ALL Digital Phase Locked Loop (ADPLL)

By

Nada Ibrahim Afifiy

Sara Salah Abd El Mone'm

Sara Sayed Dahy

Under the Supervision of

Dr. Hassan Mostafa

A Graduation Project Report Submitted to the Faculty of Engineering at Cairo University In Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Electronics and Communications Engineering

Faculty of Engineering, Cairo University

Giza, Egypt

July 2014

List of tables	. v
List of figures	. vi
Abbreviations	viii
Abstract	. ix
Chapter 1 Introduction	1
1.1 what is PLL?	1
1.2 Types of PLL	2
1.3 AAPLL vs. ADPLL	2
1.4 Applications of PLL	4
1.5 ADPLL approach	. 4
1.5.1 Why ADPLL? ,	4
1.6 project flow	5
Chapter 2 ADPLL Architecture	6
2.1 The proposed Architecture	. 6
2.2 The DCO	7
2.2.1 The ring oscillator basic idea	7
2.2.1.1 The DCV basic idea	7
2.2.1.2 Ring oscillator and tristate buffer array	9
2.2.1.3 Ring oscillator and tristate buffer array (another arrangement)	10
2.2.2 The proposed architecture of the DCO	. 11
2.2.2.1 Coarse block	. 12
2.2.2.2 Fine block	. 12
2.2.3 The control blocks of the DCO	. 12
2.2.3.1 Phase detector	. 12

Table of contents

	2.2.3.1.1 Buffer delay line	12
	2.2.3.1.2 Inverter delay line TDC	13
	2.2.3.1.3 Synchronization block	13
	2.2.3.1.4 SR flip flop and counter PD	14
	2.2.3.1.5 Phase Frequency Detector (PFD)	14
2.2	2.4 Loop filter (Controller)	15
	2.2.4.1 Successive approximation mechanism	15
	2.2.4.2 Stepping mechanism	15
	2.2.4.3 Shift Registers	16
Chapter 3	Design Flow	
3.1 Vei	ification of the DCO function	17
3.1	.1 DCV	
3.1	.2 Coarse tuning block	
3.1	.3 Fine tuning block	20
3.2 Pha	se Frequency detector	22
3.3 Coi	ntroller (Loop Filter)	22
Chapter 4	system layout	
4.1 DC	O	
4.1	.1 Nand layout	
4.1	.2 NOR layout	
4.1	.3 AND layout	
4.1	.4 One stage INVERTER	
4.1	.5 Two-stage INVERTER	
4.1	.6 MUX block	
4.1	.7 D-latch	

4.1	1.8 Fine segment	. 29
4.1	1.9 Coarse segment	. 29
4.1	1.10 fine stage	. 30
4.1	1.11 the whole DCO layout	. 30
4.2 PFD an	d Controller	. 31
Chapter 5	Simulation Results	. 32
5.1 DC	CO	. 33
5.1	1.1 Coarse Stage	. 33
	5.1.1.1 Coarse Delay	. 33
	5.1.1.2 Coarse frquency	. 34
5.1	1.2 Fine Stage	. 35
	5.1.2.1 Fine Delay	. 35
	5.1.2.2 Fine frequency	. 35
5.2 Loc	ck Time	. 36
5.3 Jitt	er simulation result	. 37
5.4 Are	ea calculation	. 38
5.5 Pov	wer Calculation	. 38
Reference .		. 39
Appendix .		. 41
A1.VHI	DL AMS tutorial	41
A2.Mod	lelSim Tutorial	49
A3.Cad	ence Tutorial	58
A4.Jitte	r Calculation	84
A5.Pow	ver Calculation	89
A6.Area	a calculation	92
A7. Des	sign Compiler Tutorial	93

List of tables

Table 1-1 AAPLL Vs. ADPLL	3
Table 2-1 Bianry word Vs. Thermometer code	16
Table 4-1 Schematic Vs. Layout	25
Table 5-1 Coarse Delay steps	33
Table 5-2 Coarse Frequency steps	34
Table 5-3 Fine delay steps	35
Table 5-4 Fine frequency steps	35

List of figures

Fig.1-1 Basic block of PLL	1
Fig. 1-2 Jitter illustration	3
Fig. 1-3 Flow chart describe the whole project	5
Fig 2-1 ADPLL as a black box from the outer most level	6
Fig. 2-2 the overall architecture of the ADPLL	6
Fig. 2-3 Basic ring oscillator	7
Fig. 2-4 Ring oscillator with analog varactors	8
Fig. 2-5 Ring oscillator with DCVs	8
Fig. 2-6 ring oscillator with mux	9
Fig. 2-7 Ring oscillator and tristate buffer array(1)	10
Fig. 2-8 Ring oscillator and tristate buffer array(2)	11
Fig. 2-9 The proposed architecture of the DCO	11
Fig. 2-10 Coarse Block	12
Fig. 2-11 Buffer Delay Line	12
Fig. 2-12 Inverter Delay Line	13
Fig. 2-13 Synchronization Block	13
Fig. 2-14 SR flip flop and counter PD	14
Fig. 2-15 PFD	14
Fig. 2-16 Difference range of PFD	15
Fig. 3-1 Circuit design for DCV unit(Nand Implementation)	17
Fig. 3-2 coarse segment	18
Fig. 3-3 The case without using Latch	19
Fig. 3-4 successive coarse segments	20
Fig 3-5 Fine block	20

Fig. 3-6 Fine Segment	21
Fig. 3-7 overall Fine block architecture	21
Fig. 3-8 PFD output when ref. Clk leads	22
Fig. 3-9 PFD output when DCO. Clk leads	22
Fig 3-10 Flow chart of Coarse controller stage	23
Fig 3-11 Flow chart of Fine controller stage.	24
Fig. 5-1 graph shows the maximum frequency	32
Fig. 5-2 graph shows the minimum frequency	32
Fig 5-3 The max lock time	36
Fig 5-4 The typical lock time	36
Fig 5-5 Jitter Calculation	37
Fig. 5-6 The total area Design	38

Abbreviations

Abbreviation	Explanation
AAPLL	All Analog Phase Locked Loop
ADPLL	All Digital Phase Locked Loop
DC	Design Compiler
DCO	Digitally Control Oscillator
DCV	Digitally Control Varactor
DRC	Design Rule Check
HDL	Hard ware Description Language
HLS	High Level Synthesis
LVS	Layout Vs. Schematic
MUX	Multiplexer
PD	Phase Detector
PFD	Phase frequency detector
PEX	Parasitic Extraction
PR	Place and Route
PVT	Process, Voltage and Temperature
RTL	Register Transfer Logic
SDC	Synopses Design Constrains
SDF	Standard Delay Format
SR	Shift Register
VCO	Voltage Controlled Oscillator
VLSI	Very Large Scale Integrated circuit

Abstract

Phase Locked Loops are used in almost every communication system. Some of its uses include recovering clock from digital data signals, performing frequency, phase modulation and demodulation, recovering the carrier from satellite transmission signals and as a frequency synthesizer.

PLL is generally implemented using analog components, which called analog PLL (APLL).

APLLs have been widely used for clock generation, and frequency synthesis with good performance and high frequency range but the main challenge is the high power consumption, large area, scalability and high noise due to matching, process variations in the layout.

All digital PLLs (ADPLL) solve the problems of the APLL where the ADPLL have low power consumption, small area and scalability across different technology nodes.

Fully digital PLLs have better noise immunity and better tolerance to bias drifts and PVT variations. They also provide the advantage of implementation using automatic CAD tools which reduces the turnaround time and is also easier to integrate and migrate over various applications and fabrication processes.

ADPLL uses a phase-frequency detector instead of a phase detector, DCO instead of a VCO, control circuit imitating the functionality of a loop filter and a fixed frequency detector. This design is very much suitable for SoC applications and can be automatically implemented with standard cell libraries.

The proposed ADPLL is implemented on the TSMC 65nm technology with 1.8v operating voltage, covering frequency range from 100 MHz to 700 MHz, with lock time around 0.2 μ s, the total area is 0.01 mm2, power 1mwatt, ,peak-to-peak jitter 6.396 psec, RMS jitter 1.035 psec.

Chapter 1: Introduction

1.1 What is PLL?

A PLL circuit is used to synchronize an output signal, which is usually generated by an oscillator, with a reference or input signal in frequency as well as in phase. In the synchronized state, the difference (error) between the reference and the oscillator output is zero or at least very small. So it is called 'locked'. Some of its uses include recovering clock from digital data signals, performing frequency, phase modulation and demodulation, recovering the carrier from satellite transmission signals and as a frequency synthesizer.

The below figure(1-1) shows the basic block diagram of PLL which consists of PFD, loop filter and VCO.



Fig.1-1Basic block of PLL

The VCO is used to generate an output clock with frequency proportional to the input actuation voltage. This output clock is fed back to PFD which is used to compare the phase of the VCO clock with the phase of the reference clock and generate a voltage proportional to the phase error. The loop filter is used to stabilize the system and achieve the desired response. This control loop makes the VCO clock have the same frequency and phase as the reference clock. Hence, the output clock will have the same frequency of the reference clock.

1.2 Types of PLL:

• Analog or linear PLL (APLL)

Phase detector is an analog multiplier. Loop filter is active or passive.uses a Voltage-controlled oscillator (VCO).

• Digital PLL (DPLL)

An analog PLL with a digital phase detector (such as XOR, edge-trigger JK, phase frequency detector). May have digital divider in the loop.

• All digital PLL (ADPLL)

Phase detector, filter and oscillator are digital. Uses a numerically controlled oscillator (NCO).

• Software PLL (SPLL)

Functional blocks are implemented by software rather than specialized hardware.

• Neuronal PLL (NPLL)

Phase detector, filter and oscillator are neurons or small neuronal pools. uses a rate controlled oscillator (RCO). Used for tracking and decoding low frequency modulations (<1 kHz), such as those occurring during mammalian-like active sensing.

1.3 AAPLL vs. ADPLL

Basically advantages of ADPLL without consideration of technology difference are low power , small area and scalability across different technology nodes also the disadvantages are frequency range , jitter and lock time ,but till now VCO converts to DCO which is designed using full custom layout fashion. This is because the required oscillation in design if DCO is pure digital will result constant given delay, but layout gives difference layouts depending on Temperature, used technology and optimizing area, in other words the DCO dominates the major performance measures of the ADPLL, such as power consumption and jitter.

Analog PLLs are widely used, and show high performance characteristics in terms of jitter and frequency range. However, the high power consumption and large area have always been disadvantages for this type of PLLs. Moreover, as the technology scales down into deep submicron, the design of these analog circuits becomes very sensitive and increases the design cycle time and time to market. The last decade has shown a lot of interest in replacing the analog PLLs with an All Digital PLL (ADPLL). The main advantages driving this new trend are the fast time to market, and the small design effort required to migrate between different technology nodes.

	ADPLL	AAPLL
Lock time	Large	Small
Frequency range	Small	High
Jitter	Worst	Better
Area	Small	Large
Power	Low	High
Scalling down to another technology	Easy	Hard
Ability of change in frequency	Discrete	Continous

Table 1.1 AAPLL Vs. ADPLL

What is Jitter?

Jitter is the undesired deviation from true periodicity of an assumed periodic signal, Deviation (expressed in \pm ps) can occur on either the leading edge or the trailing edge of a signal. Jitter may be induced and coupled onto a clock signal from several different sources and is not uniform over all frequencies.



Jitter is composed of two parts, random jitter and deterministic jitter.

Random jitter

Random Jitter, also called Gaussian jitter, is unpredictable electronic timing noise. Random jitter typically follows a Gaussian distribution or Normal distribution.

Deterministic jitter

Deterministic jitter is a type of clock timing jitter or data signal jitter that is predictable and reproducible. The peak-to-peak value of this jitter is bounded, and the bounds can easily be observed and predicted.

1.4 Applications of PLL:

Phase Locked Loops (PLLs) are widely used for frequency synthesis in many Applications:

- Microprocessors
- Digital signal processing circuits
- Communication systems as in the high speed serial links
- In the clock and data recovery circuits.

1.5 ADPLL approach

ADPLLs are designed by replacing the analog blocks in the AAPLL by their digital counterpart the RC LPF is replaced by a digital controller, and the VCO is replaced by a Digitally Controlled Oscillator (DCO).

1.5.1 Why ADPLL?

Because of its main advantages which are :

- Low power consumption
- Small total area
- Easy way for scale down process to deep submicron technology
- Small lock time.

1.6 project flow:

The following figure shows the flow chart of how project will go through from the proposed plan to simulations and get results.



Chapter 2:ADPLL Architecture

2.1 The proposed Architecture

The figure below shows the ADPLL as a black box from the outer most level.



The figure below shows the overall architecture of the ADPLL.



2.2 The DCO

The DCO (Digital Controlled Oscillator) block is considered the core of the PLL, it is the block that contains the oscillator and generates the required frequency. The basic idea of the DCO operation is demonstrated in the next few points

2.2.1 The ring oscillator basic idea

The proposed DCO architecture is build on using the ring oscillator, the ring oscillator is a very simple circuit which consist of an odd number of successive inverters connected together in a chain



There are two approaches to design DCO

First approach

by changing fixed capacitance but this consume more power and area

2.2.1.1 The DCV basic idea

Now after introducing the ring oscillator idea, it is clear that the previous figure will oscillate at only a single frequency its value depends on the introduced delay by each inverter cell. In order to make the delay of the ring oscillator controlled, we have to find a method that enables us to control that delay digitally, so we introduce a new approach where ring oscillator is connected to a variable capacitive load so that by changing the value of the varactors we can get a large combinations of different delays and hence different frequencies.



So we need to find replacement for that analog varactors to make the design completely digitally controlled, hence the DCV (Digitally Controlled Varactor) can give us a good solution for this problem

This is the block diagram of the ring oscillator and more details will be discussed later in chapter 3



Second approach

by changing number of stages of ring oscillator by using multiplexer but this approach has poor linearity and low frequency range



2.2.1.2 Ring oscillator and tristate buffer array

In order to have good resolution, an array of tristate inverters is used . Each stage has 127 tristate inverters connected as an array. These 127 tristate inverters are divided into seven stages. The first stage consists of only one tristate inverter, second stage has two $(2^{(2-1)})$, third stage has four $(2^{(3-1)})$, fourth stage has eight $(2^{(4-1)})$, and the last stage has $2^{(6-1)}$ tri-state buffers. The number of tristate inverters per stage increases by a factor of two.

It is important to note that the first stage of tri-state inverters for all the three stages of ring oscillator is controlled by En1. Similarly second stage of tri-state inverters for all three stages of ring oscillator is controlled by En2. Likewise the third, fourth to eighth stage are controlled by En3, En4 to En7, respectively. The frequency of the DCO depends on the number of tri-state inverters that are on. The more the number of tri-state inverters that are on, the higher the oscillating frequency.

<u>The disadvantage</u> of this design is that it takes very long time to oscillate also it works in high frequency range



2.2.1.3 Ring oscillator and tristate buffer array (another arrangement)

In the design, DCO[7:0] is input control word and CLK_OUT is the DCO output clock signal. Respectively, DCO[0], DCO[1] until the DCO[7] controls one three-state inverter, two three-state inverters up to 128 three-state inverters to achieve the function that DCO input word encodes the three-state inverters array with binary method

<u>The obvious disadvantage</u> is that it takes a long time to oscillate ,hence large lock time and it works in high frequency range



2.2.2 The proposed architecture of the DCO

The DCO mainly is consisted of two main parts first is the coarse tuning block and second is the fine tuning block. Here the proposed architecture of the DCO with its two main blocks.



2.2.2.1 Coarse block

The coarse block consists of the a chain of 16 coarse segment blocks successively.



2.2.2.2 Fine block

The fine block consists of the a chain of 8 fine segment blocks connected in parallel with 5 inverters.

2.2.3 The control blocks of the DCO

2.2.3.1 Phase detector

2.2.3.1.1 Buffer delay line

The start signal ripples along the buffer chain and flip-flops are connected to the outputs of buffers. On the arrival of stop signal the state of delay line is sampled by flip-flops. One of the obvious advantages of this TDC is that it can be implemented fully digital. Thus it is simple and compact. However, the resolution is relatively low since it is the delay of one buffer.



2.2.3.1.2 Inverter delay line TDC

The resolution in this TDC is the delay of one inverter which is doubled compared to buffers delay chain. In this case, the length of measurement intervals is not indicated by the position of high to low transition but by a phase change of the alternation of high to low sequence. Consequently, the rise and fall delay of inverter should be made equal which requires highly match of the process. In addition, the resolution is still limited by technology and therefore not high enough in our application of ADPLL.



2.2.3.1.3 Synchronization block

The synchronization block is used to sample the reference clock by the output fast clock of the DCO.

The main disadvantage of this design is that reference frequency must be larger than DCO frequency to allow sampling.



2.2.3.1.4 SR flip flop and counter PD

Reference frequency is connected to S and DCO frequency to R. When there is a phase difference then Q goes one so enables the counter to start, the counter resets when a new cycle of reference clock starts. The output of the counter is a number proportional to phase error, It counts as long as the pulse of phase error is high

The main disadvantage that it needs very high clock to run the counter and it can not detect the small frequency differences.



2.2.3.1.5 Phase Frequency Detector (PFD)

The PFD architecture using two DFFs and a AND gate . The DFFs are triggered by the inputs to the PFD. Initially, both outputs are low. When one of the PFD inputs rises, the corresponding output becomes high. The state is held until the second input goes high, which in turn resets the circuit and returns the PFD to the initial state.



From the characteristics, the input linear range is from -2π to 2π .



2.2.4 Loop filter (Controller)

Loop filter decides the stability of the PLL design. The locking time and the tracking phenomenon is dependent on the filter used in the design. The general purpose of the loop filter is to eliminate higher order frequencies and any harmonics.

2.2.4.1 Successive approximation mechanism

The controller is divided to two stages coarse stage and fine stage, According to Up/Down signal from PFD the controller takes its decision and generates required control bits for the DCO

The coarse stage based on successive approximation idea where if the UP/Down signal is high which means that the DCO frequency needs to be increased so the controller generates control bits in the middle of the range between reference frequency value and the maximum frequency value in the range

The fine stage works after the coarse stage it steps up and down according to UP/Down signal also to approach reference frequency.

Disadvantage of this design is that algorithm does not help some definite frequency values but its advantage is the small lock time

2.2.4.2 Stepping mechanism

The idea in this controller is to take step up or down according to the Up/Down signal. It divided into coarse stage and fine stage also, but here both stages based on stepping idea .

Disadvantage of this controller is that it takes relatively high lock time compared to successive approximation mechanism but the advantage that no problem in locking on definite frequency values.

2.2.4.3 Shift Registers

Binary Word	Thermometer Code
0000	000000000000000
0001	100000000000000
0010	110000000000000
0011	111000000000000
1100	111111111110000
1101	111111111111000
1110	11111111111100
1111	111111111111110

These shift registers are considered the interface between the control blocks and the DCO, they are designed to generate the Thermometer Code.

Table 2-1:Bianry word Vs. Thermometer code

Two shift registers are used for controlling the DCO, a 16 bits shift register is used to control the coarse tuning stage, a 8 bits shift register is used to control the fine tuning stage . When the shift register reaches the minimum value (all zeros) or the maximum value (all ones) and an extra decrement or increment is performed. The shift register will keep its state.

Chapter 3 Design Flow

3.1 Verification of the DCO function

As mentioned before it is considered the CORE of the whole system because it is responsible for generating the different frequency ranges according to any change in control bits.

3.1.1 DCV

A Digitally Controlled Varactor is considered the basic unit of the design process for DCO as it's the way for constant step change in the delay generated by DCO as mentioned before.

It has many ways to implement all of them depending on changing the gate capacitance by constant step according to the applied voltage.

In this design, capacitance is being changed by varying input capacitance of Nand gate by varying voltage applied on gates of transistors forming nand gate.



3.1.2 Coarse tuning block

The coarse tuning block is the block responsible for changing the DCO frequency by a large delay step.

Coarse block consists of 16 identical coarse segments with different control inputs, all segments work in parallel.



Each coarse segment consists of a (NAND gate, inverting multiplexer and 3 active high enable D Latches). When the first coarse control bit is 0 the multiplexer selects the output of the previous segment on the left, resulting in a total delay equivalent to the delay of an inverting multiplexer.

This delay is constrained to be very close to 60ps. On the other hand, if the first coarse control bit is set to 1 and the second bit is 0, then the total delay is equivalent to the delay of 2 inverting multiplexers and a NAND gate. This delay is constrained to be approximately 195.6ps.

Whenever an extra coarse bit is enabled the delay is increased by approximately 150ps, which is the delay of a NAND gate and an inverting multiplexer. The delay step of each coarse segment is determined by the synthesis constraints. For the delay values mentioned above, enabling an extra coarse bit will increase the clock period by 150ps. When all the coarse control bits are disabled, the minimum delay is achieved through the coarse tuning part that is 60ps. This means that in order to achieve the basic operation of the segment is come from changing the bit applied for it and it will generate the fixed value for the delay required to cover certain frequency range.

Why we use Latch?

Changing the fine-tuning control bits can be done at any instant without any condition on the phase of the clock signal. However, this is not the case for the coarse-tuning control bits. Changing the coarse tuning control bits can introduce an undesired edge on the clock signal. If the control bit is changed while the output clock is 1 as in Figure 3-3 (a), and (c) the clock will not be affected, also if the control bit is changed from 1 to 0 while the output clock is 0 as in Figure 3-3 (d). However, if the control bit is changed from 0 to 1 while the output clock is 0 as in Figure 3-3 (b) a glitch will appear on the clock path that will change the clock frequency and make the system unstable. In order to avoid these problems, the control bit is stored in a transparent latch that is controlled by the output clock



Number of coarse segments used depends on maximum and minimum delays required, Here we used 16 coarse segments.



3.1.3 Fine tuning block

The fine tuning block is the block responsible for changing the DCO frequency by a small delay step.

Fine block consists of 5 inverters and 8 identical fine segments with different control inputs, it based on DCV idea, In this design DCV is implemented by NAND gate.





Each fine segment consists of 2 nand arrays, each array composed of 4 nand gates

Fine stage uses 8 thermal bits to control 64 DCV cells. Each control bit from this stage is connected to 8 DCV cells to reach wide range of variation.



3.2 Phase Frequency detector

This part was written as a VHDL code. It based on normal PFD which consists of two active high reset D flipflops and AND gate. The DFFs are triggered by the inputs to the PFD. Initially, both outputs are low. When one of the PFD inputs rises, the corresponding output becomes high. The state is held until the second input goes high, which in turn resets the circuit and returns the PFD to the initial state. This PFD is followed by a small circuit to convert the two output signal to one Up/DOWN signal.

When reference clock leads DCO clock ,Up/DOWN goes high.



When DCO clock leads reference clock $\overline{,Up/DOWN}$ goes low



3.3 Controller (Loop Filter)

The controller is responsible for setting control bits of DCO to achieve required response and to stabilize the system. This part is also written as a VHDL code.

It is divided into two stages, coarse stage controller and fine stage controller.

It works according to the Up/Down signal comes from PFD. When the signal is high which means that DCO frequency needs to increase, the coarse stage makes one step up and when the signal is low which means that DCO frequency needs to decrease, the coarse stage makes one step down.

When the DCO frequency approaches reference frequency the coarse stage stops working and a signal goes high, this signal is responsible for running fine stage controller.





The fine stage controller also steps up and down also depending on Up/Down signal just like the coarse stage but its step is smaller than the step of the coarse stage.

Chapter 4 system layout

4.1 DCO

In this chapter we discuss the full custom layout of DCO block, and

Layout simulation add small delay to your system, so the acheived schematic results are not the same as layout results. The main result that affected by layout simulation is frequency range, Here we show the differnce between the two simulation results.

	Schematic	Layout
Trial 1	187.10 M : 1.049 G	73.0 M : 550.0 M
Trial 2	206.00 M : 1.515 G	91.8 M : 694.2 M
Trial 3	394.78 M : 2.400 G	98.9 M : 720.5 M
Table 4-1 Schematic Vs. Layout		

4.1.1 Nand layout

VMD B ∰ohl<mark>M</mark>Ø ₽ ₿ 9ach I<mark>⊮</mark>1 Ø 177 DIE N/ 9ach I<mark>M-2</mark> Ø.1270.06 B €) -ØL127Ø.1 ₿ ₿ ØF GX

Dimensions: X=6.8 Y=3.9

4.1.2 NOR layout

Dimensions: X=10.6 Y=9.5



4.1.3 AND layout



4.1.4 One stage INVERTER





4.1.5 Two-stage INVERTER

Dimensions: X=11 Y=3.1


4.1.6 MUX block





4.1.7 D-latch

Dimensions: X=6.4 Y=7.8



4.1.8 Fine segment

Dimensions x=15.5 y=8.1



4.1.9 Coarse segment

Dimensions $x=40.01 \quad y=15.7$



4.1.10 fine stage

Dimensions x=6.1 y=51



4.1.11 the whole DCO layout

Dimensions: x=176.1 y=63



4.2 PFD and Controller

We work on standerd cells layout for PFD and controller blocks, there is no final results found but there is many steps are taken, Here is the final layout but with some errors we will work on solving it later.



Chapter 5 Simulation Results

In this chapter the simulation results will be discussed in details.

5.1 DCO

It is the core of PLL as mentioned before, Here is the result of its simulation, Mentioned two graphes from Virtuoso simulation show the maximum ana minimum frequency (Frequency Range) of our PLL.





5.1.1 Coarse Stage

It is reponsible for the lareg frequency steps,

5.1.1.1 Coarse Delay

Here is table shows the delay steps of Coarse Stage

Coarse control word	Delay
0000 0000 0000 0000	1.388 nsec
1000 0000 0000 0000	1.873 nsec
1100 0000 0000 0000	2.357 nsec
1110 0000 0000 0000	2.837 nsec
1111 0000 0000 0000	3.704 nsec
1111 1000 0000 0000	4.190 nsec
1111 1100 0000 0000	4.600 nsec
1111 1110 0000 0000	5.157 nsec
1111 1111 0000 0000	6.027 nsec
1111 1111 1000 0000	6.519 nsec
1111 1111 1100 0000	7.004 nsec
1111 1111 1110 0000	7.477 nsec
1111 1111 1111 0000	8.361 nsec
1111 1111 1111 1000	8.848 nsec
1111 1111 1111 1100	9.33 nsec
1111 1111 1111 1110	9.817 nsec
Table 5-1 Co	arse Delay steps

5.1.1.2 Coarse frquency

Here is table shows the frequency steps of Coarse Stage

Coarse control word	Frequency
0000 0000 0000 0000	720.46 MHz
1000 0000 0000 0000	533.90 MHz
1100 0000 0000 0000	424.14 MHz
1110 0000 0000 0000	352.48 MHz
1111 0000 0000 0000	269.97 MHz
1111 1000 0000 0000	238.64 MHz
1111 1100 0000 0000	213.87 MHz
1111 1110 0000 0000	193.89 MHz
1111 1111 0000 0000	165.90 MHz
1111 1111 1000 0000	153.39 MHz
1111 1111 1100 0000	142.77 MHz
1111 1111 1110 0000	133.74 MHz
1111 1111 1111 0000	119.60 MHz
1111 1111 1111 1000	113.02 MHz
1111 1111 1111 1100	107.18 MHz
1111 1111 1111 1110	101.86 MHz
Table 5-2 Coarse	Frequency steps

5.1.2 Fine Stage

It is reponsible for the small frequency steps.

5.1.2.1 Fine Delay

Here is table shows the delay steps of Fine Stage

Fine control word	Delay				
0000 0000	1.388 nsec				
1000 0000	1.420 nsec				
1100 0000	1.460 nsec				
1110 0000	1.492 nsec				
1111 0000	1.529 nsec				
1111 1000	1.566 nsec				
1111 1100	1.601 nsec				
1111 1110	1.639 nsec				
1111 1111	1.677 nsec				
Table 5-3 Fine delay steps					

5.1.2.2 Fine frequency

Here is table shows the frequency steps of Fine Stage

Fine control word	frequency					
0000 0000	720.49 MHz					
1000 0000	704.21 MHz					
1100 0000	684.93 MHz					
1110 0000	670.24 MHz					
1111 0000	654.02 MHz					
1111 1000	638.56 MHz					
1111 1100	624.53 MHz					
1111 1110	609.90 MHz					
1111 1111	596.31 MHz					
Table 5-4 Fine	Table 5-4 Fine frequency steps					

5.2 Lock Time

Here is graph shows the worest case of Locking time



The typical locking time is around 60nsec



5.3 Jitter simulation result

Using the way of calculation mentioned in the appendix below

After simulating pss and pnoise analysis

The RMS jitter = 1.035 psec

Peak to peak jitter =6.396 psec

Design Variables	Analyses	
Nomo - Valua	Type - Enable Arguments	
Nallie Value	2 ncc 252 2M 10 /not4 /and	C DC
	a proise V 10.1.1G /net4 /gnd	
		T\$1
		× .
		~
	Outputs	
	Name/Signal/Expr - Value Plot Save	Sav 🝙
	1 net4	allv
	2 v /net4; pss (V) 🛛 wave 🗹 📃	<u>₩</u>
	3 Jcc[Second][k=1]@(1,1G 1.035p	
	4 Jcc[Second][k=1]@(1,1G 6.396p 🖉 📃	RMS Jitter
		P-to-p jitter
	Plot after simulation: Auto Plotting mode: Replace	
mouse I :]	R'

5.4 Area calculation

With the way of calculation mentioned in the appendix

Area = dy* dx= 171.9*58.4*10^-6 = 0.01 mm2



5.5 Power Calculation

By the appendix method the value last generated by calculator indicates the power of the $DCO = Vsource^*$ Isource = 0.952 mm2

Refrences

[1] Jayashree Nidagundi, Harish Desai, Shruti A., Gopal Manik "Design and Implementation of Low Power Phase Frequency Detector (PFD) for PLL".

[2] Prashanth Muppala B.Tech., Gayatri Vidya Parishad College of Engineering, 2008 "I

HIGH-FREQUENCY WIDE-RANGE ALL DIGITAL PHASE LOCKED LOOP IN 90 NM CMOS".

[3] Master of Science Thesis In System-on-Chip Design By Chen Yao Stockholm, 08, 2011 "Time to Digital Converter used in ALL digital PLL".

[4] Graduation project thesis ,Cairo University ,2013 ,"All Digital Phase Locked Loop (ADPLL)".

[5] Anitha Babu, Bhavya Daya, Banu Nagasundaram, Nivetha Veluchamy University of Florida, Gainesville, FL, 32608, USA "All Digital Phase Locked Loop Design and Implementation".

[6] Kusum Lata and Manoj Kumar ,survey ,"ALL Digital Phase-Locked Loop (ADPLL)".

[7] José A. Tierno, Alexander V. Rylyakov, Member, IEEE, and Daniel J. Friedman, Member, IEEE,"A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI".

[8] A Thesis Presented by Moon Seok Kim ,"0.18_x0016_m CMOS Low Power ADPLL with a Novel Local Passive Interpolation Time-to-Digital Converter Based on Tri-State Inverter".

[9] Ran Sun1, a, Lijun Zhang1, b, Hao Wu2, c, Jianbin Zheng2, "Design of the High Speed All Digital PLL for SRAM BIST Based on 55nm Process".

[10] Jingcheng Zhuang, Qingjin Du, Tad Kwasniewski Department of Electronics, Carleton University Ottawa, Ontario, Canada, "A 4GHz Low Complexity ADPLL-based Frequency Synthesizer in 90nm CMOS".

[11] A. V. Rylyakov1, J. A. Tierno1,G. J. English2, D. J. Friedman1,M. Meghelli3,"A Wide Power-Supply Range (0.5V-to-1.3V) Wide Tuning Range (500 MHz-to-8 GHz) All-Static CMOS ADPLL in 65nm SOI".

[12] Gursharan Reehal, M.S. The Ohio State University, 1998 Steve Bibyk, Adviser Phase,"A Digital Frequency Synthesizer Using Phase Locked Loop Technique".

[13]ECE 126 – Inverter Tutorial: Identifying Static and Dynamic Power in a CMOS Inverter

Appendix

A1.VHDL AMS tutorial

It used to convert a VHDL code to schematic

After writing the code and save it .vhd apply the following steps to use the code as symbol in cadence.

Some hints about the code to be successfully imported :

- Write end "entity name" instead of end "entity". example: Entity mux is port(....); End mux; Instead of Entity mux is port(.....); End entity;
- 2. In architecture part write "architecture behavioral of entity name is" The word after architecture should be behavioral And also to end architecture it should be " end behavioral".

In case a component is recalled in the code :

- 1. Take in consideration hints (1) & (2).
- 2. Put the code of the component above the code that is recalling it
- Remove the "is" word when recalling component Example: component counter port(.....);

end component;

4.put the codes of the components with the same arrangement of recalling them in the total code

1. Open cadence and from cds.log press file > import> vhdl

New	script		
Open	root@localhost:~	_ □	1
Import	rminal <u>H</u> elp		
Export	Varian \$ su -		
Refresh	Verlog		
Make <u>R</u> ead Only	ce		
<u>B</u> ookmarks	DEF OSO		
I don lavout NOR schematic	LEF		
2 den lavout NOR lavout	Stream		
S GR NOR schematic	Netlist View		
 <u>a</u> GP nor gate schematic 			
F dee lowert AND lowert			
C des lavest AND rayout			
B dco_layout AND schematic			
E 2 GP DCO conig			
Schematic			
9 fine new_fine_tb config			
<u>C</u> lose Data			
E <u>×</u> it	uoso® 6.1.4 - Log: /root/CDS.log _		
Eile Tools Options Sonnet H	elp cādence		
andian Witan Cin ant			
Joading AMSOSS.cxt			
oading AMS.cxt			-
.oading apsi.cxt			
mouse L:	M: R:		
VHUL			

2. Write the file location in file name then press enter Note: remove the word "sample" from reference libraries

VHD	L Import
Import Options Schematic Gener	ration Options
File Name	Target Library Name
/home/Desktop/	
InstallAnywhere/ artist_states/ .cache/ .cache/ .config/ .dbus/	Add >> << Remove
/root/*.vhd	
Import Structural Architectures As	schematic
Reference Libraries	basic US_8ths ieee std sample
Symbol View Name	symbol
Overwrite Existing Views	⊻
Case Sensitive Symbol Matching	_
User Specified Standard Libraries Maximum Number of Errors	10
Compile VHDL Views After Import Compiler Options	
VHDL WORK Library Name	
Summary File	. /vhdlin. summary
Compatibility Option	
v93 Option	
Power	
Net Name wdd L	Value 11

3. Choose the file and press add , it will appear in right widow Then write your library's name in the target library name

VHD	L Import ×
Import Options Schematic Gener	ration Options
File Name	Target Library Name
	TRY
D_FF.vhd MUX2x1.vhd PD_cap.vhd SR_16bit.vhd SR_9bit.vhd coarse.vhd	Add >> <remove< td=""></remove<>
/home/eslam/Desktop/controller/	/*.vhd
Import Structural Architectures As	schematic
Reference Libraries	basic US_8ths ieee std sample
Symbol View Name	symbol
Overwrite Existing Views	
Case Sensitive Symbol Matching	
User Specified Standard Libraries	
Maximum Number of Errors	10
Compile VHDL Views After Import Compiler Options	
VHDL WORK Library Name	
Summary File	. /vhdlin. summary
Compatibility Option	
v93 Option	
Power	
Net Name [vdd]	OK Cancel Defaults Apply Help

4. To create your library press yes in the following window



Then

New Library			
Library Name Directory (non-library directories) ARCH Desktop Documents Downloads ENTI Music /root	Technology File Compile an ASCII technology file Reference existing technology libraries Attach to an existing technology library Do not need process information Design Manager: No DM		
	OK Cancel Defaults Apply H	lelp	

5. To make sure that the file.vhd is successfully imported you need to see the log file



After viewing the log file make sure it is like this

	VHDL ToolBox Log File _ 🗆 🗧	×
<u>F</u> ile	Help cādence	2
INF0 @(#)\$	(VHDLIN-284): VHDL In Run Summary SCDS: vhdlin version 6.1.4 11/17/2009 20:49 (sjfnl007) \$ Mon Jul 7 10:57:45 2014	
INFO INFO INFO INFO INFO INFO INFO	<pre>(VHDLIN-238): Processing VHDL source file: /home/eslam/Desktop/controller//MUX2x1. (VHDLIN-264): Done. (VHDLIN-244): Vhdl Design Unit: mux2x1 : Entity (VHDLIN-243): Created symbol view of type symbol. (VHDLIN-245): Created entity view of type Vhdl (VHDLIN-251): Vhdl Design Unit: mux2x1.behavioral : Architecture (VHDLIN-245): Created behavioral view of type Vhdl (VHDLIN-255): -> (/home/eslam/Desktop/controller//MUX2x1.vhd,11) Signal a</pre>	
****	***************************************	
INF0 *****	(VHDLIN-229): Number of file(s) processed in this round is 1	

4		

6. Now you have a symbol with entity name in your library

I		Library Manager: WorkArea: /root				
	<u>Eile Edit V</u> iew <u>D</u> esign Manager <u>H</u> elp		cādence			
	Elle Edit View Design Manager Help Show Categories Show Files Ubrary TRM FET_VCO_wLayout FSK_receive_jub GP IEEE IEEE_vhalams IEEE IEEE Catoline Catoline Catoline Catoline Catoline Catoline Catoline Catoline IEEE IEEE IEEE IEEE IEEE IEEE IEEE IE	Cell mux2x1 mux2x1	Cādence View symbol behavioral entty symbol TRV/mux2x1/symbol			
	cds_spicelib coarse_positive_edge connect.lb dco_layout dco_layout					

7. create a new cell view and browse your library to get the symbol

N 0			Virtuoso® Sc	hematic Editor L E	diting: TR	RY mux2	2x1_tb schema	tic				_ = ×
Launch E	ile <u>E</u> dit <u>V</u> iew <u>C</u> reate	Chec <u>k</u> O <u>p</u> tions <u>M</u> igrat	e <u>W</u> indow Calibre <u>H</u> elp								c	ādence
16 🖬		X 🛈 Ty 🛤 🤇) e Q Q Q	🕅 🎼 🕄 1	abc 🕕							
							Ad	d Instance	×			
						Library	TRY		Browse			
	<u>8</u>	Library Browser -	Add Instance	>	×	Cell	mux2x1					
	✓ Show Categories					View	embol		-			
	Library	Category	Cell	View		VIEW	SYNDOT		_			
	TRV		mux2v1	cumhal		Names						• • • •
	FET VCO wi avout	-	mux2x1	symbol		Array	Rows	1 Columns	1			
	FSK_receiver_lib		mux2x1_tb	TRY/mux2	x1/symbol							
	GP IEEE				-		Rotate	🚹 Sideways 🦯 🥞 Up	side Down			
	IEEE_vhdlams											
	Nangateiny											
· · · · · ·	US_8ths				1.1.1.1.1.1							
	adsLib									· · · · · ·		• • • •
	andiLip											
	analogLib											
	analog_digital											
	cdsDefTechLib											
	cds_assertions 🦷											
					1.1.1.1.1.1							
• • • •												• • • •
	Close	Filters	Display	Help								
				<u> </u>								
							_	_				
							•	ide Cancel Defa	aults Help)		

8. After connecting inputs to the symbol then save schematic and close it then create a new cell view with the same name but with view type :config

-	New File ×
File	
Library	TRY
Cell	mux2x1_tb
View	config
Туре	config 🧧
-Application	
Open with	Hierarchy Editor 🧧
🔲 Always use	this application for this type of file
Library path file	3
/root/cds.li	b
	OK Cancel Help

9. press on use template in the window appeared then choose ams Then choose view:schematic in the next window

Virtuoso® Hierarchy Editor	_ 🗆 ×
<u>Eile Edit View Plugins H</u> elp	cādence
Top Cell Library: Cell: Cell: View: Open Edit Table View: Cell: Table View: Open Edit View: Open From File: Other> From File: Mame: Other> Prom File: Massic Oper Spectre spectre/enlog spectre/enlog verilog Verilog <td>7 6 ×</td>	7 6 ×

10. when this widow appear press save and close it

	Virtuoso	® Hierarchy I	ditor: New Co	nfigura	ation (S	Save M	leeded)	- [⊐ ×
<u>F</u> ile <u>E</u> i	dit <u>V</u> jew <u>P</u> Jugins	s <u>H</u> elp						cāder	ıce
	Save the c	O- 5 6		5					
Top Cel	I Save the c	oninguration (Nee		? 🗗 🗙	Global	Binding	IS	7) & ×
Library	TRY				Library	List:	myLib		
Cell:	mux2x1_tb				View Li	st:	əriloga vho	II vhdlams wreal)
View:	schematic								
					Stop Lis	st:	spectre)
Open	Edit				Constra	int List:			
Cell E	le View Tree Bindings Library	e View	View Found	View	To Use	Inherite	ed View Lis	Inherited Lib Lis	t
	TRY	mux2x1	behavioral			spectr	e spice	myLib	
686	TRY	mux2x1_tb	schematic			spectr	e spice	myLib	_
	anaiogLib	Vac	spectre			spectr	e spice	MyLIB	
Nam	espace: CDBA – F	Filters: OFF							
5(14) S	ave the configurat	ion							

11. now your cell has a config view type open it.

M	Library Manager: WorkArea: /root	_ 0
<u>Eile E</u> dit <u>V</u> iew <u>D</u> esign Manager <u>H</u> elp		cādeno
Show Categories Show Files		
Library	Cell	~ View
TRY	mux2x1_tb	config
FET_VCO_wLayout FSK_receiver_lib GP IEEE_vhilams Nangaleinv IEW US_8ths adsLib ahdILib analog_tib analog_tib analog_tib cds_preceibic cds_preceibic cds_preceibic cds_preceibic coarse_positive_edge connect.ib dcc_layout	mux2x1 mux2x1_tb	config schematic TRV/mux2x1_tb/config

Press ok in the following window

Open Configuration or Top (CellView ×
Open for editing	
Configuration "TRY mux2x1_tb config"	🔾 yes 💿 no
Top Cell View "TRY mux2x1_tb schematic"	🖲 yes 🔾 no
ОК	Cancel Help

12. simulation is applied for this cell first check and save > Launch > ADE L > setup> simulator/directory/host > choose ams in simulator > press ok



13. then setup >connect rules

	am	s1: Customize Built-in Rules
escription Thi	s is the de	escription for ConnRules_18∀_full_fast
Connect Module (Declarations	
Module E2R R2E_2 ER_bidir L2E_2_CPF Bidir_2_CPF	Mode	Parameter/Values vdelta=`Vsup/64 vtol=`Vdelta/4 ttol=`Tr/20 vsup=1.6 vdelta=`Vsup/64 tr=`Tr/20 tf=`Tr/20 rout=200 vdelta=`Vsup/64 vtol=`Vdelta/4 ttol=`Tr/20 tf=`Tr/20 tf=`Tr/20 vsup=1.8 vlo=0 tr=0.2n tf=0.2n rlo=200 rhi=200 rx=40 rz= vsup=1.8 vthi=1.2 vtlo=0.6 tr=0.2n tf=0.2n rlo=200 rhi=200
		View connect module View defines
Mode	•	
Parameters		
Parameter	vsup vlo tr	1.8 0
	tf rlo rbi	200
Parameter	[Value Change
Direction1	-	Discipline1
Direction2	-	Discipline2
Connect Resoluti	ons	
		OK Cancel Apply Disciplines H

Select all connect mode declarations > press customize

In the parameters vsup=1.2 > press change, Vthi =0.8 > press change, vtlo=0.4 > press change

14. Now choose analyses type and plot the outputs

	Choosing Analyses Virtuoso® Analog Design ×	💐 📑 🕺 1 🚈 ·
tie Virt	Analysis 💿 tran 🔾 dc 🔾 ac 🔾 envlp	config _ □ ×
S <u>e</u> ssion Set <u>u</u> p		cādence
	Transient Analysis	? 5 ×
Design Variables	Stop Time	ts 🛛 🛃
Name	Accuracy Defaults (empreset - Spectre Only)	©AC ODC OTrans
	🗌 conservative 🔲 moderate 🔛 liberal	89
	Transient Noise	×
	Enabled Doptions	
	OK Cancel Defaults Apply Help	ave Save Options
	1 out	no 📈
	Plot after simulation: Auto Plotting	mode: Replace
mouse L:	M:	R:
2(4) Plot Outputs	Status: Selecting outputs to be plotted T=27 C Simu	lator: ams(Spectre)Mode: batch

A2.ModelSim tutorial

This program to deal with HDL laguagues, we will focus on VHDL.

1. Open modelsim then choose file>new>project



2. This window will appear, write your project name and location you want to save it in, note that your library name is "work" "it is a defult name", then OK.

M Creat	e Project	×
Project Name ADPLL Project Location		Model
Default Library Name	Browse	
Copy Settings From /opt/modeltech/modelsim.ini Copy Library Mappings	Browse	s
	OK Cancel	

3. Select "Create New File" to write your code in new file Or "Add Exisiting File" to include files to your project.

🔣 Add items t	o the Project ×	1
Click on the icon to	add items of that type:	
Create New File	Add Existing File	ModelSim.
Create Simulation	Create New Folder	
	Close	

4. Write your project name and select VHDL as file type "Or any file type", then OK.

Click on	I items to the Project the icon to add items of that type	×
File Name	Create Project File	
Add file as type	Folder Top Level	Browse
VHDL PSL Text SystemC TCL Macro	<u>ОК</u>	Cancel
SDF XML		

5. After creating your files this subwindow will appear with names of your files.

Workspace ==== >>>>=	= + 🖻 🗙
▼ Name	∆ Sta
📕 sara.vhd	?
📕 sss.vhd	?
] 🐺 Project 🛛 📶 Librar	au ≪ ≫

6. Then write your code and save it.



7. Select your file from Workspace then click compile.



8. If there were errors, it would be written in Transcript subwindow, Here there is no errors, Note that when the file is compiled successfully, checkmark is written beside its name in Workspace.

Workspace 💳 🛲 🛨 🖬 🗙		
▼Name △ Sta	ln #	
H sara.vhd H sss.vhd	$ \begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\\5\\16\\17\\18\\20\\21\\22\\23\\24\end{array} $	<pre>library IEEE; use IEEE.std_logic_1164.all; entity D_FF is port(</pre>
		M
Project 👖 Library 🔍	H sar	a.vhd
Transcript		
ModelSim> # Compile of sara.vhd was successfu	1.	
ModelSim>		

9. Click simulate, this window will appear, choose your library name"work", you will find the name of your entity, choose it then OK.

°o 🗍 🕸 🎟 🛺 🛙	Contains 2
n/Desktop/try/s <mark>ara.v</mark> hi	t construction of the second se
IEEE.std_logi	ic_1164.all;
ity D_FF is	🔣 Start Simulation ×
Rst,D,c Q:out s	Design VHDL Verilog Libraries SDF Others
); entity	Name Tune Path
hitecture str	⊒- <mark>∭u</mark> work Library /home/e <mark>a</mark> m/Desktop/try/work
<pre>nal temp:std_</pre>	
in cess(clk Pst)	TECH//sv_std Library \$MODE TECH//sv_std
in	tital2000 Library \$MODEL_TECH//vital2000
if(Rst='1')	Library \$MODEL_TECH/./ieee
elsif(Rst='0	The modelsim_lib Library \$MODEL_TECH//modelsim_lib
end if	std Library \$MODEL_TECH/./std
end if	tibrary \$MODEL_TECH//std_developerski
process.	
struct.	Design Unit(s) Resolution
501400,	work.d_ff 🛛 🗸 🗸
	✓ Enable optimization Optimization Options
J	
	OK Cancel

10. If there is errors it will be written in Transcript subwindow, Here there is no errors.



11. In Workspace, right click on your entity name then add>add to wave.

Workspace	= ::::: = + a ×				
▼ Instance		ln	#		
	/iew Declaration /iew Instantiation			library IEEE; use IEEE.std_lo	ogic_1164. all;
📕 🗾 stc 🗚	Add	•	Add	All Signals to Wave	
sta C	Create Wave	I	Add	to Wave	<pre>clk:in std_logic; std_logic</pre>
F	Copy Find		Add Add Add	to Dataflow to List to Watch	ruct of D_FF is Llogic ;
E	Expand Selected Collapse Selected Expand All Collapse All	1	Log	if(Rst='1') elsif(Rst= if(clk)) then temp<='0'; '0') then 'event and clk='1') the
S	Save List			end if; end process;	
C	Code Coverage	۶		Q<=temp; end struct;	
E	End Simulation				

12. This window will be opened with the names of outputs and inputs.



13. To set input to specific value '1' or '0' right click on its name, then force.

📰 wave - de	erault							
	Object Declaration View Assertion Cover Directive View	×						
✓ /d_f	Radix Format	*						
	Cut							
	Copy							
	Delete							
	Create/Modify Waveform Map To Design Signal							
	Insert Divider							
	Insert Breakpoint							
	Force		l ns					
T H) sara.vhi	NoForce	NoForce						
	Clock							
	Properties							

Write value. Then OK.

М	₩ Force Selected Signal ×											
Signal Name: sim:/d_ff/rst												
Value: 0												
				ī								
	 Freeze 	 Drive 	 Deposit 									
De	elay For: 0											
Can	cel After:											
		<u>о</u> к	<u>C</u> ancel									

- 💼 wave default U U /d_ff/rst /d_ff/d /d_ff/clk
 /d_ff/q
 /d_ff/temp ່ບ Object Declaration View Assertion Cover Directive View Radix Format Cut Cntl-X Cntl-C Сору Paste Cntl-V Delete Create/Modify Waveform Map To Design Signal N Insert Divider Curso Insert Breakpoint H] sara.vhd 💼 wave Force... NoForce Properties...
- 14. To set clock, right click on the wanted signal then choose "clock".

15. Set its period in nsec and edge type then OK.

Define Clock											
Clock Name											
sim:/d_ff/clk											
offset	Duty										
0	50										
Period	Cancel										
100											
Logic Values											
High:	Low: U										
First E	dge										
 Rising 	g 🔿 Falling										
	OK Cancel										

16. To simulate your code, click run after assigning all input signals, you can set the simulation period in ns.

C	8 12 🗅 M 🖺 😘	L ≯ ¥ ± ±	N 🖳 🗉 🖪 😣 縃		🔒 🔶 🛸 📑 100 ns 🛊	🖬 27 (6 () 🖬 💷
				Layout Simulate	C ntains	Run
	wave - default					
	✓ /d_ff/rst✓ /d_ff/d	0 1				
		1				
	✓ /d_ff/temp					
	Now	200 ns	200	400	600	10000000000000000000000000000000000000
	Cursor 1	0 ns	0 ns			

17. Check your simulation results. 😳

A3.Cadence Tutorial

After running your program

1- create new library





2- create new cell

<u>File E</u> dit <u>V</u> iew <u>D</u> esign	Manager <u>H</u>	<u>f</u> elb	
<u>N</u> ew → <u>O</u> pen Open (<u>R</u> ead-Only) → Open With	Ctrl+O Ctrl+R	Library Cell View Category	
Load Defaults Save Defaults			New File ×
O <u>p</u> en Shell Window	Ctrl+P	Library	test
analogLib basic cdsDefTechLib	Carrie	Cell	Inv
		View	schematic
		Туре	schematic 🔽
		Application —	
		Open with	Schematics L
		🔲 Always use	this application for this type of file
		Library path fil	e
		/root/kits/1	CSMC_65nm/cds.lib
			OK Cancel Help

3- To add component in your cell

20 20	oso® Schematic Editor L Editing: test Inv schematic												
Launch File Edit View Create Check Options Migrate Win	dow Calibre <u>F</u>	<u>H</u> elp											
P 🖌 🖯 🏟 X 🛈 Y 🕫 🦻 🥐 🔍 🔍 🍳 🍳 🎘 🔤 1, 1, 🛎 🛥 E													
Add Instance	×												
Library	Browse												
Cell													
View symbol													
Names		· · · · · · · · · · · · · · · · · · ·											
Array Rows 1 Column	s 1												
🕼 Rotate 🖉 🕼 Sideways 🖉 U	pside Down	-											
Hide Cancel Det	faults Help												
		-											

🐲 Applications	s Places	System 🥣 🧝	j 📕 🔛						Sun Jui	13, 1:26 A	.M (1 <mark>6</mark>
»		Add	Instance	1	2	Library Browser	r - Add Instance	-	. 🗆 ×		
Launch <u>F</u> ile <u>E</u> o	Library	tsmcN65		Browse	Show Categories						
	Cell	nch			Library	Cell		View			
	View	symbol			tsmcN65	nch		symbol			
	Names				COARSE	nand3_hvt_	dnw_mac	ADVance_MS ams			
					GP-dahy	nand3_lvt		auCdl			
	Array	Rows 1	Columns	1	NEW_layout	nand3_lvt_c	dnw_mac	eldoD			
		🖹 Rotate 📄 🖉 👍 S	iideways 🛛 ╡ Ups	ide Down	US_8ths ahdILib	nand3_lvt_r nand3_mac	nac	hspiceD spectre			
					analogLib basic	nand4 nand4 25		symbol			
	Model nar	ne r	nch		cdsDefTechLib	nand4_25_0	dnw dnw mac				
	description	n a	lard VT NMOS tran	sistor	dco_layout	nand4_25_0	dnwod33				
	L(M)	ē	50n M		fine	nand4_25_0	dnwud18				
	w (M)	2	200n M		functional new_sch	nand4_25_0 nand4_25_r	dnwud18_mac				
	total_width	(M) 2	200n M	_	rfExamples rfLib	nand4_25or	d33 mac				
	Number of	Fingers 1	1	_	stdc	nand4_25uc	d18 d18 mac				
	Multiplier	Ĩ	1	_	tsmcN65	nand4_dnw	aro_mac				
	total m	G	1		tt	nand4_dnw nand4_hvt	_mac				
	Hard con	strain 💌	2			nand4_hvt_ nand4_hvt_	dnw_mac				
	Calc Diff F	'arams 💌	2			nand4_hvt_ nand4_lvt	mac				
	Source a	ea S	3.5e-14			nand4_lvt_c	dnw dnw mac				
	Drain are	. 3	3.5e-14			nand4_lvt_r	nac				
	_					nand4_mac	T				
		Hie	de Cancel De	efaults Help		25 No. 25					
 . .	· · · · ·		• • • • • •		Close	Filters	Displa	ay	нер		
∝mouse L: mouseA	ddPt()				1	/I: Rotate 90					R: sch
(3) Point at locat	tion for the i	nstance.									Cm
		1	1 rans	<u> </u>	••)(£730).						

4- Select component from library manager

5- place the component in your schematic



6- To connect between components choose Then connect



7- To name the wires

9	¢			l ∥ " 8 1 1	abc 🖚 🗄								
· ·				Add \	Add Wire Name								
			Wire Name	Net Expression				÷					
			Names	in out ydd ydd									
			1 damoo		g			ŀ					
			Font Height	0.0625	Bus Expansion	🖲 off 🔾 on						•	
			Font Style	stick 🔽	Placement	🖲 single 🔾 multiple		·					
		· ·	Justification	lowerCenter	Purpose	🖲 label 🔾 alias							
			Entru Stule	fixed offset	Bundle Display	horizontal O vertica	al						
		· ·	Entry otyle			aus Official Defaulte							
					Sil	ow Oliset Delauits	_						
			🛛 🐴 Rotate					ŀ					
· ·		· ·			Hide Ca	ncel Defaults Hel	ql						
				i i i incli	MØ "neh"								
				╘╘╍┙┥╒╸	w:200n								

Then place the name on the desired wire



8- To set the parameters of components, Right click on component => Properties



9- After connectin your design click "check and save" to check the errors



10- To simulate your design

	N.																					Vi
	<u>L</u> aunch	<u>F</u> ile	<u>E</u> di	it <u>)</u>	<u>V</u> iew	<u>C</u> r	reat	e	Cł	nec	<u>k</u>	Op	tior	ns	M	gra	te	W	ind	ow	С	alit
	<u>A</u> DE	EL							*	٤	()	T	1	ę۵		\$	(è		Q	L
ï	A <u>D</u> E	= XL					_			_							_		_			
I	AD <u>E</u>	<u>=</u> GXL																				
I	ADS	s Dyn	amic	Lin	ĸ																	
I	La <u>y</u>	out XI	L																			
I	Lay	out <u>G</u>)	XL																			
I	Sch	ematio	os <u>L</u>																			
I	Sch	ematio	os <u>X</u> L	-																		
I		orchu	Edit	or		-																
I	<u>n</u> iei Mise	archy ad Sie	r Eulo	UI Dinti	000									•								•
I		eu oig Hetier	inar c	opu	ons																	
I	<u>S</u> imi	ulatior -	1			1																
I		a 																				
I	Pa <u>r</u> a	asitics																				
						•																
															•							
															5,							
																1/1						
															- <i>C</i> 4	л у Х У	1=1	.z				
														1		2.4	2:Ø					
1																						
📲 Virtuoso® Analog Des	ign Environment (1) - test Inv scher	matic _ 🗆 ×																				
---	---	--																				
S <u>e</u> ssion Set <u>u</u> p <u>A</u> nalyses <u>V</u> ariables <u>O</u>	utputs <u>S</u> imulation <u>R</u> esults <u>T</u> ools <u>H</u> elp	cādence																				
Design Variables Name Value	Analyses Type - Enable Argument Outputs Name/Signal/Expr Value Plot Sa	Is Is																				
	Plot after simulation: Auto Plotting r	mode: Keplace																				
3(4) Choose Design	Status: Ready	T=27 C Simulator: spectre																				

11- this window will appear (ADE)

12- Set simulation type

Virtuoso® Analog Desigr	n Environment L Editing	j: test Inv schematic			
O <u>p</u> tions <u>Migrate W</u> indow Calibre <u>H</u> elp					
ign Environment (1) - test Inv schematic	_ = × "_	Choosing Analyse	s Virtuoso® Ana	log Design 🗙	
itputs <u>S</u> imulation <u>R</u> esults <u>T</u> ools <u>H</u> elp	cādence	Analysis 💿 tran	🔾 dc 🔾 ac 📿	noise	
Analysia	251	⊖ xf	🔾 sens 🔾 dcmatch 🤇) stb	
Tune – Enchiel Argumente		🔾 pz	🔾 sp 🔾 envlp 🧲	pss	
Type Ellable Alguinents		🔾 pac	🔾 pstb 🔾 pnoise 📿) pxf	
	DC	🔾 psp	🔾 qpss 🔾 qpac 🛛 🤇) qpnoise	
	<u> </u>	⊖ qpxf	🔾 qpsp 🔾 hb 🛛 🤅) hbac	
		🔾 hbnoise			
	▶®→	т	ransient Analysis		
	×				
		Stop Time 2n			
Outputs		Accuracy Defaults (errpl	reset)		
Name/Signal/Expr Value Plot Save Save		Conservative	, nderate 🔲 liberal		
	W				
	= 1	Transient Noise			
	1 <u>.</u>				
		Dvnamic Parameter			
		Enabled 🗹		Options	
Plot after simulation: Auto Plotting mode: Repla	ice 🔽 🔤 i i i i i	ОК	Cancel Defaults	Apply Help	
M:	R: <mark>=1 * *</mark>				
Status: Ready T=27 C	Simulator: spectre				



13- To select outputs to be plotted

To plot the voltage signal select the wire from schematic, to plot the current signal select the red terminal from schematic

-								Р	•																
-								×	<u> </u>																
									L VB																
								0	ŧ٦	/de=	=1.2														
								2	₹^																
•								•	÷																
-									· ·																
-																1.1									
															-										
									9						ž										
-																									
-																•									
-														P	ch da	- 6	och'								
														in	~	W	20	Øп							
												.		q	<u> </u>	Ye	id Sian								
															out.	L fi	nge	rs:1							
-																si	mM	:1							
-									ìn			•				10	xun	WE I		O	đ.				
-		ĥ	· ·									•			1	•									
			T V													- 4.4									
	п	ъĆ	Þ,	v1=	1.2									_	sut	11	nch	ee							
		1	~	v2:9	3									in		W	:2Ø	Øn							
												<u> </u>			E	- 9'									
-			· ·											à	nd!	fi	nge	rs:1							
-			- I													l si	mN	11							
			1.1														AUG I	mie i							
				nd											idi										
			\checkmark												Ъ										
-																									
-																									
-																									

14- Then click run on ADE

🐒 Virtuoso® Analog Des	ign Environment (1) - test Inv schematic _ \Box ×
S <u>e</u> ssion Set <u>u</u> p <u>A</u> nalyses <u>V</u> ariables <u>O</u>	utputs <u>S</u> imulation <u>R</u> esults <u>T</u> ools <u>H</u> elp cādence
Design Variables Name - Value	Analyses Image: Construction of the state of the s
	Plot after simulation: Auto Plotting mode: Replace
mouse L:	M: R:
3(4) Netlist and Run	Status: Ready T=27 C Simulator: spectre

15- when program finish simulation, a graph of output signals will appear



16- to display the parameters of component, select the component then Eidt => compnent display

<u>L</u> auno	h <u>F</u> ile	<u>Edit View Create</u> Chec	<u>k</u> O <u>p</u> tions	<u>M</u> igrate	<u>W</u> ind	low C	alibre	<u>H</u> elp)								
		∲ Undo	U Shift+U	ei 🥎	¢						5	1	1	abc	-0		
		💠 <u>M</u> ove	Shift+M														
		🛑 <u>С</u> ору	С														
		🔯 Stretch	M				ν										
		🗙 <u>D</u> elete	Del					VØ .	10								
		R <u>o</u> tate					- 0	₽ ^{vae}	=1.2								
		🏷 Direct <u>T</u> ext Edit	т														
		Sheet Size															
		Sheet <u>Title</u>										1	2 .				
		<u>H</u> ierarchy	•														
		<u>P</u> roperties	•									net	м				
		Net Expression	•									, vd	1	sch"			
		Select	•										- ¥	Id Id			
		Boute Flight	5									00	t 🎁	ngers:			
		Renumber Instances		2 J 2 2 2									to	talM:1			
		Compo <u>n</u> ent Display		.5									6				
		<u>A</u> lternate View								· ·							
		Update Pins From <u>V</u> iew		v1=1								net	- M	Ø hch″			
		Eind		V2:Ø								in	W	:2ØØn			
		Elfiu											- 1	60n nders:	1		
		n <u>e</u> piace										- grie	si to	mM:1 talM:1			

Then follow this figure

Edit Component 'M0' Display ×
Select Label 🛛 terminal 🗹 parameter 🛄 instance
Auto Redraw 💿 yes 🔾 no 🛛 Save Load Attach Detach
Set Simulation Data Directory
Parameter Labele
Apply Tn Ibrary Cell Cinstance
○ none ○ parameter ○ model parameter ● operating point
● DC ◯ transient
Dienlau Volue Onlu
vgs v
vth N
vdsat
region
none
none
OK Cancel Previous Next Help

rate	<u>W</u> indow	Calibre <u>H</u> elp		
\$	¢ [l 🍱 1 1 🚈 📼 🗈	
	· · · ·	· · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
			Add Pin	×
•	· · ·	Pin Names		
•	· · ·	Direction Usage	input 🔽 Bus Expansion 💿 off schematic 🔽 Placement 💿 sir	f 🔾 on ingle 🔾 multiple
		Attach Net Expres	ion: 🖲 No 🔾 Yes	· · · · ·
•	· · ·	Property Name		
•	· · ·	Default Net Name	0.0005 Font Style Stick	
•	· · · ·	A Rotate	▲ Sideways 🕞 Upside Down Sho	w Sensitivity >>
•	· · ·		Hide Cancel E	Defaults Help
•			gnd! yds=7.6u gnd! vth=419.3m vdsat=323m	
	· · ·			

17- To add pin, select add Pin Then write the pin name and select its direction

Then place the pin in your schmatic



Hint : vdd and ground pins are of direction "InputOutput"

18- To move to the layout level



This window will appear, do as shown then OK



19- defults are right, click OK

-	New File	×
_ File		
Library	test 🔽	
Cell	Inv	
View	layout	
Туре	layout 🧧	J
Application		
Open with	Layout L 🔹	
🔲 Always use th	nis application for this type of file	
Library path file		
/root/kits/TS	MC_65nm/cds.lib	
	OK Cancel He	lp)

20- If LSW window is empty as shown

	LSW	_	. 🗆	×	
Sort	Edit		Н	lelp	
ba	ckgrou	ınd	d	cw	
C	dsDef	Tech	Lib		
Layer	Obje	ct Gi	rid		
AV	NV	AS	<u> </u>	NS	
Ň					
bac	kgrou	nd	dr	w	1
edg	eLaye	r	dr	w	
				-	
				- P	1

Click tools=> Technology File Manager



This window will appear, click attach. Then choose your library from "Technology Library" bar.

1				Virtuo	so® I	.ayout Su	ite XL E	diting: test	Inv	layout
<u>L</u> aunch <u>F</u> ile <u>E</u> dit <u>V</u> iew	<u>C</u> reate Ver <u>i</u> fy Co <u>n</u> nectivity	<u>O</u> ptions	Atl	tach Tech	nolog	yy Library	to Des	ign Library	×	
11 🖻 🗉 🔊 🥐 -	÷ C 💷 🗙 🛃 (C	esign Libra	ry	test				5.0
C Technology To		0 » :	Techn	ology Librar	у	cdsDefTech	nLib 🔽			dY: E
File <u>H</u> elp	cādence				ОК	Cancel	Defaults	Apply	lelp	
Editor				· · · ·	· ·	· · · ·			· ·	
DEFT										
Manager										· · ·
New	Attach									
Load	Dump									
Save	Discard									· · · ·
⊂ Utilities			· · ·							· · ·
QCell	Edit Layers									· · ·
Graph										
11										
· · • · · · • • · ·										

LSW is OK :)

	LSW	-		×
Sort	Edit		He	elp
re	f		dr	a
	tsmo	:N65		_
Layer	Obje	ct Grid	3	
AV	NV	AS		IS
Ĭ				
ref			dra	
DNW	r		dra	
PW			dra	
NW			dra	
DD 🚺			dra	
PDI	FF		dra	
NDI	FF		dra	
tt OD	18		dra	
0D_	25		dra	
0 D	33		dra	
0D_	DECAP		dra	
PO			dra	
P 0			ru1	
VTH	_N		dra	
VTH	_P		dra	
VTL	N		dra	
VTL	Р		dra	
NT	N		dra	
NT	N		NCP	
PP			dra	
NP			dra	
Ter	0		la	17

21- In layout window, do as shown



This window will appear, verify that PR Boundry is unchecked

Generate Layout	×
Generate I/O Pins PR Boundary Floorplan	
Generate	1
 Instances Automatic Chaining Automatic Folding VO Pins Except Global Pins Except Pad Pins PR Boundary Snap Boundary 	
Device Correspondence	
Connectivity Extraction	
Extract Connectivity After Generation	
)
OK Cancel Defaults Help	5

In "IO Pins" tab select all pins then make all of it M1 pn =>> check "create label"==> options==> hieght=0.1, then click update, as shown below



22- To move freely in your layout window do as shown



This window will be appear, change the marked values to be smaller, change the snape mode to "anyAngle"

Display Optic	ons x
Display Controls	Grid Controls
🗹 Open to Stop Level 📃 Nets	Type 🔾 none 💽 dots 🔾 lines
🗹 Axes 📃 Access Edges	Minor Spacing
🔲 Instance Origins 🛛 🔲 Instance Pins	Million Opacing
🗹 EIP Surround 🛛 🗌 Array Icons	Major Spacing 5
🔲 Pin Names 🛛 🗹 Label Origins	X Snap Spacing 0.005
🔲 Dot Pins 📃 Use True BBox	Y Snap Spacing 0.005
🗹 Net Expressions 🛛 🗌 Cross Cursor	
🔲 Stretch Handles 🛛 🔲 Row Name	Filter
🔲 Via Shapes 🛛 🔲 Row Site	Size 6 Style empty 🔽
🗹 Dynamic Hilight 🛛 🗹 True Color Drag	Shon Madae
Transparent Group	Shap Modes
Maximum Number of Drag Figures 500	Create anyAngle
Instance Drawing Mode BBox	Edit anyAngle
Path Display Borders and Centerlines	Dimming Scope none
Show Name Of 🥥 instance 🧕 master 🔾 both	Automatic Dimming
Array Display Display Levels	True Color Selection only 📃
● Full Start 0	Dim Intensity:
🔾 Border	50
⊖ Source Stop 32	
🧕 Cellview 🔾 Library 🔾 Tech Library 🔾 File	~/. cdsenv Browse
Save To Load From	Delete From
ОК	Cancel Defaults Apply Help

This figure shows transistors of your schematics, Begin to connect as your schematic



This is the final layout of inverter cell



23- To run DRC



This window will appear,

In rules section browes your calibre.drc file

In input section check "Export From Layout Viewer"

Then run DRC

<u>F</u> ile <u>T</u> ranscript	<u>S</u> etup		Helb
Rules	Bun:	DRC (Hierarchical) 📃	 ☐ Incremental
Inputs			
Outputs	Layout	Waivers	×
Run <u>C</u> ontrol	File:	INV1 STAGE.calibre.db	
Tr <u>a</u> nscript]		
[– Format:	GDSII 🛁	Export from layout viewer
Run <u>D</u> RC	T 0	UNH OTA OF	
	Top Cell:	INVISTAGE	
Start R <u>V</u> E	Area:		
	Library:	NEW_layout	View: layout

This window shows DRC errors



24- To run LVS

Virtuoso®	Layout Suite L Ed	iting: N
ools <u>W</u> indow	Calibre <u>H</u> elp	
) <u>12</u> SL (Run DRC Run DFM	» ∥Wo
lect:0 Sel(N):	Run LVS	X:6.2550
	Run PERC	
	Run PEX	
	Start RVE	
	Clear Highlights	V.
	Setup 🕨 🕨	
	About	

Do as DRC, then click "Run LVS"

	Calibre Interactive - nmLVS v2011.2_34.26 : LVS65	_ O X
<u>F</u> ile <u>T</u> ranscript <u>S</u>	<u>j</u> etup	<u>H</u> elp
Rules	♦ Hierarchical 🔷 Flat 🔷 Calibre CB	
Inputs	♦ Layout vs Netlist 🧠 Netlist vs Netlist 🧠 Netlist Extraction	
<u>O</u> utputs Run <u>C</u> ontrol	Layout Netlist H-Cells Signatures Waivers	
Tr <u>a</u> nscript	File: INVISTAGE.calibre.db	
Run <u>L</u> VS	Format: GDSII -	Export from layout viewer
Start R <u>V</u> E	Top Cell: INVISTAGE	
	Library: NEW_layout View: layout	
	Layout Netlist: INVI STAGE.sp	

This window will appear.

If it's smily Face then your LVS is right, else it's not.

≁Navigator d"×	😃 Comparison Res	ults ×					
Results	🔄 🖾 Layout Cell / T	уре	Source Cell	Nets	Instances	Ports	
か。Extraction Results	INV1STAGE	•	INV1STAGE	4L, 4S	1L, 1S	4L, 4S	
Comparison Results							
ERC							
Secure							
ERC Summary							
Reports							
Rules File							_
E Extraction Report							M
LVS Report	Cell INV1STAGE	Summary (Cle	an)				
View		CELL CO	PARISON RESULTS (TO	> LEVEL)			<u>^</u>
A Finder		#	*******	*####			
Schematics			#	# * *			
Setun		# #	# CORRECT #	# \/			
Ontions		#	******	****			
wopuons							
	LAYOUT CELL NAM SOURCE CELL NAM	E: : E: :	INVISTAGE INVISTAGE				
	INITIAL NUMBERS	OF OBJECTS					- 1
		Layout :	Source Compon	ent Type			- 1
	Ports:	4	4				- 1
	Nets:	4	4				- 1
	Instances:	1	1 MN (4)	oins)			

25- To run PEX



Do as DRC, then click "Run PEX"

<u>R</u> ules	Layout Netlist H-Cells Blocks Probes
Inputs	File:
<u>O</u> utputs	
PEX Options	Format: GDSII - Export from layout viewer
Run <u>C</u> ontrol	
Tr <u>a</u> nscript	
	Library: NEW_layout View: layout
Run <u>P</u> EX	
Start R <u>V</u> E	

This window will appear, do as shown

	Calibre View Setup	×
Window Menu	NEW_layout	Â
Schematic Library:	NEW_layout	
Cellmap File:	*cell Path*/calview.cellmap	
	View Edit	
Log File:	*cell path#/calview.log	
Calibre View Name:	calibre	
Calibre View Type:	🔾 maskLayout 🥑 schematic	
Create Terminals:	$ullet$ if matching terminal exists on symbol \bigcirc Create all terminals	
Preserve Device Case		
Execute Callbacks		
Reset Properties:	m=1	
Magnify Instances By:	1	
Device Placement:	🔾 Layout Location 🧕 Arrayed	
Parasitic Placement:	Layout Location Arrayed	
Show Parasitic Polygons		
Open Calibre CellView:	🔾 Read-mode 💿 Edit-mode 🔾 Don't Open	L
Always Show Dialog	✓ OK Cancel Help	2

Click "Auto MAp Pins" then OK

	Map Cal	ibre Device	e x
Device:	nch	Library:	tsmcN65
Pins:	bdgs	Cell:	nch
Pin Map:	b=B d=D g=G s=S	View:	symbol Browse
		Terminals:	B D G S
	ОК	Cancel	Auto Map Pins Help

For cap & Res Browes analogLib then select them => Auto Map Pins => OK

	Map Cal	ibre Device	e x
Window	Menu	Library:	analogLib
Pins:	pin1 pin2	Cell:	res
Pin Map:	pin1=MINUS	View:	symbol
1	pinz=pios		Browse
		Terminals:	MINUS PLUS
	ОК	Cancel	Auto Map Pins Help

0 warnings and 0 errors :) => OK

Click "check and save"

•	•	• •	٠	٠	• •	•	٠	• •	•	•	٠	•		٠	٠	•		•	٠	٠	٠	•			٠	• •	•	٠	٠	• •	•	٠	٠	• •	٠	• •	٠	•	•		•	•	٠			•	• •
÷		• •			• •			•		-											Ca	alii	hr	e l	nf	6										×			•	• •				• •	•		
1		• •			•			•	-												~~		× 1	· ·		۳.													•	• •				• •	•		
•		• •			• •			•																															•	• •				• •	•		
		• •			• •			•		N	8	C	alik	ire	Vi	ew	ge	ne	rati	ion	10	om	ple	eted	W	'ith ∣	ο γ	/Ał	RN	IN	Gs	ani	10	ER	RC	Rs.			•					• •	•		
		• •			• •			•	~	/ 1	7	PI	ea	e 0	cor	nsi	ilt t	he	CI	w	tra	insi	eri:	nt fo	hr r	mes	e a	nec											•					• •	• •		
						•				~	1		ca.		001	100		ne	~	···		anor	Cinq	pric	~ `	nes	000	900	· ·																		
							•																_																								
																							Clo	ise																							
																						-	_	_	٢.																						
1		•••			•			•		_	_	_	_	_	_	_	_	_	_	_	_		_		_	_																		• •			•••
1	•	•••	•	•	•	•	•	• •	•	•	•	•	•••	•	•	•		•	•	•	•	•••		••	1	• •	•	•	•	• •	•	•	•	• •	•	• •	•	•		•••	•	*	•	• •	•	•	• •
		• •			•			•									• •					1	• •	••	*	• •				• •				• •					•	• •				• •	+		
•		• •			• •			• •									• •					•	• •	•	1	• •				• •				• •					•	• •				• •	•		
		• •			•			• •									• •					• ! ·	• •	÷ •	4	• •				• •				• •					•	• •				• •	•		• •
		• •			• •			• •									• •					• • •	• •	÷ •	4	• •				• •				• •					•	• •				• •	•		• •
		• •			• •			• •														- Č-								• •				• •											•		• •
		• •			• •			• •																						• •									•								
				•																					1				•				•														
																									1																						
						- '				1				- 1	- L						1						-				- •										- 1					- ·	
1	•	• •	•	•	•	•	•	•	•	•	*	•	* *	•	•	*		•	•	•	•			••	1	• •	•	•	•	•	•	•	•	• •	•	• •	•	•		• •	•	+	•	• •	*	•	• •
1		•••			•	•		• •	•••				•••			•		•					•	•••	1	• •	•			• •	•		•	•••		•••			•	• •				• •	•		• •

26- To simulate your design with PEX

From ADE choose setup=> Environment

🦉 🛛 Virtuoso® Analog Design	Environment (1) - NEW_layout test schematic _ 🗆 🗙
S <u>e</u> ssion Set <u>up A</u> nalyses <u>V</u> ariables <u>O</u>	utputs <u>S</u> imulation <u>R</u> esults <u>T</u> ools <u>H</u> elp cadence
Design V Nan Nan Nan Model Libraries Temperature Stimuli Simulation Eiles <u>M</u> ATLAB/Simulink Environment	Analyses Type - Enable Arguments
mouse L: 2(3) Environment	Plot after simulation: Auto Plotting mode: Replace M: Status: Ready T=27 C Simulator; spectre

Write "calibre" after ".sch" then OK

•	Environment Options	×
Switch View List	spectre cmos_sch cmos.sch calibre schematic veriloga	
Stop View List	spectre	
Parameter Range Checking File		
Print Comments		
userCmdLineOption		
Automatic output log	✓	
savestate(ss):		
recover(rec):	□ Y □ N	
Run with 64 hit binary		
	OK Cancel Defaults Apply Hel	lp)

Then choose simulatin type and run :)

A4.Jitter Calculation

To calculate RMS jitter and peak to peak jitter in cadence you have to do the following steps

1.Run transient simulation to know the definite accurate frequency

2.Choose pss analysis

For beat frequency write the frequency value you got in transient analysis

Choose 10 for number of harmonics

From transient analysis you knew when the system stabilize write this time in Additional time for stabilization

🔲 Choosir	ng Analys	es Vir	tuoso® A	nalog Design En	×		
Analysis	🔾 tran	🔾 dc	🔾 ac	🔾 noise			
	🔾 xf	🔾 sens	🔾 dcmatch	🔾 stb			
	🔾 pz	🔾 sp	🔾 envlp	🖲 pss			
	🔾 pac	🔾 pstb	🔾 pnoise	🔾 pxf			
	🔾 psp	🔾 qpss	🔾 qpac	🔾 qpnoise			
	🔾 qpxf	🔾 qpsp	🔾 hb	🔾 hbac			
	🔾 hbnoise				-		
	Periodi	ic Steady	State Analysi	s			
Engine	🖲 Shoot	ing 🔾 Ha	armonic Balar	nce			
Fundamer	tal Tones				≣		
# Name	Expr	Value	s Signa	l SrcId			
	-						
			Large				
			(3-				
Clear/A	dd Delet	te	Update From	Hierarchy			
🖲 Beat F	requency						
🔾 Beat P	eriod	252.215	/151M	Auto Calculate 🔄			
Output harmonics							
Number of harmonics 🔽 10							
Accuracy	Defaults (errj	oreset)					
🔲 conse	rvative 📃 n	noderate 🛛	🗹 liberal				
Additional T	ime for Stabi	lization (ts	tab) 10n	L			
		ок Са	ancel Defa	aults Apply Hel	p		

3.In case of oscillator

Check the oscillator box then choose node of the output of the oscillation for oscillator node and select ground node for the reference node.

Then press ok

Oscillator V Oscillator node /net4 Select Reference node /gnd! Select Osc initial condition default linear	Accuracy Defaults (errpreset) □ conservative □ moderate ☑ liberal Additional Time for Stabilization (tstab) 10n Save Initial Transient Results (saveinit) □ no □ yes					
Sweep	Oscillator 🗹	Oscillator node /net4 Select Reference node /grd! Select Osc initial condition default linear				
Enabled Options	Sweep Enabled 🖌	Options				

4. Now Choose phoise analysis

Put the parameters as shown below and choose node of the output of the oscillation for positive output node and select ground node for the negative output node.

🔲 Choosin	g Analys	es Viı	tuoso®	Analog De	esign En	×
Analysis	🔾 tran	🔾 dc	🔾 ac	🔾 noise		
	⊖ ×ſ	🔾 sens	🔾 dcmate	:h 🔾 stb		
	🔾 pz	🔾 sp	🔾 envlp	🔾 pss		
	🔾 pac	🔾 pstb	💌 pnoise	🔾 p×f		
	🔾 psp	🔾 qpss	🔾 qpac	🔾 qpnois	e	
	🔾 qp×f	🔾 qpsp	🔾 hb	🔾 hbac		
	🔾 hbnoise					
	Per	iodic Nois	e Analysis			
PSS Beat Fre	quency (Hz)	252.2	157151M			
Sweeptype	default	P R	elative Hari	monic 1		
Output Fre	quency Swe	ep Range	(Hz)			
Start Stan				-		
Start-Stop		stant 1		Stop 16		
Sweep Typ)e					
Automatic						
Add Specific	e Points 🔲					
Sidebands				_		
Maximum si	deband 🧧	10				
When using	g shooting ei	ngine, def	ault value is	\$ 7.		
Contract						
Output	Positive	e Output N	lode /ne	et4	Select	
voltage	Negativ	A Output	Node Zoo	lbe	Select	
	riegani	e Output	Vgr		Select	
Input Sourc	ce					$\overline{\nabla}$
		DK Ca	ancel De	efaults Ap	oply Help	p)

5.choose none for input source , and jitter fot Noise Type

Then press ok

Output voltage	Positive Output Node Negative Output Node	/net4 /gnd!	Select Select
Input Source			
Noise Type jitt jitter: jitter me	ter 💌		
FM jitter f	or autonomous circui	t	
Enabled 🗹			Options
	Cancel	Defaults	Apply Help

6.After running the simulation

From ADE results > direct plot > main form

Choose pss and the parameters as shown

Direct Plot Form ×						
Plotting Mode	Append					
Analysis						
🔾 tran	🖲 pss					
pnoise	🔾 pnois	e modulated				
🔾 pnoise jitter						
- Function						
💌 Voltage		Current				
Power		🔾 Voltage Gain				
🔾 Current Gain		🔾 Power Gain				
 Transconductance 		Transimpedance				
Compression Point		IPN Curves				
Power Contours		Reflection Contours				
Given Harmonic Frequency		Over Added Eff.				
Power Gain Vs Pout		🔾 Comp. Vs Pout				
🔾 Node Complex Imp.		○ THD				
Select Net		•				
Sweep						
🔾 spectrum 🧕	time					
Add To Outputs	⊻	Replot				
> Select Net on	schemati	o				
		OK Cancel He	lp)			

7. From ADE : results > Direct plot > main form choose pnoise jitter for analysis and the parameters as shown

To calculate RMS jitter choose rms for Signal level

If peak to peak choose peak to peak for Signal level

Check Add to outputs box select the output from the schematic and press on plot

	Direct Plot Form ×						
Plotting Mode Analysis	Append						
🔾 tran	O pss						
🔾 pnoise	pnoise modulated						
💿 pnoise jitter							
Function							
O Phase Noise	○ -20dB/dec Line						
⊖ Jc	● Jcc						
Number of Cycles	Number of Cycles [k] 1 Signal Level • rms • peak-to-peak						
Modifier							
🖲 Second 🔾 L	JI 🔾 ppm						
Freq. Multiplier	1						
Integration Limits	3						
Start Frequency ((Hz) 1						
Stop Frequency ((Hz) IG						
Add To Outputs	Plot						
> Press plot butt	on on this form						
	OK Cancel Help						

🐮 Virtuo	oso® Analog Desi	ign Enviro	onment	(3) - (GP try0 schem	atic		- 0	x
S <u>e</u> ssion Set <u>u</u> p <u>A</u> na	alyses <u>V</u> ariables <u>O</u> u	ıtputs <u>S</u> imul	ation <u>R</u> e	sults <u>T</u>	ools <u>H</u> elp		cā	d e n c	e
Design Variables Name -	Value	Analyses Type - 1 tran 2 pss 3 pnoise	Enable V (V 2 V 2	0 20n 252.2M 1 10 1 1G .	Arguments 10 /net4 /gnd! /net4 /gnd!		?		
		Outputs Outputs Name 1 net4	/Signal/E	kpr 🔻	Value	Plot	Save		■ -
		2 v /net4; p: 3 Jcc[Secol 4 Jcc[Secol	ss (V) nd][k=1]@ nd][k=1]@	(1,1G (1,1G	wave 1.035p 6.396p	Y Y Y			M
		Plot after si	mulation:	Auto	Plotting mo	ode: Rep	lace		
mouse L:			M:	1	Chattan Danada Liz		Loinut		R:
ZZ(35) Setup Outputs	3				Status: Ready T	=27 C	Simula	ator: spec	ctre

8. Now you will have the results

A5.Power Calculation

1-Using the SPECTRE simulator, set up a DC analysis ensure you check: Save DC Operating Point (DO NOT SETUP A SWEEP VARIABLE):



2-In addition, setup a transient analysis of length=10us

Analysis	🖲 tran) dc	🛈 ac	💛 noise			
	⊖ xf	🔵 sens) dcmatch	💛 stb			
	🛈 pz	🔵 sp	🔾 envlp	💛 pss			
	🔵 pac	🔵 pstb	🔾 pnoise	💛 pxf			
	i) psp	i qpss	🔾 qpac	💛 qpnoise			
	i) qpxf	i) qpsp					
Transient Analysis							
Stop Time	100						

3-From the simulator menu, choose: Ouptuts->Save All...

o Check off "Save pwr" and "Save currents" as shown here:

🔀 Save Options	×
Select signals to output (save)	🗆 none 🔲 selected 🛄 t vip ub 🔲 ivi 👱 alipub 🛄 ali
Select power signals to output (pwr)	🗆 none 💷 total 🖃 devices 🔚 subckts 👱 all
Set level of subcircuit to output (nestivi)	
Select device currents (currents)	🔄 selected 📃 nonlinear ⊻ all
Set subcircuit probe level (subcktprobelvl)	

4-Run the simulation, ensure there are no errors in the CIW window

o There will not be anything plotted during the simulation

5-When the simulation is complete, from the simulator menu choose: Tools->Calculator

o When the calculator pops up, click on the "vt" button

1 Virtuoso (R) Visualization & Analysis L Calculator	- 🗆 ×
Eile Tools View Options Constants Help	cādence
Results Dir: /home/grad/tfarmer/cadence/simulation/inv2_tb_dynamic_power_dissipation/spectre/schematic/ps vt O vf O vdc O vs O op O var O vn O sp O vswr O hp O zm it O if O idc O is O opt O mp O vn2 O zp O yp O gd O data	sf

6-You will be returned to the schematic, click on the blue wire connecting VDD to

its VDC source

o The calculator will be populated with: VT("/vdd!")

Meaning 'transient voltage of net vdd!

o Next, click on the "it" button



7- You will be returned to the schematic, click on the top red terminal connecting of

the VDC connected to VDD

- o The calculator will be populated with: IT("/V1/PLUS")
 - Meaning 'transient current of the instance V1 (note your source may have a difference instance name, like V2, V3, etc)
- o From the calculator pad, click on the * symbol to multiply these two signal together



8-Under the "Special Functions" category, click on the "average" function

Special Funct	tions 🗸							7897
average bandwidth clin	compressionVRI convolve cross	deriv dft dfthh	evmQAM evmQpsk eveDiagram	freq freq_jitter frequency	getAsciiWave groupDelay harmonic	iinteg integ intersect	lshift overshoof neak	123-

9-You have now built the expression shown in the calculator below. Press the

evaluate button to evaluate the expression.

⊖vt ⊖vf ⊖vdc ⊖vs op ⊖var ⊖vn ⊖sp ⊖vswr ⊖hp ⊖zm ●it ⊖if ⊖idc ⊖is ⊖opt ⊖mp ⊖vn2 ⊖zp ⊖yp ⊖gd ⊖data					
⊖ Off ⊖ Family ⊖ Wave 🔽 Clip 🦏 🐗 Append 🔽 📄					
average(VT("/vdd!")*IT("/V1/PLUS"))					

10-The total power should be calculated .

A6.Area calculation

Measure it (with the ruler, or just by reading coordinates off the banner) and multiply the width by the height.



A7.Design Compiler

1- Opening the program

E eslam@localhost:~	_ □	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>S</u> earch <u>T</u> erminal <u>H</u> elp		_
[eslam@localhost ~]\$ lmli2		^
∑ root@localhost:~	_ 0	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>S</u> earch <u>T</u> erminal <u>H</u> elp		
[eslam@localhost ~]\$ lmli2		^
[eslam@localhost ~]\$ su - Password:		
<pre>[root@localhost ~]# cd *your destination folder*</pre>		
🕫 root@localhost:/home/eslam/Desktop/saraaa	_ 0	×
<u>File Edit View Search Terminal Help</u>		
[root@localhost saraaa]# dc_shell		^
DC Professional (TM)		
DC Expert (TM)		
DC Ultra (TM) FloorPlan Manager (TM)		
HDL Compiler (TM)		
VHDL Compiler (TM)		
DesignWare Developer (TM)		
DFT Compiler (TM)		
Power Compiler (TM)		
Versier B 2000 00 fer linut Aug 25 2000		
Copyright (c) 1988-2008 by Synopsys, Inc.		
ALL RIGHTS RESERVED		Ξ
This software and the associated documentation are confidential and		
proprietary to Synopsys, Inc. Your use or disclosure of this software		
between you, or your company, and Synopsys, Inc.		
The above trademark notice does not imply that you are licensed to use		
all of the listed products. You are licensed to use only those products		
for which you have lawfully obtained a valid license key.		
Initializing		
uc_snett> start_gut		¥)

This is its GUI interface

2											D	esign \	Visio	n - Top	Level.	L	×
Fil	e <u>E</u>	dit <u>V</u> iew	<u>S</u> elect	<u>H</u> ighlig	ght L <u>i</u> st	<u>H</u> ierar	rchy <u>E</u>	<u>D</u> esign	Attribute	s S <u>c</u> hemat	tic]	<u>T</u> iming	<u>T</u> est	Power	Windo	w He	/ Help
	2 6	86] @ 2	Q (1)	0.0	≛ ≜			53 BH 🗍 I				_		•	0 🕘 🤇	۲. ا
		Hier.1 ogical Hie	Cells (Cell N	Hierarch	nical)	me	Cell F	Path									
		dc_she dc_she Log F dc_shell	ll> sta ll>	rt_gui						_							Options: v
Rea	ady																

2- file => setup



This window will appear

	Application Setup x
Categories	Defaults
<mark>Defaults</mark> Variables	Search path: psys/B-2008.09/libraries/syn /usr/synopsys/B-2008.09/dw/syn_ver /usr/synopsys/B-2008.09/dw/sim_ver
	Physical library:
	Link library: * * your_library.db
	Target library: * your_library.db
	Symbol library: * your_library.sdb
	Synthetic library:
	* = required
	Reset • OK Cancel Apply

3- Add then wc,bc, and tc in Link Library

And the wc library in Trget Library

And sdb in symbol library

Select Files ×
Look <u>i</u> n: 🔄 /home/eslam/Desktop/encounter/tt/ 💽 🗢 🗈 💣 🏢
uk65lscllmvbbr_132c0_bc.lib
uk65lscllmvbbr_090c125_wc.db 🗋 uk65lscllmvbbr.sdb
uk65lscllmvbbr_090c125_wc.lib
uk65lscllmvbbr_120c25_tc.db
uk65lscllmvbbr_120c25_tc.lib
uk65lscllmvbbr_132c0_bc.db
File <u>n</u> ame: hvbbr_120c25_tc.db" "uk65lscllmvbbr_132c0_bc.db" Open
File type: All files (*)

As shown, then OK

	Application Setup ×
Categories	Defaults
- Defaults	Search path: psys/B-2008.09/libraries/syn /usr/synopsys/B-2008.09/dw/syn_ver /usr/synopsys/B-2008.09/dw/sim_ver
Vullubics	Physical library:
	Link library: * unter/tt/uk65lscllmvbbr_120c25_tc.db /home/eslam/Desktop/encounter/tt/uk65lscllmvbbr_132c0_bc.db
	Target library: * /home/eslam/Desktop/encounter/tt/uk65lscllmvbbr_090c125_wc.db
	Symbol library: * /home/eslam/Desktop/encounter/tt/uk65lscllmvbbr.sdb
	Synthetic library:
	* = required
	<u>R</u> eset v OK Cancel <u>Apply</u>

4- file => Analyze

4	_								
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>S</u> elect	<u>H</u> ighlight	L <u>i</u> st				
🗳 <u>B</u>	ead				1				
F	lemo <u>v</u>	e All D	esigns						
A	nal <u>y</u> z	e							
E	labora	ate			aı				
Setup									
Ŀ	ink De	esign							
Ŀ	mport				+				
C	reate	MW <u>L</u> i	brary						
S	et MV	/ Libra	ry <u>R</u> efere	ence					
C) <u>p</u> en N	1W Lib	rary						
C	lose l	∕IW Li <u>b</u>	rary						
C	Cop <u>y</u> №	1W Lib	rary						
S	et TLL	J <u>+</u>							
	iave			Ctrl+	S				

5- select your files

Analyze Designs ×
Look in: 🔄 /home/eslam/Desktop/dc_enc/ 💽 🗢 🗈 💣 🏬 🎬
PD_cap.vhd coarse.vhd SR_16bit.vhd crs_counter.vhd SR_9bit.vhd D_FF.vhd total.vhd MUX2x1.vhd tri.vhd new_fine.vhd tri.vhd
File <u>n</u> ame: d" "SR_16bit.vhd" "SR_9bit.vhd" "total.vhd" "tri.vhd"
File type: Database Files (*.v *.vhd *.sv *.vhdl) Cancel

6- file => Elaborate



7- select "WORK" Library

	Elaborate D	esigns	×
Library:	WORK		•
Design:	DEFAULT		
<u>D</u> esign.	GTECH		
Parameters:	WORK		
<u>R</u> eanalyze	out-of-date libraries		
		ОК	Cancel

8- choose your top Entity Design

	Elabora	ate Designs X
<u>L</u> ibrary:	WORK	•
<u>D</u> esign:	TOTAL(BEHAVI	IORAL)
<u>P</u> arameters:	Name	Value
□ <u>R</u> eanalyze	out-of-date lib	raries
		OK Cancel
now 'coarse	e_1'.	

9- Attiributes => operating environment => operating conditions

			Des	ign visi	on -	юргем	el.1 (tota	II)	
: L <u>i</u> st <u>H</u> iera	rchy <u>D</u> esign	<u>A</u> ttributes	S <u>c</u> hematic	<u>T</u> iming	<u>T</u> est	<u>P</u> ower	<u>W</u> indow	Help	
) 🖸 🛛 🕹 📤	D 🖸 🔤	<u>S</u> pecify	Clock			-			
	_	<u>O</u> perati	ng Environm	ent 🕨	Ir	put Dela	ау		
al)		Optimiz	ation <u>C</u> onstr	raints 🕨	<u>o</u>	utput De	elay		
Ref Name	Cell Path	Optimiz	ation <u>D</u> irect	ives 🕨	D	rive Stre	ength		
FD	M1	u			L	oad			
oarse	M2	u			С	haracter	ize		
ew_fine	МЗ	u			0	perating	Conditio	ns	
R_9bit	M4	u			10	lire Leas			
R_16bit	M5	u			<u></u>	nie Load	1		
ristate	M6	u			<u> </u>	iming Ra	ange		

Then OK

Dperating	g Conditions ×
Current design: total	
Analysis condition	
© <u>S</u> ingle	C <u>M</u> in/max case
Maximum operating condition	Minimum operating condition
Library: uk65lscllmvbbr_090c125_wc	Library: uk65lscllmvbbr_090c125_wc
Condition: uk65lscllmvbbr_090c125_wc	Condition: uk65lscllmvbbr_090c125_wc
	OK Cancel <u>Apply</u>

10- write the following commands to generate the netlist, sdf , and sdc files

compile_ultra

write -format verilog -hierarchy -output layout.v

write_sdf ADPLL.sdf

write_sdc -nosplit ADPLL.sdc

-	1	_
	dc_shell> write_sdf ADPLL.sdf	
	Information: Annotated 'cell' delays are assumed to include load delay. <u>(UID-282)</u>	
	Information: Writing timing information to file '/home/eslam/Desktop/dc_enc/ADPLL.sdf'. (WT-3)	
	Information: Timing loop detected. (OPT-150)	
	U122/A U122/Z M1/Mb/temp_reg/RB M1/Mb/temp_reg/Q	<u> </u>
	4	Þ
	Log History	Options: 💌
	compile_ultra -no_boundary_optimization	
	dc chells write -format verilog -hierarchy -output layout.v	
	write_sdf ADPLL.sdf	
F	write_sdc -nosplit ADFLL.sdc	•
с Г. с		