Digital System Design Using FPGA (E3 231)

Mini Project

Submitted by

Aravind Penta (04-01-01-10-51-20-1-18114)

Shubham (04-01-01-10-51-20-1-18276)

Problem Statement

Implement a DPLL, this should be designed to have minimum synchronization time, i.e. if input Clock changes, within minimum clock cycles output should be synchronized.

Phase Locked Loop

It is a feedback control system which maintains constant phase between the input and the output signals while having the same output frequency as that of input.

Design Specifications

Clock period = 20 ns
Centre frequency = 340 ns

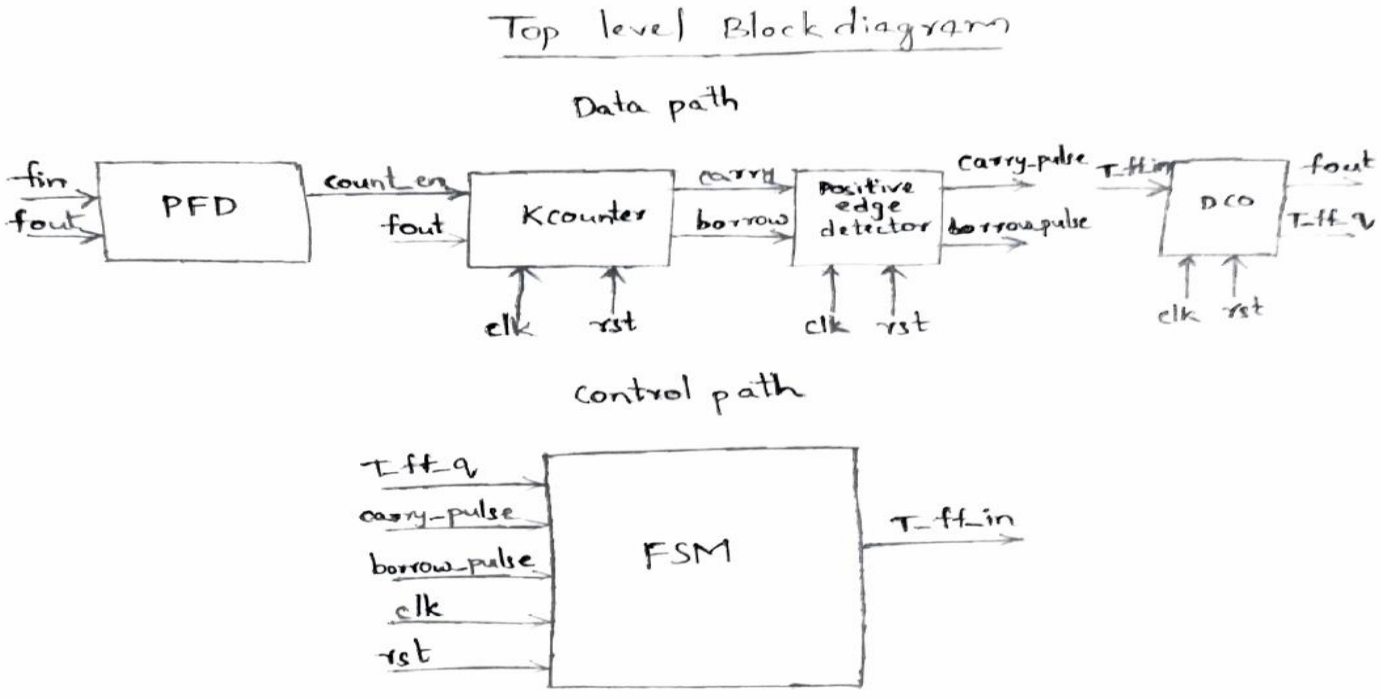
Block Diagram and Implementation Details

Design details of total design and components used are summarized in this section.

Top Level Module: -

Entity Name: - DPLL_Top

Sl No	Port Name	Direction	Description
1	clk	Input	Clock input to circuit
2	rst	Input	Reset signal to circuit
3	fin	Input	Reference input to circuit
4	fout2	Output	Synchronized output



The phase difference between the input and output signals is used to generate counter_en signal by PFD and the carry and borrow signals are generated by Kcounter. Carry and borrow signals are used to generate carry_pulse and borrow_pulse by "Positive Edge detector" module. These pulse signals along with T_ff_q (output of DCO) are used to generate the control signal T_ff_in which controls the outputs of DCO.

PD Module:-

Entity:- PD

This module accepts reference signal and the output signal as inputs and generates counter_en signal which serves as input to KCounter and enables either up or down counter.

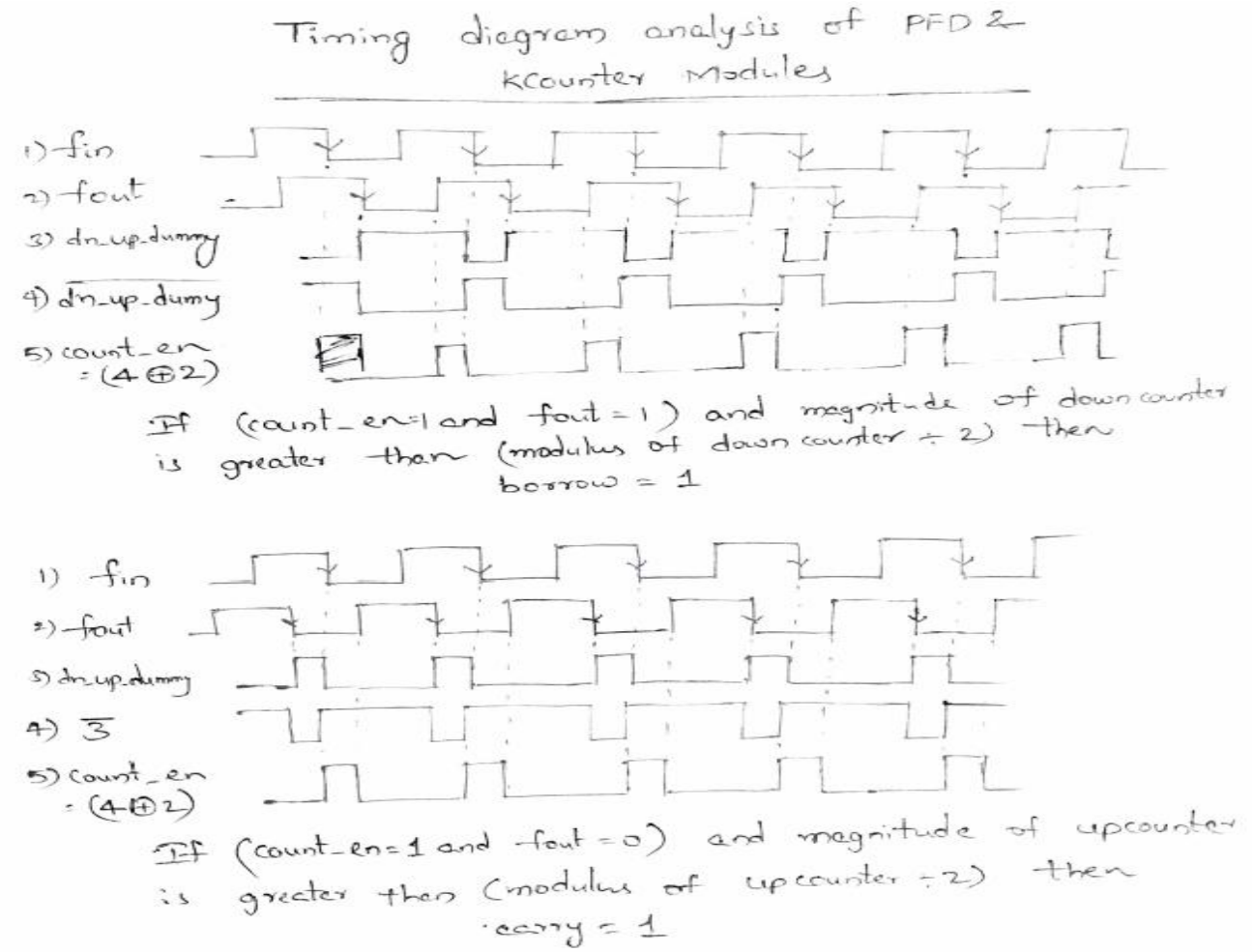
Sl No	Port Name	Direction	Description
1	fin	Input	Reference input
2	fout	Input	Feedback signal(output of DPLL)
3	Count_en	Output	To enable either up or down counter in next module

KCounter Module: -

Entity:- KCounter

KCounter is used to generate carry and borrow signals based on the fout and count_en inputs to the module.

Sl No	Port Name	Direction	Description
1	clk	Input	Clock input
2	rst	Input	Reset input
3	count_en	Input	PD module output
4	fout	Input	Feedback signal(output of DPLL)
5	carry	Output	Signal used to generate carry pulse by positive edge detector.
6	borrow	Output	Signal used to generate borrow pulse by positive edge detector.



Positive Edge Detector Module: -

Entity:- Pulsedet

This module accepts carry and borrow signals along with clk and rst as inputs to generate carry_pulse and borrow_pulse as outputs when a positive transition occurs in inputs.

Sl No	Port Name	Direction	Description
1	clk	Input	Clock input
2	rst	Input	Reset input
3	carry	Input	Used to generate carry_pulse
4	borrow	Input	Used to generate borrow_pulse
5	carry_pulse	Output	Pulse of one clock period is generated when a positive transition in carry occurs
6	borrow_pulse	Output	Pulse of one clock period is generated when a positive transition in carry occurs.

FSM Module: -

Entity:- FSM

This module is used to generate control signal(T_ff_in) by accepting carry_pulse, borrow_pulse, clk,rst and T_ff_q(output of Toggle FF in DCO) as inputs.

Sl No	Port Name	Direction	Description
1	clk	Input	Clock input
2	rst	Input	Reset input
3	carry_pulse	Input	Used to generate control signal T_ff_in
4	borrow_pulse	Input	Used to generate control signal T_ff_in
5	T_ff_q	Input	Used to generate control signal T_ff_in
6	T_ff_in	Output	If this becomes 0, then the output of TFF in DCO maintains the previous state.

The working of FSM can be better explained by state diagram and timing diagram.

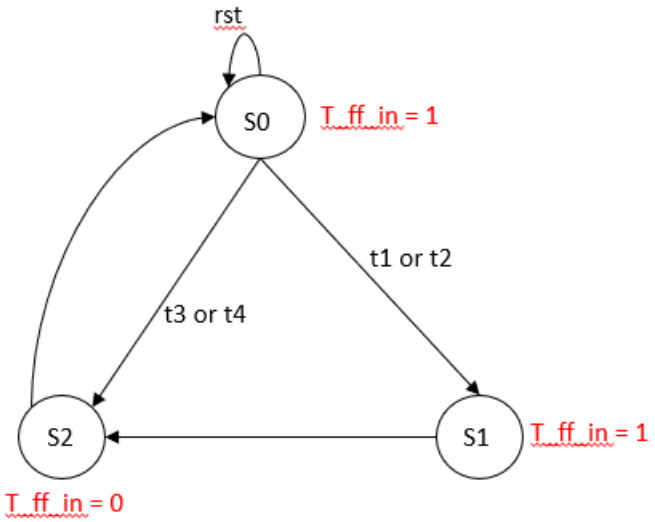
State Diagram

$t1 = \overline{rst} \cdot carry\_pulse \cdot \overline{T\_ff\_q}$

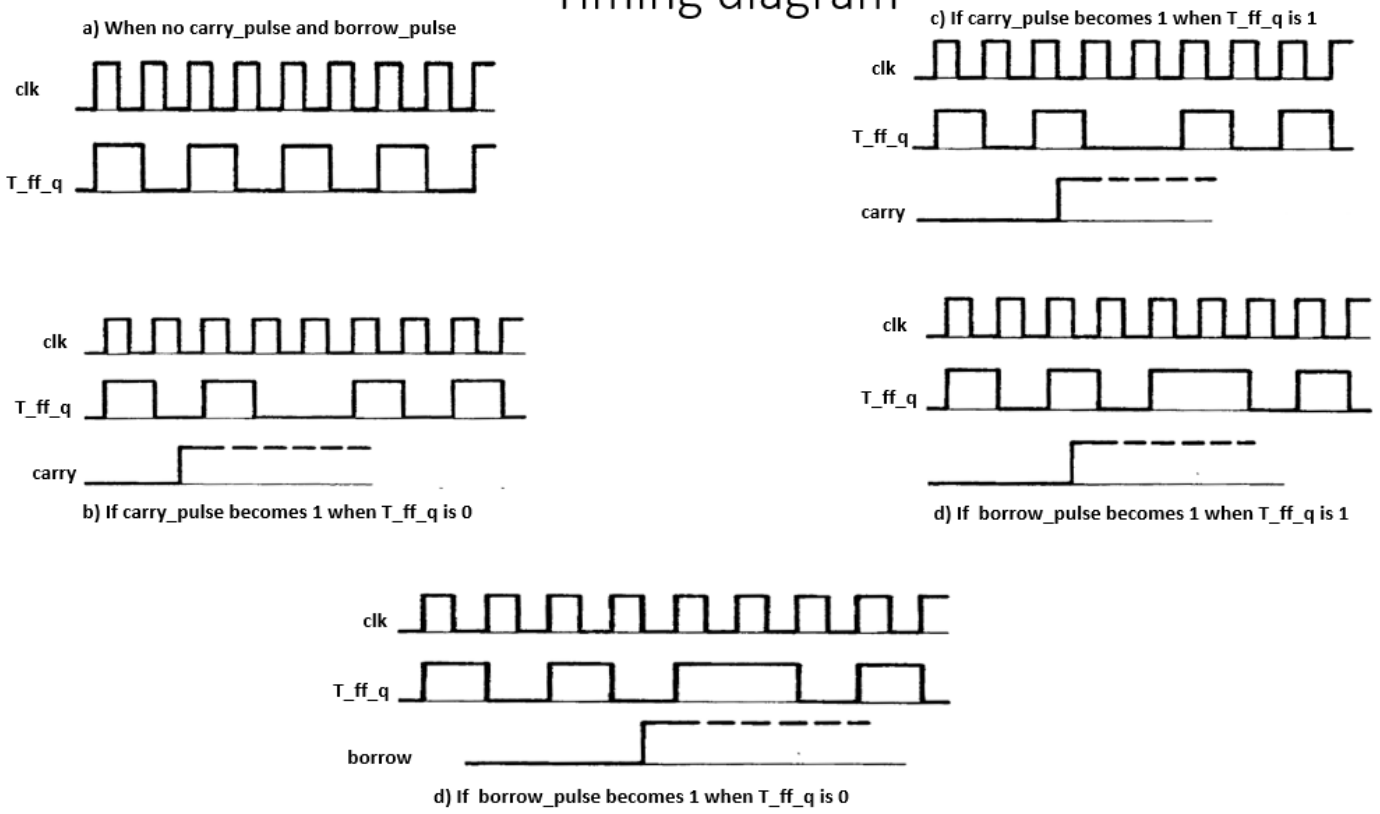
$t2 = \overline{rst} \cdot \overline{carry\_pulse} \cdot borrow\_pulse \cdot T\_ff\_q$

$t3 = \overline{rst} \cdot carry\_pulse \cdot T\_ff\_q$

$t4 = \overline{rst} \cdot \overline{carry\_pulse} \cdot borrow\_pulse \cdot \overline{T\_ff\_q}$



Timing diagram



- If there are no carry and borrow pulses, the TFF output continuously toggle.
- If carry pulse comes when the T_ff_q is 0, then the next state is s1(where T_ff_in is 1, which means the TFF toggles in next clock cycle) and then unconditionally the state transits to s2(where T_ff_in is 0, which means the TFF remains same in next clock cycle) and then unconditionally transits to s0(where T_ff_in is 1, which means the TFF toggles in next clock cycle).
- If carry pulse comes when the T_ff_q is 1, then the next state is s2(where T_ff_in is 0, which means the TFF remains same in next clock cycle) and then unconditionally the state transits to s0(where T_ff_in is 1, which means the TFF toggles same in next clock cycle).
- If borrow pulse comes when the T_ff_q is 0, then the next state is s2(where T_ff_in is 0, which means the TFF remains same in next clock cycle) and then unconditionally the state transits to s0(where T_ff_in is 1, which means the TFF toggles same in next clock cycle).

- If carry pulse comes when the T_ff_q is 1, then the next state is s1(where T_ff_in is 1, which means the TFF toggles in next clock cycle) and then unconditionally the state transits to s2(where T_ff_in is 0, which means the TFF remains same in next clock cycle) and then unconditionally transits to s0(where T_ff_in is 1, which means the TFF toggles in next clock cycle).

DCO Module: -

Entity:- DCO

This module consists of Toggle FF, the output of TFF(T_ff_q) and the clk are applied as inputs to NOR gate and the output of the gate is applied to the frequency divider to generate the output. This module also generates output of TFF as the module output which serves as input to FSM.

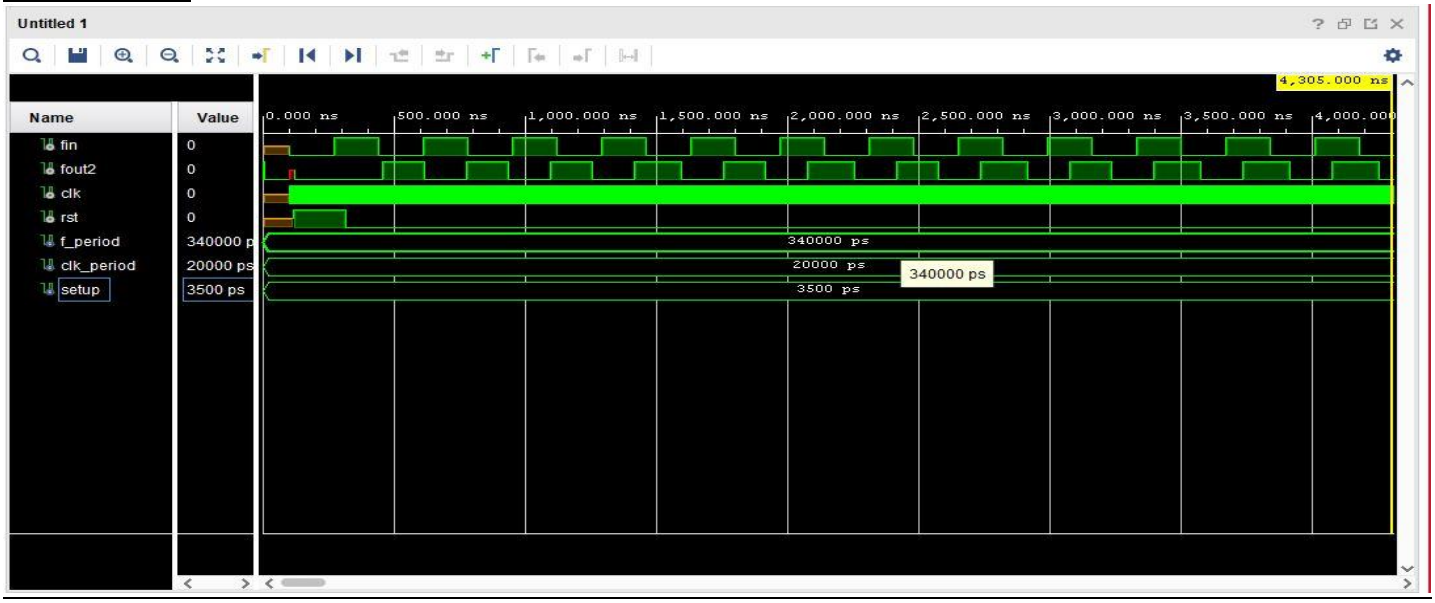
Sl No	Port Name	Direction	Description
1	clk	Input	Clock input
2	rst	Input	Reset input
3	T_ff_in	Input	Input to TFF in DCO module
4	fout	Output	Overall DPLL output
5	T_ff_q	Output	Used to generate control signal T_ff_in

Simulation results

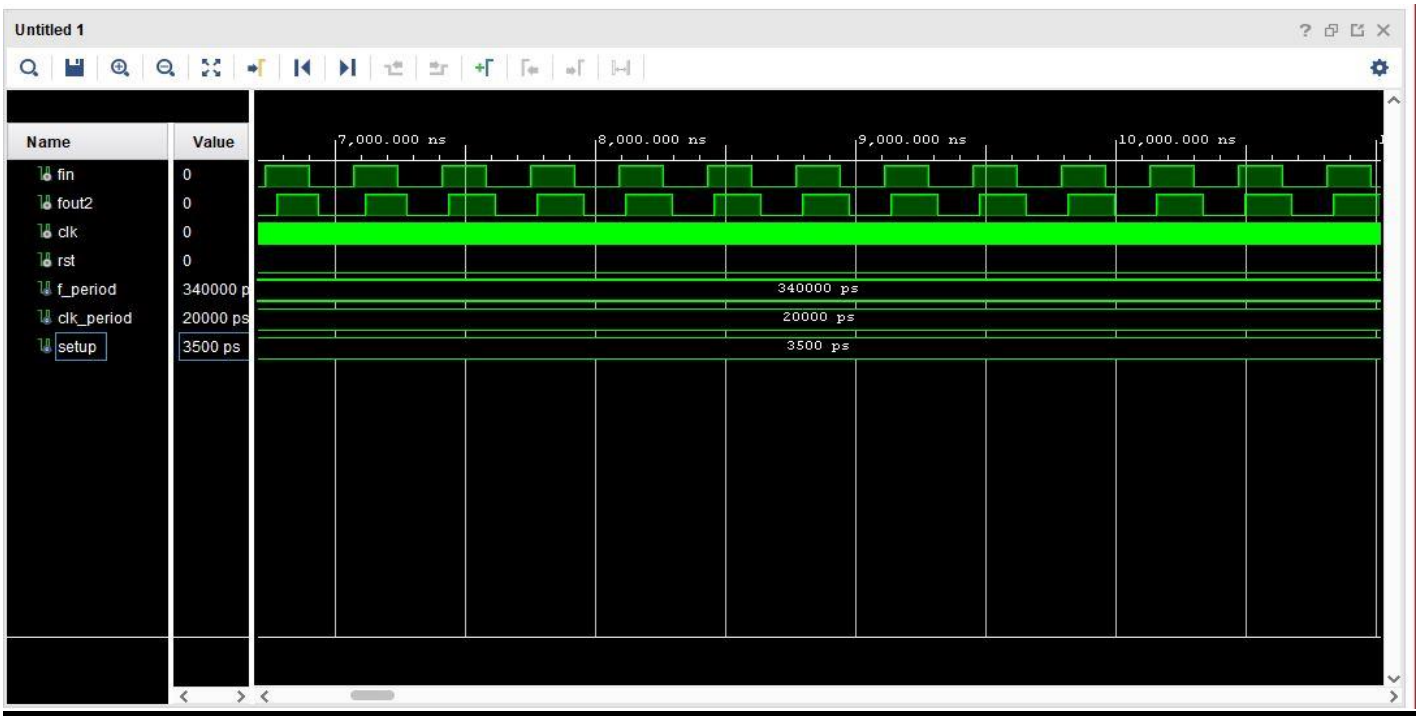
Test bench was designed in a way to include different reference frequencies to check whether the DPLL is meeting the specifications.

Post implementation Timing Simulation

Unlock state



Lock state



Post Implementation Summary and Results

Resource Utilization Summary

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	Bonded IOB (106)	BUFGCTRL (32)
DPLL_Top	13	20	8	13	4	2
u1 (PD)	1	2	2	1	0	0
u2 (KCounter)	4	8	2	4	0	0
u3 (Pulsedet)	2	4	2	2	0	0
u4 (FSM)	3	2	2	3	0	0
u5 (DCO)	4	4	4	4	0	0

Timing Summary

Clock constraint was given as 100MHz. Constraint file is attached in submission folder

Specification	Value
Worst Negative Slack (WNS)	6.534 ns
Worst Hold Slack (WHS)	0.179 ns
Setup between clocks	1.568 ns(clk) 1.445 ns(fin)
Input port to setup	3.266 ns
Output Port: Clock to Out delay	Max_delay=8.351 ns, min_delay=2.738 ns

Conclusion

In this project, we have designed and implemented an ADPLL with center frequency of 2.94MHz (time period of 340ns), and the implemented design is working as per the specifications.

Reference

[1] *Phase-Locked loops design, simulation and applications* by R.E.Best.

[2] *Design of All Digital Phase Locked Loop in VHDL*" *Gayathri M G / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 ,Vol. 3, Issue 4, Jul-Aug 2013, pp.1074-1076*

[3] *K.T. lbrahim, and A.E Salama, "Digital Of ADPLL for Good Phase and Frequency Tracking Performance", Nineteenth National Radio Science Conference, Alexandria, March 2002.*

PS: VHDL Source Code, Testbench Constraint File attached in the Zip file