

Design analysis of Frequency Divider circuit for Phase-Locked Loop application using 65nm CMOS Technology

A DEP report (Course code: CP 301) submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Technology

by

Lokesh Jassal
(Entry No. 2021EEB1185)

Under the guidance of
Dr. Mahendra Sakare



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROPAR

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Lokesh Jassal

Entry No. 2021EEB1185

Date: 09-05-2024

Abstract

This work presents the design of a frequency divider circuit implemented in 65nm CMOS technology for phase-locked loop purposes. The divider targets an input frequency range of 4.6GHz and aims to achieve a modulus of 460. The design focuses on low power consumption, high operating frequency by implementing a fast End of Count architecture for counters used in frequency divider. Then design is put under test in an ideal PLL generated using VerilogA and a conventional PLL. The circuit schematic is presented along with simulations to verify its functionality.

Contents

Abstract

List of Tables	ii
List of Figures	iii
1 Introduction	1
2 Literature survey	2
3 Dual Modulus Prescalar	3
4 Program Counter	6
5 Swallow Counter	9
6 Pulse Swallow Divider	11
7 Simulations	14
7.1 Ideal PLL	15
7.2 Conventional PLL	17
8 Conclusion and Future Work	23
8.1 Conclusion	23
8.2 Future work	23

List of Tables

3.1	MC=0, divide by 4	4
3.2	MC=1, divide by 5	4

List of Figures

3.1	4/5 Prescalar	3
3.2	NAND and NOT	5
3.3	Clocked-CMOS D-flipflop	5
4.1	Program Counter	6
4.2	3-bit Synchronous Counter	7
4.3	Set-Reset D-flipflop	7
4.4	EOC-100	8
5.1	Swallow Counter	9
5.2	EOC-60	10
6.1	Pulse Swallow Divider	11
6.2	Frequency of MC vs time	12
6.3	Modulus Control vs time	13
7.1	Phase-Locked Loop	14
7.2	Transient of Output Frequency in Ideal PLL	15
7.3	Transient of Control Voltage in Ideal PLL	16
7.4	PFD using 65nm CMOS Technology	17
7.5	Charge Pump and Loop Filter using 65nm CMOS Technology	18
7.6	VCO using 65nm CMOS Technology	19
7.7	Transient of Output Frequency in Conventional PLL-4GHz	20
7.8	Transient of Control Voltage in Conventional PLL-4GHz	21
7.9	Transient of Feedback Frequency in Conventional PLL	22

Chapter 1

Introduction

Frequency dividers employed in feedback path of Phase-Locked Loops are necessary for high locking frequencies from relatively low frequency clocks generated using crystal oscillators. Crystal Oscillators provides the cleanest clocks but they synthesize clocks at low frequencies and frequency synthesized can not be easily manipulated according to our needs as frequency synthesized by crystal oscillator is highly hardware dependent. To serve this purpose of high frequency clock requirements and varying frequency at will there comes the concept of programmable dividers in feedback of PLL. By passing the digital input to programmable divider its modulus can be varied and hence the output frequency.

In this work a modulus 460 divider using Pulse Swallow Divider is presented to operate at 4.6GHz. However, divider presented in this work has a fixed modulus but this can be scaled to a programmable divider with just a bit of modification in end of count architecture of Swallow counter. Improving this end of count architecture and a trade off between synchronous and asynchronous counter for speed and complexity is the highlight of this work.

Chapter 2

Literature survey

A variety of frequency dividers are available including Miller divider, Voucher divider, Injection-Locked dividers, Pulse Swallow dividers along with various methodologies to execute them like clocked-CMOS, differential latching, Current-Mode logic.

These many architectures were purposed in textbook by B. Razavi out of which Pulse Swallow divider was the most soothing one. Bottle-neck of this divider is end of count architecture of counters employed in divider.

Work of I. Som, S. Sarangi, and T. K. Bhattacharyya served the way for executing a high speed EoC architecture by pre detecting the state of slow asynchronous counter and integrating the same with state of fast synchronous counters for high speed reset trigger for counters in divider.

Chapter 3

Dual Modulus Prescalar

A dual modulus prescalar is a frequency divider block which either divides the input frequency by N or N+1 depending on modulus control. In this work we have made a 4/5 prescalar, refer to fig. 3.1

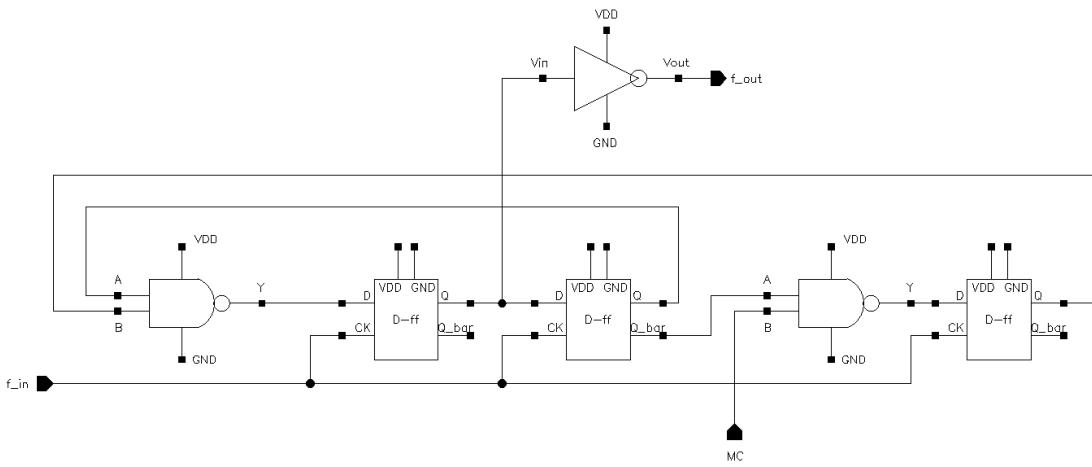


Figure 3.1: 4/5 Prescalar

It consists of three synchronous D flipflops for high frequency operations and combinational logic with MC (Modulus Control) to dictate whether it should divide by 4 or 5. When MC is set to 0 this block divides by 4, and when MC is set to 1 it divides by 5.

Truth Table when MC is set to 0, same pattern repeats after four clock pulses

Q0	Q1	D0	D1	Q0*	Q1*
0	0	1	0	1	0
1	0	1	1	1	1
1	1	0	1	0	1
0	1	0	0	0	0

Table 3.1: MC=0, divide by 4

Truth Table when MC is set to 1, same pattern repeats after five clock pulses

Q0	Q1	Q2	D0	D1	D2	Q0*	Q1*	Q2*
0	0	1	1	0	0	1	0	0
1	0	0	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	0	1

Table 3.2: MC=1, divide by 5

where Q0 is LSB, and Qi* represents corresponding next state

We take inverted output from Q0 to get a clean wave and to drive the further connected loads. From this topology we wouldn't get 50% duty cycle when it is dividing by 5.

Refer to fig. 3.2 for NAND and NOT execution.



Figure 3.2: NAND and NOT

We have used Clocked-CMOS technique for making D-flipflop. Refer to fig. 3.3 for circuitry. This technique would work for our current requirement of operation at 4.6GHz, but for further high speed operations we should look for faster topologies like CML (Current Mode Logic). Here we presisted with clocked-cmos because CML consumes a lot of power as it draws static current.

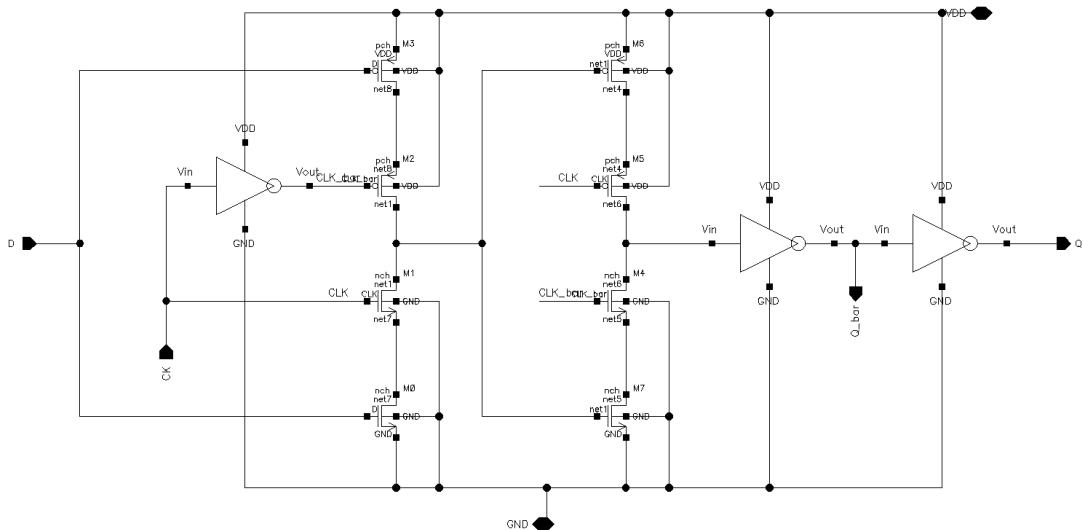


Figure 3.3: Clocked-CMOS D-flipflop

Chapter 4

Program Counter

A program counter in a Pulse Swallow divider is a fixed-modulus divider, for our purpose that fixed modulus is 100. Hence, this block resets after every 100 input clock pulses. Counter designed, refer to fig. 4.1 consists of a synchronous 3-bit counter for LSB further connected to a 4-bit asynchronous counter, and an End of Count (EOC) architecture to reset the counter. This EOC architecture is the bottle neck of our work for high frequency operations.

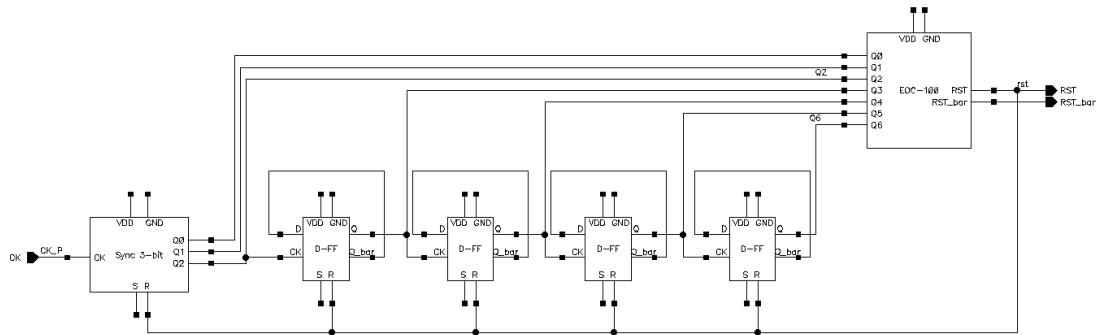


Figure 4.1: Program Counter

At first, let's discuss counter architecture. As mentioned earlier, our up-counter is a combination of synchronous 3-bit for LSB and asynchronous 4-bit. Later, in EOC architecture you will realise we need high speed generation of three LSBs that are Q2 Q1 Q0. We know synchronous counters are faster than asynchronous, but are complex to design and consume a lot more hardware. So, in trade off between speed and complexity we took a combination of both synchronous and asynchronous counters designed with set-reset D-flipflops. Refer to fig. 4.2 for design of 3-bit synchronous counter.

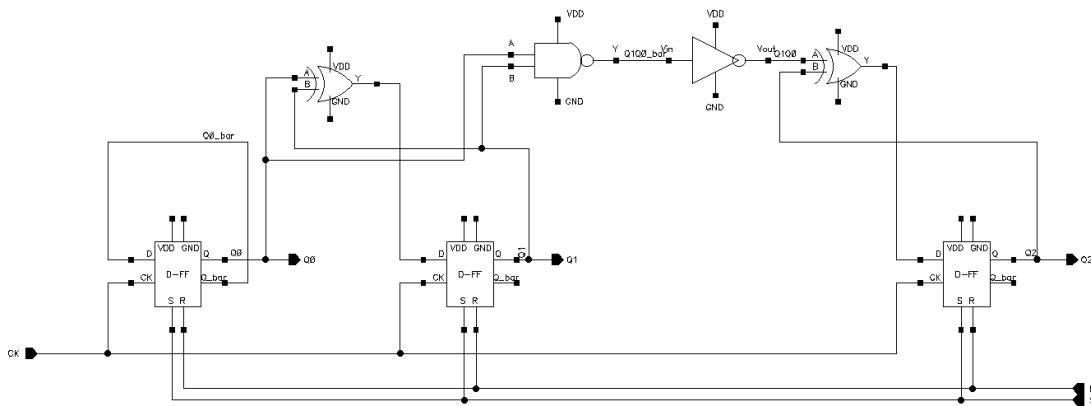


Figure 4.2: 3-bit Synchronous Counter

Refer to fig. 4.3 for design of Set-Reset D-flipflop using Clocked-CMOS Technology.

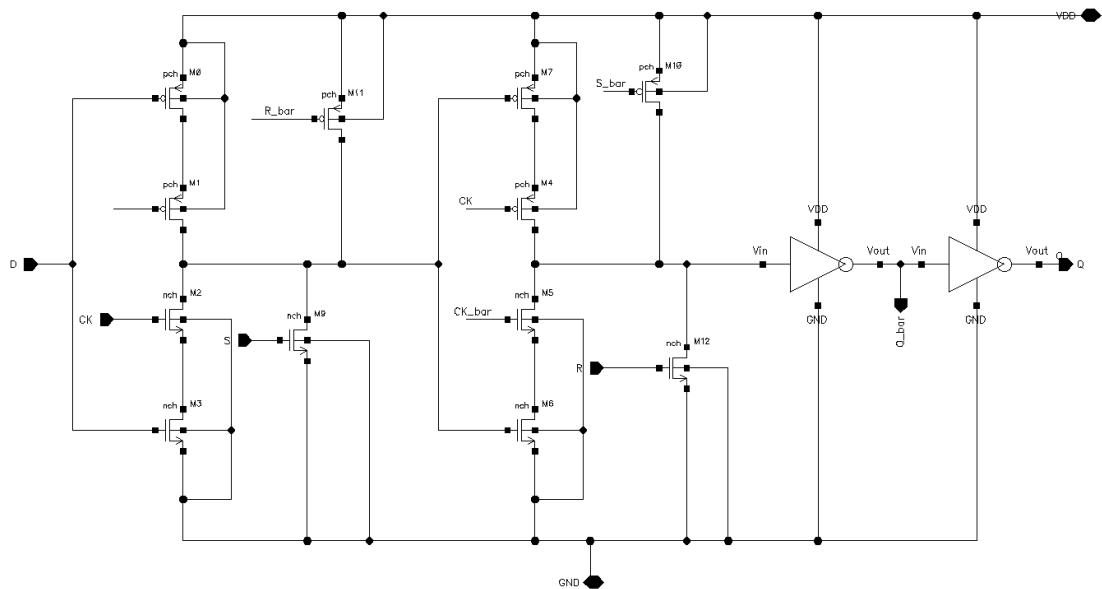


Figure 4.3: Set-Reset D-flipflop

EOC Architecture

EOC takes current state of counter as input that is Q6...Q2Q1Q0 and generates a reset for counter if it has reached the end of count that is 100 for this case. We need this EOC to generate reset signal as quickly as possible. Faster this EOC generates, higher the frequencies at which counter can be operated.

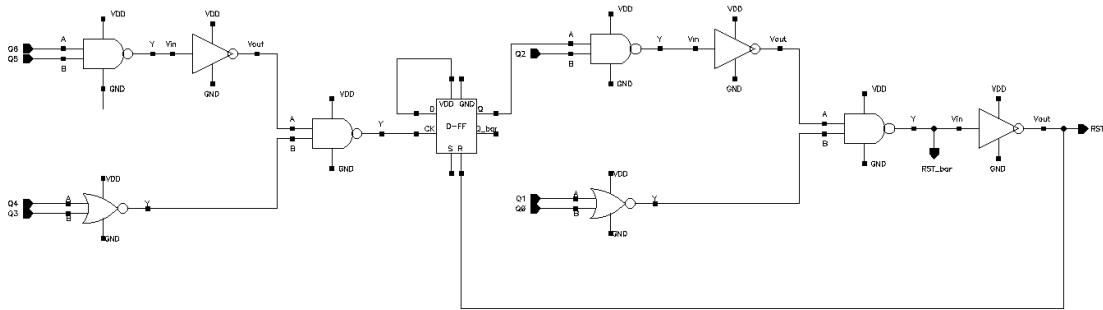


Figure 4.4: EOC-100

We are designing a modulo 100 divider which in binary is 1100100. Since, it is an up-counter it will go through 1100000 before getting to end of count, what our architecture does is it pre-detects MSB 1100 which are states of slow asynchronous counter and store this detected state as a single bit in D-flipflop whose D is connected to VDD (always high). $Q_6Q_5Q_4Q_3 = 1100$ is detected using combinational logic and is used to give falling edge to our clock of D-flipflop (negative-edge triggered) to set it high, refer to fig. 4.4. Hence, $Q_6Q_5Q_4Q_3 = 1100$ is collectively represented by a single bit which is stable way before getting end of count.

Now, this set bit along with LSBs Q2Q1Q0 is used to generate final reset trigger for our counter. As already discussed Q2Q1Q0 are outputs of fast synchronous counter and we already had stable set bit for slow asynchronous counter, this increased speed of reset generation by EOC. We executed combinational logic to detect $Q_s Q_2 Q_1 Q_0 = 1100$, where Q_s represents the output of flip flop which is high when 1100 at MSB has passed. Once end of count is reached it resets the D-flipflop in EOC and also resets all the flipflops in Program Counter.

Chapter 5

Swallow Counter

Generally modulus of Swallow Counter is programmable by passing digital input to EOC architecture of Swallow Counter. But for our case we have fixed its modulus to 60 hence, it works almost similar to previously discussed Program Counter except for a change that it generates MC (modulus control) for Dual-Modulus Prescalar and it is reset in two cases, one when counter reaches its end of count and other when Program Counter resets.

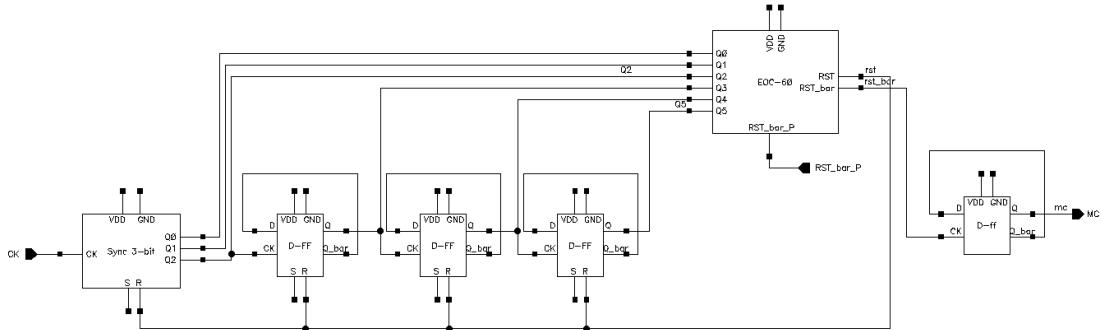


Figure 5.1: Swallow Counter

In fig. 5.1 you can observe counter architecture is similar to that of program counter. EOC of Swallow counter contains additional input from program counter as swallow counter should reset whenever program counter resets. Later, it will be explained that modulus control MC is to be inverted every time Swallow Counter resets. EOC architecture outputs a trigger when it needs to reset, this trigger is fed to clock of D-ff whose output is set to invert whenever it receives a falling edge and hence, MC is generated.

EOC Architecture

It is similar to that of Program Counter. Methodology is same but combinational logic is different because end of count for Swallow Counter is 60 which in binary is 111100. Again same concept output of slow asynchronous counter $Q_5Q_4Q_3 = 111$ is pre detected and stored in D-ff which is again passed to combinational logic with outputs of fast synchronous counter $Q_2Q_1Q_0$ as inputs to generate reset triggers. One change here is that it has additional input in form of reset-bar to dictate Swallow Counter should reset whenever a reset comes from Program Counter. See fig. 5.2 this reset-bar is in NAND with output of combinational that is whenever reset-bar goes low EOC triggers a reset.

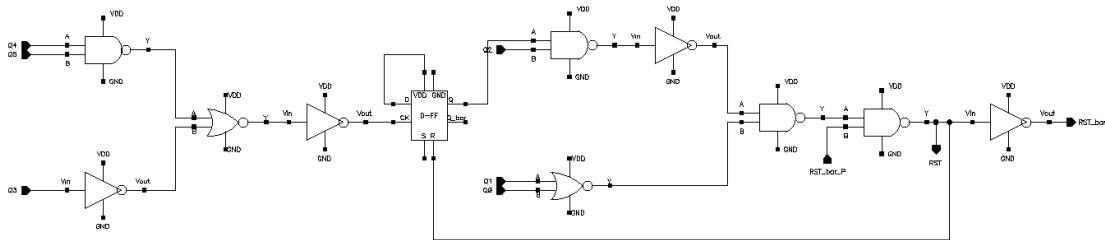


Figure 5.2: EOC-60

Chapter 6

Pulse Swallow Divider

A Pulse Swallow Divider is formed by integrating Dual-Modulus Prescalar ($N/N+1$), Program Counter (modulus P) and Swallow Counter (modulus S), P is greater than S see fig. 6.1.

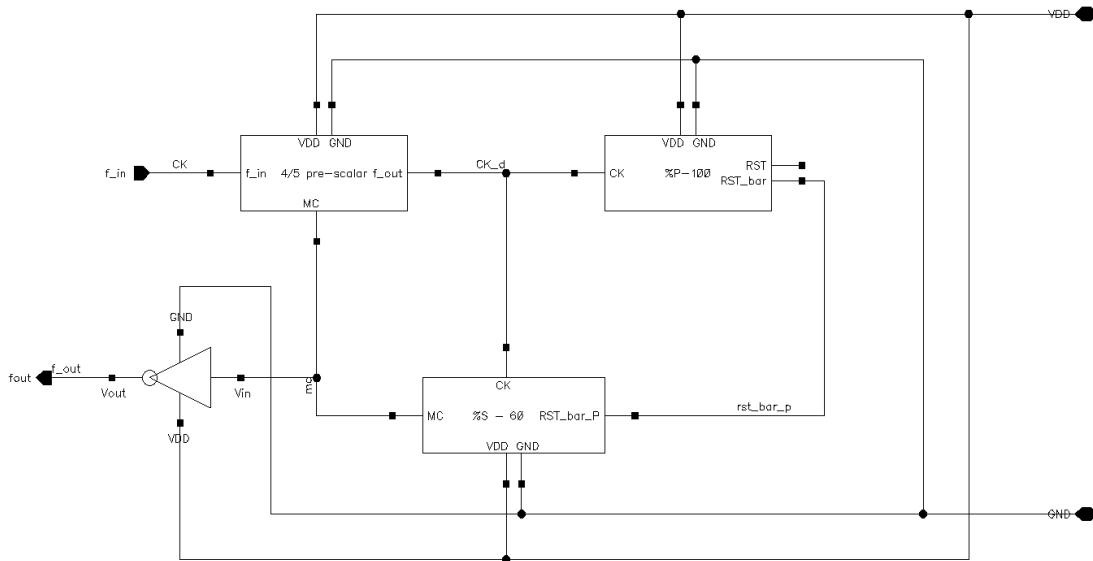


Figure 6.1: Pulse Swallow Divider

Initially MC is 1, hence Prescalar takes $N+1$ input pulses to output one pulse which is fed to program counter and swallow counter until swallow counter resets and MC is inverted. So, we can say initially when MC is 1, Pulse Swallow Divider take $(N+1)(S)$ input pulses to reset Swallow Counter for first time and invert MC to 0. Till now, S pulses has also passed to program counter and it will take $(P-S)$ more pulses to reset. As already discussed when end of count reaches at program counter it resets both program counter and swallow counter. For this

P-S pulses it will take $(N)(P-S)$ pulses at input of Prescalar (MC is 0). After end of count has reached at program counter it resets both counters and MC is again inverted to 1 and same cycle repeats. Hence, it can be concluded that to generate one complete pulse at MC it takes,

$$\text{Total input pulse} = (N + 1)(S) + (N)(P - S) = NP + S \quad (6.1)$$

$$f_{\text{out}} = \frac{f_{\text{in}}}{NP + S} \quad (6.2)$$

In our design $N=4$, $P=100$, and $S=6$

$$f_{\text{out}} = \frac{f_{\text{in}}}{NP + S} = \frac{f_{\text{in}}}{4 \cdot 100 + 60} = \frac{f_{\text{in}}}{460} \quad (6.3)$$

We have simulated this Pulse Swallow Divider in Cadence Virtuoso, an input clock of frequency 4.6GHz is given to Pulse Swallow Divider and output is taken from modulus control. Frequency of MC is stable at 10MHz which is expected ($4.6\text{GHz}/460 = 10\text{MHz}$) see fig. 6.2

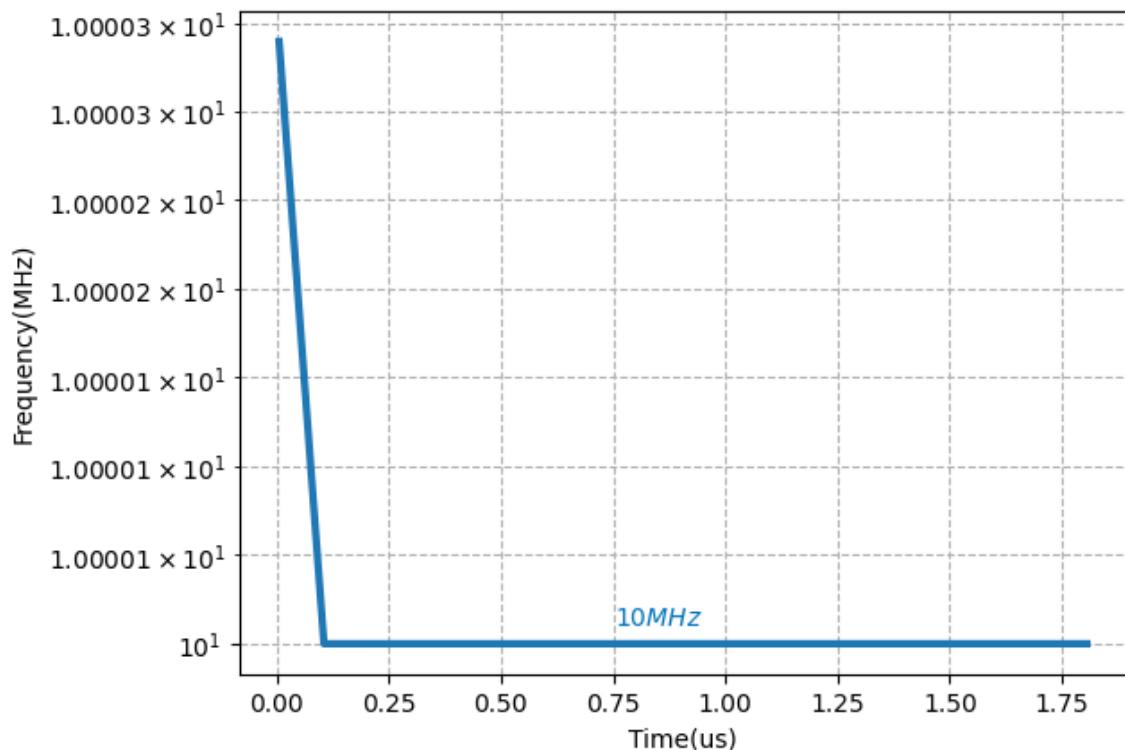


Figure 6.2: Frequency of MC vs time

Modulus Control with time is plotted below pulsating at 10MHz

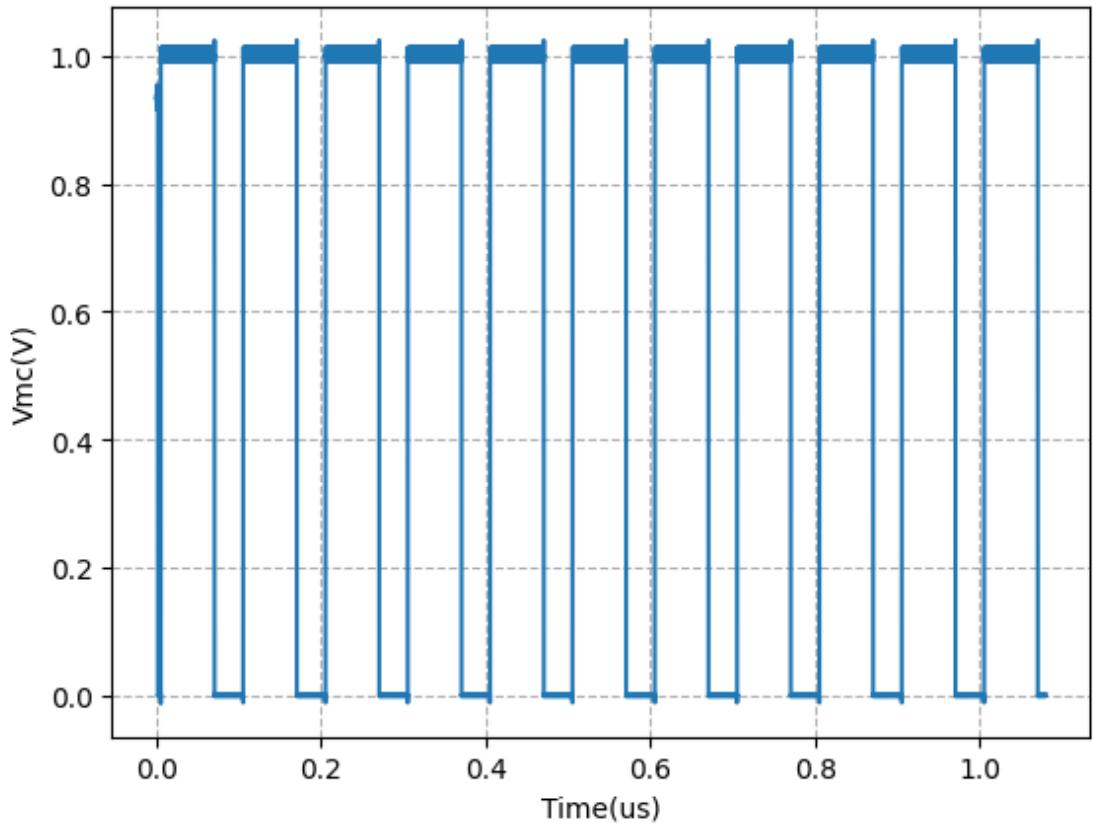


Figure 6.3: Modulus Control vs time

Chapter 7

Simulations

We have designed the frequency divider for PLL purposes. A typical Phase-Locked Loop (PLL) consists of Phase Frequency Detector (PFD), Charge Pump, Loop Filter, VCO and finally our Frequency Divider in feedback path. In fig. 7.1 loop filter is included in charge pump (cp) itself.

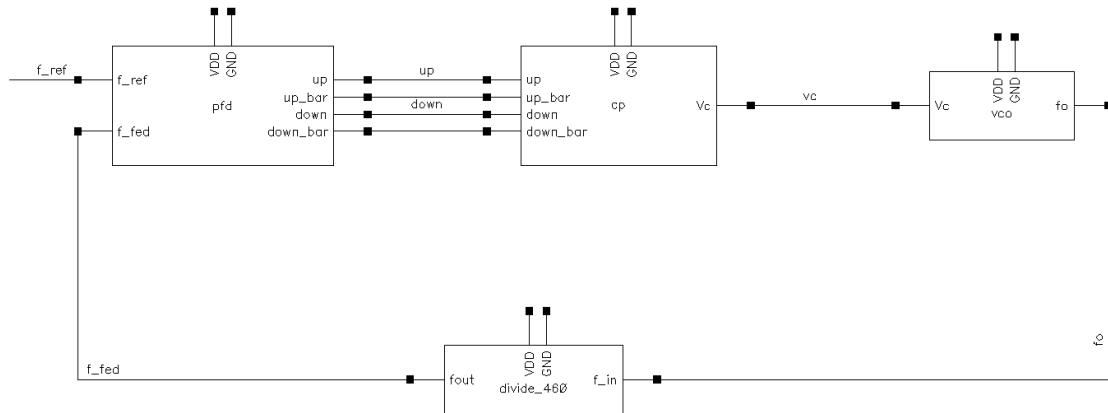


Figure 7.1: Phase-Locked Loop

We have put our design under test in an ideal PLL generated using VerilogA as well as in a conventional PLL designed using 65nm CMOS Technology to check whether PLL locks or not. Following are the simulation results depicting control voltage V_c and output frequency f_{out} .

7.1 Ideal PLL

An ideal PLL is generated using VerilogA, and frequency divider of ideal PLL is replaced with our designed divider keeping other blocks the same. A reference clock of 5MHz is given to PLL divider is of modulus 460 so output frequency of PLL should lock to $(5\text{MHz})(460)$ that is 2.3GHz.

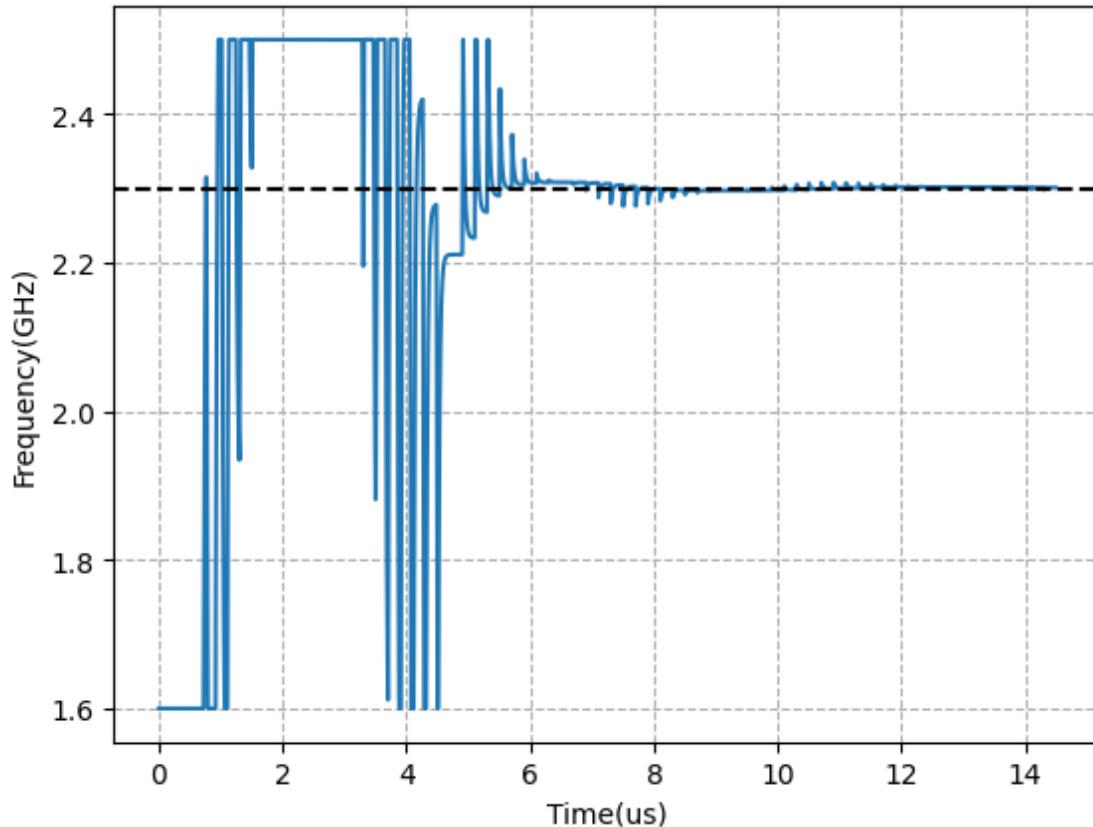


Figure 7.2: Transient of Output Frequency in Ideal PLL

VCO is having a maximum frequency limit of 2.5GHz this clips the output frequency at 2.5GHz and we can observe PLL eventually locking to 2.3GHz in fig. 7.2 .

Control Voltage (V_c) generated by PFD and charge pump feeds the VCO. VCO output frequency is directly proportional to V_c and hence we observe a similar pattern in plot of V_c in fig. 7.3 .

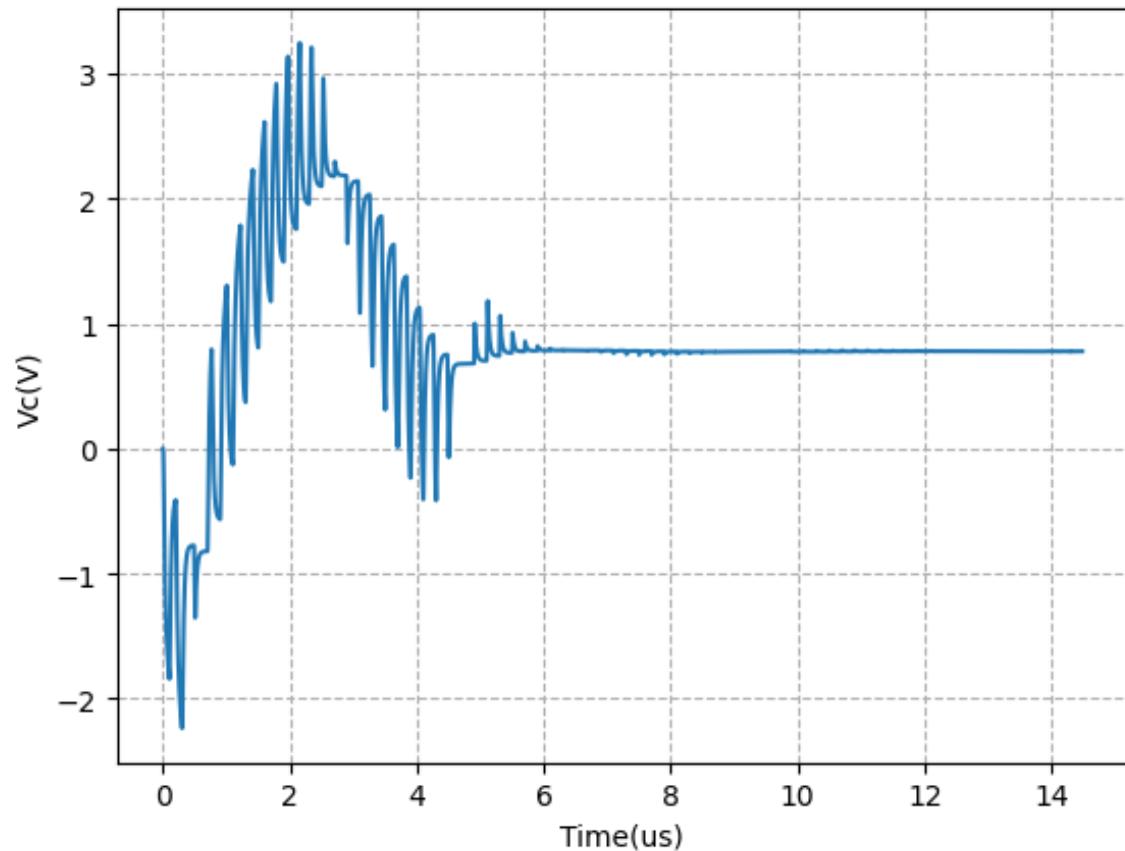


Figure 7.3: Transient of Control Voltage in Ideal PLL

7.2 Conventional PLL

At first let us discuss function of all blocks of PLL other than frequency divider in brief.

Phase Frequency Detector

PFD inputs reference clock and clock through feedback path. We have designed a Type-II order-3 PLL which eliminates both phase error and frequency error between reference clock and feedback clock. PFD generates gating pulses (see up and down in fig. 7.4) for charge pump which in combination with loop filter generates control voltage. When PLL is locked that is there is no phase error and frequency error output of PFD are just spikes. Negative feedback here locks the PLL.

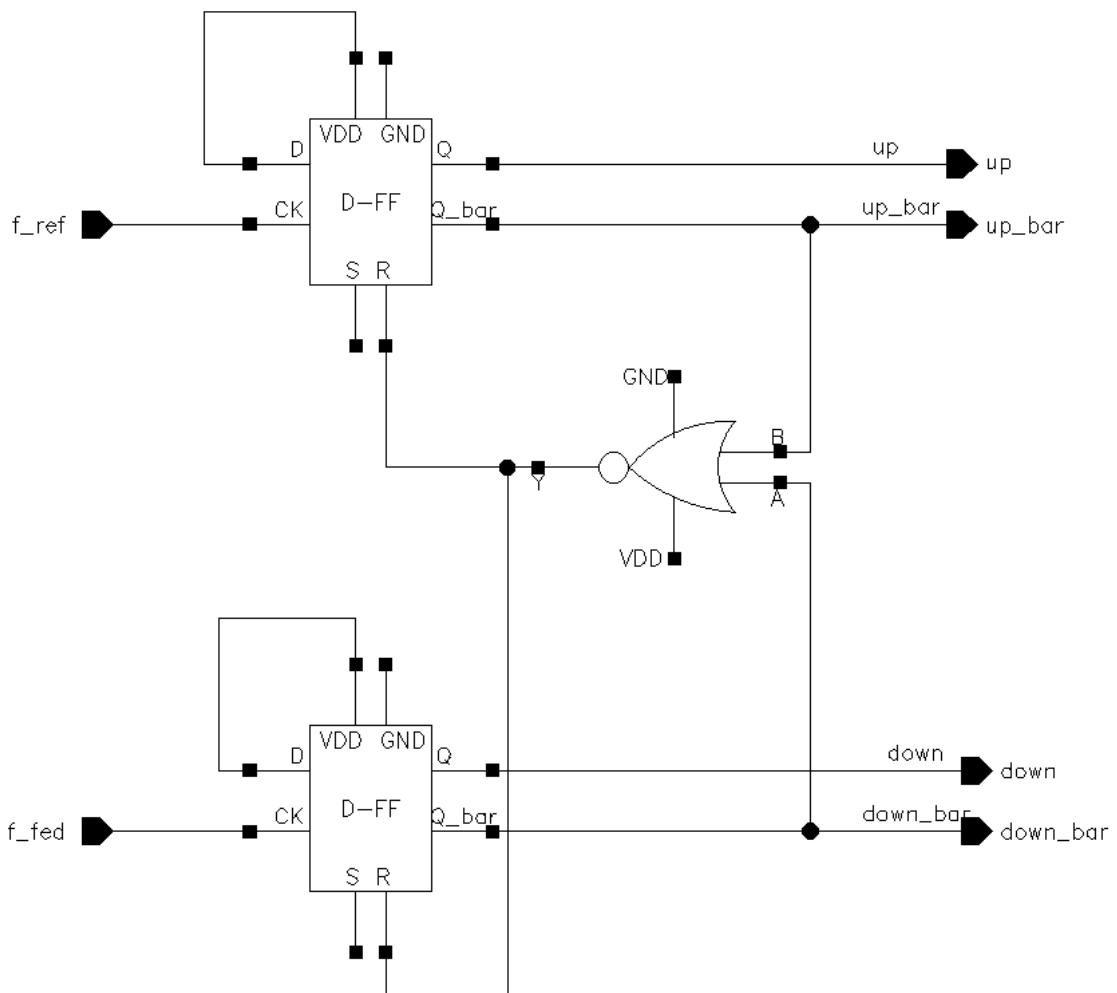


Figure 7.4: PFD using 65nm CMOS Technology

Charge Pump and Loop Filter

Charge Pump takes input as gating pulses from PFD and generates control voltage for VCO. When up is high it connects Loop Filter capacitors to VDD hence charges them and increases the control voltage this further increases the output frequency. In other case when down is high it connects Loop Filter capacitors to ground which discharges these capacitors reducing the control voltage and output frequency. We have used a transmission gate in loop filter to avoid discharging when both up and down are inactive. Loop filter also serves the purpose of low pass filter.

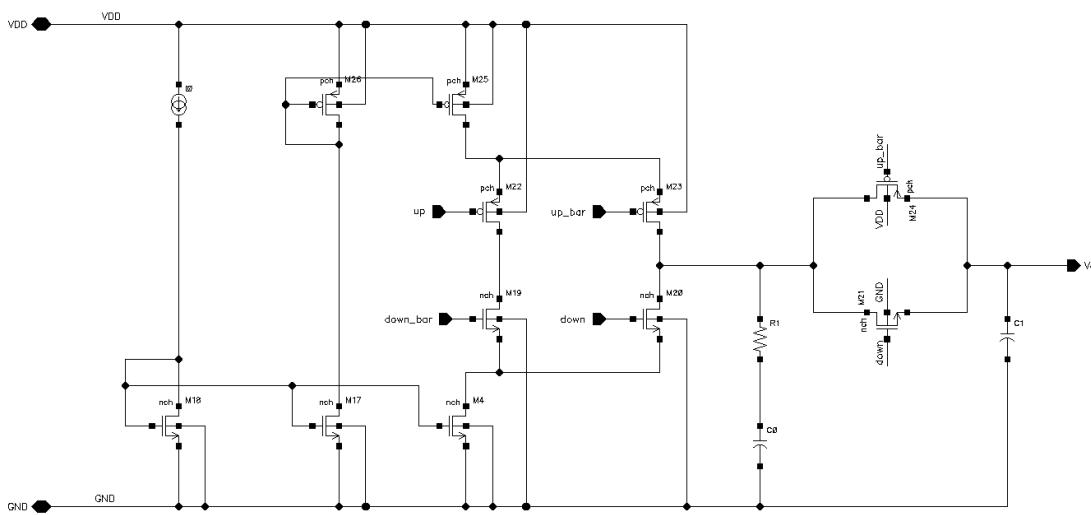


Figure 7.5: Charge Pump and Loop Filter using 65nm CMOS Technology

Voltage Controlled Oscillator

A VCO varies its output frequency based on control voltage provided to it. In an ideal VCO output frequency is directly proportional to control voltage and there are no frequency limits. We have current starved inverters as delay cells to serve the purpose. Increasing the V_c increases the current, all these delay cells are current mirrored and hence current increases in all delay cells. As current increases, charging and discharging time decreases resulting in increase in output frequency. There must be a 180 degrees phase shift to be maintained. Also, increasing the number of cells decreases the output frequency.

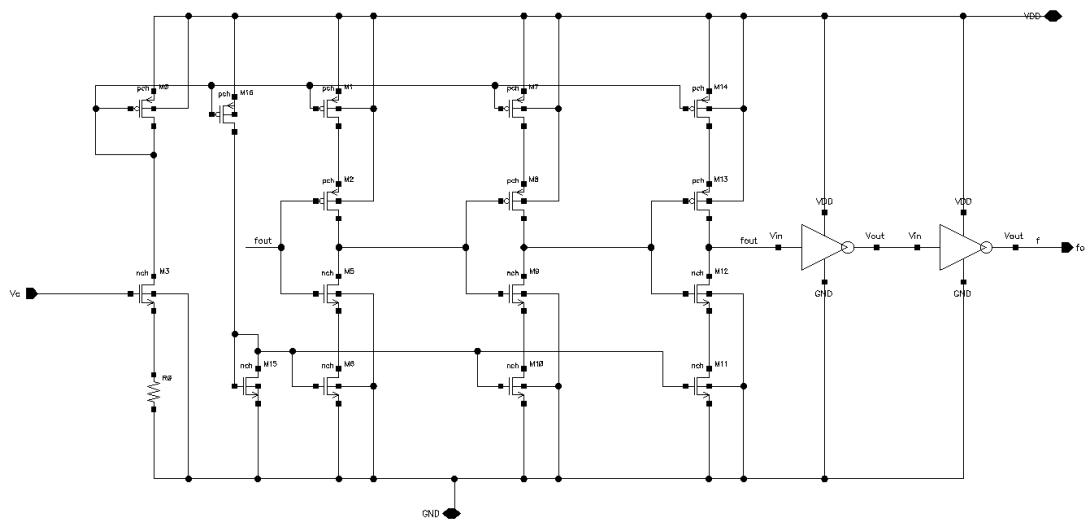


Figure 7.6: VCO using 65nm CMOS Technology

Testing the Conventional PLL

Conventional PLL designed is now put under test for verification. Reference frequency of 1GHz is given with a divider of 4. So, PLL output frequency should lock to 4GHz. As discussed earlier, control voltage should be in similar pattern.

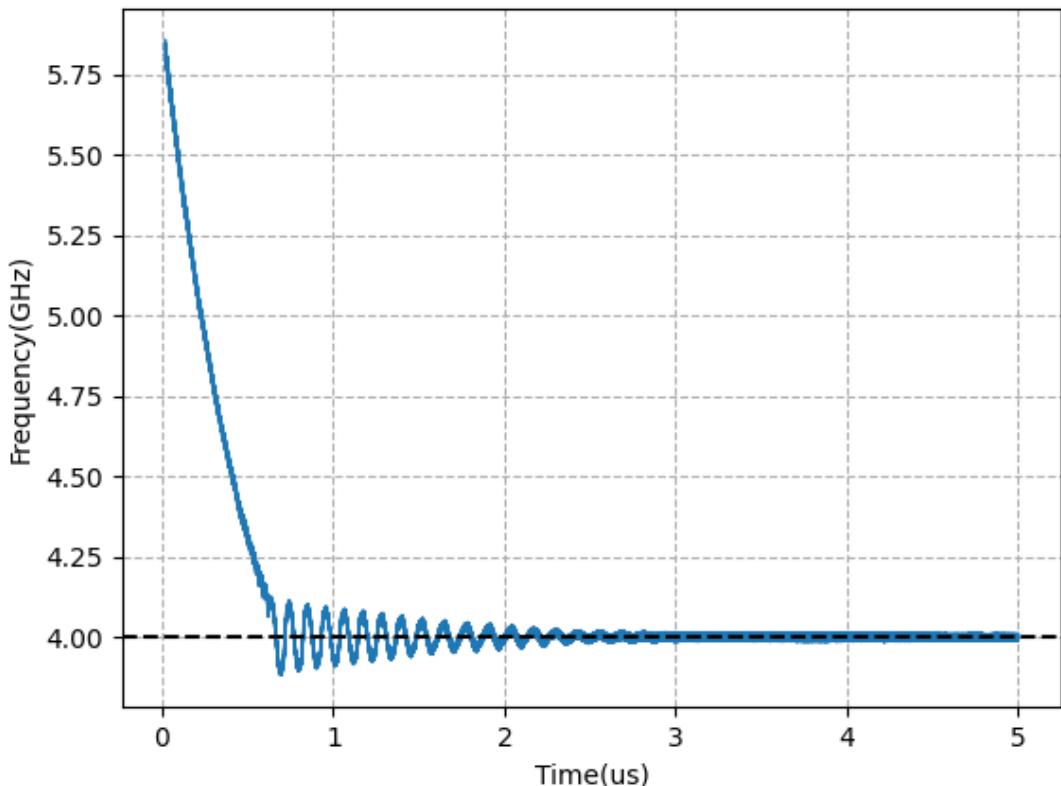


Figure 7.7: Transient of Output Frequency in Conventional PLL-4GHz

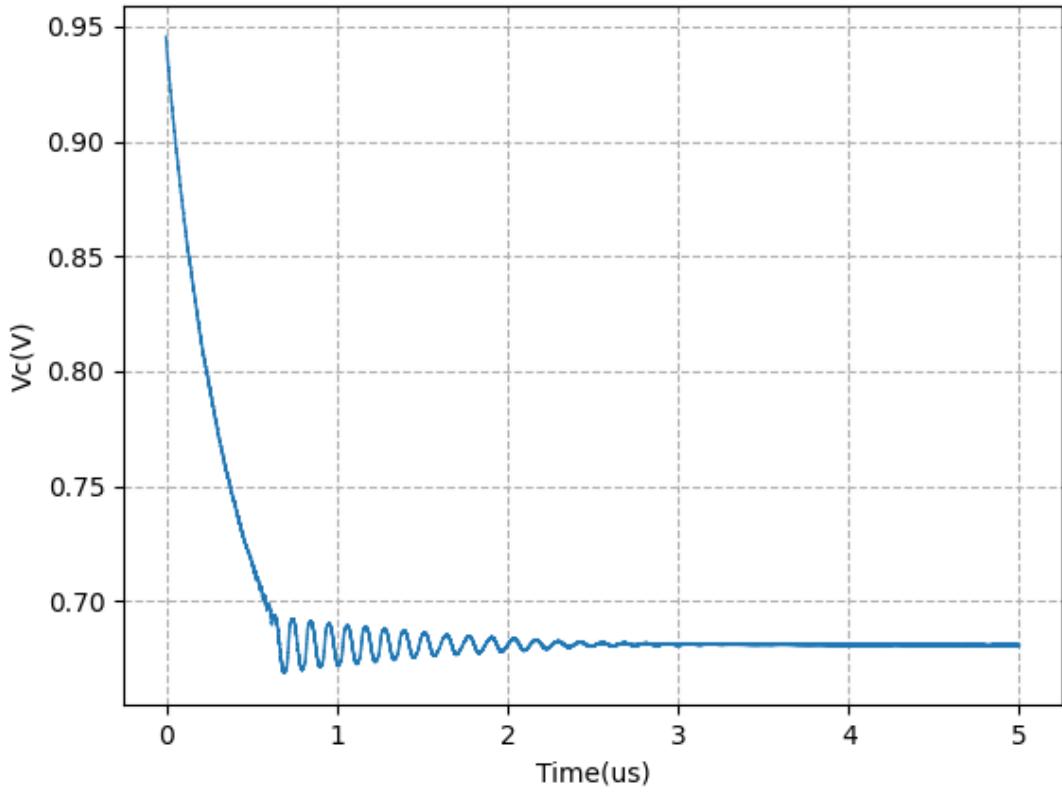


Figure 7.8: Transient of Control Voltage in Conventional PLL-4GHz

Testing Pulse Swallow Divider in Conventional PLL

In Conventional PLL we have put our designed modulus 460 divider and simulated. We can observe feedback frequency has locked to input reference frequency of 10MHz.

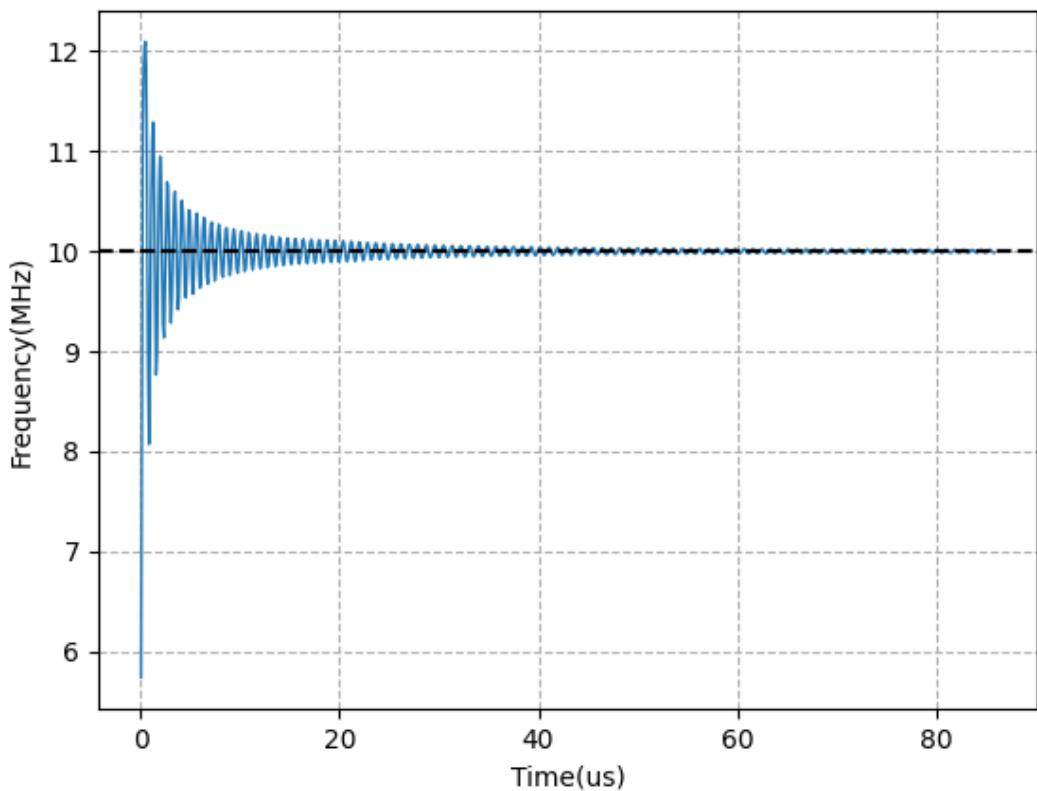


Figure 7.9: Transient of Feedback Frequency in Conventional PLL

Chapter 8

Conclusion and Future Work

8.1 Conclusion

We have executed a modulus 460 Pulse Swallow divider and tested it by integrating it in an ideal PLL and in a conventional one later. Divider perform well in desired operating frequency of 4.6GHz giving out a divided clock of frequency 10MHz.

8.2 Future work

This divider can be scaled to operate at even higher frequencies by using CML (Current-Mode Logic) in End of Count architecture, Dual-Modulus Prescalar and Synchronous counter. Flip flops and combinational logic generated using CML logic operates at much better speed than clocked CMOS (used in this work). We persisted with clocked-CMOS because of its simplicity in design and low power consumption. CML logic draws static current and hence consumes high power. Another thing that can be done is integrating the EOC of swallow counter with an ADC to give digital input to EOC. This modification will result in an externally programmable counter without rendering the hardware.

References

- [1] I. Som, S. Sarangi, and T. K. Bhattacharyya, "A 7.1-GHz 0.7-mW Programmable Counter With Fast EOC Generation in 65-nm CMOS," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2397-2401, Nov. 2020, doi: 10.1109/TCSII.2020.2966373.
- [2] Razavi, Behzad. Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level. Cambridge University Press, 2020.