis, “Low jitter process-independent DLL and PLL based on self-biased techniques”, IEEE J. SSC, Vol. 31, No.11, pp1723-1732, Nov, 1996

[LARS96] P. Larsson, “High-speed architecture for a programmable frequency divider and a dual-modulus prescaler”, IEEE J. SSC, Vol. 31, No.5, pp744-748, May, 1996

[BEST97] R.E. Best, “Phase-locked loops, design, simulation and applications”, McGraw-Hill, New York, 1997, 3rd Edition

[TI99] Texas Instruments, “Fractional/integer-N PLL basics”, SWRA029, edited by Curtis Barrett, wireless communication business unit, 1999

[HAJI99] Ali Hajimiri, Thomas H. Lee, “The design of low noise oscillators”, Kluwer Academic, Boston, 1999