Frequency Synthesizer for a ZigBee Transceiver

A Project Report

submitted by

ARUN M

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

MAY 2007

THESIS CERTIFICATE

This is to certify that the thesis titled **Frequency Synthesizer for a ZigBee Transceiver**, submitted by **Arun M**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Nagendra Krishnapura
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Prof. Shanthi Pavan
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai
Date: 7th May 2007

ACKNOWLEDGEMENTS

I am extremely grateful to Dr. Nagendra Krishnapura and Dr. Shanthi Pavan for guiding me through the whole of the project. The weekly meetings with them proved very useful. I should also mention the fact that it was the 'Analog IC Design' class of Dr. Nagendra which gravitated me towards analog design.

I would like to thank the other members involved in the design of the Zigbee transceiver, Raja, Sagar, Kapil and Sriganesh for all the discussions we had, technical and otherwise. I also reserve special thanks to Reddy, IR, Murali and Laxmi Nidhi for helping me out with the Cadence software. Reddy also helped me a lot with layouts. I would also like to thank all the faculty and staff of Electrical dept in IIT Madras for making my stay here, a very enjoyable one.

Finally, I would like to thank IC&SR for sponsoring the 'Zigbee Transceiver' project.

ABSTRACT

'Zigbee' is a standard that was developed as a guideline to designing ultra low-power trasceivers working at low data rates. Wireless devices developed with this standard will be very low cost and have a very long battery life.

The current effort is a part of a bigger project aimed at developing a fully functional single chip RF transceiver with as low off-chip components as possible and subscribing to the 'Zigbee' standard. This work delves into the design of a very important component of any wireless transceiver, the local oscillator. An integer-N PLL was designed to be used as the local oscillator for both transmitter and receiver. The standard required that the PLL be programmable to 16 channels in the 2.4-2.5 GHz range. Throughout the whole design cycle, the emphasis was on minimizing power consumption and chip area as much as possible. This design was able to meet all the required specifications laid down by 'Zigbee', while maintaining a very low power consumption and without any off-chip components.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

PLL	Phase Locked Loop
LO	Local Oscillator
VCO	Voltage Controlled Oscillator
PD	Phase Detector
PFD	Phase Frequency Detector
CP	Charge Pump
LPF	Low Pass Filter
NF	Noise Figure

CHAPTER 1

INTRODUCTION

1.1 PLL as a frequency synthesizer

This thesis report discusses the design and implementation of a low-power phase locked loop frequency synthesizer to be used as the local oscillator in a wireless transceiver operating in the frequency range of 2.4-2.5 GHz.
Obviously, an open loop oscillator is not a suitable candidate for being used in a local oscillator because the frequency cannot be controlled precisely. The other possibility that one can think of is a fully digital frequency synthesizer whose frequency can be accurately controlled by digital logic. But this option is also not viable since the power consumption at high frequencies becomes prohibitively large. A phase locked loop(PLL) comes in very handy when we need to control the frequency accurately. A PLL is a negative feedback system with a low-frequency crystal oscillator at the input and a frequency divider circuit in feedback. As in any negative feedback system working with a large enough loop gain, we can expect the gain of the system to be the inverse of the feedback factor. So, the output frequency will be the input signal multiplied in frequency by the divide factor of the feedback divider. By changing the divide value, we can accurately control the output frequency.

1.2 Emphasis

The synthesizer is meant to be used in a low-power transceiver. So, the emphasis of the design will be on minimizing the power consumption as much as possible. The focus will be to try and meet the required specifications with as low a power as possible.

1.3 Organisation of the thesis

Chapter 2 introduces the intended application of the transceiver and also gives a block diagram level introduction to the working of the PLL.
Chapters 3 through 5 deal with the design and implentaion of the various modules inside the PLL.
Specifically, chapter 3 deals with the design of the Voltage Controlled Oscillator(VCO).
Chapter 4 deals with the design of the Programmable Frequency Divider
Chapter 5 deals with the PLL loop analysis followed by the design of the low frequency blocks, namely, the Phase Frequency Detector(PFD), Charge Pump(CP) and filter and concludes with noise analysis.
Finally, chapter 6 shows the layouts of the various components and summarizes the simulation results and power consumption breakup.

CHAPTER 2

ZIGBEE STANDARD AND PLL INTRODUCTION

2.1 Zigbee standard

'Zigbee' (officially 'IEEE 802.15.4') was a standard developed for building low-data rate wireless transceivers with emphasis on very low power consumption. It was developed as a wireless standard for Wireless-Personal Area Networks working at data rates upto 250 kbps. Typical operational range of target appliances is around 10 metres. It can be easily incorporated into devices used in home automation, PC peripherals and health care appliances.

2.2 System architecture

The prime concern in the transceiver design is low cost and low power. So, it is important to minimize the use of off-chip components as much as possible. We decided that a Direct Conversion architecture (where the local oscillator and RF signal are the same frequency so that the signal gets transformed directly to the baseband) was best suited for this application. Figs. 2.1 and 2.2 show the transmitter and receiver architectures respectively.

2.3 Relevant specifications

The modulation used here is O-QPSK with sine pulse shaping. Transmit data rate is 250 kbps and a spread factor of 1:8 is applied, so the chip rate going in is actually 2 Mcps. The channel bandwidth is approx. 2 MHz. The band extends

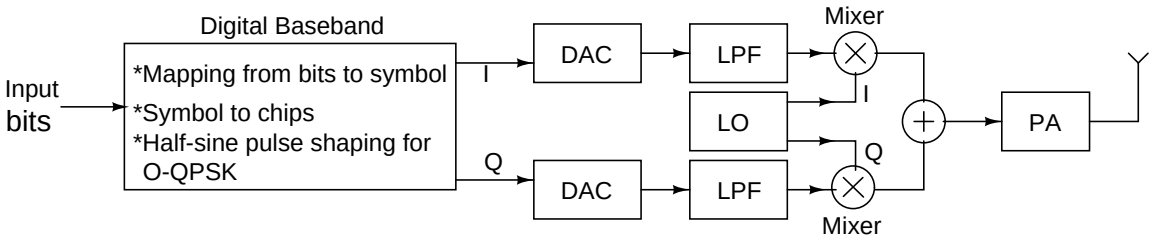

Figure 2.1: Zigbee transmitter

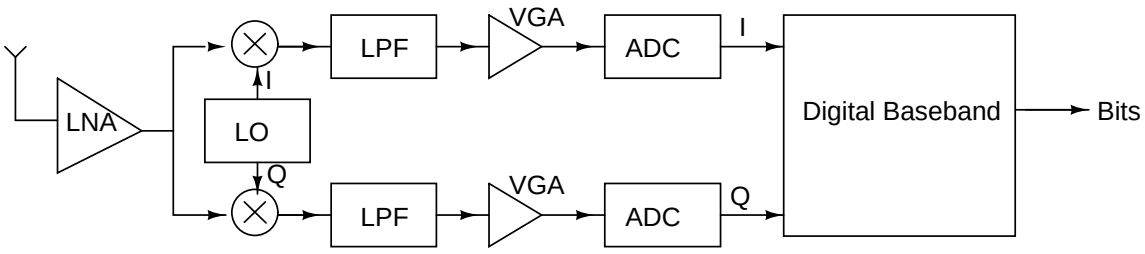

Figure 2.2: Zigbee receiver

from 2.405 GHz to 2.480 GHz with 16 channels spaced at 5 MHz each. Zigbee specifies a packet error rate of less than 1% which equates to a bit error rate of 0.00625%. This translates to an output SNR requirement of 0 dB after considering a spreading gain of 9 dB. This, coupled with an input sensitivity requirement of -85 dBm results in a receiver noise figure requirement of around 20 dB.

As far as the PLL is concerned, the relevant specifications are:

- the settling time must be less than 192 μs
- the final settling value must have an accuracy of $\pm$40 ppm. Here, with a center frequency of around 2.5 GHz, it translates to a frequency deviation of $\pm$100 kHz
- phase noise must be below -92 dBc/Hz at an offset of 3.5 MHz from the carrier
- additional constraints posed by the trasmit mask in the band and receiver noise considerations

2.4 PLL block diagram

The PLL, in simple terms, is a block that locks the frequency and phase of a locally generated signal to a reference signal. It is a negative feedback loop which tends to make the phase error between the actual and the reference signal zero. By putting a programmable frequency divider in the feedback loop, we can generate any frequency which is a multiple of the reference frequency. We decided to use an integer-N PLL as opposed to a fractional-N PLL('N' denotes the divide value used in the programmable divider), the reasons being its simplicity and the fact that the specifications here are not too tight to warrant the use of fractional-N PLL. The basic blocks in an integer-N PLL are the Voltage Controlled Oscillator(VCO), programmable divider, phase detector and low pass filter. The design and analysis of all these blocks are given in the subsequent chapters.

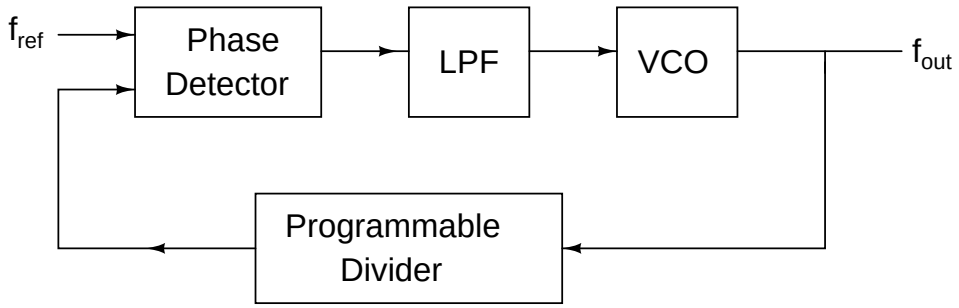

Figure 2.3: PLL block diagram

CHAPTER 3

VOLTAGE CONTROLLED OSCILLATOR

3.1 Introduction

The voltage controlled oscillator(VCO) is generally the most difficult component to design in the PLL since it works at the highest frequency. It basically performs the function of voltage to frequency conversion in the PLL. The VCO consumes a major chunk of the total PLL power since it operates at the highest frequency. So, its design must be given due importance. The factors governing VCO design are the frequency tuning range, phase noise limit, power and area occupied.

3.2 Choice of VCO

The two widely used VCOs are ring and LC based oscillators. LC based oscillators have inherently lower phase noise due to their resonant nature. They also have a lower sensitivity to supply and gnd. So, we have decided to use LC oscillator for this design.

Zigbee uses offset-QPSK as its modulation scheme,which needs both in-phase(I) and quadrature(Q) components. Hence, the choice of VCO is influenced by the ability to produce I and Q signals. There are three ways by which one could think of generating I and Q signals. They are

- VCO followed by polyphase ($+45°$, $-45°$) filters
- VCO running at double the required frequency followed by a Master-Slave flipflop which will divide the frequency by two and also provide the I and Q outputs
- Two cross-coupled VCOs operating in quadrature

The first option uses RC, CR filters to generate the quadrature signals. But we cannot guarantee exact quadrature due to the tolerances of R and C. Also, it needs output buffers which take up a lot of power[1].
The second option requires the VCO to be running at twice the required frequency, but is quite simple to implement and it gives good quadrature. Also, one major advantage of using this configuration is that the dividers acts as buffer between the mixer and VCO, hence isolating the VCO from noise from the mixer. Also, the fact that it runs at twice the frequency prevents pulling by the transmitter; all noise leaking from transmitter will get rejected by the oscillator resonant circuit. One issue of concern though, is that the flip-flops have to run at 5 GHz which can potentially consume a lot of power.
The third option is better as far a power consumption is concerned, but it comes at the cost of twice the VCO area.
We have chosen option two since it gives a good compromise between area and power and also good isolation of the VCO.

3.3 LC oscillator - principle

An LC parallel circuit without any resistance produces sustained oscillations. But a practical LC circuit will have an effective parallel resistance, which will damp the oscillations. So, if we can somehow generate a negative resistance to cancel out the parasitic resistance of the tank, then the ciruit will oscillate.

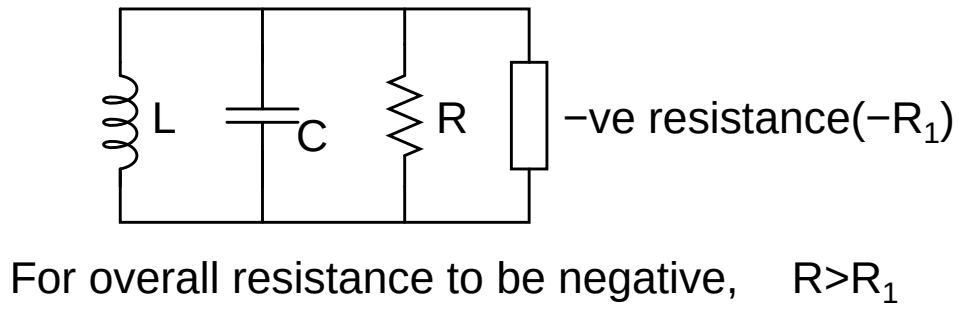

Figure 3.1: LC oscillator model

3.4 LC Oscillator - cross coupled differential pair

One way of generating negative resistance is by using a cross-coupled differential pair. A cross-coupled diff pair with transconductance g_m will result in a differential resistance of $-\frac{2}{g_m}$ which can be split as $-\frac{1}{g_m}$ on each side. So, if the LC tank has an effective parallel resistance of R_p on each side, then the condition for oscillation is $g_m > \frac{1}{R_p}$ (the overall resistance must be negative)

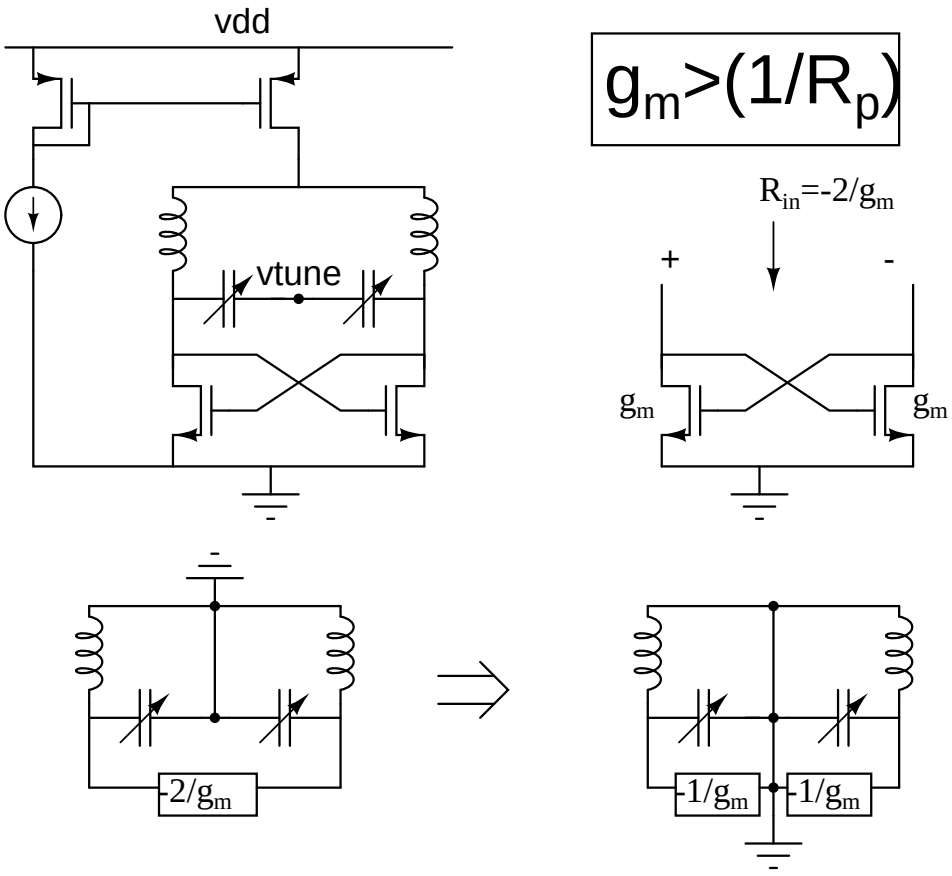

Figure 3.2: LC oscillator with cross-coupled pair

3.5 Amplitude and frequency of oscillation

Frequency of oscillation $= \frac{1}{\sqrt{LC}}$

The amplitude of oscillation can be estimated by knowing the amount of non-linearity in the circuit. The transconductance of the diff pair is non-linear and

because of this, there is a limit to how much the amplitude can be. When the oscillation starts, $g_m > \frac{1}{R_p}$ and hence the amplitude starts to increase, but as it increases, g_m starts to decrease due to the non-linearity and the amplitude settles at a point where $g_m = \frac{1}{R_p}$. But in our context, we can calculate the amplitude of oscillation easily by using the fact that the current in the diff pair switches completely every half cycle. So, the amplitude of oscillation will just be the first harmonic component of the square wave of amplitude $\pm I_0 \times R_p$ where I_0 is the tail current of the diff pair.
So, amplitude of oscillation $= \frac{4}{\pi} I_0 R_p$ (V_{p-p},single-ended)

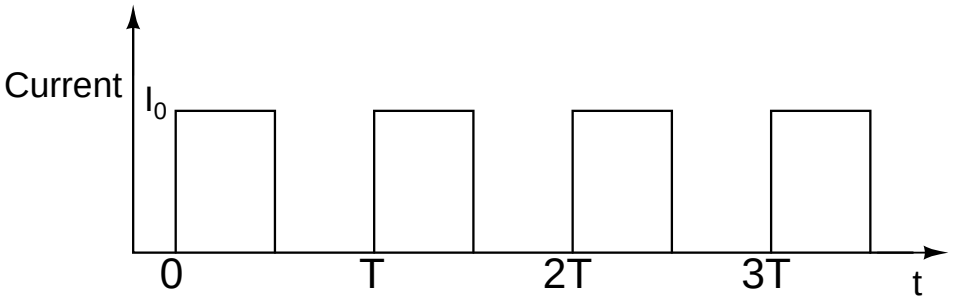

Figure 3.3: Current in one arm of the diff pair

3.6 VCO design

Design of the ocillator boils down to choosing the appropriate values of tail current, g_m of diff pair, values of L(and its quality factor) and C. The resonant frequency fixes the LC product. Generally, we don't have a lot of freedom in choosing the values of L and C independently due to the following factors:

- the parasitic capacitances of the diff pair, inductor, load and the required tuning range required set a lower limit on the value of the capacitance
- practical values of on-chip inductances with a sufficiently high Q are not more than around 10nH

It can be shown that using as high an inductor value as possible(while maximising Q) results in the best power efficiency and phase noise [1].
The absolute lower limit on g_m of the diff pair is $\frac{1}{R_p}$ where $R_p = Q^2 R_s$ and

$Q = \frac{L\omega_0}{R}$ (if the capacitance Q is neglected). The amplitude of oscillation needed sets the lower limit on the tail current.

Next part of the design is choosing whether to go for an nMOS or pMOS diff pair or a combination of both. There are implementations (eg. [2]) which use both nMOS and pMOS, which gives a higher effective g_m, but leads to higher parasitic caps, thereby restricting the tuning range (we may run out of capacticance required for tuning). This design uses an nMOS diff pair with a current source on top since it was found to give the lowest phase noise.

3.6.1 Inductor Modelling

A symmetric square inductor was used since it saves a lot of area compared to two separate inductors. This process comes with the thick top metal option(to decrease series resistance), so the inductor was laid out in this layer(metal 6). Modelling was done using ASITIC[3] and FASTHENRY[4]. An inductance value of 4nH and Q of 8 were obtained using a 140 μm spiral. The ASITIC estimation of the parasitic capacitance was less accurate. So, we extracted the capacitance using ASSURA and used a distributed model of the inductor(R and L in the series arm and C in the shunt arm). We found that five segments were enough and distributed the total inductance, resistance and capacitance equally among the five segments. A series resistance of 16 Ω was used for simulations (which is slightly higher than the series resistance that was obtained from ASITIC), to be on the safer side.

Inductor Dimensions:

length: 140 μm, Turns: 6, width: 6 μm, spacing(between turns): 2 μm

We fabricated a test chip with this inductor and the test results conformed to what the model predicted.

3.6.2 Varactor Modelling

We decided to go for a MOS capacitor working in the accumulation and depletion regions (also known as accumulation mode varactor) over the inversion mode

varactor because of the following reasons:

- the C-V curve is more linear here
- n^+ on n-well gives us the option of decoupling substrate noise from the varactor and it also gives lesser series resistance since the channel is made of electrons

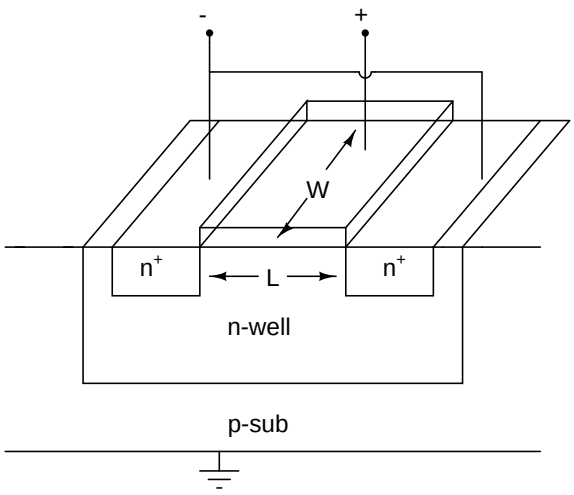

Figure 3.4: Accumulation Varactor

The accumulation varactor was modelled in Verilog-A using the basic channel charge-voltage equations. The model used is given in Appendix- A.

In the actual oscillator circuit, the gate was tied to the oscillating node while the tuning voltage was applied to the body (n-well). The average value of the gate bias was around 0.65 V. So, a variation of body voltage from 0 V to 1.8 V translated to a variation of V_{gb} from 0.65 to -1.15 V. The capacitor size was chosen to be W=1.44 μm, L=0.26 μm and NF=36. From the C-V plot, we see that this corresponds to a capacitor variation of around 110 fF to 42 fF. This gave a tuning range of around 400 MHz (required range is 200 MHz)

Parasitic resistance

There are two parasitic resistances associated with the varactor, the gate resistance and the channel resistance. The gate resistance is proportional to $\frac{W}{L}$ and the channel resistance is proportional to $\frac{L}{W}$. So, we have to make sure that we choose

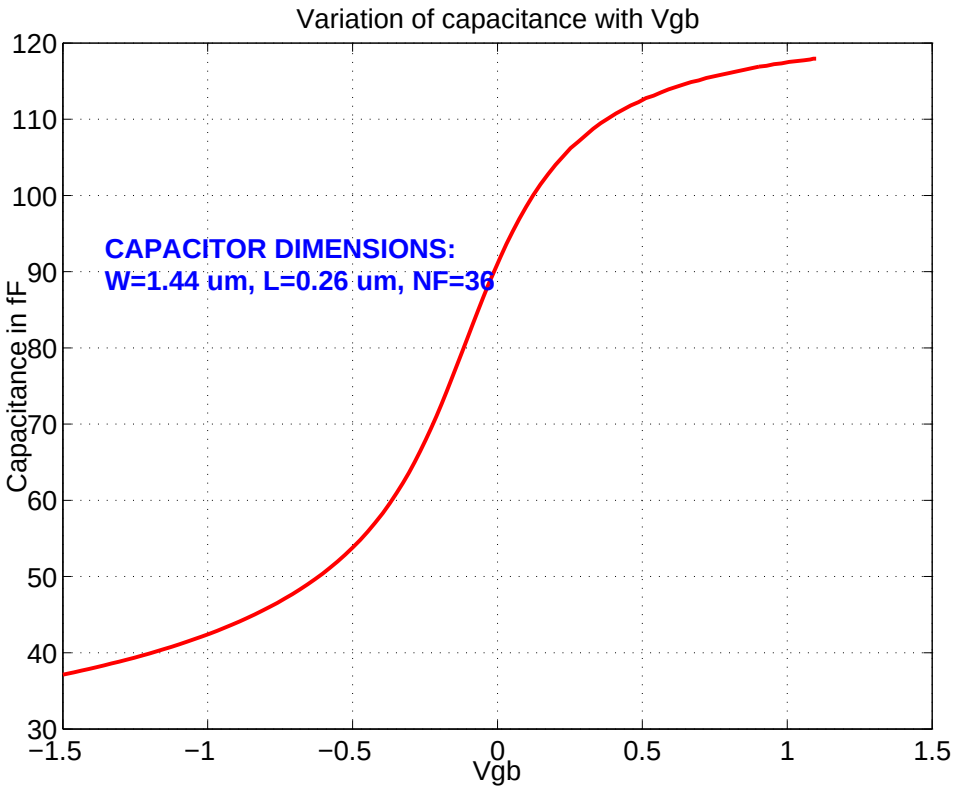

Figure 3.5: C-V curve

the appropriate values of W and L (with the constraint that WL=const) so that neither one completely dominates the other. We chose $W_{total} = 48.96\,\mu$m(NF=34) and $L = 0.26\,\mu$m. This gave a Q for the capacitor high enough that the parasitic resistance of the varactor could be neglected when compared to the inductor parasitic.

Layout of varactor

We chose a differential layout(Fig. 3.7) for the varactor as suggested in [5]. Here, the gates of the two capacitors are interleaved so that they see differential signals. This creates ac grounds at the mid-points of the n^+ diffusions. It offers the following advantages over the traditional layout:

- increase in layout density because we can now take n^+ contacts just at the two ends and those in the middle can be left unconnected
- The series resistance decreases because of lesser contacts(shorter gaps) and hence the cap has a better Q

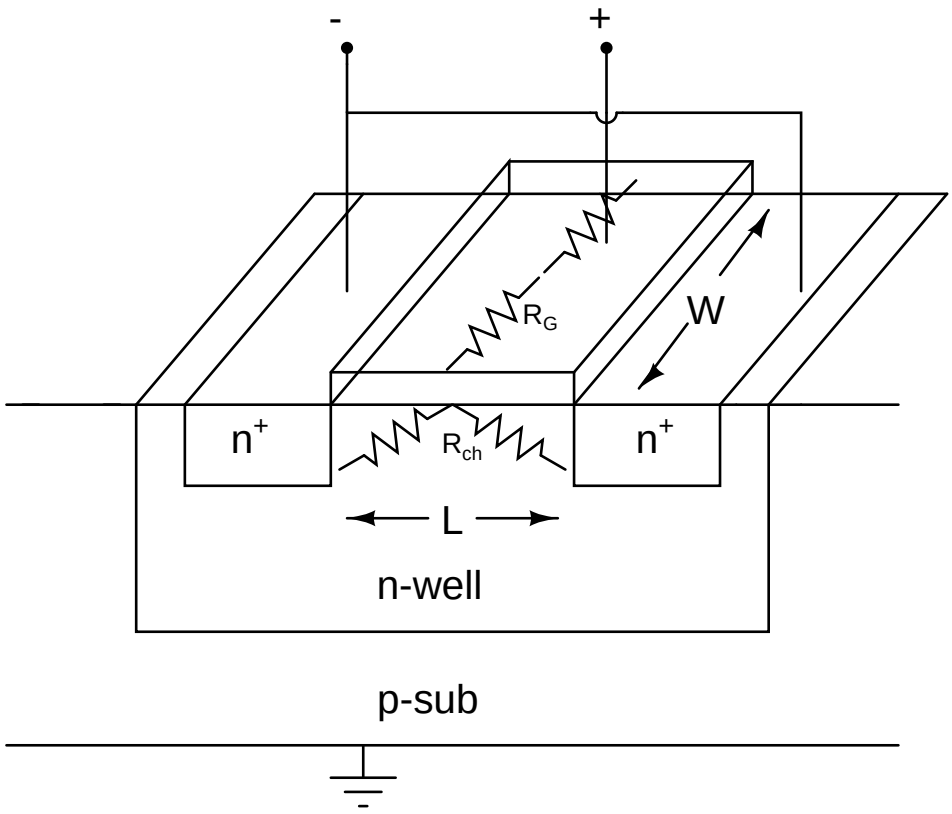

Figure 3.6: Parasitic resistances

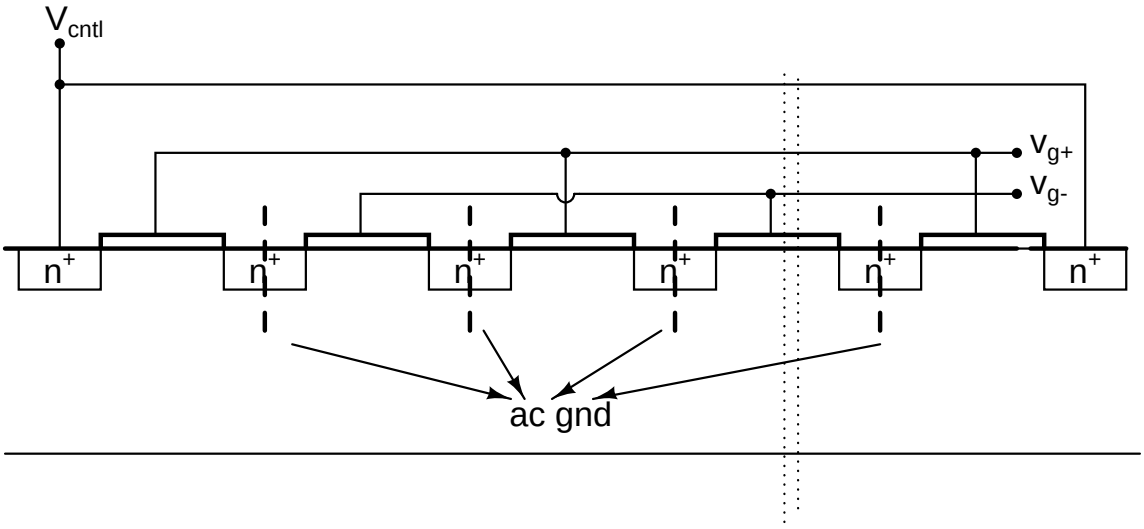

Figure 3.7: Differential layout of Varactor

3.7 VCO-Schematic

Finally, coming to the complete VCO schematic, the only values that need to be fixed are the tail current value and the sizes of the main differential pair.

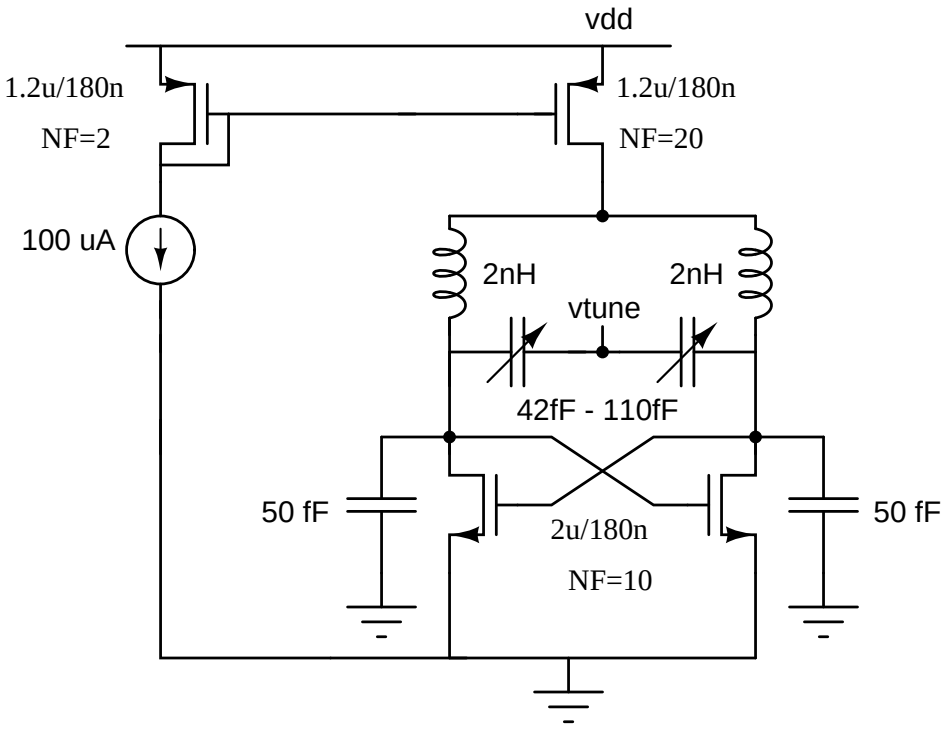

Figure 3.8: VCO schematic

A series resistance of $8\,\Omega$ in each inductor leads to an effective parallel resistance of $500\,\Omega$ (approx). A tail current of $1\,\text{mA}$ was sufficient to get a single-ended peak-to-peak swing of around $700\,\text{mV}$

The diff pair size used was $\frac{20\,\mu\text{m}}{0.18\,\mu\text{m}}$. This gave a g_m of $5\,\text{mS}$ so that $g_m R_p$ turns out to be 2.5

An extra 50 fF MIMCAP was finally added to the capacitance to get the frequency range to the required value after taking into account all the parasitic caps of VCO, load and the wiring parasitics.

CHAPTER 4

DIVIDER CHAIN

4.1 Introduction

The divider chain follows the VCO and divides the 5 GHz signal from VCO right down to 5 MHz which is the reference frequency for the PLL. First comes the 5 GHz to 2.5 GHz divider which feeds the mixer. It is a Master-Slave configuration which produces I and Q outputs. This is followed by the programmable divider chain which can be programmed to any divide value from 481 to 496.

4.2 First Divider(5 GHz to 2.5 GHz)

This is the most power hungry divider because of the fact that it operates at the highest frequency and also because it sees the Mixer load in addition to the subsequent divider. CML divider was used here because it is more power efficient at this high frequency when compared to CMOS.

The basic structure is a Master-Slave flipflop, where the Q_{bar} output of the Slave is fed to the input of the Master. The Master and Slave outputs then are in quadrature. Figs. 4.1 and 4.2 show the divider configuration and the waveforms respectively.

The other divider configuration which works very well at high frequencies is the Injection Locked Frequency Divider(ILFD) [6]. An implementation of ILFD divider to generate quadrature signals is given in [7]. In principle, it is just an oscillator circuit which is driven at an appropriate node by the input signal. The oscillator's resonant frequency is set close to half the frequency of the input signal. The fact that the ciruit is driven makes it oscillate exactly at half the input signal frequency and the other harmonics are rejected by the resonant circuit. The circuit naturally has a very high gain at this frequency and hence frequency division

can be realized at low power. But we didn't go for this circuit since it required two inductors which needed to be accurately characterised and also will take up a lot of area.

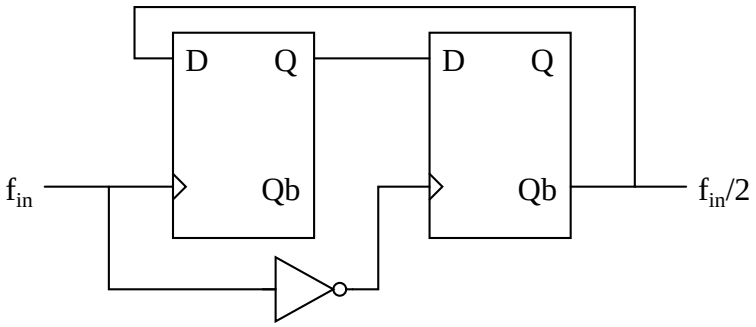

Figure 4.1: Divide by 2 Structure

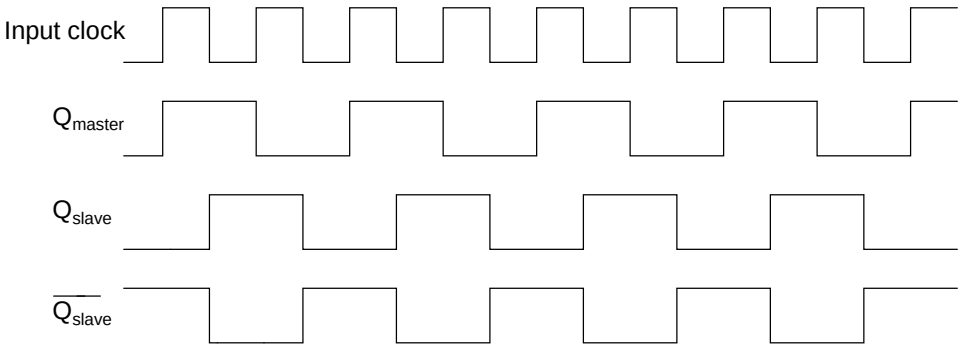

Figure 4.2: Master and Slave waveforms

4.2.1 Divider Design

The basic CML latch is shown in Fig 4.3. The main differential pair switches the current depending on the input when clock goes high and the cross-coupled differential pair reinforces the outputs when the clock is low. This circuit is transparent only when the clock is high.

We can build the Master-Slave flipflop by using two such latches with complementary clock inputs. The basic trade-off in this circuit is between speed and current. For a particular output swing, the product of I_{tail} and load resistance is fixed. If we decrease I_{tail} and increase the load resistance, the bandwidth (given by $\frac{1}{RC}$) decreases, thus decreasing the maximum speed of operation.

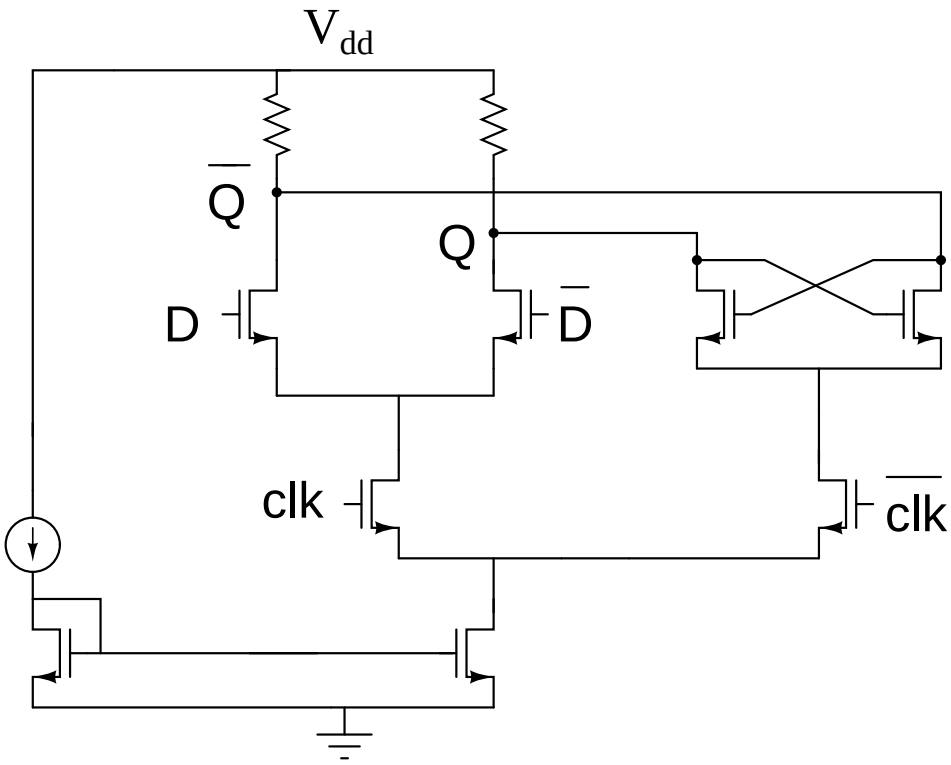

Figure 4.3: Basic CML Latch

One clever way of increasing the bandwidth of such circuits is by using an inductor in series with R. Increasing the value of this inductance will increase the bandwidth and for a particular inductance, the response will become maximally flat. If the inductance is increased further, the response starts to peak. So, the inductance value can be chosen to be around the value where the response becomes maximally flat. It can be derived that the value of inductance which gives a maximally flat response is $L = 0.414CR^2$. To get a ballpark value of L, let us assume a resistance of 400 Ω and a capacitance of 150 fF. This gives an inductance of 10 nH. We need four such inductors for the divider. But the Q of the inductors can be small since the series resistance can be incorporated into the load resistance. Passive, on-chip inductors were unsuitable for this because, firstly, they took a lot of area and also, the parasitic capacitance associated with the inductor turned out to be way too high that the inductor was not giving bandwidth improvement at all when used in the circuit. So, we went for an active-inductor realization with a single MOS transistor.

Active Inductor Design

A simple way of realizing an active inductor is shown in Fig 4.4

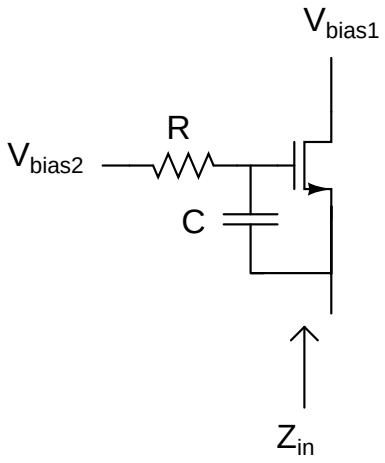

Figure 4.4: Active inductor realization

The small signal equivalent circuit is given in Fig 4.5

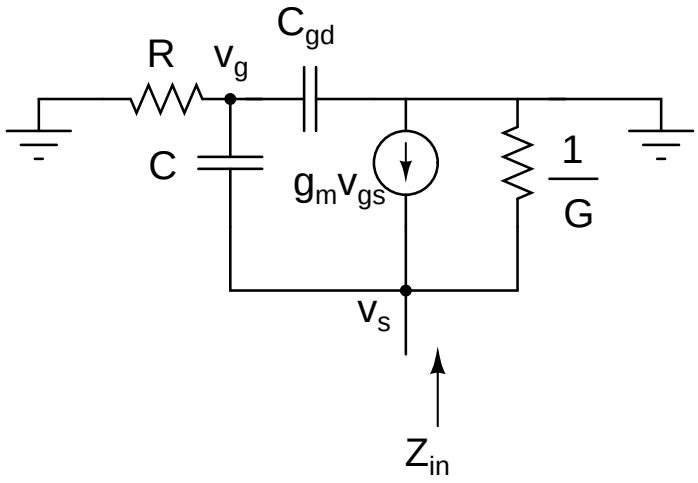

Figure 4.5: Small signal equivalent circuit

Neglecting C_{gd}, we get the input impedance as

$$Z_{in} = \frac{1 + sCR}{(g_m + G) + sC\,(GR + 1)} \tag{4.1}$$

This corresponds to a dc impedance of $\frac{1}{g_m + G}$, a zero at $\frac{1}{RC}$ and a pole at $\frac{g_m + G}{C(GR+1)}$. Now, if we chose R, g_m, C and G such that the zero comes first and the pole next, we have an inductor for the frequency range between the zero and the pole. The inductance value is $\frac{RC}{g_m + G}$. Also, we can increase the pole frequency by increasing g_m, thereby increasing the usable frequency range of the inductor. Fig. 4.6 shows the bode plot of Z_{in}

But we have to account for the fact that C_{gd} may not be neglected if C itself is

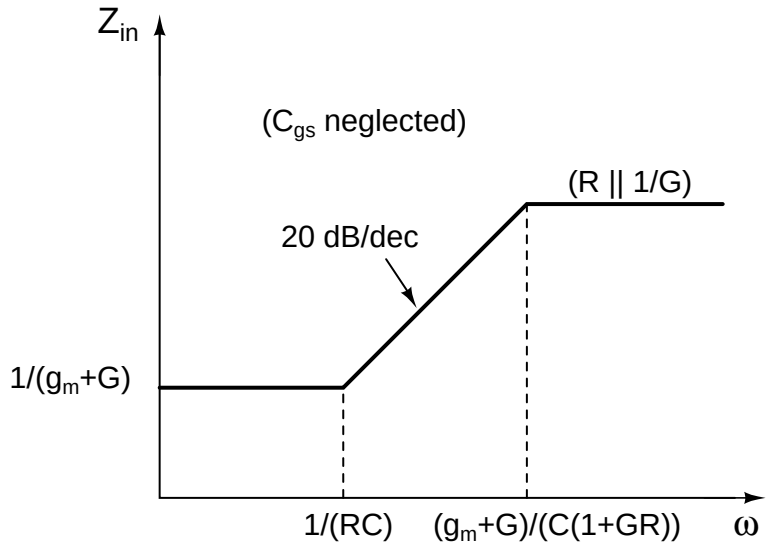

Figure 4.6: Bode plot of Z_{in}

small. A non-zero C_{gd} introduces one more pole in the circuit. So, when we do a practical inductor out of this, we cannot have an arbitrarily large frequency range and we can have a good inductor only in a small band of frequencies. Even within the bandwidth, it will start to behave lesser and lesser as an inductor as frequency increases. This is because the slope of Z_{in} decreases from 20 dB as the poles start approaching.

The $\frac{W}{L}$ of the transistor has to be fixed while taking into account the g_m and C_{gs} required. The present circuit necessitated that we use a small size transistor to get the required frequency range. The finalised circuit had:

$R = 8.7\,\mathrm{k}\Omega$, $\frac{W}{L} = \frac{8\,\mu}{180\,\mathrm{n}}$ with no external cap.

There was a slight dip in the inductor performance at the higher end of the frequency range due to the non-idealities mentioned earlier.

Another practical issue that comes up while using the active inductor in a CML ciruit is that we may not be able to afford the head-room it needs because CML already has multiple levels of transitors. The gate of the active inductor transistor needed to be biased at 2.4 V(V_{dd} = 1.8 V) in our circuit. To do this, we need a voltage boosting circuit to get 2.4 V from 1.8 V. The design of the voltage boosting circuit was not complete at the time of writing of this thesis.

Switching pair sizing

While sizing the main switching pair, we have to realize the fact that the two latches form an oscillator (there is no input driving them, they are self-driven). So, if there is insufficient loop gain, the voltages may remain at the dc solution itself and there will be no output swing. Choosing a transistor size larger than required will unnecessarily increase the load capacitance.

Constant g_m biasing

With constant current biasing, the circuit was not giving the required output swing at slow corners and high temperatures. This was basically because of a reduction in g_m. So, we decided to use a constant-g_m bias circuit so that the current increases at slow corners and high-temperature to compensate. With this, the output voltage swing was fairly constant across all corners and temperatures. RC low pass filters were used to shunt out noise from bias transistors. Since we are concerned with noise above 3.5 MHz, the RC cut-off frequency was set to around 1.5 MHz.

4.2.2 Buffer Design

The divider circuit is loaded by the Mixer and the subsequent divider stage. The divider circuit was taking up a lot of current to drive this load and also there was not enough room for all the transistors. So, we decided to go for a differential pair as buffer between the two transitors. We used an n-MOS differential pair

with a gain slightly greater than one. Here also, an active inductor was used to increase the bandwidth. The total current consumed by the divider circuit and the two differential pairs turned out to be lesser than just the single divider circuit appropriately scaled up to drive the load directly. One thing to be taken care here is that the buffer should see identical loads in the I and Q arms. So, though the subsequent divider will be connected to one arm only, the other arm too needs to be loaded by a dummy circuit which will simulate the same load.

4.2.3 Final Schematic

The final schematic of the divider and differential pair buffer are given in Figs 4.7 and 4.8. The divider consumed around 1.8 mA current and the two buffers consumed 700 μA each, so the total consumption was around 3.2 mA

4.3 Programmable Divider Chain

The divider chain following the first divider must be programmable from 481 to 496 (16 channels with center frequencies from 2.405 GHz to 2.480 GHz). A completely synchronous approach will require a lot of power. Using asynchronous dividers initially for high frequencies and then switching to synchronous circuits for low frequencies seems a plausible approach. A popular approach to implement a programmable divider is the architecture given in Fig 4.9. To see how this works, assume that the M/M+1 divider is working in div by (M+1) mode initially. After MQ pulses, the div by Q will give an output which will make the M/M+1 divider to go to div by M mode. The div by P counter will give an output only after a further M(P-Q) pulses. So, the output of the div by P counter will be running at (M+1)Q+M(P-Q)= MP+Q. Now, if we make P and Q programmable, we have a programmable divider over a large range of values.
Another architecture[8] which has become quite popular due to its modular architecture is given in Fig 4.10.

Figure 4.7: Divide by two schematic

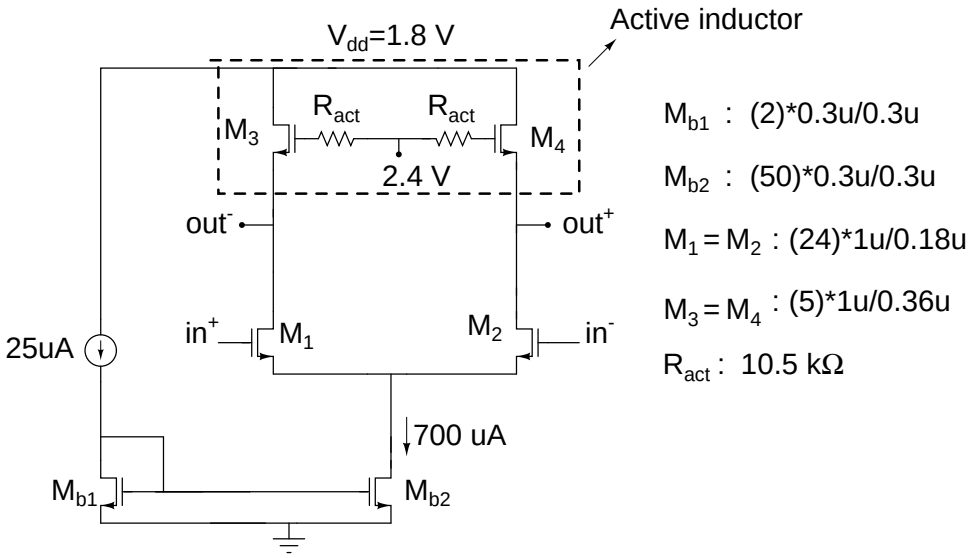

Figure 4.8: Buffer circuit- Differential pair

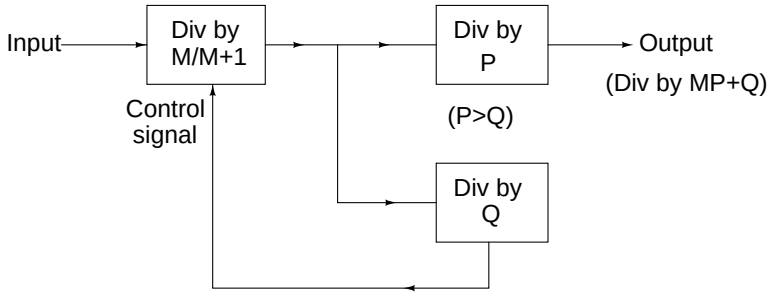

Figure 4.9: Programmable divider architecture

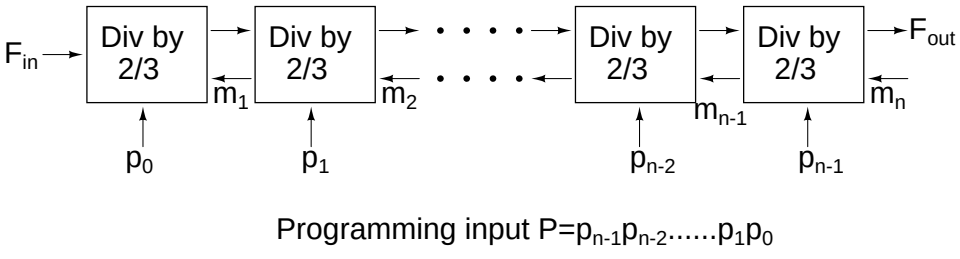

Figure 4.10: Modular Programmable divider architecture

The programming bits control whether the cells divide by 2 or 3. Here, $p_i = 1$ implies the i^{th} cell divides by 3. Let F_{out} be the final output signal from the divider. The thing to note here is that the cells don't divide by 3 throughout if 'p' is set. They divide by 3 only once every cycle of F_{out}. Its equivalent to each cell adding one input period to its output for one period of F_{out}. This functionality is achieved with the 'm' signal. It makes the cells divide by three only once every cycle of F_{out}. The 'm' signal input to the last cell is made '1'. So, the divide value doesn't range from 2^N to 3^N as one might expect, but from 2^N to $2^{(N+1)}$ corresponding to $P = 0$ to $P = 2^N - 1$.

We decided to choose this architecture for the divider mainly because of ease of implementation offered by its modularity.

Here, we need divide values from 481 to 496. So, we need atleast 8 cells. Further, the last 3 cells in the chain can have their programming bits set to '1' permanently. This restricts the programmability to all integer values between 480 and 511. If Q is the 4-bit channel select register, we can add 1 to this and store it in a 5-bit register. These can be used as programming bits, LSB through MSB go to the first five cells respectively. This gives the required programmability.

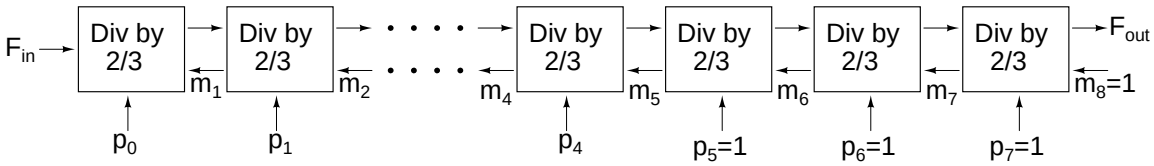

Figure 4.11: Programmable divider implementation with N=481 to 496

4.3.1 Implementation of a single 2/3 cell

Each 2/3 cell can be implemented using four latches and three AND gates as suggested in [8] (refer Fig 4.12).

Each latch can be implemented using CML logic. Also, the AND gate can be embedded into the latch quite easily. But there is a fundamental problem with the architecture in Fig 4.12. There is only half a cycle time for the 'm_{in}' signal

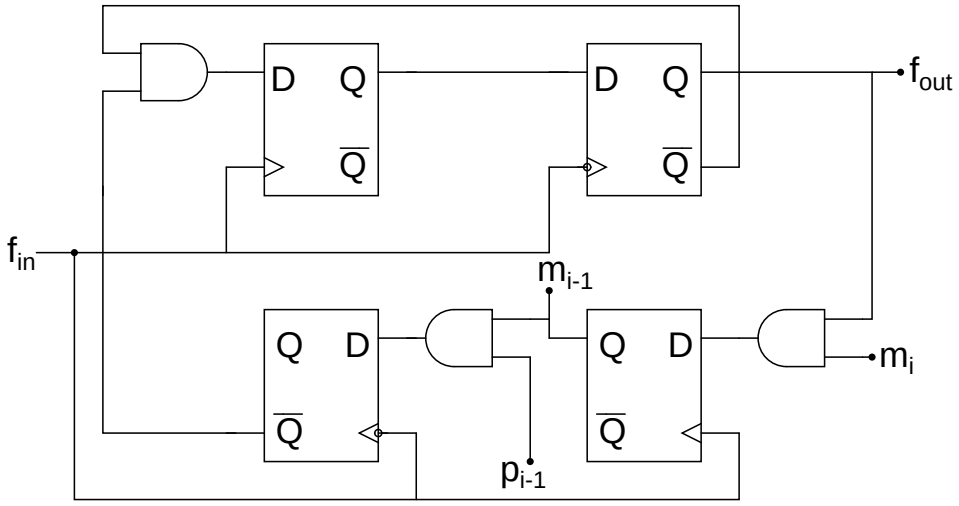

Figure 4.12: Implementation of a single divide by 2/3 cell

to settle before the positive edge of the clock. This is because the flipflops are negative edge triggered. This presents a bottleneck and the divide cells working at high frequencies may not be able to respond fast enough. Now, instead of taking f_{out} from the top right latch in that figure, we can take it from the top left latch, making the output to change at the positive clock cycle itself.This leaves one clock cycle for the latch to respond, hence removing the bottleneck. This modification was suggested in [9].

4.3.2 Implementation of the whole chain

We decided to use active inductor load for the first divider (working at 2.5GHz) and resistive loads for the others. Since each divide circuit operates at half the frequency as the previous one, we can just scale all the device sizes and current by half while doubling the load resistance. We decided to go with CML for the whole chain because of ease of implementation. If we go for CMOS logic at lower frequencies, the both the actual signal and control signals need to be passed through SE to differential converters. Control signal generation also has to be synchronized between CMOS and CML circuits. Anyway, at low frequencies, the current consumption is small enough that we don't gain much by changing to CMOS.

The 'latch with the AND gate embedded' used in the first divider in the chain(working at 2.5 GHz) is shown in Fig 4.13.

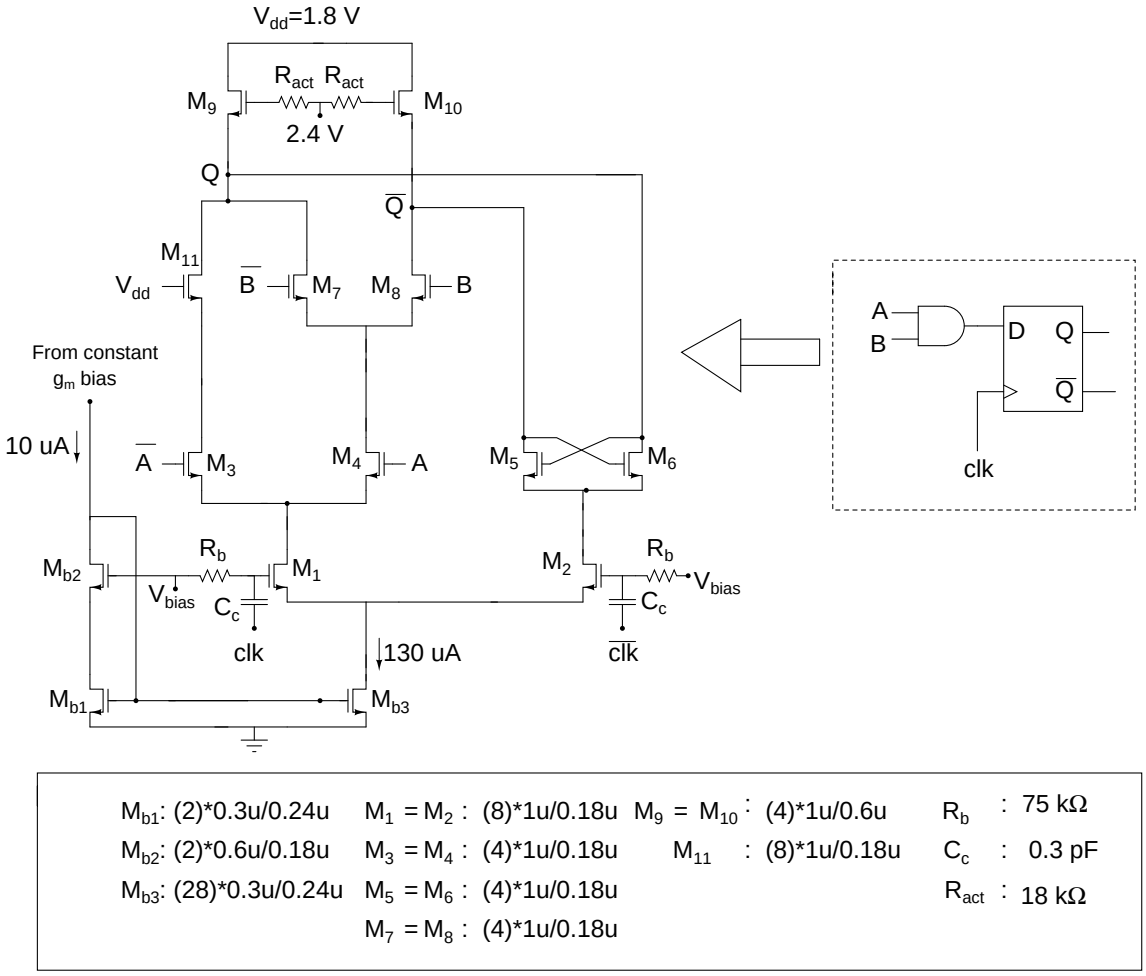

Figure 4.13: AND gate + Latch used in the first divider

One problem with the current architecture is that it is not possible to directly cascade cells since the input and output dc levels are different. So, the signal was ac coupled using a large capacitor and the dc bias was provided through a large resistor. For successive stages, the load resistor gets bigger and hence the dc coupling resistor also must get bigger. So, the first four cells were coupled using R-C and subsequent stages used source follower circuits to shift down to the required dc level.

Table 4.1 shows the transistor sizes and load resistor values for cells 2 to 8.

The bottom differential pair (where the clock is input) was sized twice that of the main differential pair in all cells.

Table 4.1: Sizing of the cells in the divider chain

Cell no.	Current	Main differential pair	load resistance
2	$265\,\mu$A	$\frac{(2)*1\,\mu\text{m}}{0.18\,\mu\text{m}}$	$8.5\,\text{k}\Omega$
3	$130\,\mu$A	$\frac{(1)*1\,\mu\text{m}}{0.18\,\mu\text{m}}$	$20\,\text{k}\Omega$
4	$58\,\mu$A	$\frac{(1)*0.5\,\mu\text{m}}{0.18\,\mu\text{m}}$	$32\,\text{k}\Omega$
5	$30\,\mu$A	$\frac{(1)*0.25\,\mu\text{m}}{0.18\,\mu\text{m}}$	$75\,\text{k}\Omega$
6	$14\,\mu$A	$\frac{(1)*0.25\,\mu\text{m}}{0.36\,\mu\text{m}}$	$150\,\text{k}\Omega$
7	$14\,\mu$A	$\frac{(1)*0.25\,\mu\text{m}}{0.36\,\mu\text{m}}$	$150\,\text{k}\Omega$
8	$14\,\mu$A	$\frac{(1)*0.25\,\mu\text{m}}{0.36\,\mu\text{m}}$	$150\,\text{k}\Omega$

No scaling was done for the last two stages because the current became so insignificant in the sixth stage itself and further scaling only increased the resistor size (and hence area) without giving any current saving. The same constant g_m bias circuit that was used before was used to bias all dividers.

4.4 Differential to Single-ended converter

Finally, the differential output signal has to be converted to single-ended form which has to be CMOS compatible to be given as input to the Phase-Frequency detector(PFD). The circuit has a differential pair with a current mirror in the load which converts the differential signal to single ended form. The single ended peak-to-peak value was kept to around 1 V. Then it was level converted using a source follower and given to an inverter which converted the signal to CMOS levels.

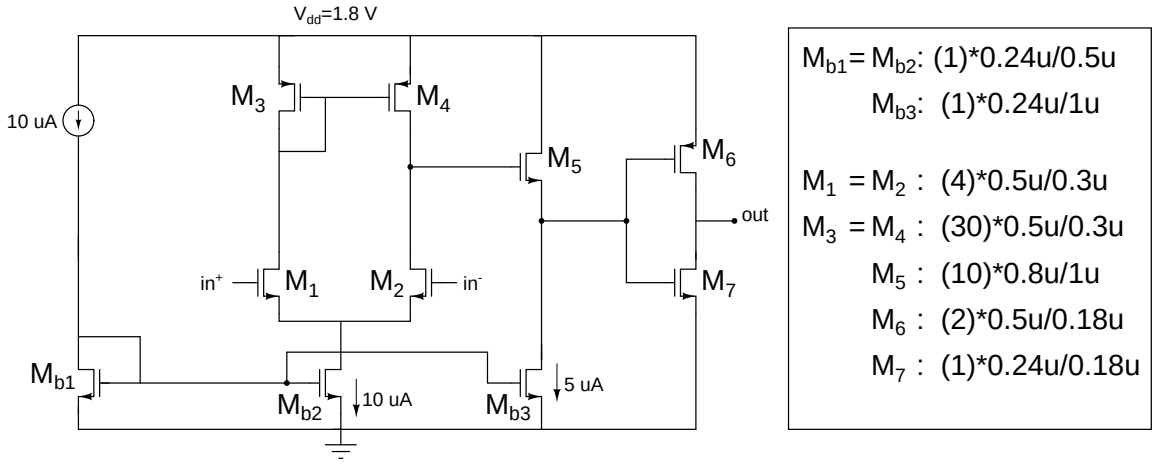

Figure 4.14: Differential to Single ended converter

CHAPTER 5

PHASE LOCKED LOOP DYNAMICS

5.1 Introduction

The remaining components of the PLL are the low frequency components. They comprise the Phase-Frequency Detector(PFD), charge pump(CP) and low pass filter(LPF). These components determine the stability, noise and settling time of the loop. So, they have to be designed carefully.

5.2 PLL loop dynamics

5.2.1 Loop Design

We have used a type-2, third order PLL for our design. Type-2 refers to the fact that there are two integrators in the loop and third order means that there are three poles in the closed loop transfer function of the PLL. The block diagram of a type-2 PLL is given in Fig. 5.1

The PFD basically produces UP or DOWN pulses whose widths correspond to the difference between the phases of the two inputs. At any point of time, only one of UP or DOWN will be present. A charge pump can be thought of as a digital version of transconductance. Current either flows in or out of the charge pump based on which of its inputs is high. If both inputs are high or low, then output current is zero. The UP and DOWN inputs can be connected to the positive and negative inputs of the CP. The charge pump current then flows into the R-C circuit, thereby giving the required poles and zeros.

The design parameters here are the charge pump current and the values of R, C_1 and C_2. We have to choose these values based on the required specs. The relevant specs are (repeated here for convenience):

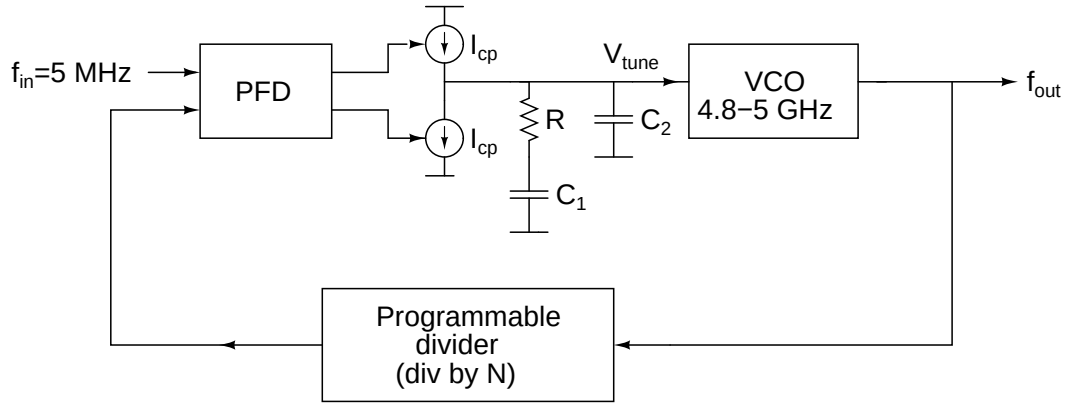

Figure 5.1: Type-2 PLL

- the settling time must be less than 192 μs
- the final settling value must have an accuracy of ± 40 ppm. Here, with a center frequency of around 2.5 GHz, it translates to a frequency deviation of ± 100 kHz
- phase noise must be below -92 dBc/Hz at an offset of 3.5 MHz from the carrier

Linear Model

A small signal, linear model can be written for the PLL, with phase as the variable. The transfer functions for the various blocks are given in Fig. 5.2

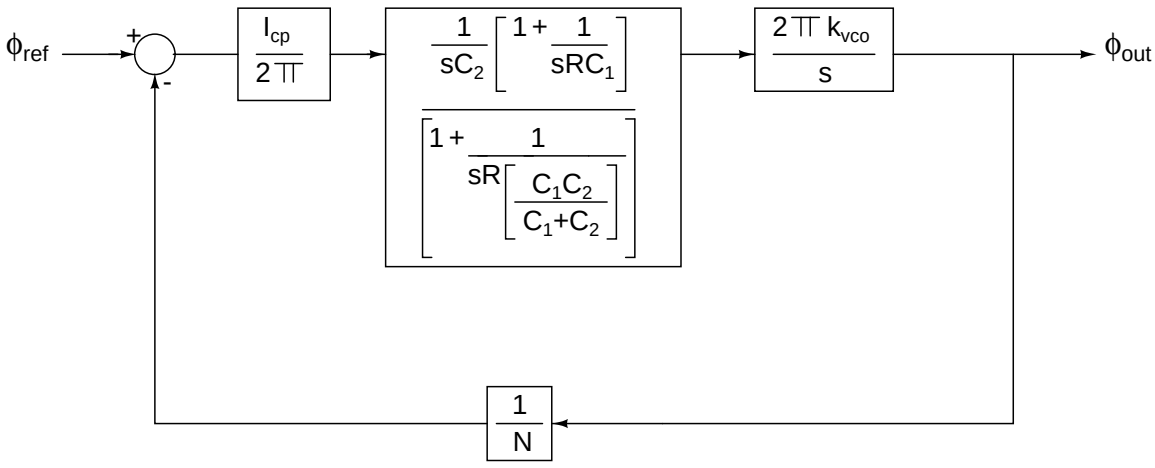

Figure 5.2: Linear model of a Type-2 PLL

The loop gain can be written as

$$LG(s) = \frac{I_{cp}k_{vco}}{N}\frac{\left(1+\frac{z_1}{s}\right)}{s^2C_2\left(1+\frac{p_1}{s}\right)} \tag{5.1}$$

where $z_1 = \frac{1}{RC_1}$ and $p_1 = \frac{1}{R\left(\frac{C_1C_2}{C_1+C_2}\right)}$ We can estimate the approximate value of the loop gain from eqn 5.1 by assuming that z_1 is well below f_u(unity gain frequency) and p_1 is well away from f_u. These assumptions are not impractical since they have to be largely satisfied in order to maintain a good phase margin. Thus, we have

$$f_u \approx \frac{I_{cp}k_{vco}R}{2\pi N} \tag{5.2}$$

We can also compute the closed loop response using eqn 5.1 as,

$$\begin{aligned} CLG(s) &= N\frac{1}{1+\frac{1}{LG(s)}} \\ &= N\left(\frac{1}{1+\frac{N}{I_{cp}k_{vco}}\frac{s^2C_2\left(1+\frac{p_1}{s}\right)}{\left(1+\frac{z_1}{s}\right)}}\right) \\ &= N\left(\frac{1}{1+\frac{N}{I_{cp}k_{vco}}\frac{s^2C_2\frac{p_1}{s}\left(\frac{s}{p_1}+1\right)}{\left(1+\frac{z_1}{s}\right)}}\right) \end{aligned} \tag{5.3}$$

Now, we shall try and model this as a second order system by neglecting the last pole. This will make the characterisation easier.

$$CLG(s) \approx N\left(\frac{1}{1+\frac{N}{I_{cp}k_{vco}}\frac{s^2C_2p_1}{(s+z_1)}}\right) \tag{5.4}$$

Rearranging the terms, we get

$$CLG(s) \approx N \left(\frac{\frac{s}{z_1} + 1}{1 + \frac{s}{z_1} + s^2 \frac{N}{I_{cp} k_{vco}} \frac{C_2 p_1}{z_1}} \right) \tag{5.5}$$

But $p_1 \approx \frac{1}{RC_2}$, $z_1 = \frac{1}{RC_1}$ and hence

$$CLG(s) \approx N \left(\frac{\frac{s}{z_1} + 1}{1 + \frac{s}{z_1} + s^2 \frac{N}{I_{cp} k_{vco}} C_1} \right) \tag{5.6}$$

Comparing the denominator to the standard form $\left(1 + \frac{s}{Q\omega_0} + \frac{s^2}{{\omega_0}^2}\right)$, we get

$$\omega_0 \approx \sqrt{\frac{I_{cp} k_{vco}}{N C_1}} \tag{5.7}$$

and

$$Q \approx \frac{\sqrt{\frac{N}{I_{cp} k_{vco} C_1}}}{R} \tag{5.8}$$

The settling time(T_s) can be calculated by observing that the response to a step will be an exponential with time constant $\frac{2Q}{\omega_0}$ if we can neglect the effect of the extra pole (p_1). Considering that we are going to place the pole far off from the unity gain frequency, this is a reasonable assumption to make. To satisfy the specs, we need the final value to settle to $\pm 0.004\,\%$(to satisfy the ± 40 ppm constraint) of the ideal value. So, we have

$$e^{-\frac{\omega_0 T_s}{2Q}} = 4 * 10^{-5} \tag{5.9}$$

$$\Rightarrow \frac{\omega_0 T_s}{2Q} = ln\left(\frac{1}{4 * 10^{-5}}\right) \tag{5.10}$$

$$\Rightarrow T_s = (10.126) * \left(\frac{2Q}{\omega_0}\right) \tag{5.11}$$

The absolute limit for T_s is 192 μs, but to be on the safer side, we have to design for a value well below this. Armed with all these equations, we can now proceed to choose the component values. The thing to note, though, is we have made a few approximations while arriving at these equations. Also, many non-ideal effects come in when we finally implement these circuits. For example, the UP and DOWN charge pump currents need not be the same, the VCO characterisitics is not strictly linear, R and C values vary widely with process. So, a lot of safety margin is needed when we go about designing with the aid of these equations.
There is also one more phenomenon which the model doesn't represent. The phase detector has a range of 0 to 2π only. It is quite possible that, in the process of locking, the phase increases from 0 to 2π and further too. Now, the PD characteristic becomes non-monotonic, after crossing 2π, it goes down. This is generally referred to as 'cycle-slipping' in PLLs.

Verification with MATLAB

To choose the values, we can start with the assumption that z_1 is a decade below f_u and p_1 is a decade above f_u. k_{vco} was measured from the VCO characteristics and it was around 140 MHz/V. We can take N to be 500 for our design. A good starting point for Q will be around 0.5. We can also start with a value of 50 kHz for f_u. A first order estimate of settling time is $\frac{1}{f_u}$. So, by choosing 50 kHz for f_u, we are starting with a settling time of around 20 μs, keeping in mind that we are looking at a very high accuracy here. We should also try to keep C_1 as small as possible because a very large value will mean a lot of chip area.
The system was also modelled in MATLAB. After choosing the values based on initial design, we simulated the model in MATLAB to check if we indeed got the characteristics we were looking for. After a few iterations, we arrived at the following values: $I_{cp} = 10\,\mu$A, $R = 80\,$kΩ, $C_1 = 220\,$pF, $C_2 = 7\,$pF
Important loop parameters:
$p_1 = 293\,$kHz, $z_1 = 9.04\,$kHz, $f_u = 35.48\,$kHz, $f_0 = 17.95\,$kHz, $Q = 0.5$ and $T_s = 90\,\mu$s
The MATLAB simulations plots are shown in Figs. 5.3, 5.4 and 5.5 . From the

plots, we see that there is a slight peaking in frequency response and ringing in the step response. Though the phase margin is good, the zero in the closed-loop transfer function is causing this slight overshoot. Also, the step response indicates a settling time T_s of 110 μs (though the exact value is not discernable from the plot here). Also, we see from the loop gain plot that $f_u \approx 35$ kHz.

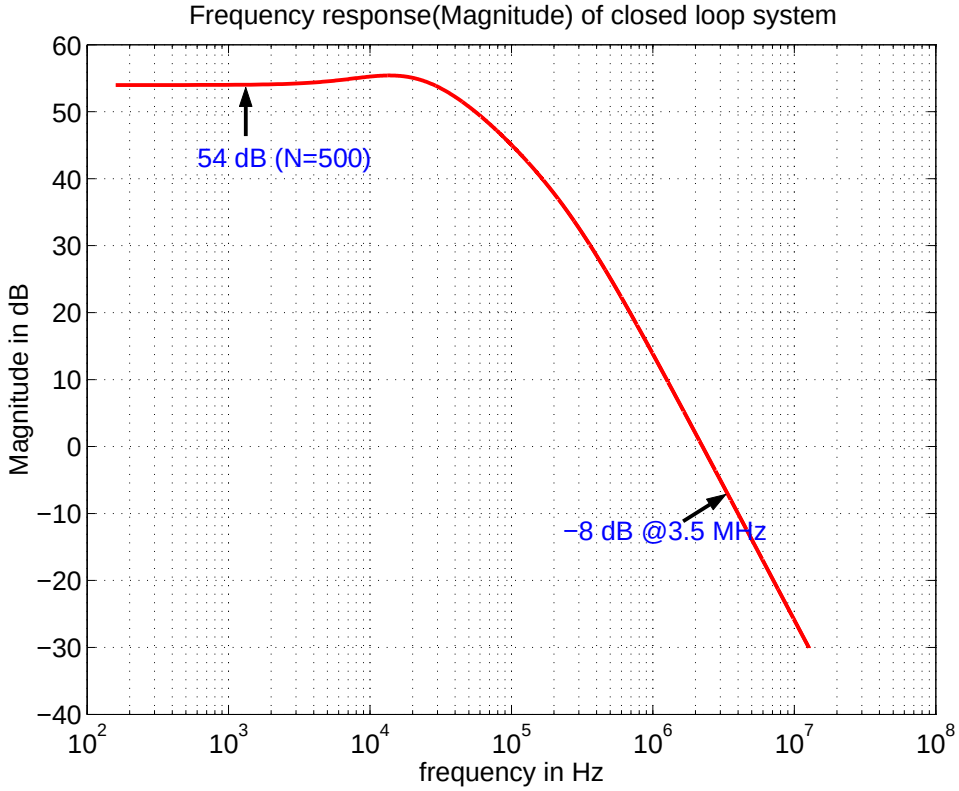

Figure 5.3: Frequency response of the PLL

5.3 Circuit Design

Now that the block level design is over, we come to the circuit design of PFD and CP.

5.3.1 Phase-Frequency Detector

PFD can be implemented quite simply by using two resettable flipflops and an AND gate (Fig. 5.6). It works as follows: If A leads B, then UP stays at '1'

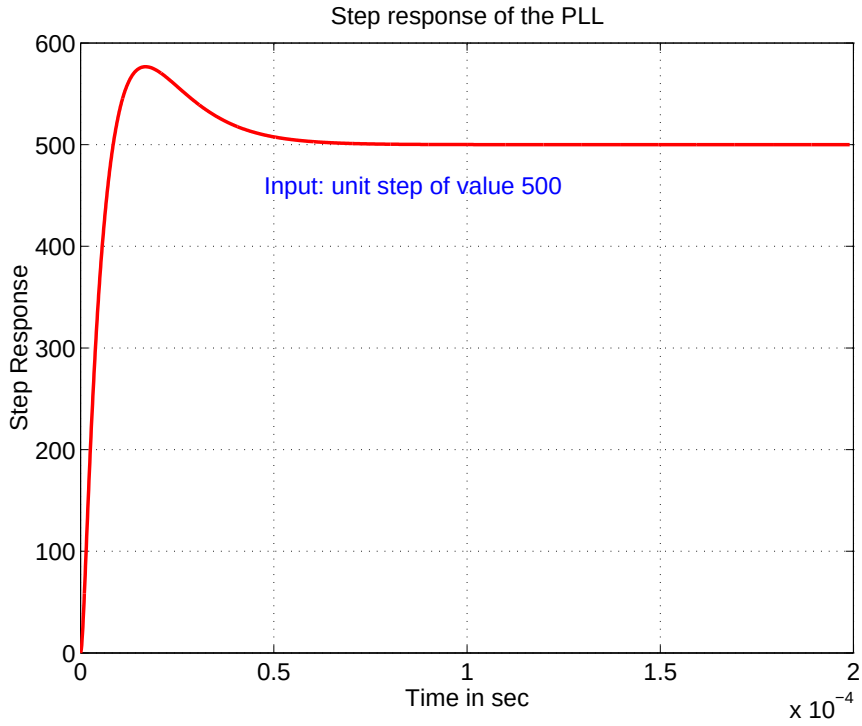

Figure 5.4: Step response of PLL

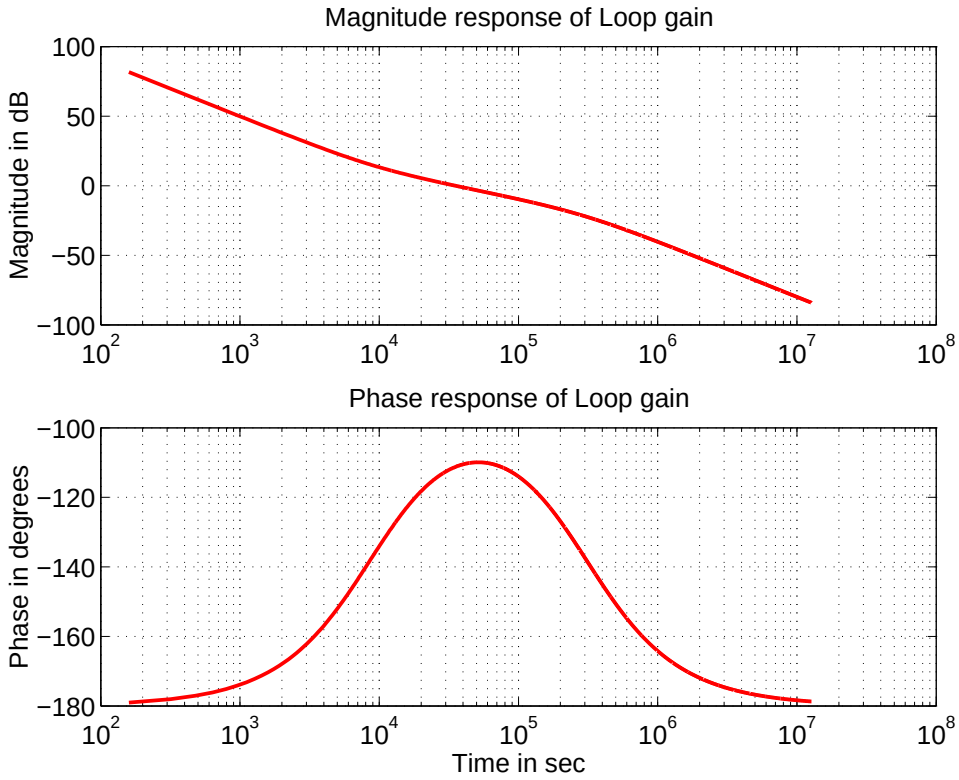

Figure 5.5: Frequency response of Loop gain

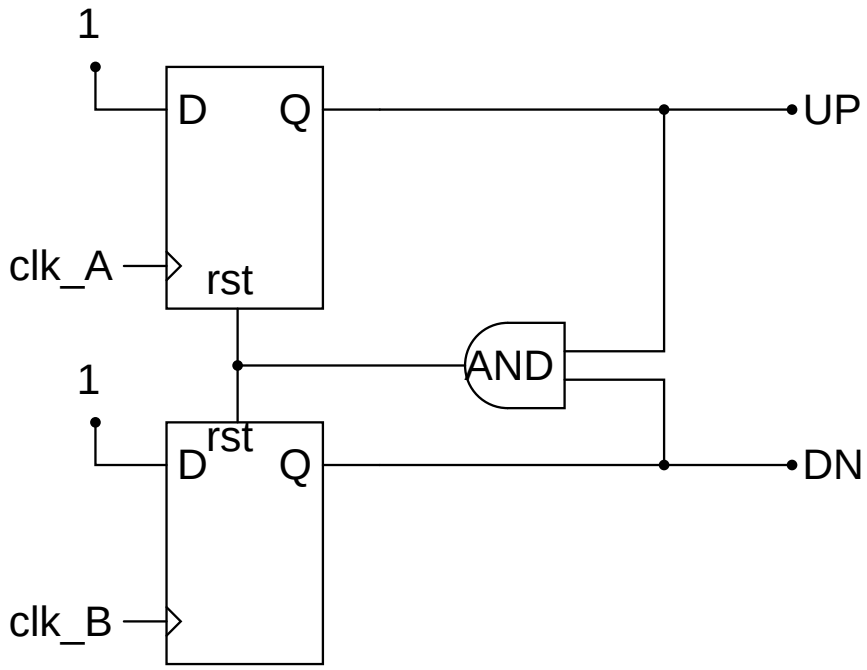

Figure 5.6: PFD block diagram

from the positive clock edge of A till the positive clock edge of B. When positive clock edge of B arrives, the AND gate output resets both the flipflops. So, ideally, there is only UP output when A leads B. The same applies when B leads A. But, due to the finite delay in the AND gate and the reset mechanism, both UP and DN might be high for a short duration. But if the charge pump (the next block) has equal UP and DN currents, this is not really a problem since the two currents cancel out and there is no current flowing into the loop filter.

Here, we have implemented the flipflops as TSPC latches. The only tricky part is the implementation of the RESET functionality into the latches.

Consider the standard D-FF implementation using TSPC latches. Here, the input is made '1' permanently as required by the PFD (Fig. 5.7). Now, we have to try and see how to add the reset functionality into this.

The reset has to be asynchronous, i.e, the clock should not come into the picture here. Reset can be applied when 1. clock is low and 2. clock is high. We shall consider each case one by one. Please refer to Fig. 5.7 for the following discussion.

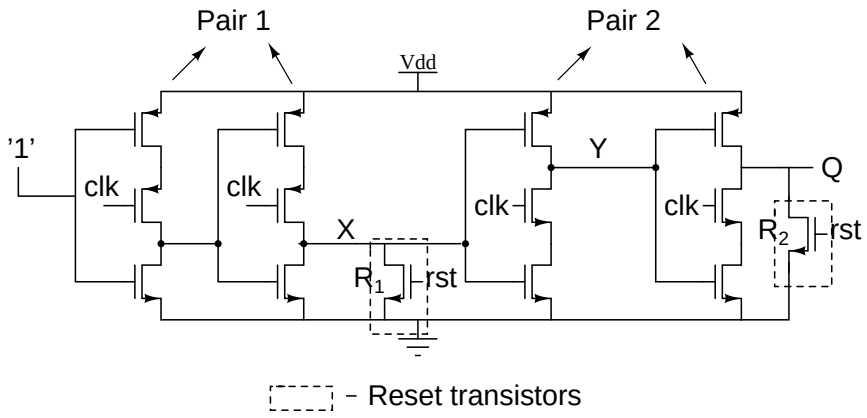

Figure 5.7: TSPC D-FF with input set to '1'

Clock is low

When reset is applied, both Q and X get reset initially through the transistors. Also, Y will go high. After this, X will go high(it will be recharged by the pMOS transistors), but Y will remain high since clock is low. So, Q will remain low till next clock high. But we should make sure that the reset transistor-R1 is strong enough when compared to the pMOS transistors charging 'X' so that X goes low for some time initially.

Clock is high

Now, if clock is high when reset is applied, let's see what happens. X goes low, Y goes high and Q goes low. But this state is maintained only till clock remains high. Now, when clock becomes low, X charges again, but the nMOS transistors(in pair 2) don't conduct now. So, there is no change at Y and Q and all seems fine. But there is one practical issue here. That is, when X starts charging when clock goes low, the pair-2 nMOS transistors don't completely switch off due to the finite rise and fall times of the clocks and hence they may transmit for sufficient time to make Y low. This will make Q go high again. To avoid this, the clocks to 'pair-1' were delayed by enough time to make sure that node X doesn't start charging till the clocks to the nMOS transistors become completely low. The delay was achieved using four inverters with 100 fF loads. Fig. 5.8 shows the two clocks.

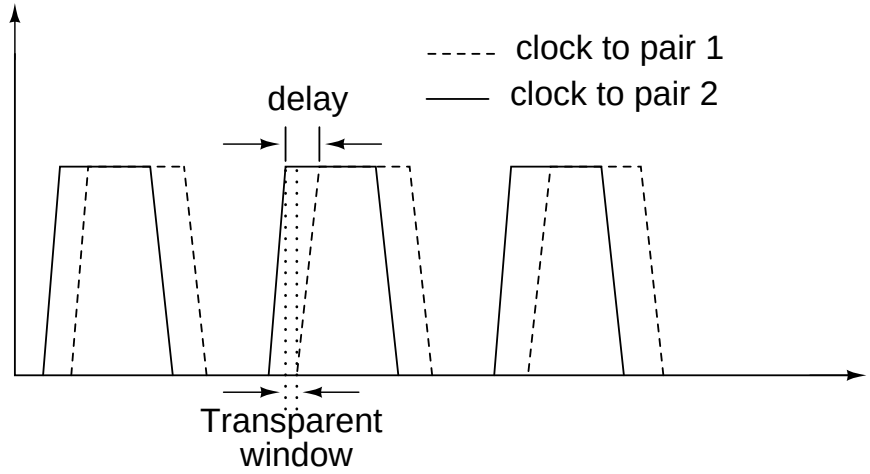

Figure 5.8: Clocks to the two pairs

There is one potential problem that one might foresee when we delay one of the clocks. Now, there is a small time window when both the latches are transparent. If the reset is given during this window, infact the flipflops don't reset. This is compounded by the fact that, when PLL is in steady state, that reset immediately follows the rising edge, bang on our problematic time window. To solve this issue, we can delay the reset path by a sufficient time (more than the time by which the clock was delayed inside the flipflop) so that the reset signal is pushed out of this window.

The circuit was simulated over all corners to make sure that the delays come out properly as expected and there is no instance where the circuit may fail.

Fig. 5.9 shows the flipflop used in the PFD and Fig. 5.10 shows the full PFD circuit. The clock delay in the flipflop was kept to around 3 ns and 'rst' signal was delayed by around 6 ns with respect to the clock, satisfying easily, the constraints that we were talking about. The NAND gate used in the PFD was a simple CMOS implementation.

5.3.2 Charge Pump

The charge pump essentially converts the voltage information from the PFD to current information. The current can then be passed through a series RC branch to get the integrator + zero combination.

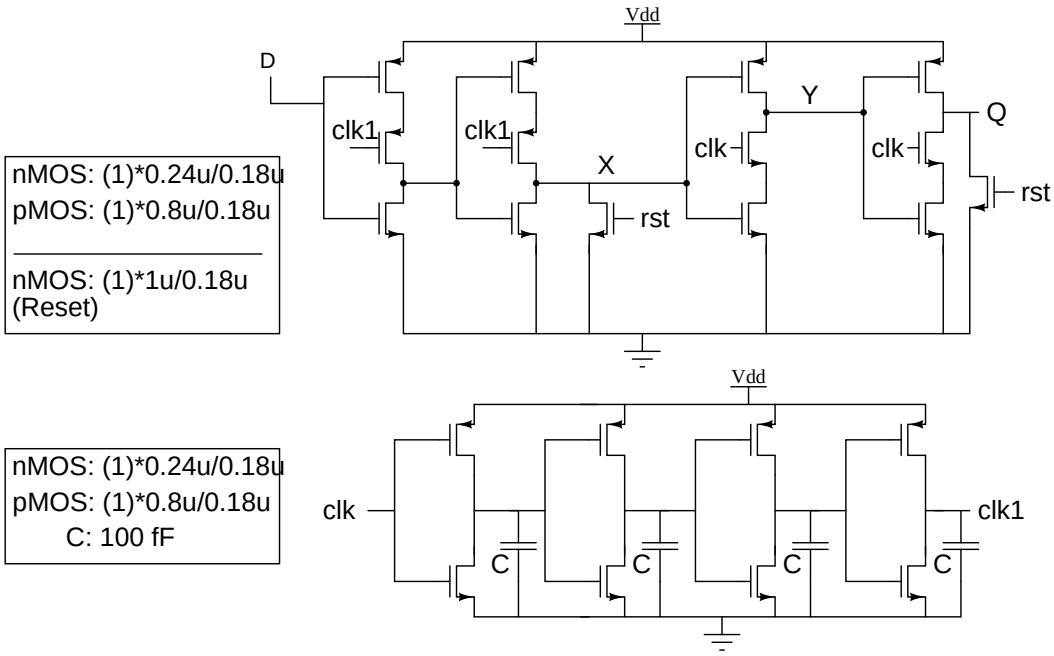

Figure 5.9: TSPC D-FF with reset

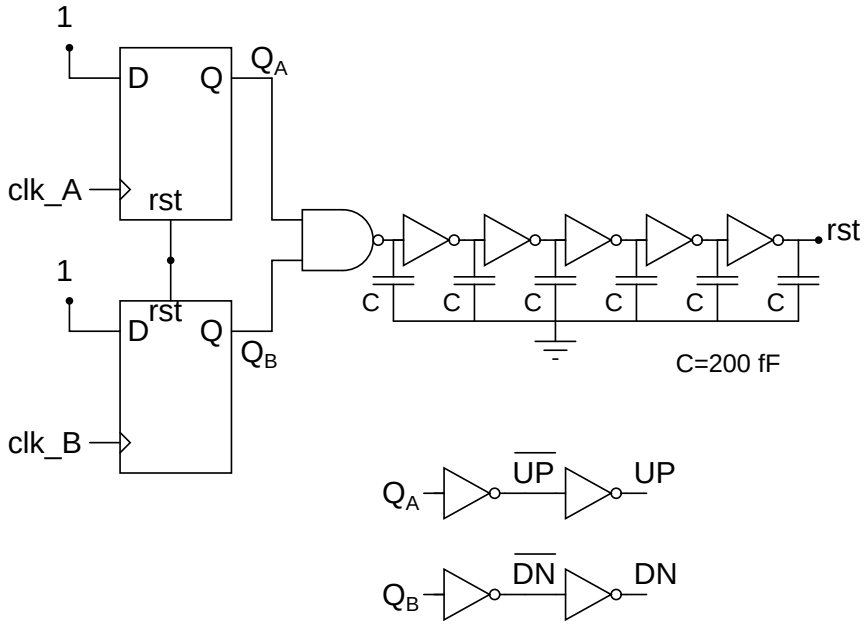

Figure 5.10: Implemented PFD

The basic requirements for a charge pump are two current sources for UP and DN currents and a control mechanism to switch these currents based on PFD outputs. Based on [10], we chose to try out the 'switch-on-source' and 'current steering' type of charge pumps. We stuck with simple implementaions since high speed operation is not the concern here. Current steering CPs have the advantage

that they don't switch off the currents but just switch them to a dummy branch. Hence, the off-current is lesser when compared to non-current-steering types. From simulations also, it was observed that distortion at steady state was slightly better for current-steering CPs. One more practical issue in CPs is the range of tuning (output) voltage over which they work properly. This is because, at voltages near the rails, the MOS transistors stop conducting. Here too, current steering CP was doing slightly better. So, we decided to go with the current-steering CP. Fig. 5.11 shows the schematic used.

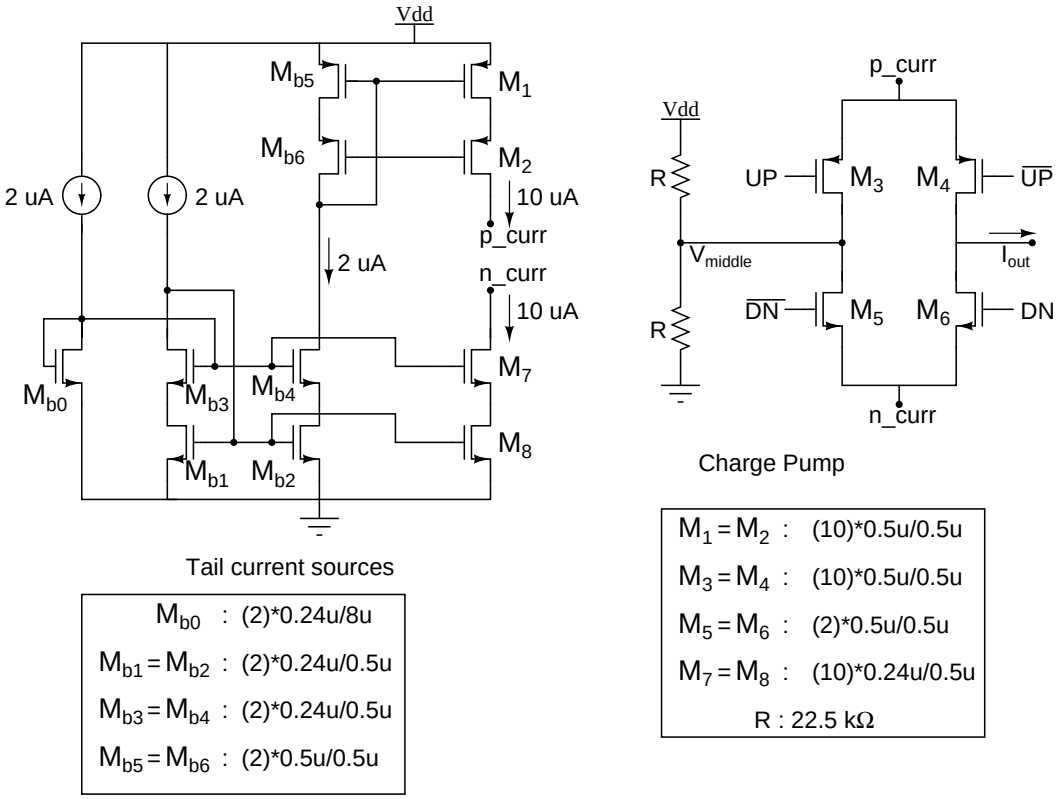

Figure 5.11: Current steering CP

5.4 Noise analysis

The fact that the transceiver is going to be used in a wireless environment with other devices leads to many restrictions on both in-band and out-of-band emissions, which directly impacts the amount of PLL noise that can be deemed acceptable.

5.4.1 Transmitter specifications

As far as the transmitter is concerned, the signal being transmitted out of the antenna must conform to the spectral mask given by the standard. The actual signal output power restriction doesn't affect the PLL specs, but the restriction on the side lobes does. The specification says that the power in a 100 kHz resolution bandwidth, if located more than 3.5 MHz away from the carrier, must not exceed -30 dBm in absolute terms and -20 dB relative to the highest spectral power. The noise coming out of LO at these frequencies must be low enough so that the signal mixing with the noise at these frequencies will satisfy this power constraint.
There is also a blanket phase noise limit for emission outside 3.5 MHz. According to this phase noise must not exceed -92 dBc/Hz @ 3.5 MHz. If phase noise is kept to this value, then power in a 100 kHz bandwidth @ 3.5 MHz is

$$power = -92 + 10log(10^5) \tag{5.12}$$

$$= -42dBm \tag{5.13}$$

So, meeting the phase noise requirement of -92 dBc/Hz has ensured that we will meet the side-lobe specification too. We also have to add the fact that there is a filter following the DAC which will further attenuate the side lobes.
Hence, we can safely say that as far as the transmitter is concerned, meeting the phase noise specification(less than -92 dBc/Hz @3.5 MHz away from the carrier) is enough.

5.4.2 Receiver specifications

In the receiver, the issue of concern is the following. There will be interferers in adjacent bands, which will mix with the phase noise at the same frequency to go back to the baseband along with the signal. This will look like noise to the signal. The same happens with spurious tones(spurs) too. So, we have to make sure both the phase noise and spurs don't exceed the required limit. The way mixing happens can be visualized using the Fig. 5.12

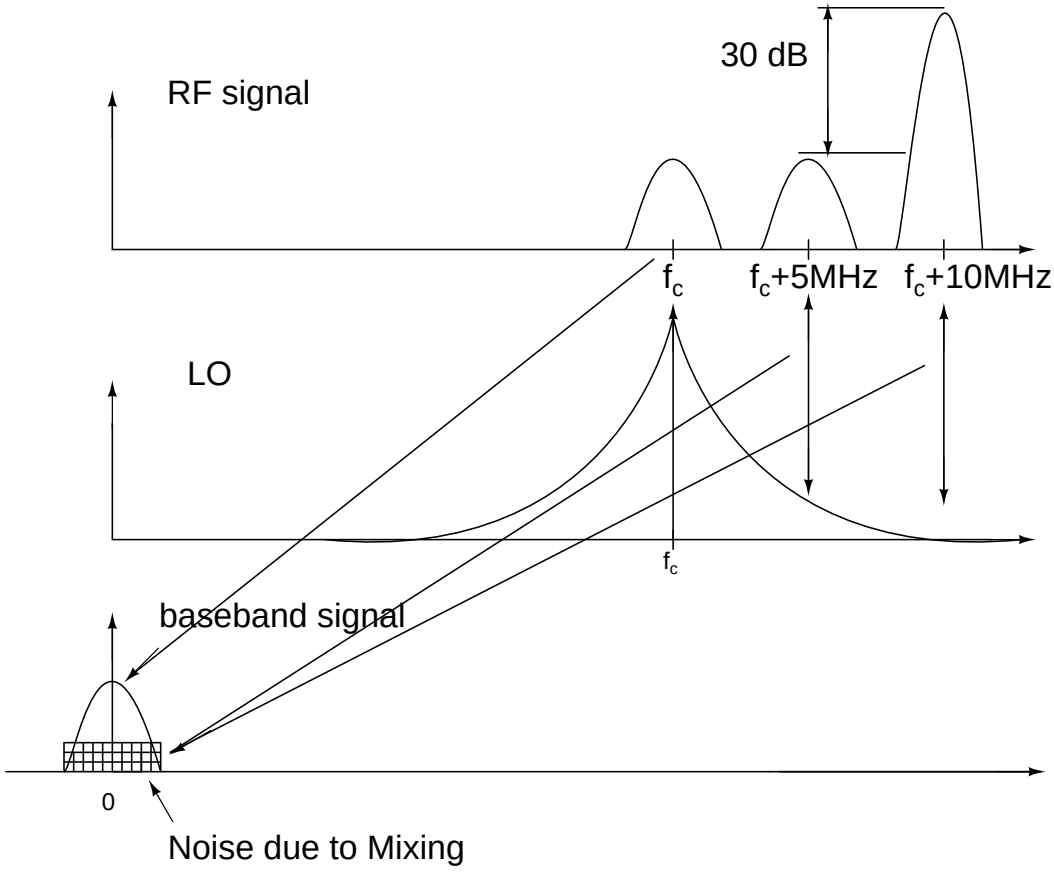

Figure 5.12: Noise due to interferers

The adjacent channel interferer can be only as strong as the signal and the alternate channel interferer can be 30 dB stronger than the signal.

First, let us calculate the spur level allowed.

1. Due to the 5 MHz interferer. Let the spur level(in dBc) be P_{5M}

$$SNR_{in} = -(P_{5M}) \tag{5.14}$$

$$SNR_{out} = -(P_{5M}) - NF(dB) \tag{5.15}$$

Assuming 20 dB noise figure, and knowing that SNR_{out} allowed is 0 dB, $P_{5M} = -20dBc$.

2. Due to the 10 MHz interferer. Let the spur level(in dBc) be P_{10M}

$$SNR_{in} = -(P_{10M}) - 30 \tag{5.16}$$

$$SNR_{out} = -(P_{10M}) - 30 - NF(dB) \tag{5.17}$$

Assuming 20 dB noise figure, and knowing that SNR_{out} allowed is 0 dB, $P_{10M} = -50dBc$.

Next, let us calculate if the -92 dBc/Hz phase noise is enough to get the required SNR. Knowing that the signal bandwidth is 2 MHz, noise power coming into signal band is

$$\begin{aligned} SNR_{in} &= 92 - 10log(2M) \\ &= 29dB \end{aligned} \tag{5.18}$$

$$\Rightarrow SNR_{out} = 9dB \tag{5.19}$$

This satisfies our SNR_{out} requirement.

So, to finalise, we just need to make sure that the spur limits specified here are satisfied. The receiver doesn't pose a tighter phase noise spec than the transmitter.

5.4.3 Modelling of various noise sources

The potential noise sources in the PLL are

- VCO phase noise
- noise due to divider chain
- noise from CP
- noise from resistor in LPF
- spurs due to non-ideal PFD+CP

Let us characterize each one by one. We can find the transfer function from each noise source to output. The power spectral density at the output can be obtained by multiplying each noise spectral density with the transfer function squared and adding them up. The end goal is to see the spectral density at 3.5 MHz away from carrier and see if it is below the required -92 dBc/Hz

VCO phase noise

Fig. 5.13 shows the model that can be used to get the transfer function of VCO noise to output. Qualitatively, we see that this has to be a high pass response with a cut-off frequency near f_u.Since 3.5 MHz is well away from f_u, the gain must be approximately 1 at that frequency. The MATLAB plot of the frequency response from vco noise to output is given in Fig. 5.14. From this, we see that the noise at 3.5 MHz comes out undisturbed. So, we have to make sure that VCO phase noise is a few dB below -92 dBc/Hz

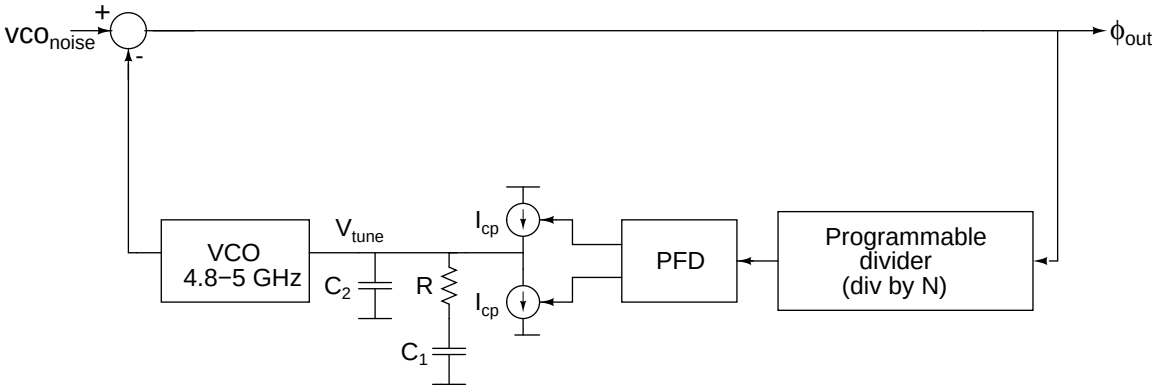

Figure 5.13: Noise due to VCO

Noise due to divider chain

To estimate the noise contribution of the divider, we can estimate the noise at the end of the divider chain by giving a pure sine wave to the first divider and applying the transfer function to the output, which by the way, is the same as the transfer function from reference to output since noise is being injected at the same point as the reference. So, we can just multiply the noise by the gain at 3.5 MHz, which is -8 dB (from Fig. 5.3

Noise due to charge pump

The charge pump noise is the noise due to the current sources. The current noise is of the order of KTg_m and here, the g_m was around 0.25 mS. We can refer this

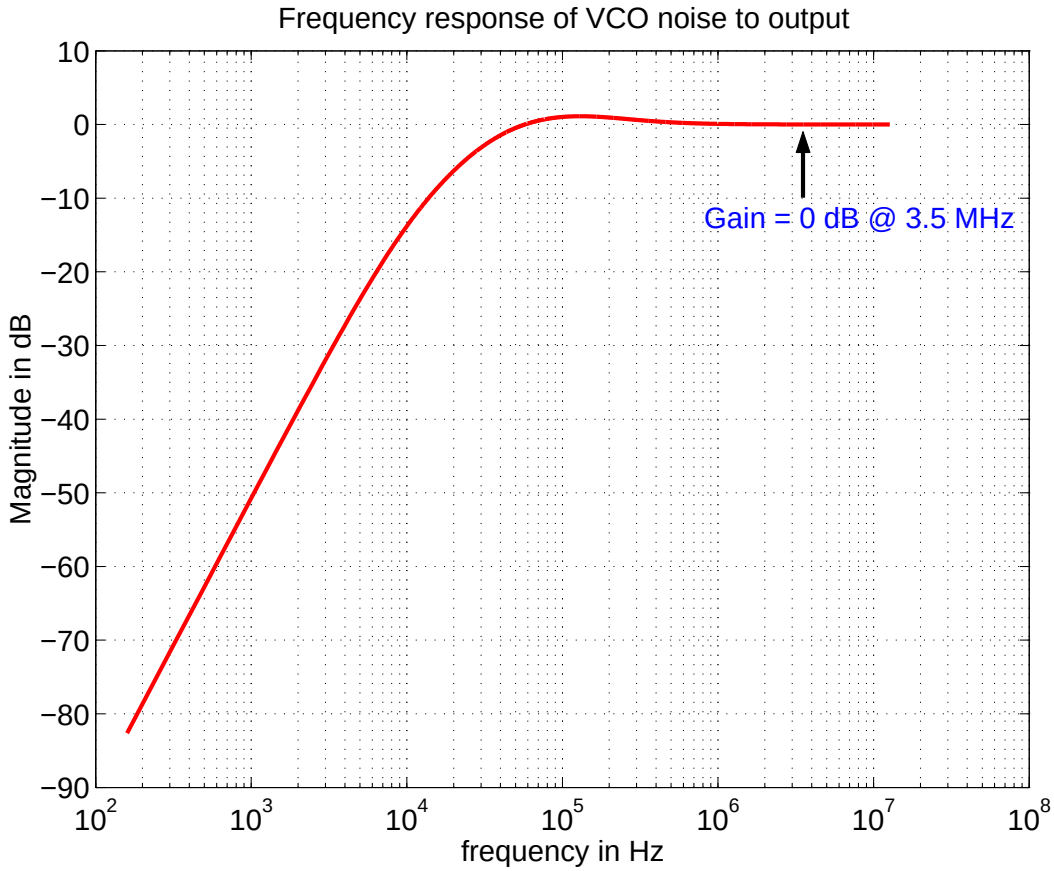

Figure 5.14: Frequency response(VCO noise to output)

current noise to the input of the phase detector(we have to divide by $\frac{I_{cp}}{2\pi}$). Now, this is equivalent to reference noise and we can apply the same gain(-8 dB) as we did in the previous case.

$$\begin{aligned} S_{out,CP} &= \frac{S_{CP}}{\frac{I_{cp}}{2\pi}^2} * (-8dB) \\ &= KTg_m \frac{I_{cp}}{2\pi}^2 * (-8dB) \\ &= -240 + 116 - 8 \\ &= -132dBc/Hz \end{aligned} \tag{5.20}$$

Noise due to resistor in LPF

We have a large resistor in LPF which contributes noise. It can be modelled as given in Fig. 5.15. The frequency response of the transfer function to output is

given in Fig. 5.16. The gain at 3.5 MHz is 8 dB.

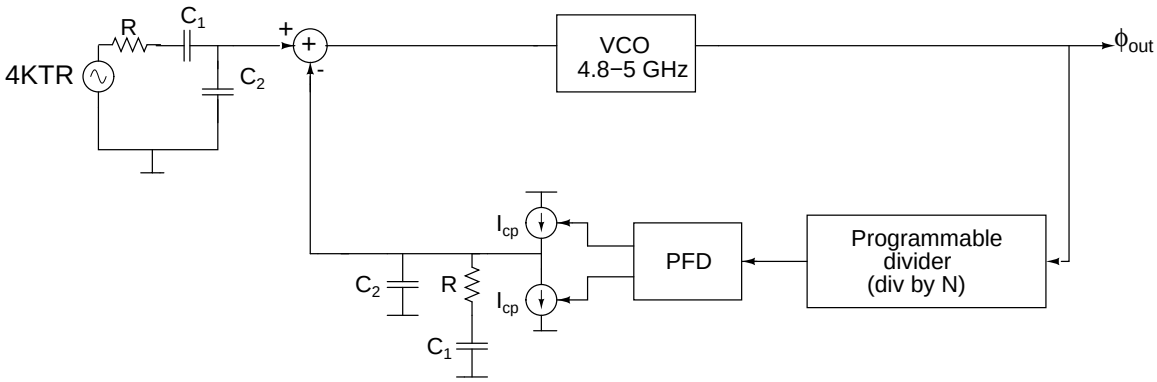

Figure 5.15: Noise due to resistor

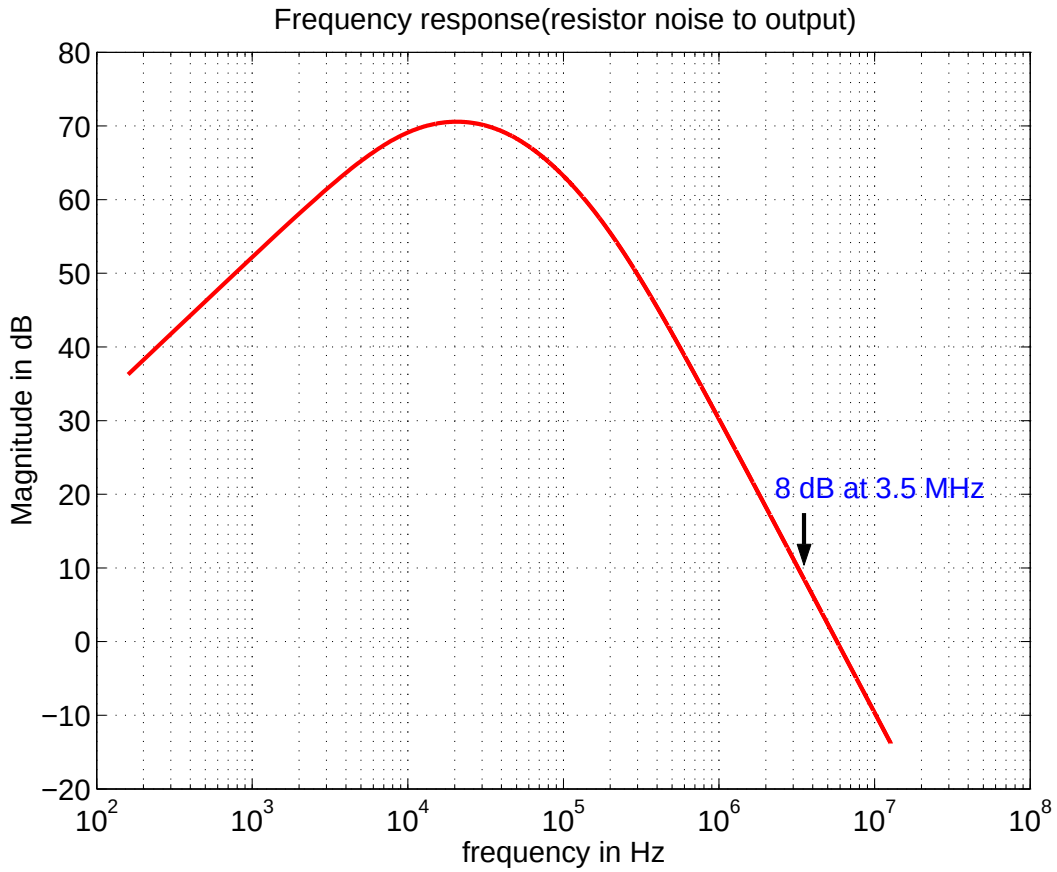

Figure 5.16: Frequency response(resistor noise to output)

$$\begin{aligned} S_{out,res} &= S_{res} * (8dB) \\ &= 4KTR * (8dB) \\ &= -204 + 55 + 8 \\ &= -141dBc/Hz \end{aligned} \tag{5.21}$$

Again, this is well below the required level.

Spurs due to charge pump current mismatch

Ideally, at steady state, there should be no charge pump current flowing into the LPF. But a combination of two factors result in some current flow. First, PFD output has both UP and DN on for a finite time due to the reset delay. Second, the UP and DN currents in the charge pump may not be equal. This means that the difference current will flow during the time both UP and DN are on. This creates spurs at harmonics of the reference frequency.

This effect was modelled as follows: PFD was driven with signals having a zero phase difference and CP currents were made to have a mismatch value of 3σ(for the transistors used here, the 3σ value comes to around 18%). The components included in the simulation were PFD, CP and LPF. Then, we took the DFT of the output voltage at LPF($\mathrm{v_{tune}}$) to find out the spur levels at 5 MHz and 10 MHz(simulation results showing the spur levels is given in chap. 6). The spurs can then be given as input to the PLL model(Fig. 5.17) to find out the spur levels in the output spectrum. The model has a bandpass response and we can expect that the transfer function will just be the VCO gain $\left(\frac{2\pi k_{vco}}{s}\right)$ at 5 MHz and 10 MHz since they far away from f_u. The MATLAB plot of the frequency response is plotted in Fig. 5.18. From the MATLAB plot, we see that the gains at 5 MHz and 10 MHz are 29 dB and 23 dB respectively.

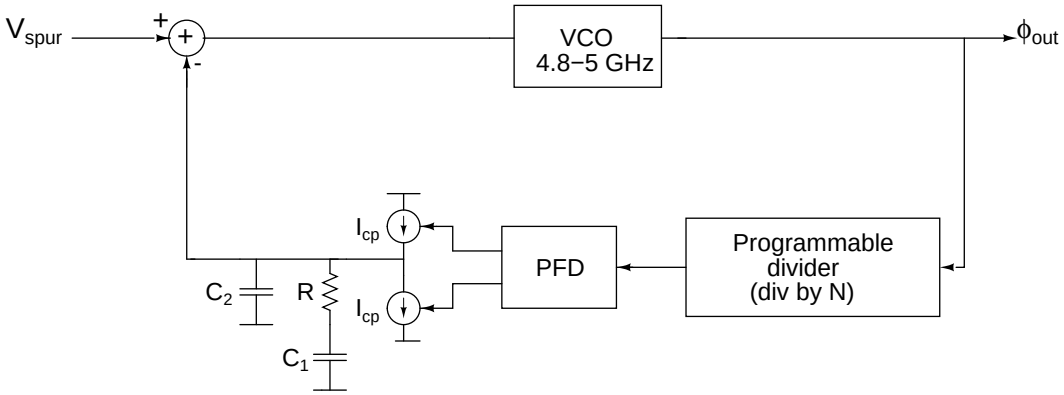

Figure 5.17: Model to characterize spurs due to CP current mismatch

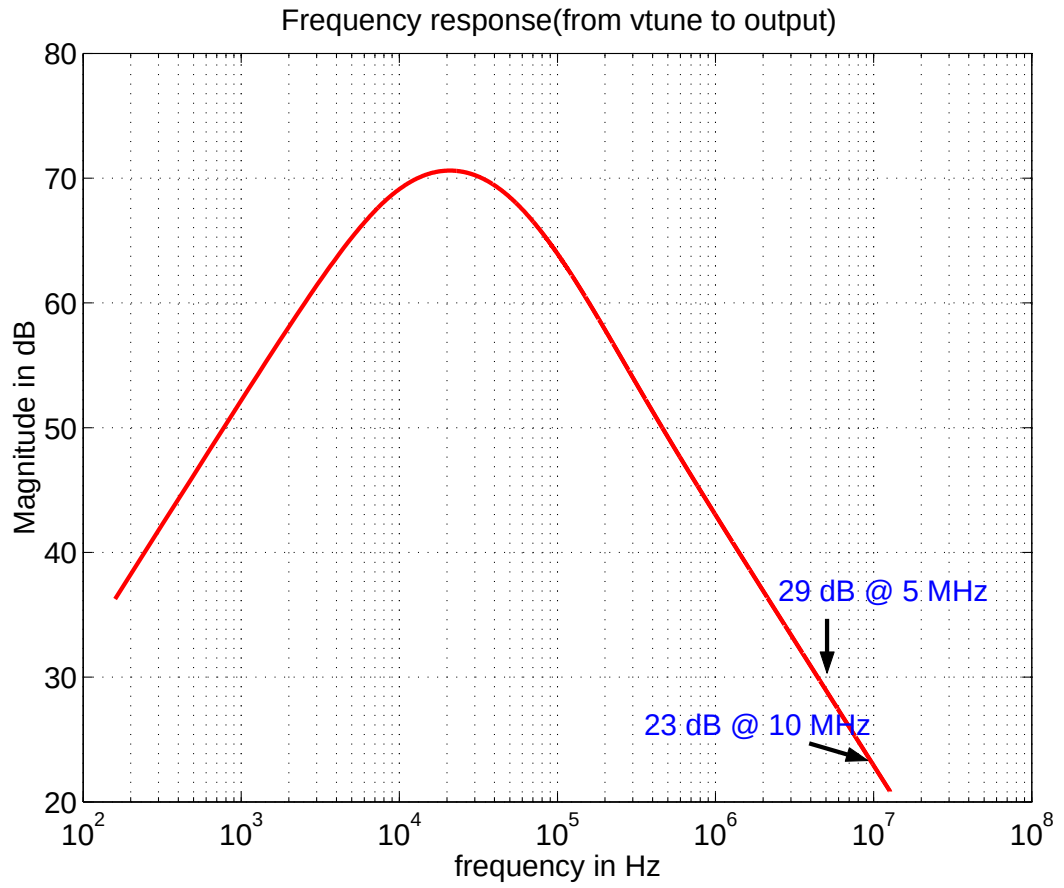

Figure 5.18: Frequency response(from v_{tune} to output)

CHAPTER 6

LAYOUT AND SIMULATION RESULTS

6.1 Layout

Fig. 6.1 shows the layout of the VCO and first divider. The other components had not been laid out at the time of writing this thesis. Extreme care has to be taken to match the circuitry in the I and Q branch. Also, we should be careful not to cross the LO lines across RF signal lines to reduce LO leakage.

6.2 Simulation Results

6.2.1 VCO and first divider

The simulations of VCO and the first divider were done with the extracted views. The oscillation amplitude was measured to be around 0.7 V SE,pp. The obtained oscillation frequency vs tuning voltage plot is shown in Fig. 6.2 We see that this covers the required range of 4.8 GHz to 4.96 GHz comfortably.

Fig. 6.3 shows the time domain waveforms at the output of the VCO and the first divider. The divider was designed such that the output amplitude was around $350-400$ mV pp,SE. The output of the differential pair buffer following the divider was also kept around 400 mV pp,SE Fig. 6.4 shows the phase noise plot at the output of the first divider. We see that the phase noise goes to around -117 dBc/Hz at 3.5 MHz away from carrier. Though this noise comes out directly to the PLL output at this frequency(as was explained in chapter 5), this meets the phase noise spec of the PLL quite comfortably(just -92 dBc/Hz required to meet the spec).

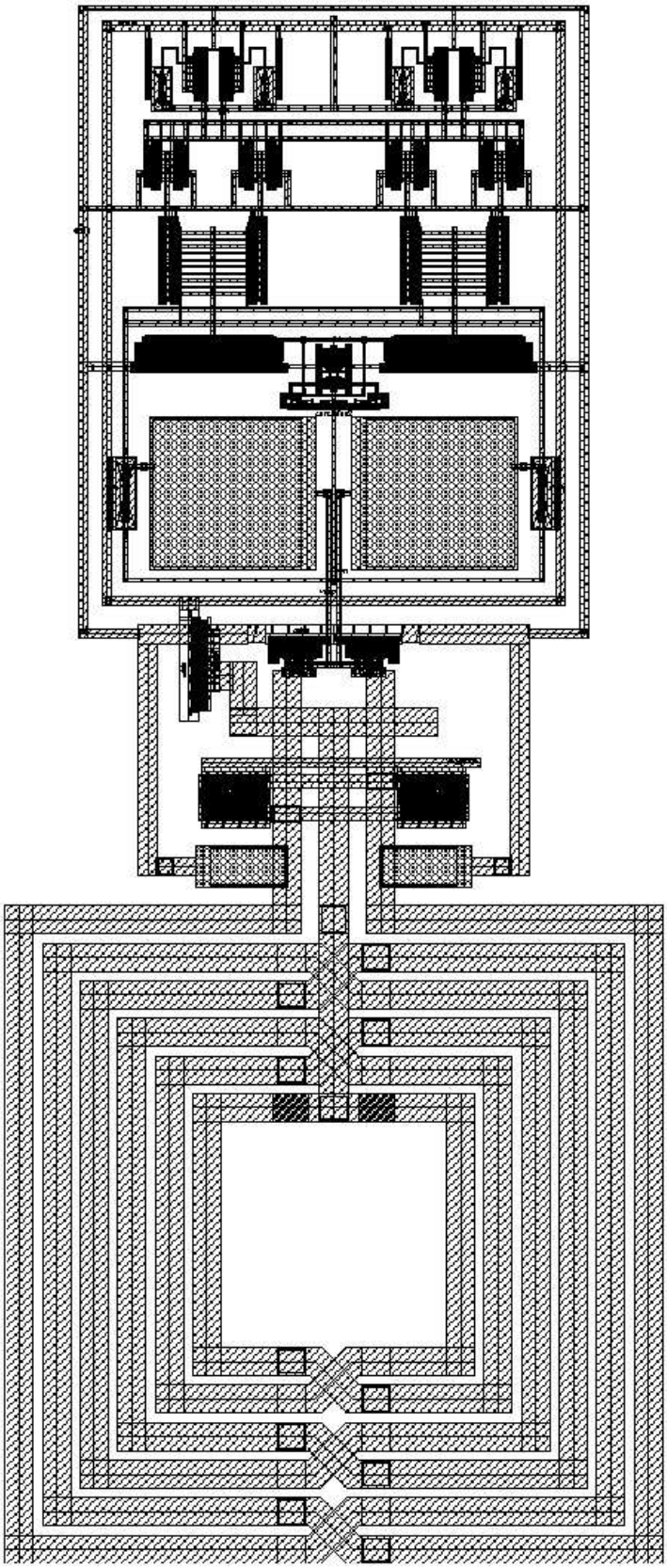

Figure 6.1: Layout of VCO and first divider

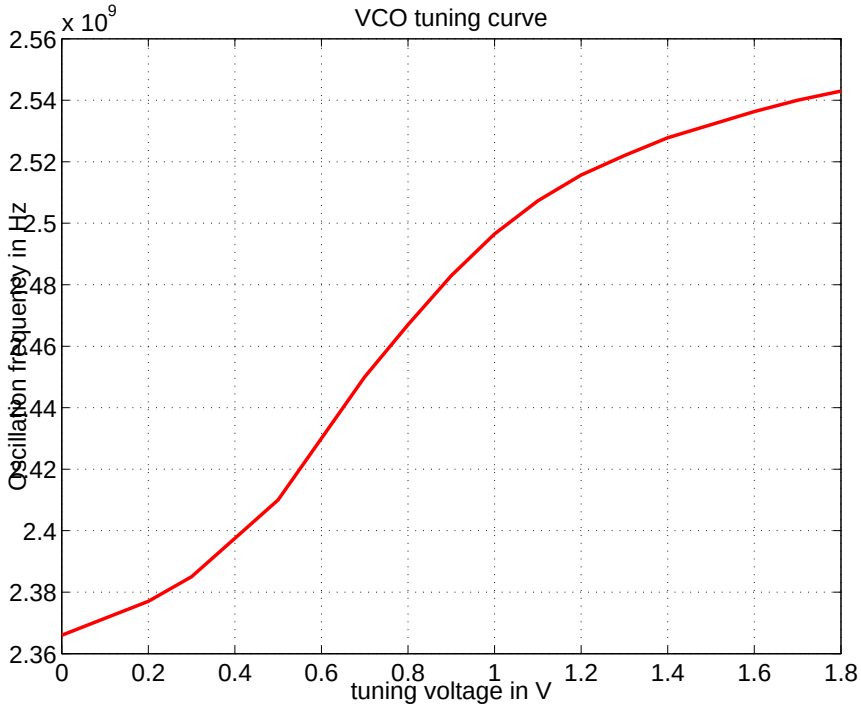

Figure 6.2: VCO tuning characteristics

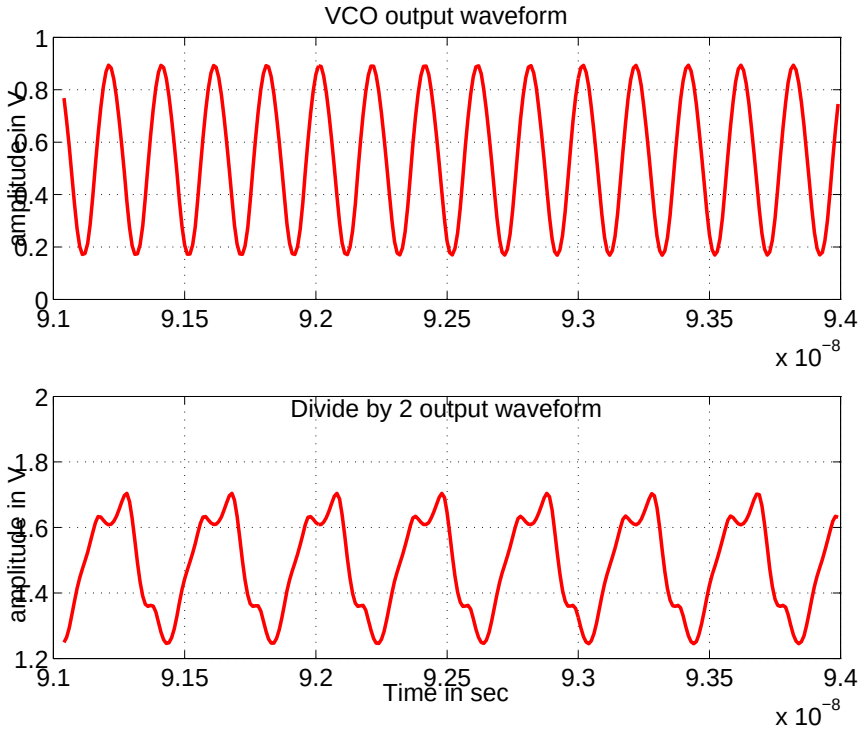

Figure 6.3: Time domain waveforms at VCO and first divider

There is a slight deviation from the linear slope at around 1 MHz. This is because of the RC filters that were used in the bias paths to shunt out noise from biasing transistors. As it turned out, these were the ones that were dominating phase noise when the filters were not used.

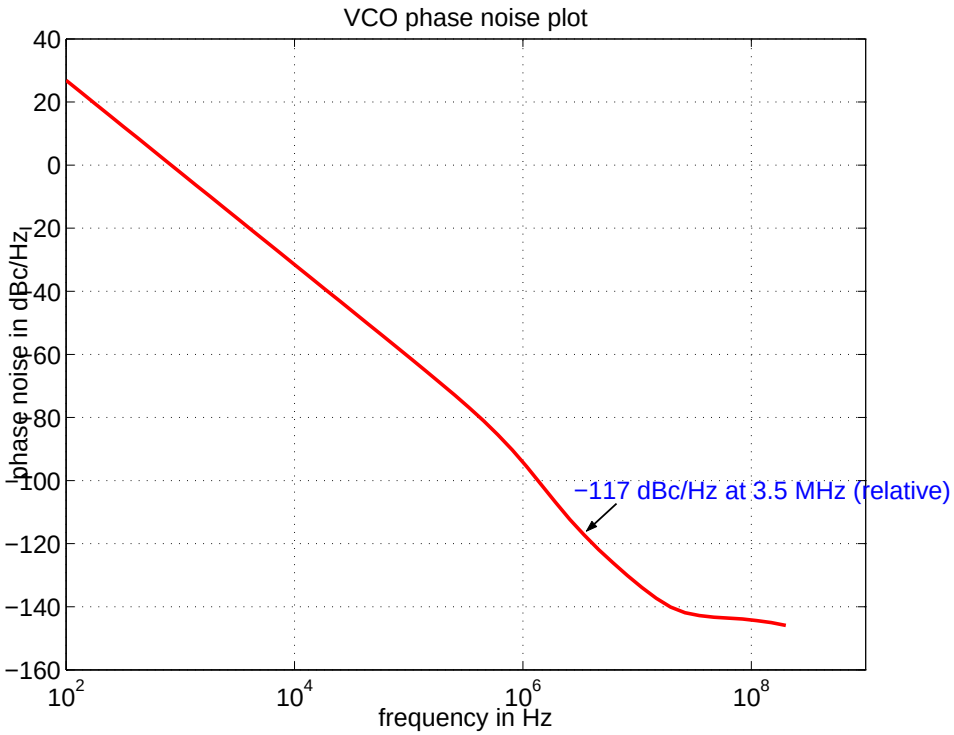

Figure 6.4: Output Phase Noise plot of the VCO

Simulations were done across all process corners and temperature variation from 0 to 80°C. The amplitude of oscillation varied from 660 mV (pp,SE) to 760 mV (pp,SE). Also, the constant g_m bias that was used to bias the divider helped to maintain the output amplitude approximately constant even at slow corners and high temperature. Phase noise variation across corners also was within acceptable range.

6.2.2 Divider Chain

The output voltage was maintained at around 350 mV pp,SE throughout the divider chain. The simulated output voltage plots(schematic view) of the successive

dividers in the chain are shown in Fig. 6.5. Here too, the divider was simulated across corners and temperature from 0 to 80°C.

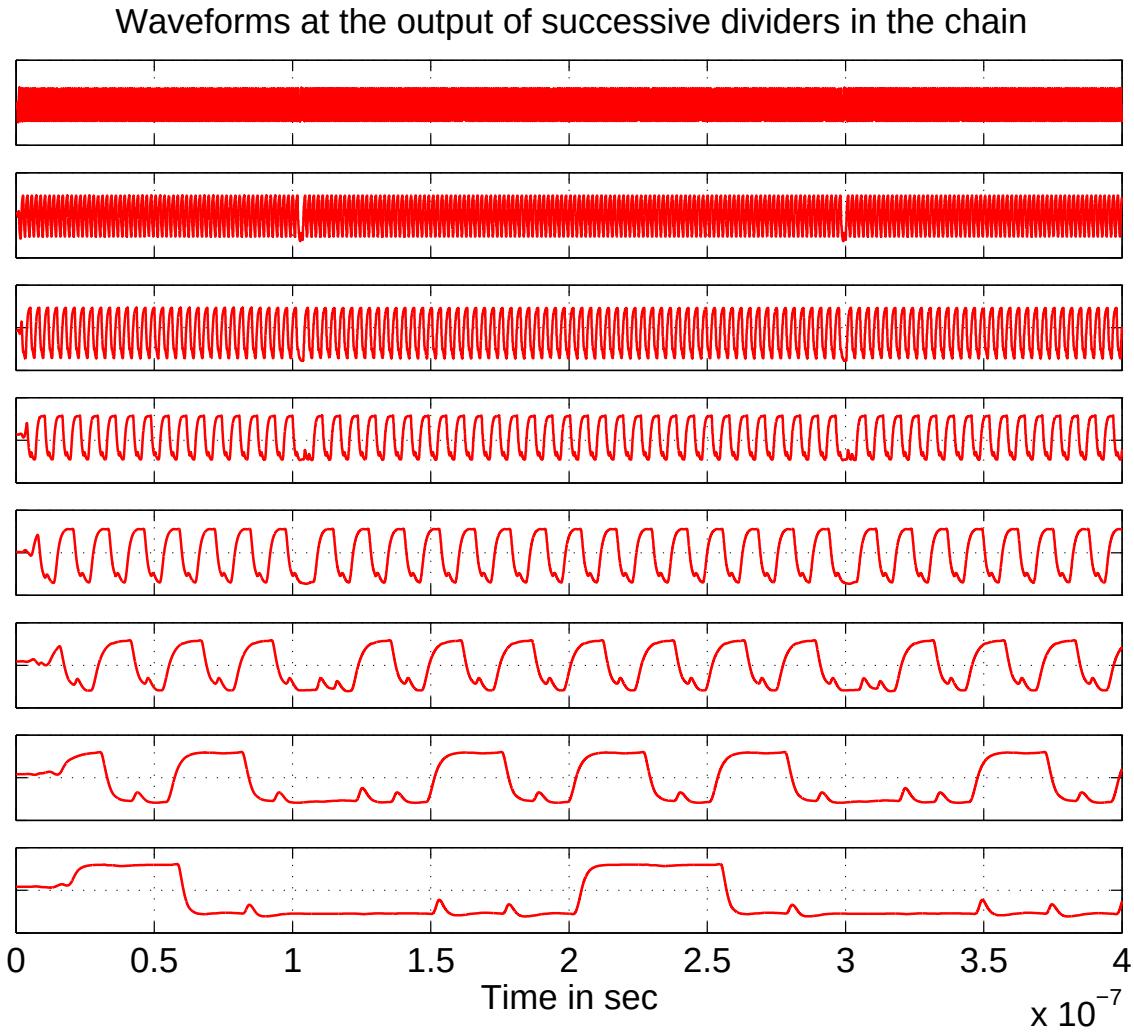

Figure 6.5: Divider Waveforms at successive stages in the chain

The noise contribution of the divider chain was measured using PSS+pnoise analysis. A pure sine wave was input to the first divider and the noise at the last divider output was measured. The phase noise plot obtained is shown in Fig. 6.6. The increase that we see around 3 MHz is because of sampling. The signal frequency is 5 MHz, so the phase noise plot above 2.5 MHz will be a mirror image of the plot below 2.5 MHz. We see that the phase noise at 3.5 MHz is around -116 dBc/Hz. This should be multiplied by the gain of this noise to output to get the divider contribution to the output noise. That is around -8 dB(refer to chapter 5). So, the phase noise at the output due to divider is around -124 dBc/Hz which

is again well below the required level.

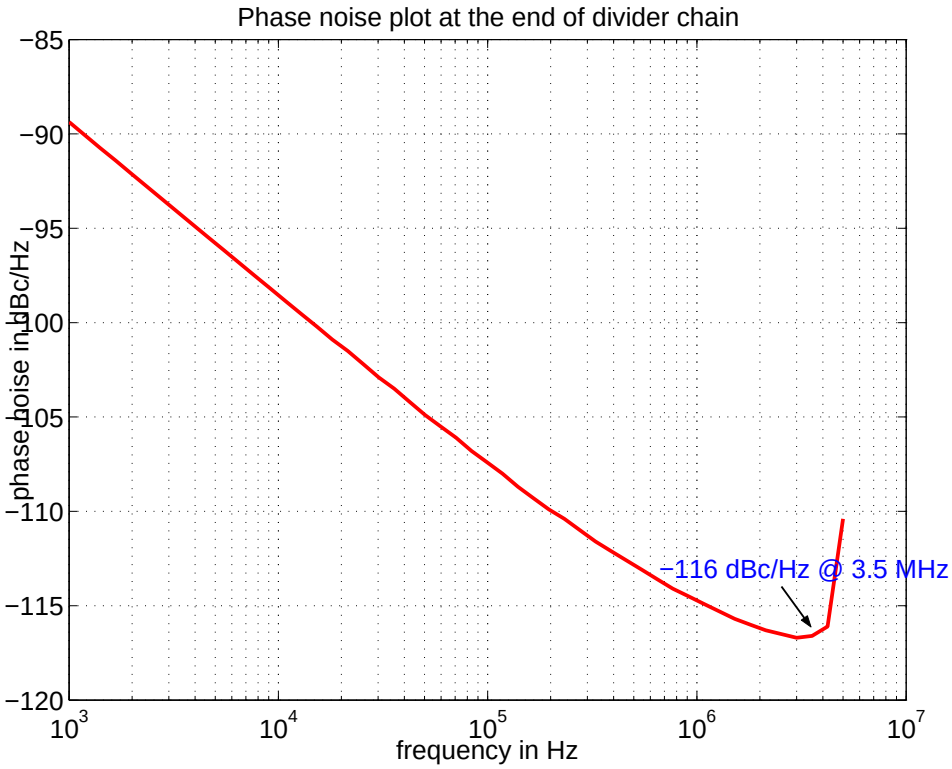

Figure 6.6: Phase noise at the output of last divider

6.2.3 Differential to Single-ended converter

The differential pair gain was around 2.5. The voltage swings from 0.9 to 1.8 V. The level converter shifts down the voltage so that the average is around 0.9 V, which is the switching threshold of the inverter. The waveforms are given in Fig. 6.7. Here again, the circuit was simulated across process and temperature.

6.2.4 PFD and Charge Pump

The PFD had a measured 'reset delay' of 11 ns.

The charge pump was designed to have UP and DN currents of 10 μA. The mismatch between UP and DN currents as the output voltage was varied from 0 to 1.8 V is given in Fig. 6.8.

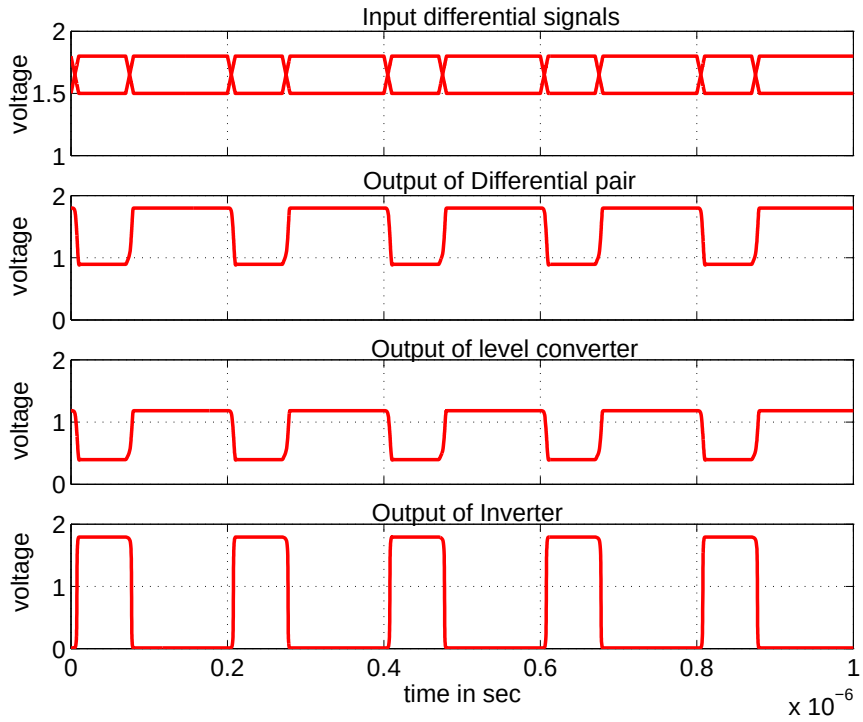

Figure 6.7: Waveforms in the differential to single-ended converter

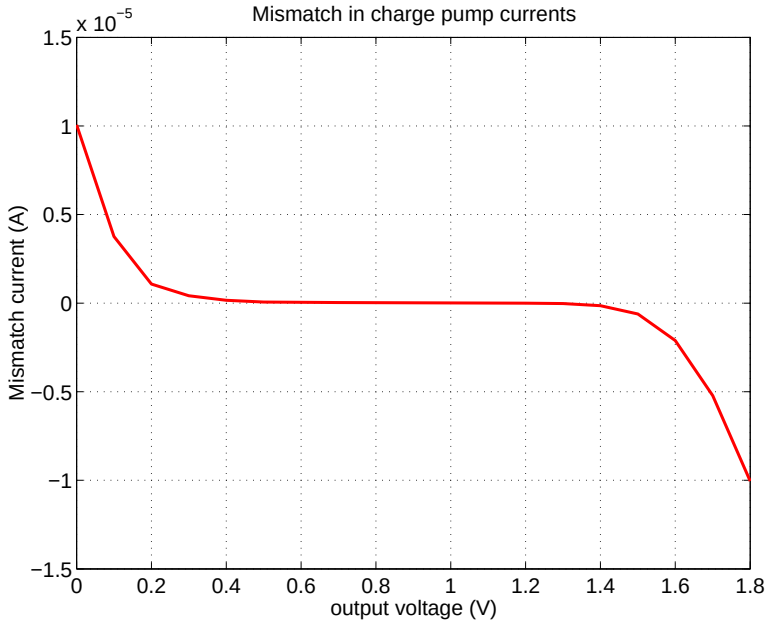

Figure 6.8: Mismatch between charge pump currents as a function of output voltage

One more simulation was done was to find out the spur levels at 5 MHz and 10 MHz due to mismatch in charge pump currents(the procedure was described in the noise analysis section of chapter 5). It was seen that the major contributor to

spurs was the spiky charge pump currents which come due to the sharp transitions of the UP and DN signals from the PFD. The spiky current was dominating till a mismatch of around 20%. The spur levels were found to be -68 dBV at 5 MHz and -73 dBV at 10 MHz. The DFT plot of the output spectrum is shown in Fig. 6.9

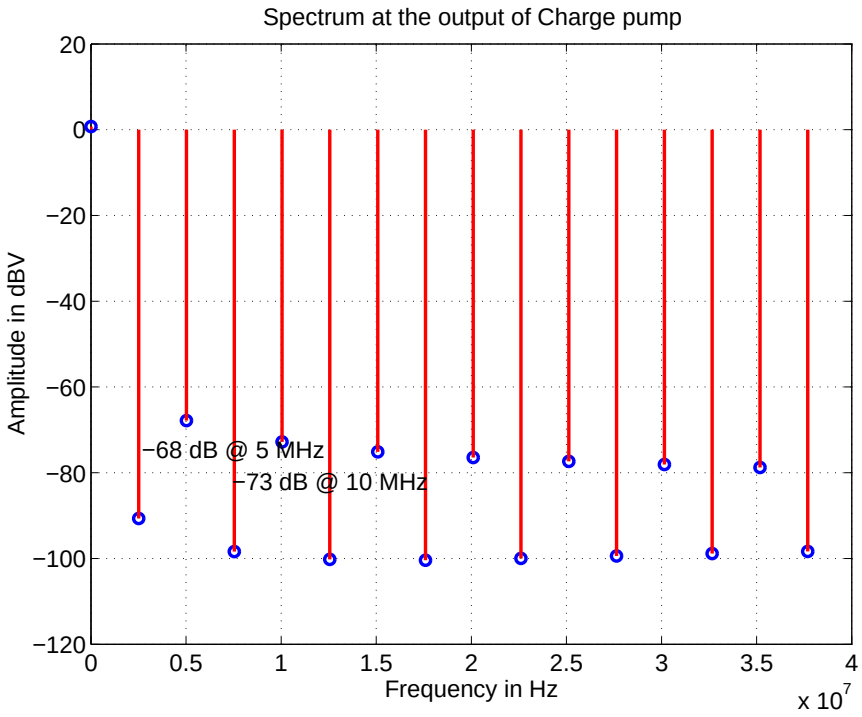

Figure 6.9: DFT at output of CP(Mismatch Spurs)

We have to add the respective gains to get the spurs at the output. At 5 MHz, we have

$$\begin{aligned} \phi_{spur-5MHz} &= V_{spur-5MHz} * \left(Gain^2\right) \\ &= (-68 + 29)dBc \\ &= -39dBc \end{aligned} \tag{6.1}$$

At 10 MHz, we have

$$\begin{aligned} \phi_{spur-10MHz} &= V_{spur-10MHz} * \left(Gain^2\right) \\ &= (-73 + 23)dBc \\ &= -50dBc \end{aligned} \tag{6.2}$$

So, we see that the spur requirement at 5 MHz(-20 dBc) is well satisfied, while the 10 MHz spur level just meets the requirement(-50 dBc).

6.3 Power Consumption

All measurements at tt, $55°C$

Table 6.1: Power consumption

Circuit	Current(A)	Power consumed(W)
VCO	1 m	1.8 m
First divider	1.8 m	3.24 m
Buffer	1.4 m	2.52 m
Divider chain	1.09 m	1.96 m
Differential to SE converter	22 μ	39.6 μ
PFD	23 μ(avg)	41 μ
CP	20 μ	36 μ
Bias circuits	350 μ	630 μ
Total	5.7 m	10 m

6.4 Measured PLL specifications

Table 6.2: PLL specifications

Measured quantity	Zigbee specification	measured value
phase noise	-92 dBc/Hz	-117 dBc/Hz
settling time	196 μs	110 μs
spur@5 MHz	-20 dBc	-39 dBc
spur@10 MHz	-50 dBc	-50 dBc

6.5 Conclusion

A low power PLL has been designed to be used as the frequency synthesizer in a Zigbee transceiver. The power consumption is minimal, in line with the

requirements of the Zigbee transceiver. All the relevant specifications laid down by Zigbee standard have been met. The circuit doesn't use any off-chip components and uses only one on-chip inductor.

The component which has used the most power is the 5 GHz to 2.5 GHz divider. So, a solution which uses a quadrature VCO or an ILFD divider in place of the CML divider might result in some power reduction. But these circuits require more inductors, which will increase the area. Accurate characterization of inductors is another issue.

APPENDIX A

Modelling the accumulation varactor using verilog-A

A.1 Relation between channel charge and surface potential

Fig. A.1 shows the accumulation varactor. The p-substrate is grounded and the gate and n-well form the two pins of the capacitor.

Fig. A.2 shows the 2-D cross section of just the relevant portions of the varactor

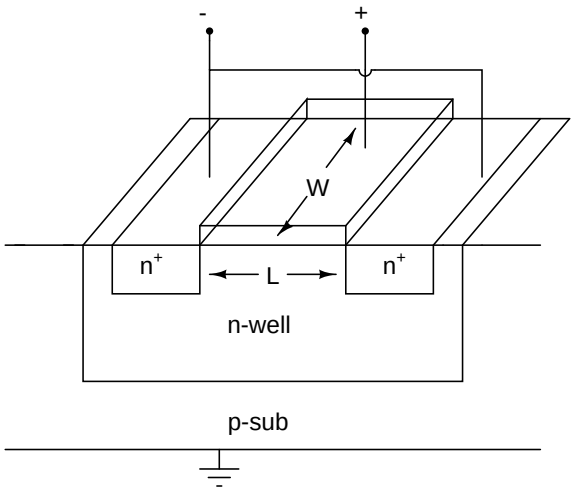

Figure A.1: Accumulation Varactor

viz, the poly, oxide and n-well. ψ_s denotes the surface potential. Using Kirchoff's law, charge conservation and Poission's equation in the y-direction, we can write the relation between the channel charge per unit area(Q_c) and ψ_s[11].

$$Q_c = \sqrt{2q\xi_s N_D}\sqrt{\left(\phi_t e^{\frac{\psi_s}{\phi_t}} - \psi_s - \phi_t\right) + e^{\frac{2\phi_f}{\phi_t}}\left(\phi_t e^{\frac{-\psi_s}{\phi_t}} + \psi_s - \phi_t\right)} \qquad \text{(A.1)}$$

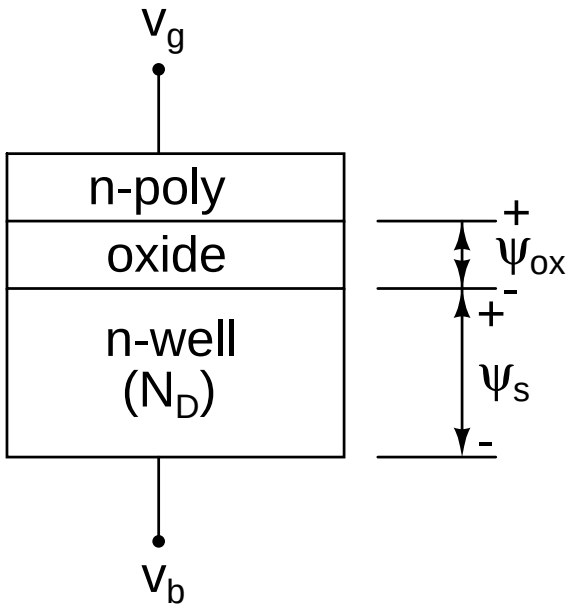

Figure A.2: Accumulation Varactor 2-D cross-section

where q is electron charge, ξ_s is silicon permittivity, ϕ_t is thermal voltage and ϕ_f is fermi potential (negative here). We should be careful with the sign of Q_c. In the accumulation region, ψ_s is positive and Q_c is negative. In depletion and inversion, ψ_s is negative and Q_c is positive.

In our case though, the MOS cannot go to inversion region because there are no source and drain regions to provide the holes(especially at such high frequencies). For accumulation and depletion regions, the equation simplifies to

$$Q_c = \sqrt{2q\xi_s N_D}\sqrt{\left(\phi_t e^{\frac{\psi_s}{\phi_t}} - \psi_s - \phi_t\right)} \tag{A.2}$$

A.2 Capacitance derivation

Now that we have the channel charge equation, it is fairly straight forward to get the capacitance of the varactor as a function of V_{gb}. Using Kirchoff's voltage law

across the MOS, we have

$$\begin{aligned} V_{gb} &= \phi_{MS} + \psi_s + \psi_{ox} \\ &= \phi_{MS} + \psi_s + \frac{Q_G}{C_{ox}} \\ &= \phi_{MS} + \psi_s - \frac{Q_c + Q_f}{C_{ox}} \\ &= V_{fb} + \psi_s - \frac{Q_c}{C_{ox}} \end{aligned} \tag{A.3}$$

where Q_G is the gate charge per unit area, Q_f is the parasitic oxide charge per unit area, ϕ_{MS} is the work function difference between the poly and n-well and V_{fb} is the flatband voltage defined by $\left(\phi_{MS} - \frac{Q_f}{C_{ox}}\right)$. Further, we know that for a n-poly - n-well combination, $\phi_{MS} \approx 0$. For modern processes, Q_f tends to be very small. So we can assume $V_{fb} \approx 0$ in our modelling.

$$V_{gb} = \psi_s - \frac{Q_c}{C_{ox}} \tag{A.4}$$

Now, we can combine this equation with the channel charge equation we derived to get an expression for V_{gb} in terms of ψ_s. Then, the capacitance can be simply calculated as

$$C_{var} = \frac{dQ_G}{dV_{gb}} \tag{A.5}$$

Here $Q_G = C_{ox}\left(V_{gb} - \psi_s - \phi_{MS}\right)$. This has both V_{gb} and ψ_s as variables, but we know the relation between the two. So, mathematically, it is possible to find out the capacitance.

A.3 Verilog-A modelling

In the last section, we concluded that we can find out the capacitance mathematically. But the problem there is that there is no closed form expression for ψ_s in terms of V_{gb}, which is needed to find out the capacitance. So, we have to make our verilog-A model to find out the expression for us.

It turns out that we can use a very simple trick to solve this problem. We know that Q_c and ψ_s are related by eqn. A.2. Let us denote this relation by $Q_c = f(\psi_s)$.

Also,

$$Q_c = C_{ox}\left(V_{gb} - V_{fb} - \psi_s\right) \tag{A.6}$$

$$\Rightarrow f(\psi_s) = C_{ox}\left(V_{gb} - V_{fb} - \psi_s\right) \tag{A.7}$$

$$\Rightarrow V_{gb} = V_{fb} + \psi_s + \frac{f(\psi_s)}{C_{ox}} \tag{A.8}$$

Making $V_{fb} = 0$,

$$V_{gb} = \psi_s + \frac{f(\psi_s)}{C_{ox}} \tag{A.9}$$

Now, this equation was implemented in Verilog-A using the circuit given in Fig. A.3. Note that f(.) in the circuit is same as the f(.) in eqn. A.9. The current flowing through the circuit is $f(\psi_s)$. So, the voltage V_x must be ψ_s(according to eqn. A.9). Now, we can use this voltage to build the actual capacitor model. The capacitor

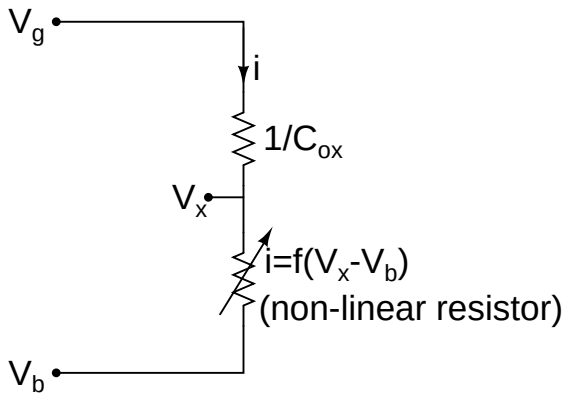

Figure A.3: Verilog-A circuit to generate ψ_s

was implemented in the current domain(as opposed to the charge domain) using verilog-A. The current flowing through the capacitor can be written as

$$I_{cap} = \frac{dQ'_G}{dt} \tag{A.10}$$

where $Q'_G = Q_G \times (W.L)$ where (W.L) is the capacitor area and $Q_G = C_{ox}\left(V_{gb} - \psi_s - \phi_{MS}\right)$. We need not know the value of ϕ_{MS} because it is constant with time and we are differentiating Q_G. One final implementation issue is that the verilog circuit takes in extra current. So, this current was measured and cancelled out by an external

controlled current source.

REFERENCES

[1] M. Tiebout, *Low Power VCO Design in CMOS*. Springer, 2006.

[2] J. Craninckx and M. S. J. Steyaert, "A fully integrated CMOS DCS-1800 frequency synthesizer," *IEEE Journal of Solid-state Circuits*, vol. 33, no. 12, pp. 2054–2065, 1998.

[3] U. of California, Berkeley, "ASITIC: Analysis and simulation of spiral inductors and transformers for ICs."

[4] MIT, "Fasthenry."

[5] A. S. Porret, T. Melly, C. C. Enz, and E. A. Vittoz, "Design of high-Q varactors for low-power wireless applications using a standard CMOS process," *IEEE Journal of Solid-state Circuits*, vol. 35, no. 3, pp. 337–345, 2000.

[6] H. R. Rategh and T. H. Lee, "Superharmonic Injection-Locked Frequency Dividers," *IEEE Journal of Solid-state Circuits*, vol. 34, no. 6, pp. 813–821, 1999.

[7] A. Mazzanti, P. Uggetti, and F. Svelto, "Analysis and design of Injection-Locked LC Dividers for Quadrature generation," *IEEE Journal of Solid-state Circuits*, vol. 39, no. 9, pp. 1425–1433, 2004.

[8] C. Vaucher, I. Ferencic, S. Locher, M. Sedvallson, U. Voegeli, and Z. Wang, "A family of low-power truly modular programmable dividers in standard 0.35 um CMOS technology," *IEEE Journal of Solid-state Circuits*, vol. 35, no. 7, pp. 1039–1045, 2000.

[9] A. Wafa and A. Ahmed, "High-speed RF multi-modulus prescaler architecture for fractional-N PLL frequency synthesizers," in *Proceedings of the 2004 International Symposium on Circuits and Systems*, IEEE, May 2004.

[10] W. Rhee, "Design of high-performance CMOS charge pumps in phase-locked loops," in *Proceedings of the 1999 International Symposium on Circuits and Systems*, IEEE, July 1999.

[11] Y. Tsividis, *Operation and Modelling of The MOS Transistor*. Oxford University Press, 1999.