Improvement to Intel's analog Design Verification Flow for 22nm and below Design

Major Project
Submitted in Partial Fulfilment of the Requirements
for the Degree of
Master of Technology(M.Tech.)
in
VLSI-Design
by

PATEL VIVEK M.
12MECV21

Department of Electronics & Communication Engineering
Institute of Technology
Nirma University
Ahmedabad
May-2014

Improvement to Intel's analog Design Verification Flow for 22nm and below Design

Major Project

Submitted in Partial Fulfilment of the Requirements
for the Degree of
Master of Technology(M.Tech.)
in
VLSI-Design
by

PATEL VIVEK M.
12MECV21

Dr. N. M. Devashrayee
Internal Guide

Mrs. Mugdha D. Alurkar
External Guide

Department of Electronics & Communication Engineering
Institute Of Technology
Nirma University
Ahmedabad
May-2014

Declaration

This is to certify that

1. I, Vivek Patel, a student of Master of Technology in VLSI-Design, Nirma University, Ahmedabad hereby declare that the project work "Improvement to Intel's analog Design Verification Flow for 22nm and below Design "has been independently carried out by me under the guidance of Mrs. Mugdha D, Alurkar, Engineering Manager, Intel Technology India Private Limited, Bangalore and Dr. N. M. Devashrayee, Program Co-ordinator, Department of VLSI-Design, Nirma University, Ahmedabad. This Project has been submitted in the partial fulfillment of the requirements for the award of degree Master of Technology(M.Tech.) in VLSI-Design, Nirma University, Ahmedabad during the year 2013 - 2014.

2. I have not submitted this work in full or part to any other University or Institution for the award of any other degree.

Vivek Patel
12MECV21

Certificate

This is to certify that the Major Project entitled “Improvement to Intel’s analog Design Verification Flow for 22nm and below Design ” submitted by Vivek Patel (12MECV21), towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI-Design of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Dr. N. M. Devashrayee
Internal Project Guide,
Institute of Technology,
Nirma University, Ahmedabad

Dr. D.K.Kothari,
Section Head of EC Dept
Institute of Technology,
Nirma University,
Ahmedabad

Mrs. Mugdha D. Alurkar
External Project Guide
Intel Technology India Pvt. Ltd.,
Bangalore

Prof. P N Tekwani
Head of EE Dept
Institute of Technology,
Nirma University,
Ahmedabad

Date:

Place:Ahmedabad

Acknowledgement

With immense pleasure, I would like to present this report on the dissertation work related to "Modeling and Budgeting of Custom Boundary timing/noise". I am very thankful to all those who helped me for the successful completion of the first phase of the dissertation and for providing valuable guidance throughout the project work.

I would like to express my deepest thanks and gratitude to Mrs. Alurkar Mugdha, Mr. Girish Agarwal, Satyanarayan Shwetha, Mitharan Anupama, Patel Jaydeep and Bathini Aruna for providing constant support and guidance required to understand the project work and the platform required to carry out the project work.

I would also like to thank Dr. K.R.Kotecha, Director, Institute of Technology, Nirma University, Ahmedabad for providing me an opportunity to get an internship at Intel Technology India Private Limited, Bangalore.

I would also Thank to my internal guide and program co-ordinator, Dr. N. M. Devashrayee, Professor, Embedded Systems, Institute of Technology, Nirma University, Ahmedabad for giving valuable support for project work.

I would like to thank my all faculty members for providing encouragement, ex-changing knowledge during my post-graduate program.

I also owe my colleagues in the Intel, special thanks for helping me on this path and for making project at Intel more enjoyable.

Vivek Patel
12MECV21

Abstract

Design of Low Power Analog Integrated Circuits on leading edge process technologies poses a significant challenge due to a number of issues like EOS, Gate Oxide Breakdown due to excess electric field on oxide of MOS gate, Aging stress, Breakdown of Inter Layer Dielectric (ILD) or Bias Temperature Stress (BTS), and Signal Integrity to measure the quality of signal. Conventional circuit simulation using SPICE is not good enough as it has an Insufficient Coverage. Extent of analysis of a circuit simulation is only as good as the input stimulus. If stimuli are not comprehensive many pitfalls in the circuit cannot be identified. The effort is also large as circuit simulation is expensive from compute time and human effort standpoint. Due to this we often need non-conventional methods such as Static Verification. It also fails to identify the Circuit Issues Early due to its requirement of human effort to setup the simulation (writing test benches). Ideally a designer would like to identify any fatal issues in circuit as soon as schematic editing is done rather than identifying them by setting up and running circuit simulation. For long operation it is necessary to include Aging affects which Conventional SPICE simulation cannot simulate on circuit due to Hot Carries and PBTI. With the increasing frequency, signal suffers with high frequency effect due to ringing, reflection, cross talk, ground bounce. It generates the jitter in the signal. Conventional signal integrity tool is not efficient and not correlate with the actual Jitter value. Due to the above reasons, design flows include 3 critical verification steps of Static Electrical Rule Checking - that is verification of circuit for problematic / risky patterns without actually applying stimuli , special Aging Simulation flow to analyze circuit degradation and signal integrity analysis to analyze jitter number. Current static verification flow at Intel has some disadvantages due to long runtime, generation of unmanageable false violations. The focus of the internship project is to improve the Intel's static verification flow efficiency. Also, there are similar inefficiencies in aging verification flows as well as signal integrity verification flow. The internship work overall involves analysis of Intel's requirements for Aging, EOS, Signal integrity and GOX robustness of circuit, developing methods to identify and mitigate them and developing and performing quality assurance of an efficient design flow.

Contents

List of Figures

Chapter 1

Introduction

At Nano scale technology, reliability becomes a serious concern. Even Though the chips achieve great performance but their operation becomes more unreliable. The whole concept covers many chip reliability issues like leakage, time dependent degradation, temperature variation, manufacturing imperfection.

There is a number of challenges like EOS, AGING, BTS, signal integrity and Process Variation when we work on the nano technology. The process variations change the attributes of transistors Length, widths, oxide thickness when the circuits are fabricated. So that decisions based on the nominal may not be correct because the models are either overestimates or underestimations of actual value. Hence, the resultant circuits may not be optimal.

EOS or Electrical overstress refers to the thermal damage that may occur when an electronics device is subjected to the current and voltage that is the beyond the limits of the device. So it generates the excessive heat during the EOS event and very high temperature even the path on low resistance path. It can damage devices. So we cannot estimates the actual behavior of circuits because EOS may have damaged the device and output is unpredictable.

During the circuit operation, there is a degradation in transistor performance accompanied with changes in threshold voltage often referred to as aging. With increasing time, effect of aging becomes worse.

So there exists a gap between the pre-silicon and post-silicon analysis which need to be bridged to predict the exact behavior of the device at a very early phase (i.e. in pre-silicon analysis) and also identify the bug or error. So it becomes customary to take necessary action to remove those errors or bugs and improve the pre-silicon flow.

1.1 PVT Variation:

1.1.1 Process Variation:

Process variations accounts for deviations in the semiconductor fabrication process.Usually process variation is treated as a percentage variation in the performance calculation. Variations in the process parameters can be impurity concentration densities, oxide thicknesses and diffusion depths. These are caused by non-uniform conditions during depositions and/or during diffusions of the impurities. This introduces variations in the sheet resistance and transistor parameters such as threshold voltage. Variations are in the dimensions of the devices, mainly resulting from the limited resolution of the photo lithographic process. This causes (W/L) variations in MOS transistors. Process variations are due to variations in the manufacture conditions such as temperature, pressure and dopant concentrations [1].

- Random Variation
  1. Occurs without regards to the location and patterns of the transistors within the chip.
  2. E.g. Random Doping Fluctuation (RDF) in MOSFET. Variation in transistor threshold voltage caused by density variations of impurities in the transistor material.
- Systematic variation
  1. Related to the location and patterns.
  2. E.g. Exposure pattern variation that occurred in lithography process.
- Intra-die or Within-die variation
  1. Variations between the elements in the same chip
- Inter-die or Die-to-die variation
  1. Variations between chips in the same wafer or in different wafers.

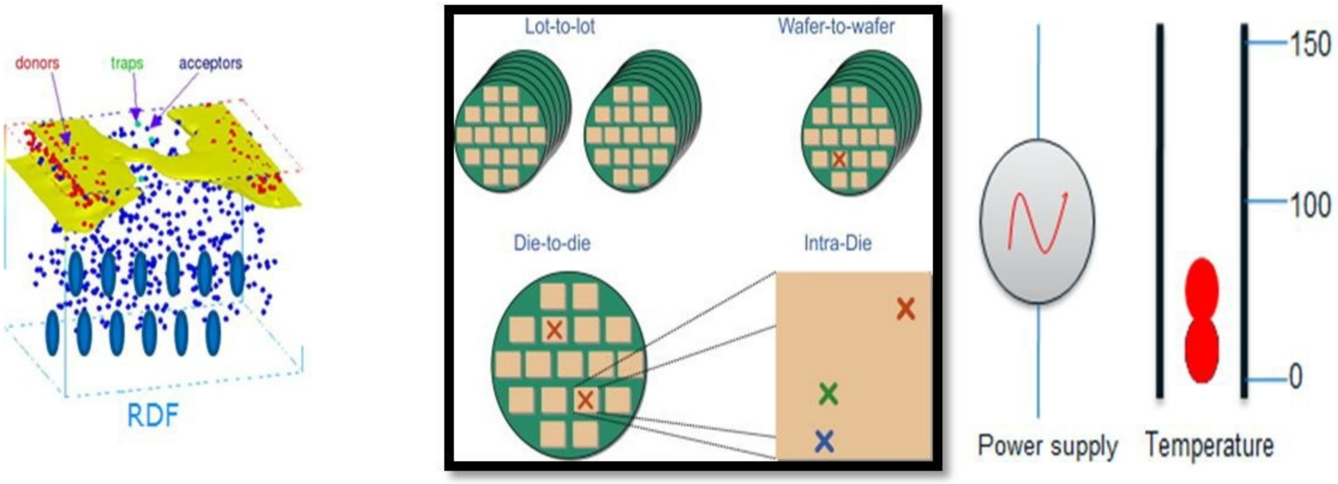


Figure 1.1: Parameter Variation [1]

1.1.2 Environmental Variable

- Supply Voltage Variation
  - The designs supply voltage can vary from the established ideal value during day-to-day operation. The saturation current of a cell depends on the power supply. The delay of a cell is dependent on the saturation current. In this

way, the power supply inflects the propagation delay of a cell. Throughout a chip, the power supply is not constant and hence the propagation delay varies in a chip.

- Operating Temperature Variation
  - Temperature variation is unavoidable in the everyday operation of a design. Effects on performance caused by temperature fluctuations are most often handled as linear scaling effects, but some submicron silicon processes require nonlinear calculations. When a chip is operating, the temperature can vary throughout the chip. This is due to the power dissipation in the MOS-transistors. The power consumption is mainly due to switching, short-circuit and leakage power consumption.

1.2 Process corner:

- Process corners represent the extremes of these parameter variations within which a circuit that has been etched onto the wafer must function correctly. A circuit running on devices fabricated at these process corners may run slower or faster than specified and at lower or higher temperatures and voltages.
- Transistors have uncertainty in parameters. Process depend on Leff, Vth, Tox of NMOS and PMOS. It will vary around typical (T) values.
- In this project we have used main corners TTTT, RSSS, cross corner like RFFF, RFSF, RFFS, RSSF, and RSFS. Here the line (tttt, rfff, rsss) is orthogonal because the ratio of NMOS and PMOS is constant.
- In RSFS, R = realistic, S = process (slow), F = NMOS (fast), S = PMOS (slow) same like other corners.
- All processes must work inside these corners. If any process lies outside this circle then they are bought back by concerned process people.

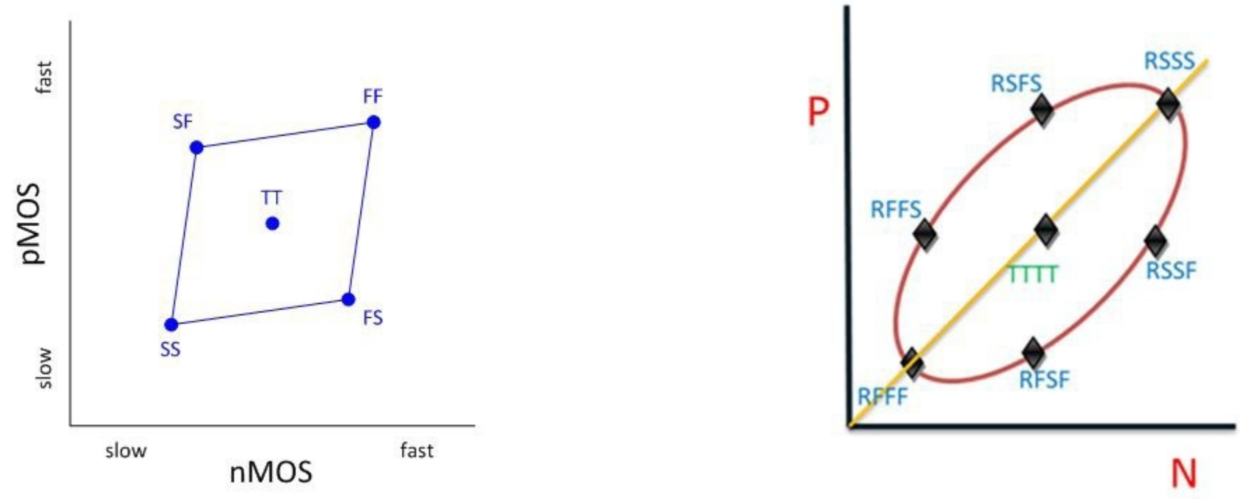


Figure 1.2: Process Corner [2]

1.3 Aging Variation:

Transistor aging effects in 22nm or below process nodes result in transistor performance degradation over the device life-time. The primary physical mechanism behind transistor aging is Bias Temperature Instability (BTI), Hot Carrier Instability (HCI) and time-dependent-dielectric breakdown (TDDB). NBTI primarily increases the threshold voltage (Vth) of PMOS devices. Such parameters shift, significantly affects circuit lifetime and performance (e.g. power, speed and failure rate), and in the worst case, may even result in a complete parametric failure of a system.

1.4 EOS:

EOS means the Electrostatic Overstress and used to describe the thermal damage that may occur due to current or voltage that is the beyond the specification limit. So it generates high heat and damages the devices. Latch up is the major sources of the EOS as it develops a parasitic SCR which increases the current and voltage drastically and damages the device. There are other sources of the EOS like Uncontrollable voltage surge on power supply, overshoot and undershoot during IO switching, poor grounding and the poor PSD protection etc.

1.5 Signal Integrity:

Signal integrity means to check quality of the electrical signal. Transmission line effect, ground bounce, Cross-talk, Dielectrical absorption are main causes of degrades quality of the signal. These parameters add noise and which creates duty cycle error and Jitter to signal. Power supply induced Jitter main causes of the Deterministic Jitter, Thermal and flicker noise add Random Jitter to signal.

1.6 References:

1. Cristiano Forzana, DavidePandini, Integration ,the VLSI journal, Statistical static timing analysis:A survey.

2. http://www.design-reuse.com/articles/20296/power-management-leakage-control-process-compensation.html

Chapter 2

AGING Simulation flow

2.1 Introduction

- We have analyzed degradation induced by BTI (Bias Temperature Instability), HCI (Hot Carrier Instability) on the NMOS and PMOS.
- The dependence of Aging effect on key process (e.g., L, Vth, Tox) and design parameters (e.g.Vdd, Vds, duty cycle, Temperature etc.)

2.1.1 NBTI:

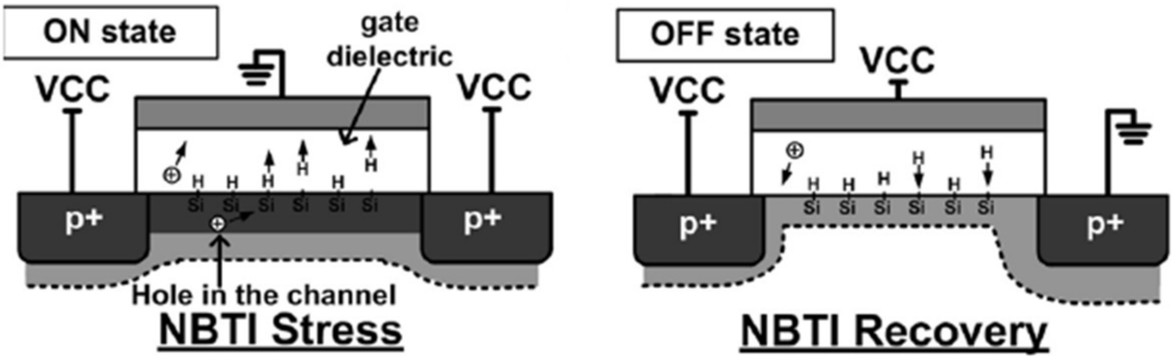


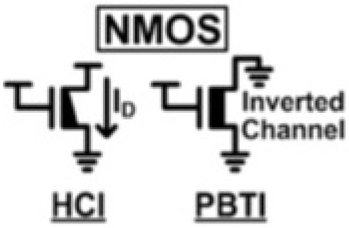
Figure 2.1: NBTI effect [2]

- Negative bias temperature instability (NBTI) in PMOS transistors is often cited as the primary reliability concern in modern processes, especially after the introduction of nitrogen into gate stacks, which reduces boron penetration and gate leakage, but leads to worse NBTI degradation. This mechanism is characterized by a positive shift in the absolute value of the PMOS Vth [1].
- The shift is generally attributed to hole trapping in the dielectric bulk, and/or to the breaking of Si-H bonds at the gate dielectric interface by holes in the inversion layer, which generates positively charged interface traps.

- When a stressed device is turned off, it immediately enters the recovery phase, where trapped holes are released, and/or the freed hydrogen species diffuse back towards the substrate/dielectric interface to anneal the broken Si-H bonds, thereby reducing the absolute value of the Vth [1].

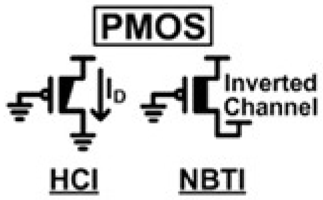
1. **PBTI effect:**

   - In Nmos vg = ”1” and source and drain taking to the ground.
   - Then the threshold voltage shift due to the PBTI.



2. **NBTI effect:**

   - In pmos Vg = ”0” and source and drain connected to Vcc.
   - Shift in the threshold voltage due to the NBTI.



2.1.2 HCI:

With designs moving into deep sub-micron (DSM) levels, shorter channel lengthscause the electric field in the channel to become larger. A channel carrier is accelerated by a high electric field of the drain near the drain’s area. It raises ionization collision causing electron-hole pairs to occur. Some of these electrons and holes are injected into the gate oxide. This effect is called hot carrier injection. It is said to produce the worst device degradation under normal operating-temperature ranges [3].

1. **HCI effect on NMOS:**

   - In NMOS Vg = ”1” and source connected to vcc and drain connected to ground.
   - Shift in the threshold voltage due to HCI.

2. **HCI effect on PMOS:**

   - IN PMOS Vg = ”0” and source connected to Vcc and drain connected to ground.

- Shift in the threshold voltage due to HCI.

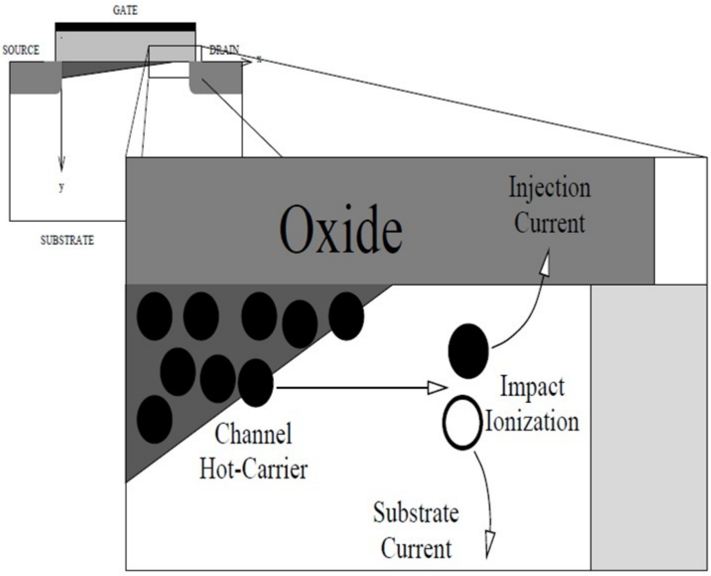


Figure 2.2: HCI effect [3]

2.2 HCI effect:

Simulation tool to predict the behavior of the circuit after arbitrary time period. For PMOS, degradation model based on the threshold voltage shift and for NMOS, degradation model is based on the uniform current scaling and combine both the degradation results are analyzed.

2.2.1 Aging-sim Flow:

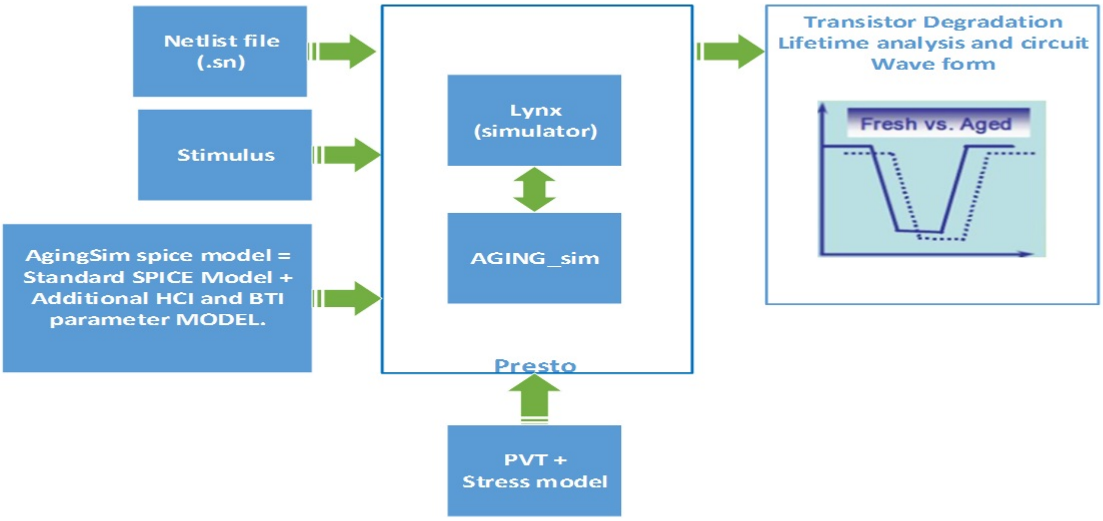
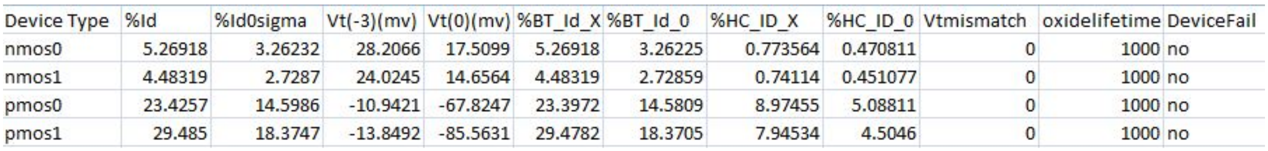

Netlist file (.sn)
Stimulus
AgingSim spice model = Standard SPICE Model + Additional HCI and BTI parameter MODEL.
Lynx (simulator)
AGING_sim
Presto
PVT + Stress model
Transistor Degradation Lifetime analysis and circuit Wave form
Fresh vs. Aged

Figure 2.3: Aging-Sim tool Flow

2.2.2 Aging-Sim Simulation Mode:

1. **Fresh mode:**
   Fresh mode calculate the current behavior of the circuit at the worst case corner like (Fast corner or slow corner) at high voltage and high or Low temperature. Temperature depends on the selection of the worst case corner.

2. **Stress Mode:**
   It calculates amount of the circuit performance degradation after arbitrary time of period. Generally we take typical corner (tttt) for this mode. Calculate circuit degradation during short transient simulation and extrapolate the degradation out to the desired degradation time. Assume that signal from short transient simulation is repeated throughout the entire degradation time. Generate Age Report shown in figure with degradation statistics for each transistor.



Device Type	%Id	%Id0sigma	Vt(-3)(mv)	Vt(0)(mv)	%BT_Id_X	%BT_Id_0	%HC_ID_X	%HC_ID_0	Vtmismatch	oxidelifetime	DeviceFail
nmos0	5.26918	3.26232	28.2066	17.5099	5.26918	3.26225	0.773564	0.470811	0	1000	no
nmos1	4.48319	2.7287	24.0245	14.6564	4.48319	2.72859	0.74114	0.451077	0	1000	no
pmos0	23.4257	14.5986	-10.9421	-67.8247	23.3972	14.5809	8.97455	5.08811	0	1000	no
pmos1	29.485	18.3747	-13.8492	-85.5631	29.4782	18.3705	7.94534	4.5046	0	1000	no

Figure 2.4: Aging Report

3. **Play-Back Mode:**
   Simulate degraded circuit performance based on Stressing Mode results. Simulate fresh and degraded waveforms at the same operating. Compare to fresh device characteristics to determine impact of degradation.

2.3 Aging-SIM Result of Buffer circuit:

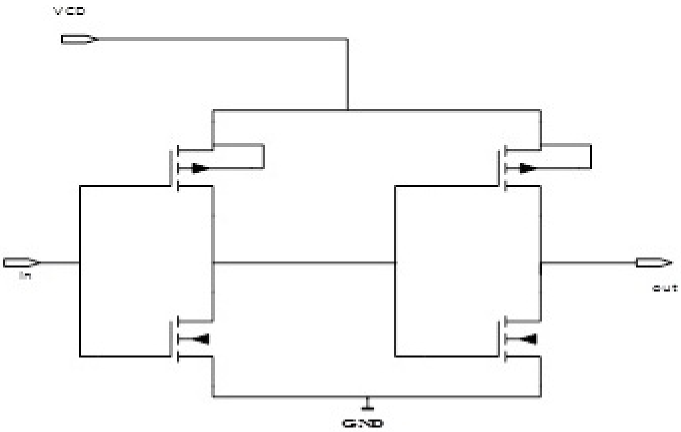


Figure 2.5: Buffer Circuit

2.3.1 Input Stimuli:

VCD $->$ DC voltage
In $->$ pulse waveform (60% duty cycle)

2.3.2 Aging-SIM Mode information:

1. **Fresh mode:**
   Process corner − > slow corner
   Temp − > hot temperature

2. **Stress mode:**
   Process corner − > typical corner
   Temp − > cold temperature

3. **PlayBack Mode:**
   Same process corner and Temp is used in Fresh mode.

For fresh mode and in aged mode for different age rise time delay between input and output calculated as shown in the figure 2.6.

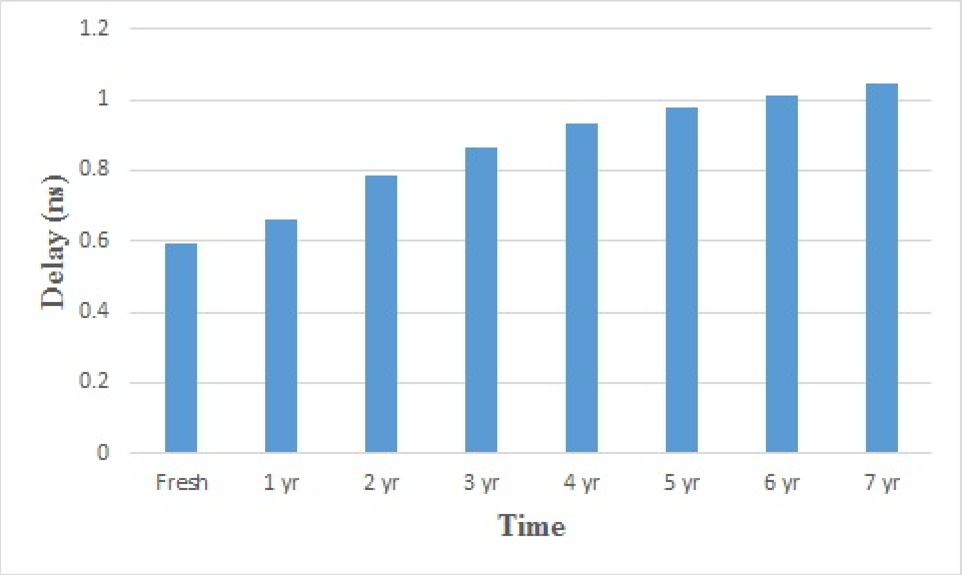


Figure 2.6: Fresh VS aging of inv chain

2.4 Conclusion:

Shift in the threshold voltage in aging because of the BTI and HCI effect. Degradation increases with increases year. NBTI effect more impact than the PBTI on the circuit.

2.5 References:

1. John Keane, Xiaofei Wang, Devin Persaud, and Chris H. Kim, an All-In-One SiliconOdometer for Separately Monitoring HCI, BTI, and TDDB.

2. Aditya Bansal a, Rahul Rao a, Jae-Joon Kim a, Su Zafar a, James H. Stathis a, Ching-Te Chuang a,b, Impacts of NBTI and PBTI on SRAM static/dynamic noise margins and cell failure probability.

3. Rakesh Vattikonda, Wenping Wang, Yu Cao, Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design.

Chapter 3

Electrical Overstress and ERC

3.1 Introduction

Electrical Over stress means the thermal damage that may occur when an electronic device is subjected to a current and voltage that is beyond the specification limits of the circuit. Thermal damage is the result of the excessive heat generated during the EOS event. The heat increase temperature and high temperature damages material used in the circuit [2].
Causes of the EOS:

- Overshoot and undershoot during IO switching.
- Poor grounding resulting in excessive noise on the ground plane.
- Uncontrollable voltage surge on the power supply.
- Latch-up may result the EOS and damage the devices.
- Voltage Spikes due to the external connection.
- ESD may damage the weaken part of the circuit and this part becoming more susceptible to future EOS event [2].

It is necessary to determine the EOS effect on the circuit at early phases otherwise at the final end (chip level) it may damage the circuit due to excessive heating and overall circuit will be fail. Designer need to be identify voltage violation between the device in circuit and transient violation if occurs, at early phases. So that we use two types of the tools used for finding EOS issues.

3.2 Static tools:

Static tool just checks the connectivity issue for the existence of high voltage path across the thin devices and does not depend on the functionality of circuit. Static tool is not used for finding the transient issue. **Advantage of static over the Dynamic tools:**

1. An input stimulus is not required.

2. Fast compared to Dynamic Tools.

Disadvantage:

1. Static tool generates the unmanageable number of false violation

2. Aging and BTI analysis is not possible.

Static tool use following algorithm:

1. **Voltage traversal algorithms**

   Voltage traversal starts from tracing the connection of each input signal and power supplies (including ground) to the first device element (transistor, diode, capacitor or resistor, etc.) in the netlist, and marks all the hierarchical nets along the path with the voltage value of that input signal or power supply.

   After first step, voltage traversal engine orderly picks input signals and power supplies on-by-one and propagates its voltage value over devices and deep into the netlist to every reachable node: For transistors, voltage is only propagated from drain to source, or vice versa. Voltage propagation terminates by gate or bulk of any transistors. Resistors are treated as short connections in voltage propagation. User can control how to propagate voltage across diodes and capacitors.

2. **Voltage checking algorithms**

   Voltage traversal engine saves the minimum and maximum voltage levels it sees for any node it trespass. Voltage checking engine calculates the minimum and maximum voltage differences between any two nodes of any transistors junctions and checks them against the official device junction voltage limits. Any junction rule violations will be reported in the output file [1].

3.2.1 HV_VERC

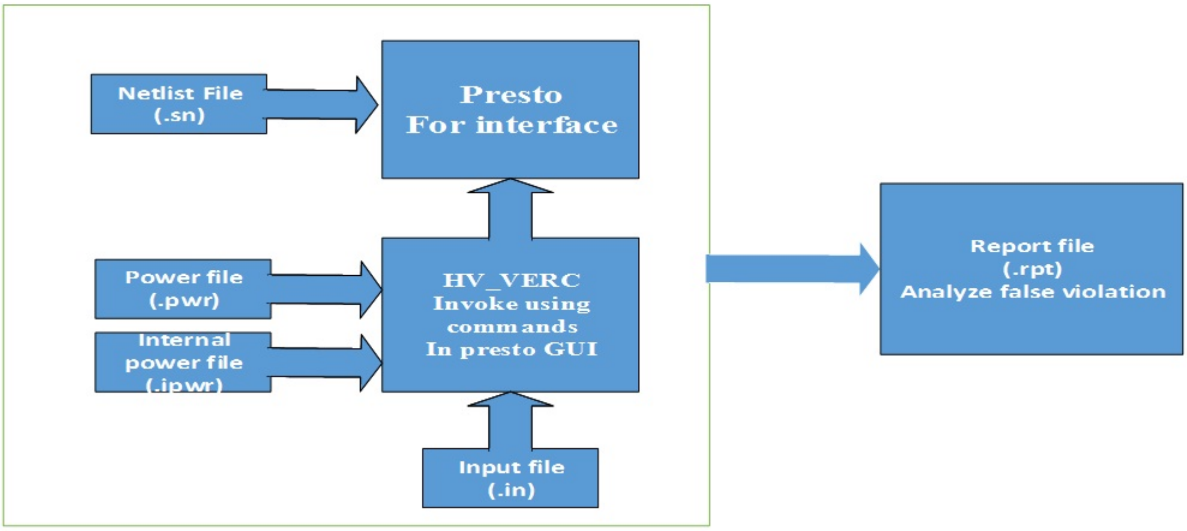


Figure 3.1: HV-VERC Flow

HV_VERC used for finding the connectivity issue.Netlist file, power file, input file are compulsory but internal power file is optional which is used when circuit has level shifter block.

3.2.2 Openrail

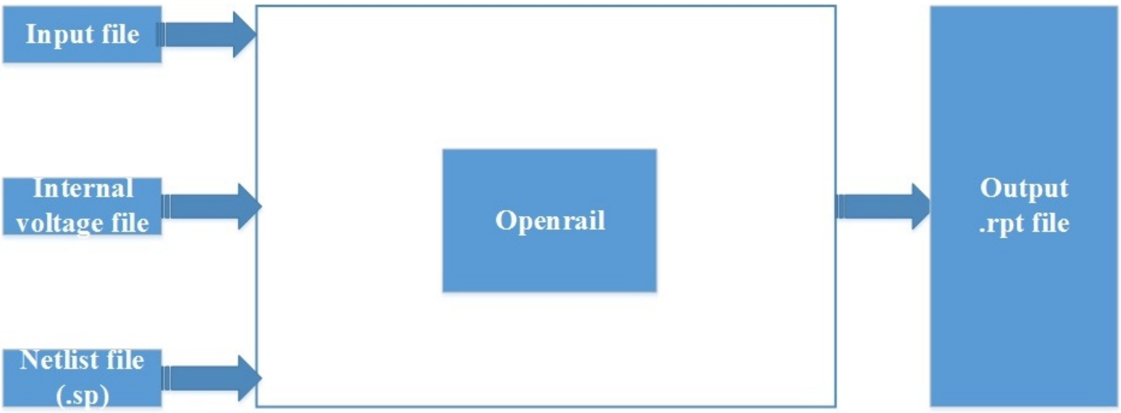


Figure 3.2: Openrail FLow

Openrail also used for finding connectivity issue. Netlist file, input file are compulsory but internal voltage file is optional.

3.3 Dynamic tool

Dynamic tool uses for finding the transient issues. Voltage propagation is depend on the state of transistor.

3.3.1 HV_AC

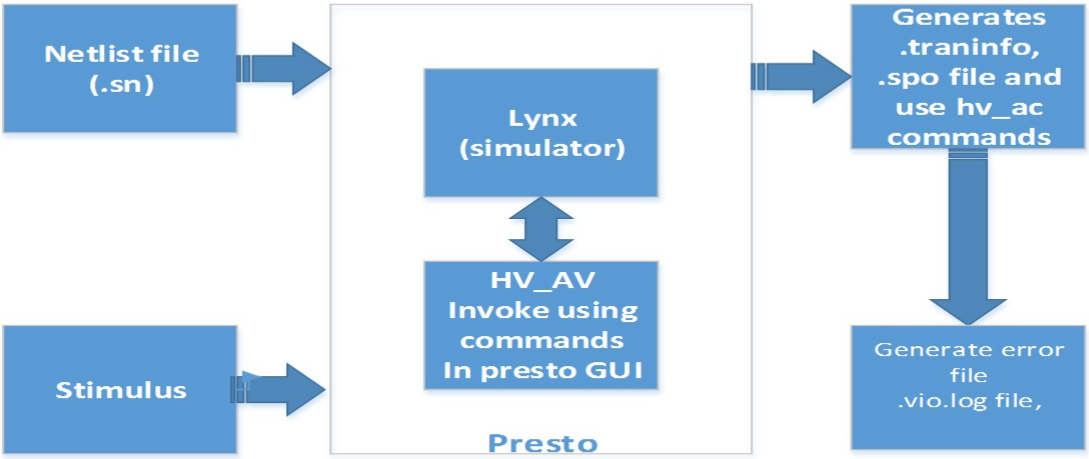

Netlist file (.sn)
Stimulus
Lynx (simulator)
HV_AV Invoke using commands In presto GUI
Presto
Generates .traninfo, .spo file and use hv_ac commands
Generate error file .vio.log file,

Figure 3.3: HV-AC FLow

Input: netlist file
Stimulus must require.
HV-AC simulation divided in to two parts after invoking HV_AC in presto environment:

1. Calculate voltage of each net using lynx simulator in presto environment.

2. **Voltage checking algorithm:** Voltage checking engine calculate voltage differences between any two terminals of any transistors junctions and checks them against the official device junction voltage limits. Any junction rule violation occurs then it will report in the report file (.vio.log file) with violation instant and violating voltage that instant [1].

3.4 Example on EOS tool:

EOS run for simple resistor based inverter chain in HV_VERC, OPENRAIL and HV_AC (dynamic) tool. In this section I described the voltage propagation algorithm and violation algorithm for each tool listed above and also described differences between these tools. Here VCC_HV and voltage as “1 ”are in extra high voltage domain and all NMOS are thin gate transistors. So voltage drop across any terminals cause the voltage violation.

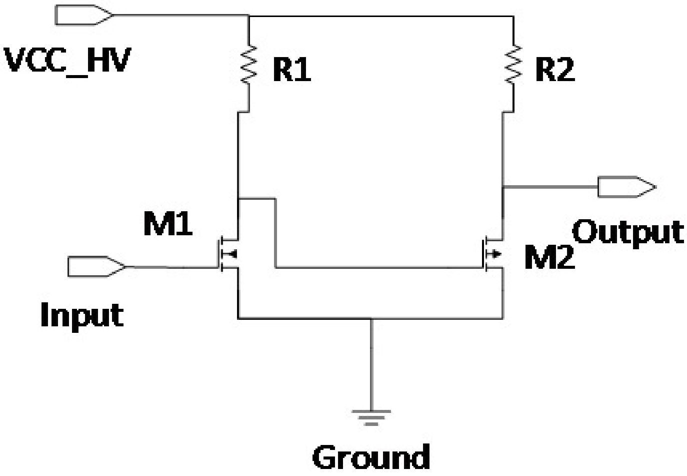


Figure 3.4: Resistive inverter chain

3.4.1 HV_VERC:

1. Netlist of circuit open in presto.
2. Create the input file, power file (having only information of global power supply) and internal power file if required.
3. Integrate the HV_VERC in presto using command and integrate input, power, netlist and internal power file to HV_VERC.
4. In HV_VERC voltage propagates as “1 ”or “0”or both through net.
5. Calculates possible maximum and minimum voltage drop between terminals of transistor.
6. If voltage drop is beyond the limit between terminals of transistor then tool list this terminals of transistor to report file.

3.4.1.1 voltage propagation Input = “1 ”(High voltage):

Voltage propagation through circuit is shown in square bracket. In hv_verc output takes the “0 ”.
from Figure 3.5

1. R1 and R2: In hv_verc resistance become shorted. At any terminal of the transistor at “1 ”then it passes “0 ”to other terminal of the resistor.
2. M1: Gate terminal has “1 ”and drain and source at “0 ”.
3. M2: all terminal of M2 transistor has “0 ”logic.

Violation summary:
From figure there is only two voltage violation across terminals GS and GD of transistor M1.

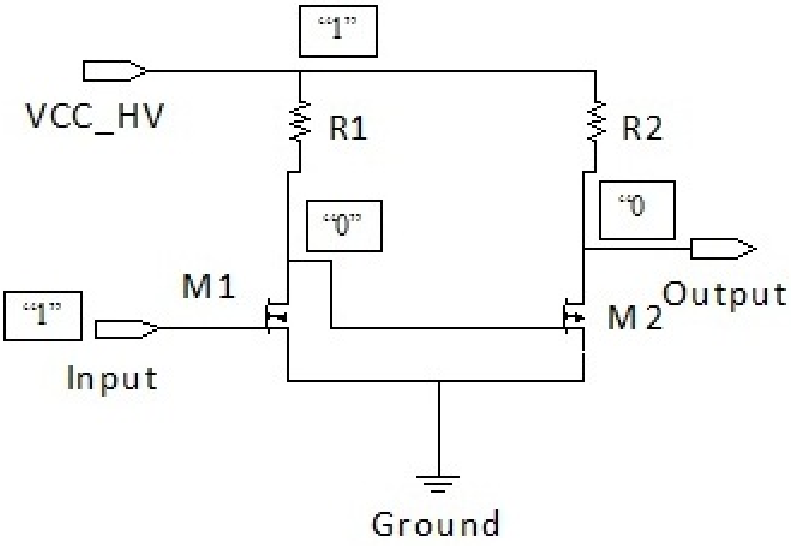


Figure 3.5: propagation in HV-VERC when input = ”1”

3.4.1.2 voltage propagation when Input = “0 ”(low voltage):

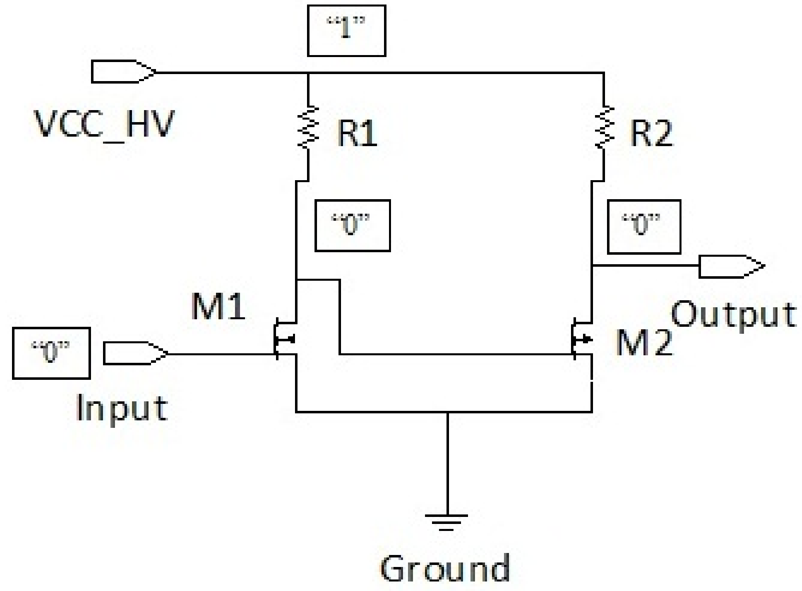


Figure 3.6: propagation in HV-VERC when input = ”0”

from Figure 3.6 ,

1. R1 and R2: In hv_verc resistance become shorted. At any terminal of the transistor at “1 ”then it passes “0 ”to other terminal of the resistor.

2. M1 and M2: All terminals at “0 ”.

violation summary:

From figure no voltage violation across terminals of any transistor.

3.4.2 Openrail:

steps followed by openrail:

1. Create the input file, internal power file and netlist file (spice format).

2. In Openrail voltage propagates as “1 ”or “0 ”or both through net.

3. Calculates possible maximum and minimum voltage drop between terminals of transistor.

4. If voltage drop is beyond the limit between terminals of transistor then tool list this transistor with maximum and minimum possible voltage for each terminal and with maximum and minimum voltage limit to violation report.

3.4.2.1 voltage propagation Input = “1 ”(High voltage):

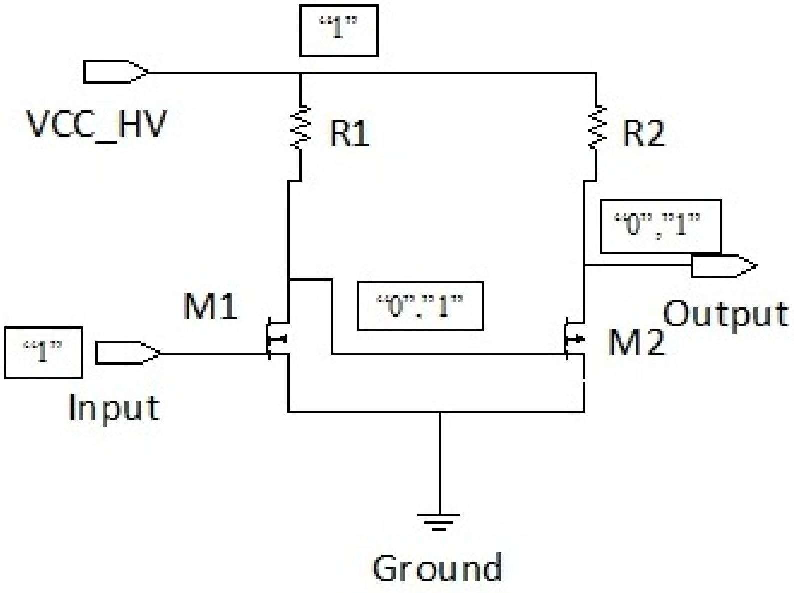


Figure 3.7: propagation in openrail when input = ”1”

From Figure 3.7;

1. R1 and R2: In open resistance become shorted.
2. M1: Gate terminal at “1 ”, source terminal at “0 ”and drain at “0 ”and “1 ”.
3. M2: Gate and drain take as “1 ”and “0 ”respectively and source take as a ground.

Violation summary:

In violation report file tool list only violation transistor. M1 and M2 are violated transistors for this experiment.
M1 nmos (vgmax = “1 ”, vgmin = “1 ”, vsmax = “0 ”, vsmin = “0 ”, vdmin = “0 ”, vdmax = “1 ”)
M2 nmos (vgmax = “1 ”, vgmin = “0 ”, vsmax = “0 ”, vsmin = “0 ”, vdmin = “0 ”, vdmax = “1 ”)

3.4.2.2 voltage propagation when Input = “0 ”(low voltage):

From figure 3.8 shows,

1. R1 and R2: Resistance becomes shorted and pass “1 ”to other terminal.
2. M1: Gate terminal at “0 ”and source terminal at “0 ”and drain terminal at “1 ”.
3. M2: Gate and drain take as “1 ”and “0 ”respectively and source take as a ground.

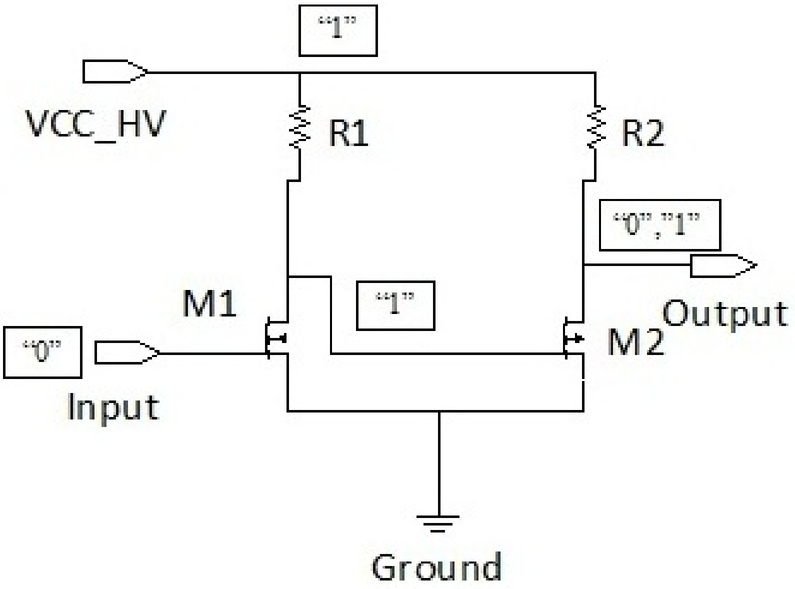


Figure 3.8: propagation in openrail when input = "0"

3.4.2.3 Difference between hv_verc and Openrail:

1. In HV_VERC always assumes output is "0 "but for Openrail output may be "0 "or "1 "or both.
2. In HV_VERC reports terminal of transistor which beyond limit but in openrail it reports only violated transistor.
3. In HV_VERC, when one terminal of resistor at "1 "then it passes to other end "0 "but in openrail it passes "1 "or "0 "or both.

3.4.3 HV_AC:

Step followed by HV_AC:

1. Netlist file open in the PRESTO and apply stimuli to circuit.
2. Integrate HV_AC with presto using command.
3. Run simulation on lynx simulator and it calculates the voltage on each net.
4. Calculate voltage drop between each terminal of transistor and check voltage within limit. If voltage between any terminal of transistor beyond the limit then terminal of violated transistor list to report file. In report file tool also list violated time and violated instant with violated terminal of transistor. Violated time and instant depend on the input stimuli.

3.4.3.1 voltage propagation when Input = “1 ”(low voltage):

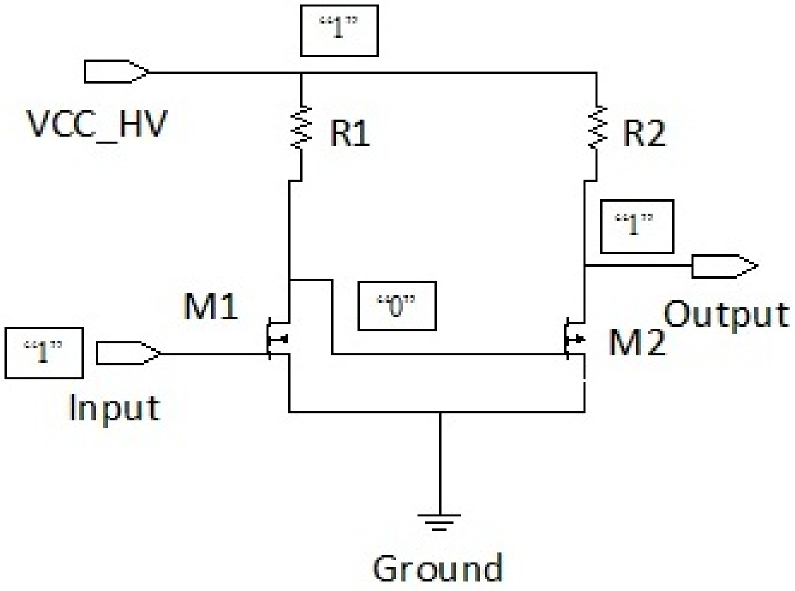


Figure 3.9: propagation in HV-AC when input = ”1”

Violation summary:

In HV_AC stimuli is compulsory and voltage at each net shown in rectangular bracket. The violation report contains:
Vgs, Vgd, Vgb, Vdb of transistor M1 with violated instant and voltage at that instant.
Vgs, Vdb, Vds, Vsb of transistor M2 with violated instant and voltage at that instant.

3.4.3.2 voltage propagation when Input = “0 ”(low voltage):

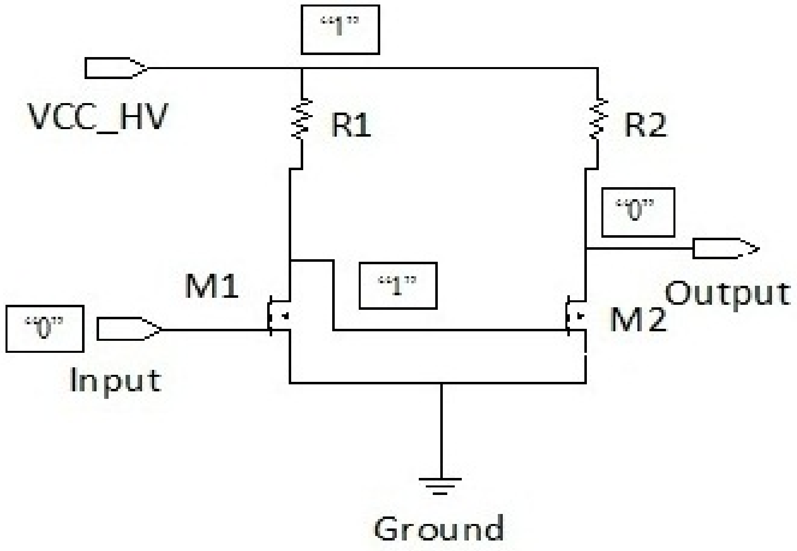


Figure 3.10: propagation in HV-AC when input = ”0”

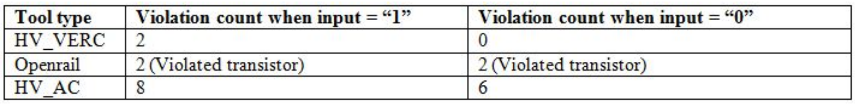
Violation summary:

In HV_AC stimuli is compulsory and voltage at each net shown in rectangular bracket. The violation report contains:
Vgd, Vdb, Vds of transistor M1 with violated instant and voltage at that instant.
Vgs, Vgb, Vgd of transistor M2 with violated instant and voltage at that instant.

3.4.3.3 difference between the HV_VERC and HV_AC:

1. In HV_VERC stimuli is not required but in HV_AC stimuli is compulsory.
2. HV_AC used for transient issue but HV_VERC used for only connectivity issue.
3. Resistor will be consider as short in HV_VERC but In HV_AC it does not consider resistor short and gives voltage drop across resistor terminal.

3.4.3.4 Violation count differences between the HV_VERC, openrail and HV_AC for resistive inverter chain:



Tool type	Violation count when input = "1"	Violation count when input = "0"
HV_VERC	2	0
Openrail	2 (Violated transistor)	2 (Violated transistor)
HV_AC	8	6

Figure 3.11: Violation count summary for HV-VERC, Openrail and HV-AC.

According to figure we conclude that static tool gives different number of violation. In static tool voltage propagation is not depend on the Gate voltage so that some time gives the wrong violation which is not actually available on the realtime circuit. So static tool generates the unmanageable voltage violation.

In Dynamic tool voltage propagation depend on the Gate voltage. But violation depend on the quality of the stimuli so efficiency of the tool result depends on the designer.

3.5 Automation work for EOS:

3.5.1 Violation summary file creation for XAEOS:

XAEOS is a dynamic fast spice EOS checking tool. It uses the XA fast spice engine from Synopsys to do device check analysis. It generates 2 main reports - DPM and PREDPM. DPM is a quality metric associated with Gate oxide (GOX) of the devices to estimate the EOS risk associated with each device. DPM is calculated by internal models based on certain criteria like device violating voltages, reference voltage, activity factor, violation fraction, etc. If the DPM criteria is met, the violation is waived else it is logged as an error. PREDPM retains all the EOS violation along with DPM info. DPM file only contains violations left out after filtering.

The script written automates the summary generation of the report based on violation type, model type, device count, etc. This allows the designer to quickly get an overview of the type of xaeos report he/she is going to review.

3.5.2 Quality check for EOS report file:

I made script for QA and archiving. Design Automation engineer use the QA and Design Engineer use the archiving. In QA-flow first we run test after that we call the script. In archiving flow automatically call script. This script checks two things: 1.) Existence of the required output files like report file, log files etc. and 2.) Quality check for EOS report file and script do following Quality check.

1. DPM parameter calculates for only gate related voltage violation. So script check that there is no DPM parameter for any non-gate terminals voltage violation for any transistor.

2. It read the DPM parameter from post violation file for gate related violation and compare with limit.

3. It check violation time is less than the transit time if not report for violation.

3.5.3 Make configuration file for XAEOS and openrail from EOS collateral:

Configuration file contain EOS limit of each transistor model. So configuration file is necessary to generate violation report of circuit for EOS tool. For each tool have different format of the configuration file and for each EOS tool we have to make the configuration file manually. So it takes lot of effort and time along with being susceptible to errors. The EOS limit can suffer due to the induced errors in the report file. So it creates the report file which take wrong EOS limits. If for one transistor model actual EOS limit is 2V but in configuration we take 3V in configuration file. So Report generated due to this configuration file miss actual EOS violation. So to avoid this manual error in configuration file, I was making script which make configuration file automatically for XAEOS and openrail tool from EOS collateral.

3.5.4 Reliability Regression:

Along with the update on tool and collateral, comparison between the old and the latest variants is done manually. Depending on the update the test case is selected and fired. It runs on the local machine. This is followed by another test case on completion of the former and requires a manual re-fire for the second test case and the flow continues for remaining test cases in a serial fashion. The last step is the comparison of the results thrown by the completed tasks which accounts for lot of time. It is necessary to automate these intermediate components by making all the selected test case run parallel. This is enabled by using a script which automatically takes selected EOS and AGING test cases and run them on different local machines.

3.6 ERC

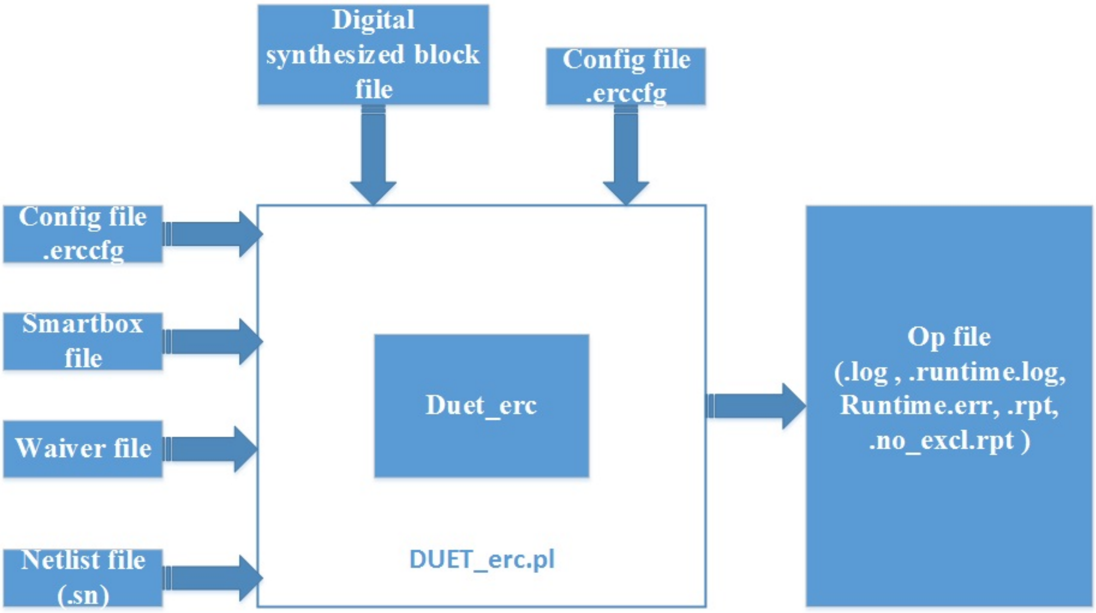


Figure 3.12: Duet-erc flow

DUET_ERC check for sizing, methodology and connectivity issues. So ERC identify the topological circuit defects against the certain predefined rules and requirements based on design and timing guideline:

1. **Configuration file:**
   Erccfg file contain input node, output node, analog cells, ground, node and upartition cells.

2. **Smart-box file:**
   Duet uses smart (schematic model abstraction and remodeling tools) for logic analysis of blocks. Smart is not capable to analyze the analog cells.

3. **Waiver File:**
   Waiver ignores the nodes and cell as per rule basis.

4. **Rule-set file:**
   For which rules we wanted to run ERC are included in Rule-set file [3].

3.7 Work done in ERC:

I show the count mismatch some of the erc rule for different version of the duet erc tool run and also for two different run for same version of the duet erc tool.
solution:

Update the waiver procedure.

3.8 References:

1. Xiaojun Lil, Alurkar, Mugdha D: HV_VERC user Manual.

2. www.cypress.com

3. Duet_erc noble training.ppt

Chapter 4

Signal Integrity

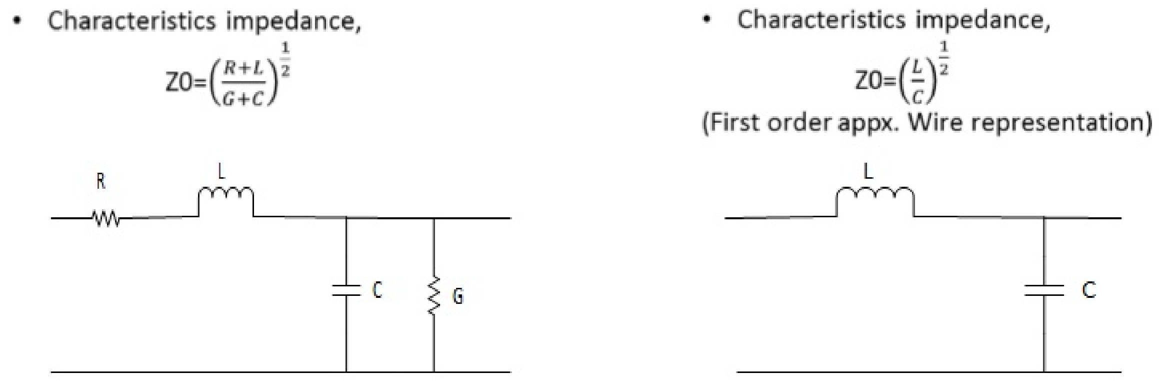
4.1 Introduction

Signal integrity means the measures the quality of the signal. At lower frequencies the signals remain within data characterization and the system performs as designed. But as system speeds increase, the higher frequency impact on the system means that the high frequency effect take over. Signal suffers at high frequency due to ringing, reflection, crosstalk and ground bounce. These parameters causes of noise, duty cycle error in signal. Noise in signal creates the jitter. Noise at the power supply causes of the deterministic Jitter and Thermal and Flicker noise cause of the Random Jitter. This Jitters affects the bit error rate of signal drastically and it seriously affects the integrity (quality) of the signal. So it is necessary to do signal integrity analysis at early phase of design and this analysis gives the deterministic and random Jitter for given circuit so we can check that jitter for each open loop analog block is within limit if not then we apply for circuit for further analysis. Here we use RavenSI as signal integrity tool and it is not use for digital and close loop blocks. It is use only for open loop analog block to calculate the Jitter.

4.2 Basic principle of the Signal Integrity:

There is various parameter which is responsible for degrades the quality of the electrical signal. These parameter shown below:

4.2.1 Transmission line effect:



Characteristics impedance,
ZO=$\left(\frac{R+L}{G+C}\right)^{\frac{1}{2}}$
R
L
C
G
Characteristics impedance,
ZO=$\left(\frac{L}{C}\right)^{\frac{1}{2}}$
(First order appx. Wire representation)
L
C

Figure 4.1: circuit model representation of transmission line

At low frequency wire or track is consider without capacitor and inductor but at high frequency wire and track consider with capacitor, resistance and inductor. A circuit model representation is shown in Figure 4.1 used to determine the characteristic impedance of the wire or track. This wire impedance is extremely important, as any mismatch within the transmission path results in a reduction in signal quality [2].

4.2.2 Impedance mismatch:

If Unequal impedance of source output Zs, load Zl and line Zo which causes the impedance mismatch. Due to this mismatch all Transmitted signal is absorbed at receiver end and some of the power reflected back to transmitter side. This process continues back and forth until all the energy is absorbed. At high data rates this can cause signal overshoot, undershoot, and ringing and stair step waveforms, which produce signal errors [2].

4.2.3 Signal attenuation:

Transmission line losses makes difficult for receiver correctly interpret the high frequency signal. The following two causes of transmission line losses are due to transmission medium:

1. **Dielectric absorption:**
   High-frequency signals excite molecules in the insulator, which causes the insulator to absorb signal energy. This absorption reduces the signal strength. Dielectric absorption relates to the PCB material and can be lessened by careful material selection.

2. **Skin effect:**
   Varying current waveforms caused by AC and high-frequency signals tend to travel on the conductor's surface. Signals traveling on the surface cause the self-inductance of the material to produce an increased inductive reactance at high frequencies, which forces electrons to the material's surface. The effective reduction

of conductive area causes an increase of resistance and, therefore, attenuation of the signal. The figure 4.2 shows cross section view of the conductor material [2].

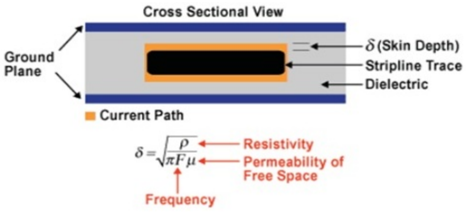


Figure 4.2: The Skin Effect [2]

4.2.4 cross-talk:

Whenever a signal is driven along a wire, a magnetic field develops around the wire. If two wires are placed adjacent to each other, it is possible that the two magnetic fields interact causing a cross-coupling of energy between signals known as crosstalk. The following two energy coupling types are the predominant causes of crosstalk:

1. **Mutual inductance:**
   A magnetic field causes induced current from the driven wire to appear on the quiet wire. This mutual inductance causes positive waves to appear near the transmitter end of the quiet line (near end inductance) and negative waves at the receiver end of the transmission line (far end crosstalk).

2. **Mutual capacitance:**
   The coupling of two electric fields when current is injected in the quiet line proportional to the rate of change of voltage in the driver. This mutual capacitance causes positive waves near both ends of the transmission line.

4.3 Jitter

Figure 4.3 shows a single period, or cycle, of a signal in which multiple different periods are evident. An ideal waveform repeats an invariable cycle. Actual waveforms, however, vary in the time domain, with signal edges rising or falling earlier (red) or later (blue) than they are supposed to jitter is produced by things such as slight instabilities in electrical signal reading devices and interference along signal carrying pathways. Jitter is the result of time-domain fluctuations in digital signals, but jitter comes in a lot of different types. Jitter is hard to assess with a single numeric value because it changes minutely over time and because there is a variety of fluctuation patterns with respect to time [3].

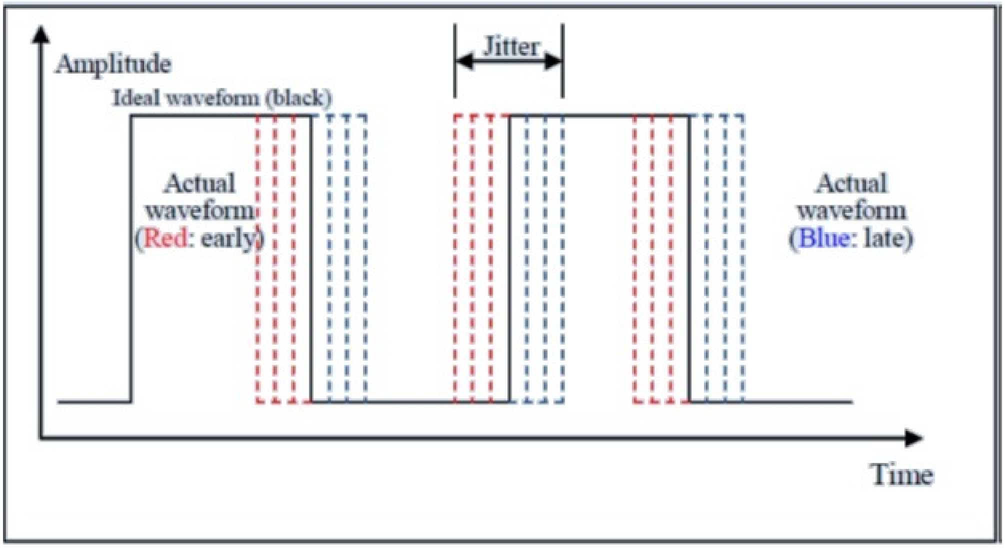


Figure 4.3: Jitter calculation [3]

Types of Jitter

1. **Period jitter (peak-to-peak)**
   It is measure in variation in the peak to peak value. Peak to peak jitter means the total jitter and find peak to peak jitter using the following equation:

   $$TJ = DJ + (n * RJ)$$

   Where TJ = total jitter
   DJ = deterministic jitter
   n = number of standard deviations corresponding to required BER.

2. **RMS jitter:**
   RMS jitter takes the standard deviation (1-sigma) from the measured result.

3. **Random jitter:**
   Random Jitter is a Jitter component which cannot be predicted and random jitter is unbounded.Principle source is the Gaussian noise within the system component. It interacts with the slew rate of signal and produces the timing error at switching point.Most of the noise and jitter in electrical system due to thermal noise, which has a Gaussian distribution.

4. **Deterministic jitter:**
   Deterministic Jitter withnon-Gaussian probability density function.It is a bounded jitter and caused by a circuit design, electromagnetic induction and etc. Deterministic jitter is sandwiched between the random Jitter as shown below.

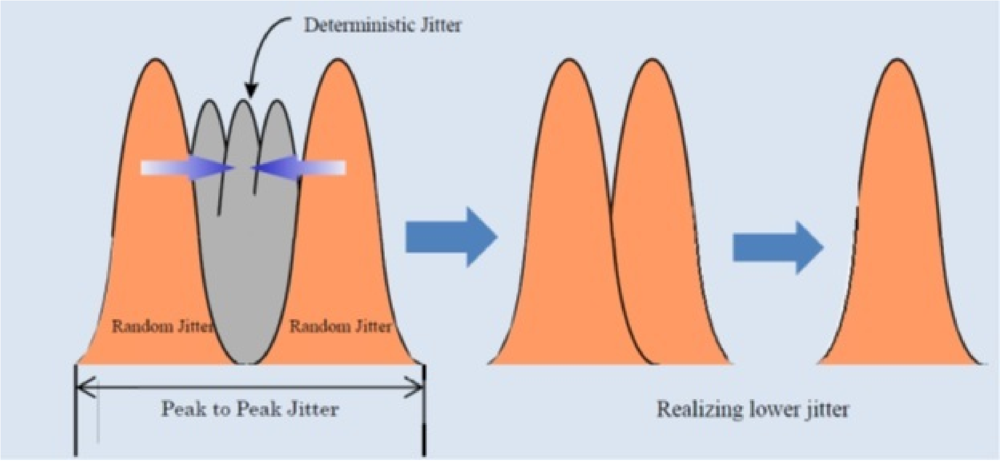


Figure 4.4: Deterministic and Random Jitter [3]

To summarize the concept of the random Jitter (RJ) and Deterministic Jitter (DJ) comprise overall jitter components in a single cycle length. The key to reducing jitter is to reduce deterministic jitter. Optimizing this component causes the gap between right and left RJ to overlap so that it can exist as an ideal normal distribution.

5. **Accumulated jitter (long-term jitter):**

   Jitter JAC(n) is the time displacement of the edges of a clock relative to the triggering edge of the same clock. This jitter is a function of n and it is the general case of JPER. It is measured by its Peak-to-Peak value or RMS. It gives the maximum discrepancy between two clock rising edges n cycles apart. For adigital system with synchronized packet data, this jitter measure may be the right one for system timinganalysis. Its mathematic form is given as:
   Accumulated Jitter:

$$Jac(n) = Tper(n) - nT0$$

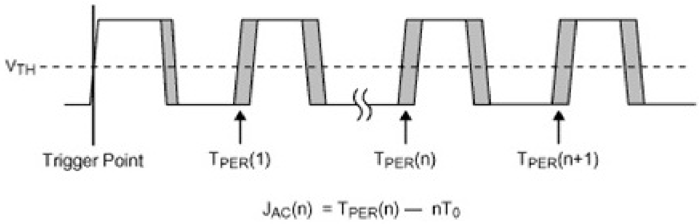


Figure 4.5: Accumulated jitter measurement [4]

4.4 Simulation Result:

4.4.1 Tool Information:

RavenSI is signal integrity tool and used for measuring the power supply induced jitter, duty cycle amplification, phase transfer function analysis, thermal and Flicker noise analysis and wideband modulation. In RavenSI we used additional special stimuli for post processing. It is a generic algorithm to automate the IO blocks. Extracted parameter is passed to a SIGSIM and evaluates the eye diagram for measure the bit error rate.

For power supply induced noise jitter RavenSI calculates the jitter phase and jitter amplitude for different number of jitter frequency and pass to SIGSIM. Number of jitter frequency depends on the number of harmonics. SIGSIM makes eye-diagram from these parameters and calculate the bit error rate.

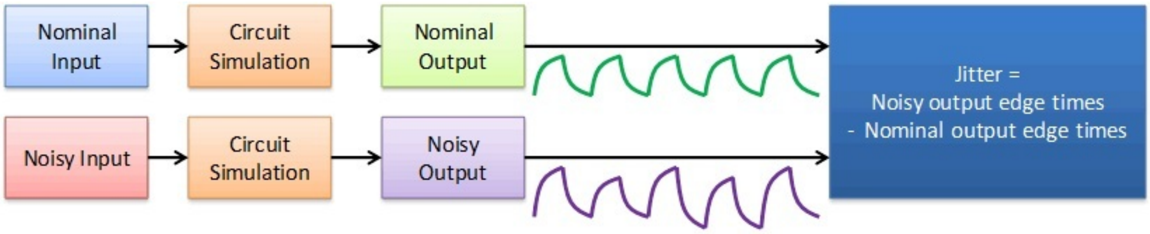


Figure 4.6: RavenSI tool flow [1]

For PSN and TFN analysis we use the below block diagram:

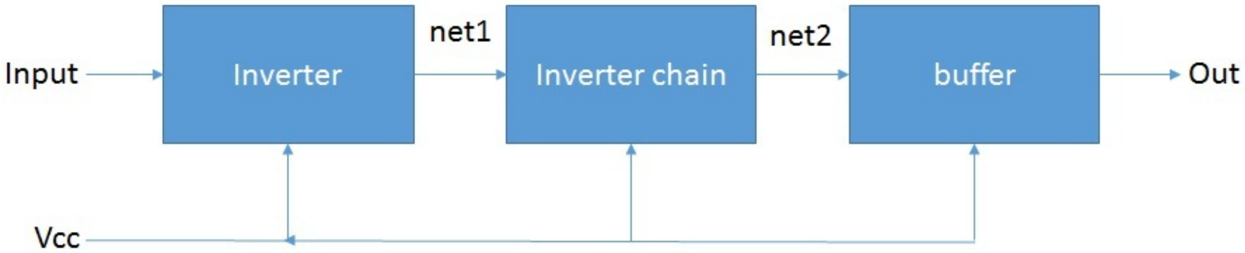


Figure 4.7: Dummy block diagram use for RavenSI analysis

4.4.2 Power supply induced noise jitter analysis:

RavenSI allows the introduction of arbitrary power supply noise waveforms, which could have been obtained via power-grid measurements or simulations. Additionally, single sinusoids of a frequency estimated by designers or DC shifts can be used. It is recommended, though, to adopt the more rigorous approach of using measured or simulated power-grid waveforms. [1]

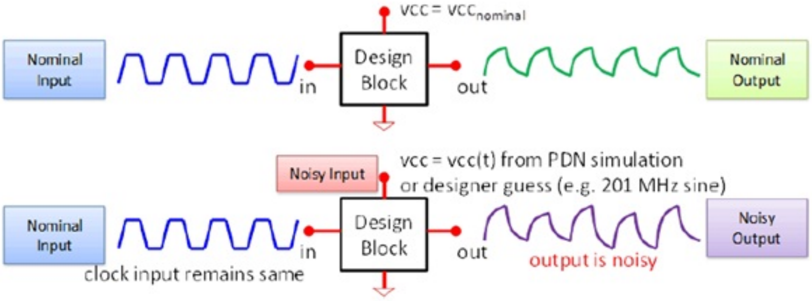


Figure 4.8: power supply noise induced jitter [1]

For PSN analysis requires two DC source one is nominal and noisy. Nominal power supply take as DC voltage and noisy set following two ways:

1. **Appling the PWL (piecewise linear model):**
   In PWL file having two column, time and voltage.

2. **Noisy sinusoidal superimpose on the DC:**
   Frequency of the noisy sinusoidal is calculated as below:

   $$Jittersensitivity = Jitteramp(f)/(p - pamp.)$$

   Then swiping the frequency and measure the jitter sensitivity. Select this frequency which having highest jitter sensitivity. In this experiment we take 201 MHz sinusoidal Frequency.

Stimuli:

- **Noisy stimuli:**
  1. **Power supply (VCC):** 201MHz 50mv (p-p) sinusoidal signal superimpose on 1.5V DC.
  2. **Input stimuli (input):** Pulse waveform with, v1 = 1.5V, v2 = 0V, period = 200ps, delay= 100ps, fall = 20ps, rise = 20ps, pulse width = 80ps
- **Nominal stimuli:**
  1. **Power supply (VCC):** 1.5V DC.
  2. **Input stimuli (input):** Pulse waveform with, v1 = 1.5V, v2 = 0V, period = 200ps, delay= 100ps, fall = 20ps, rise = 20ps, pulse width = 80ps

This section divide in to parts:

First part contains the comparison between the circuit simulation result and the RavenSI result of circuit shown in figure 4.7. These comparison is very useful because it is necessary to check that RavenSI produce same Jitter number as the actual circuit produce. So it check the stability of the RavenSI tool. RavenSI is automated tool which produce jitter (DJ, RJ) and frequency domain from the time domain analysis.

Second part contains the PSN analysis of circuit shown in figure 4.7 and try to find the open loop analog block which impact the most the overall Jitter (DJ) of the circuit. If the Jitter (DJ) is abruptly large or greater than the critical limit than circuit apply for further analysis.

1. **Comparison between the Circuit simulation with noisy power supply and RavenSI PSN analysis result for Figure 4.7.**
   Circuit simulation:

   - **Steps require for circuit stimuli:**
     (a) Set stimuli as described before run simulation in the ADE-L.
     (b) Calculate the jitter using abs_jitter command in calculator which is available in ADE-L and it generated the Jitter curve in time domain is shown in Figure 4.9.

From the figure 4.9 we calculate the worst case Jitter produce by the circuit using circuit simulation = 6.6ps

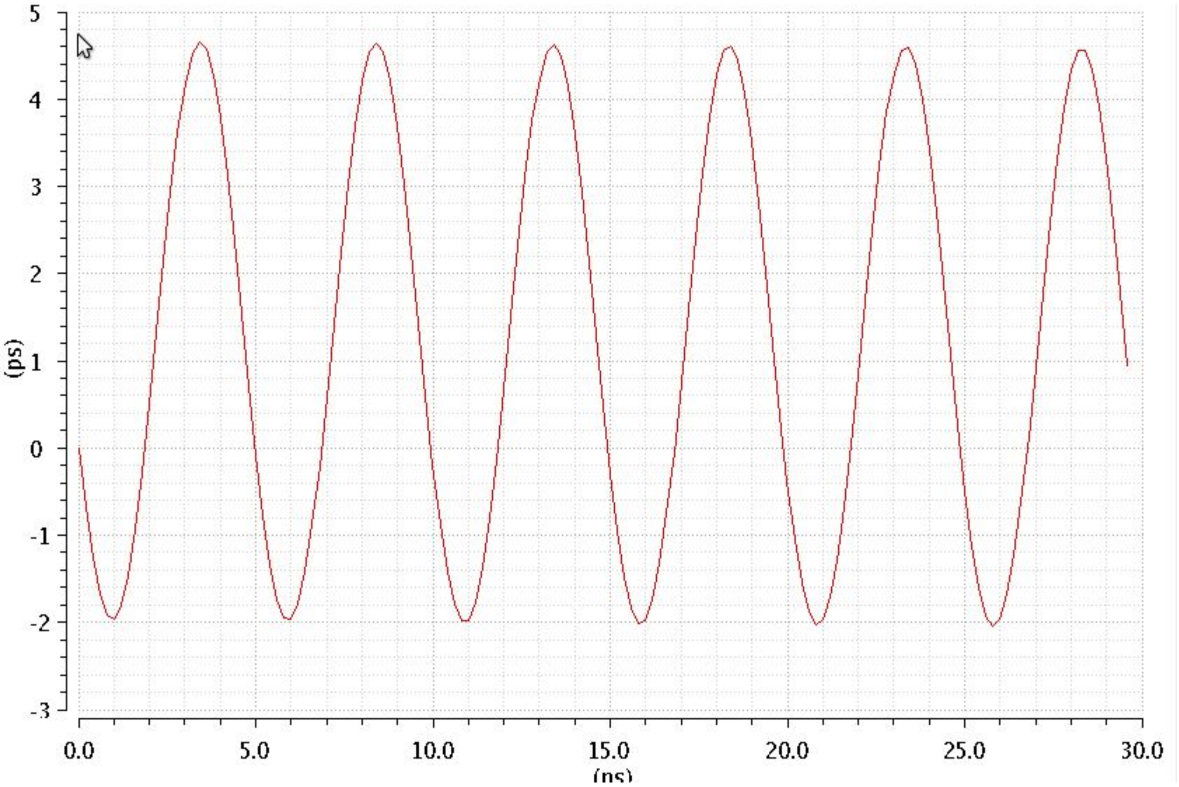


Figure 4.9: Circuit simulation for Figure 4.7

RavenSI simulation:

RavenSI require nominal as well as the noisy stimuli as explained above.

From Figure 4.10 we calculate the worst case Jitter for RavenSI PSN analysis = 6.6ps

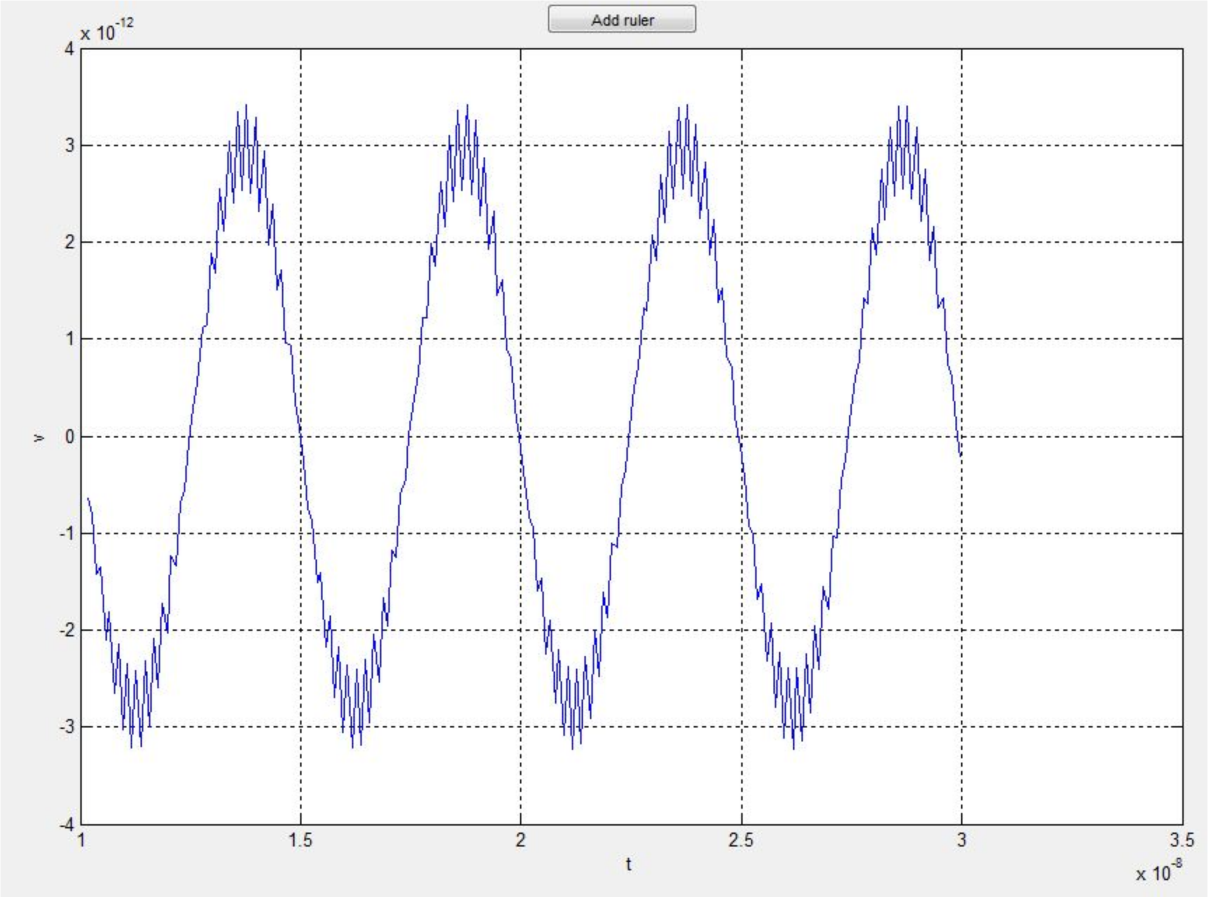


Figure 4.10: Time domain Jitter curve for PSN generated by circuit analysis

2. **PSN analysis for Figure 4.7 and analyze DJ. Try to find which block impacts most the output Jitter.**

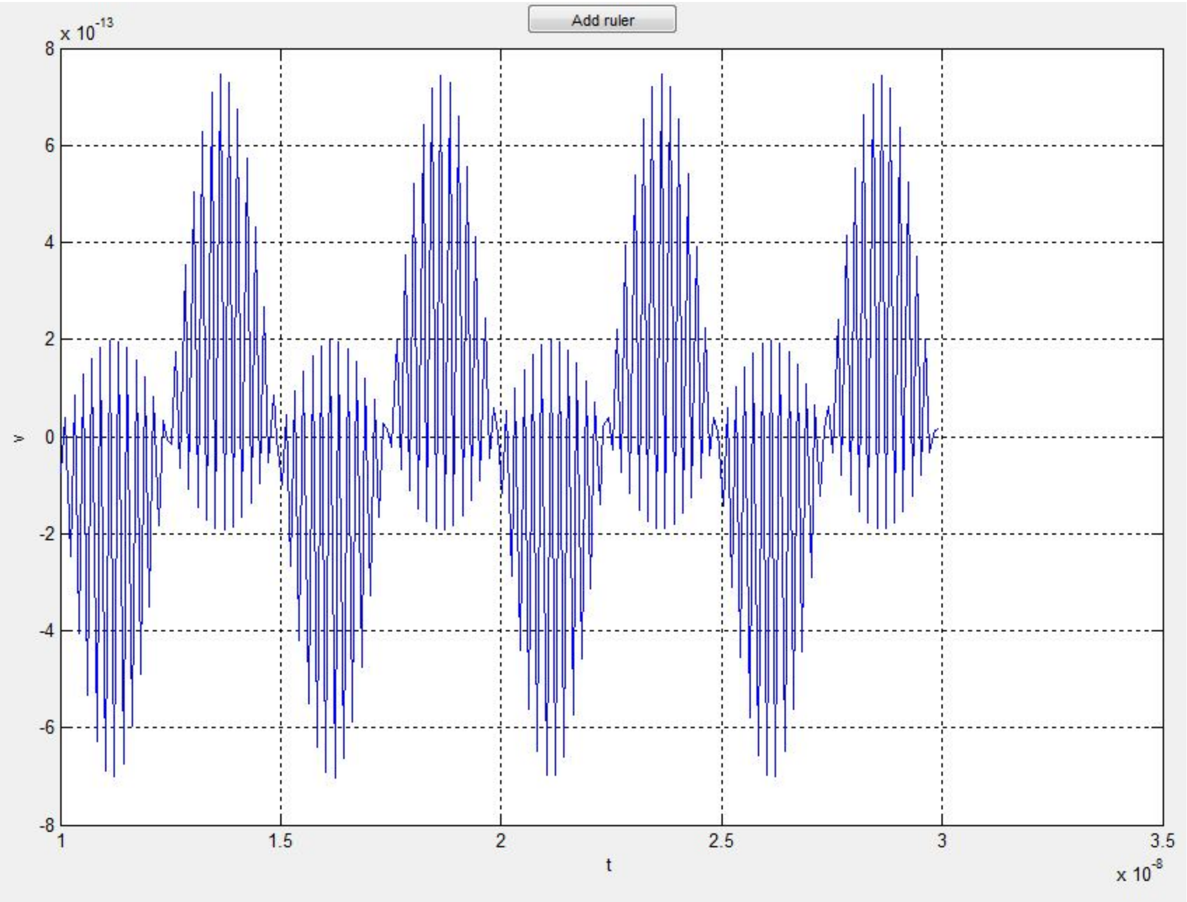


Figure 4.11: Deterministic Jitter at net1 for Figure 4.7

Deterministic Jitter at net1 in Figure 4.7 = 1.45ps So Deterministic Jitter introduced by inverter block = 1.45ps

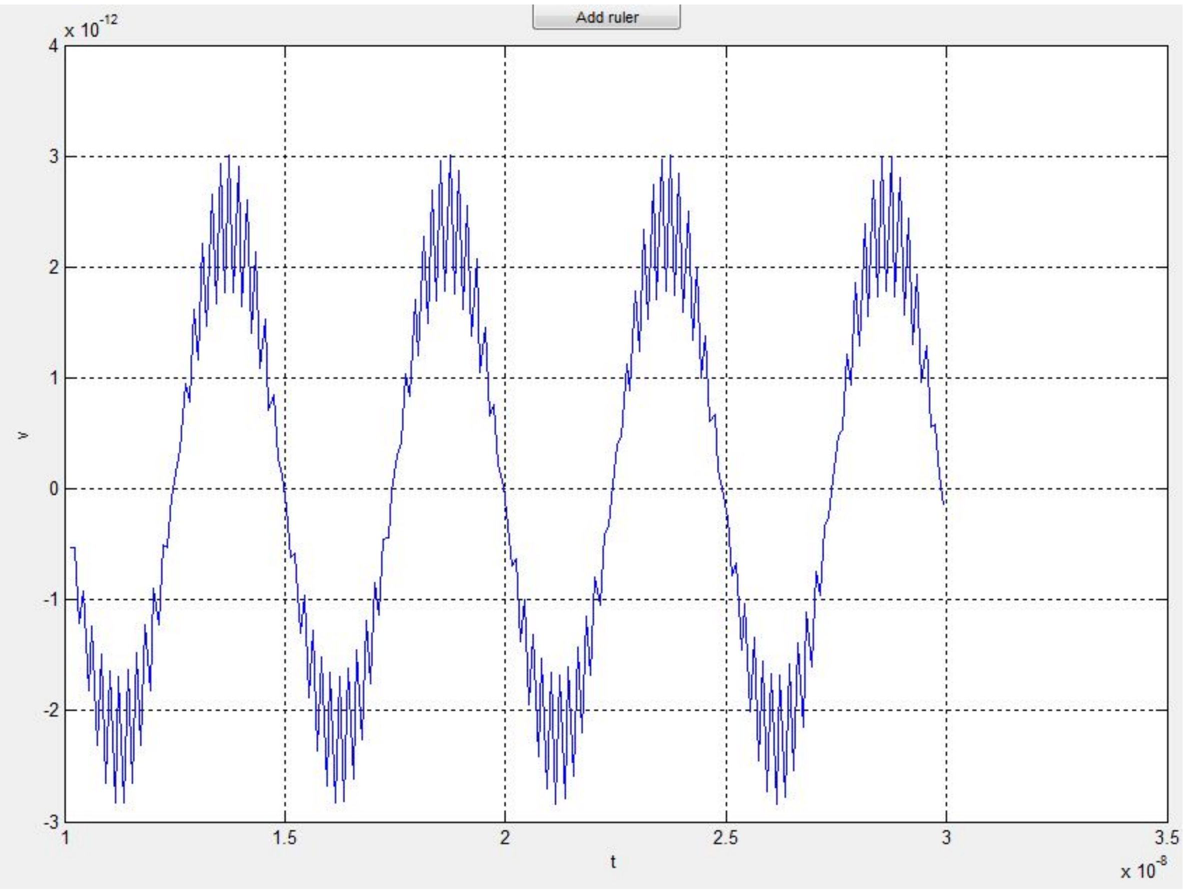


Figure 4.12: Deterministic Jitter at net2 for Figure 4.7

Deterministic Jitter net2 of Figure 4.7= 5.8ps. Deterministic is accumulated Jitter. So Deterministic Jitter introduce by inverter chain block = 5.8-1.45 = 4.25ps

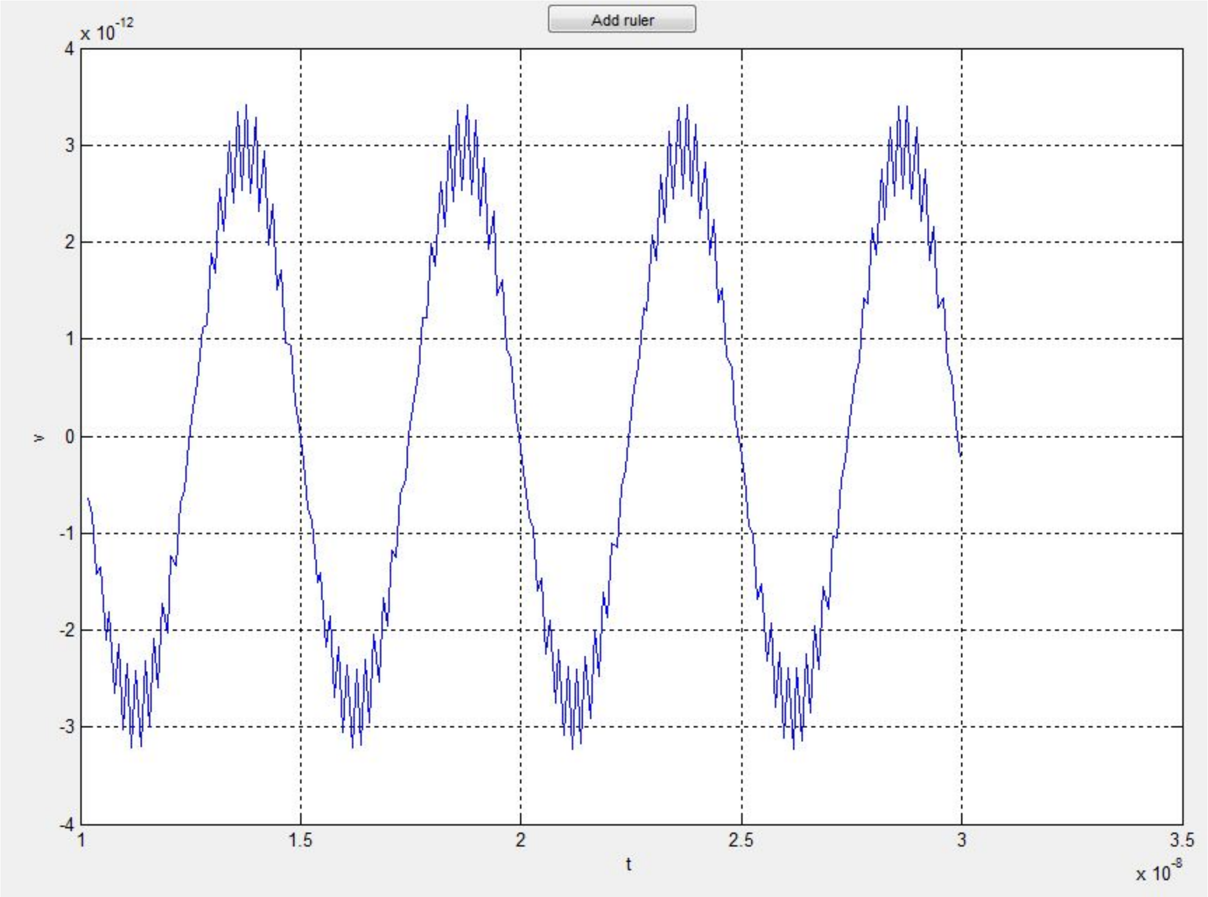


Figure 4.13: Deterministic Jitter at out net of Figure 4.7

Deterministic Jitter at out net of Figure 4.7 = 6.6ps.
So Deterministic (p-p) Jitter introduce by buffer block = 6.6 5.8 = 0.6ps

RavenSI calculates jitter phase and jitter magnitude for different jitter frequency in PSN analysis and passes these parameters to SIGSIM where it calculates the bit error rate from eye opening.

If the deterministic jitter for each block or entire circuit is abruptly large or greater than the threshold value which is given by the circuit team than circuit passes for further analysis.

Conclusion:

Jitter produce by the circuit simulation and the RavenSI analysis is almost same. In circuit the stage2 means inverter chain introduces more jitter compare to the other block.

4.4.3 Thermal and Flicker Noise induced Jitter Analysis (TFN):

Unbounded sources of jitter, such as thermal and flicker (1/f) noise, lead to random jitter, which is typically characterized by a Gaussian distribution. This distribution can be made via noise simulations in the time domain or in the frequency domain by assuming a linear transfer function mapping the thermal noise to output of the component. Such a transfer function can be obtained by regular AC analysis, which readily performs the linearization.

If such linearization is not desirable, the time-domain approach requires a time-domain noise simulation with a certain stop time. Although this approach typically requires a longer noise simulation, it has the advantage of including the non-linear device effects that could contribute to random jitter. This method also obviates the need to define a linear transfer function, which may be conceptually difficult to define for multiple devices located in different parts of the circuit. Hence, the time-domain approach has been implemented in RavenSI as shown below:

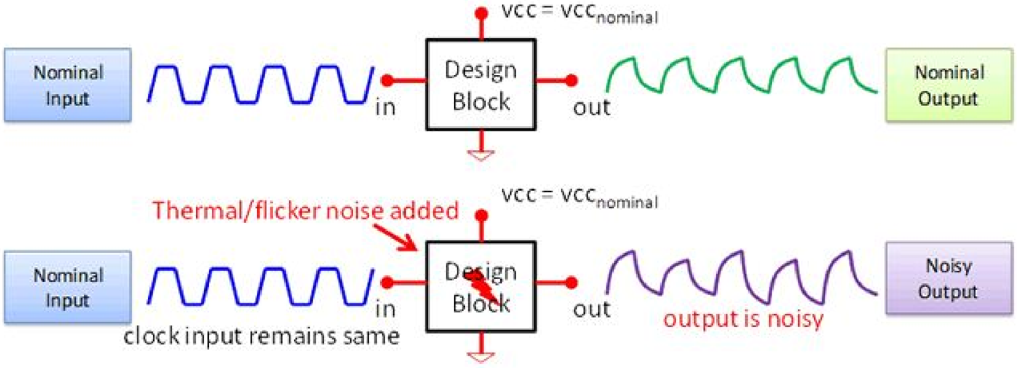


Figure 4.14: Thermal and Flicker Noise induced Jitter Analysis [1]

Simulation result of RavenSI TFN analysis for Figure 4.7:

1. **Nominal stimuli:** same as PSN analysis

2. Noisy Stimuli: For noisy stimuli set thermal noise for each block of the circuits.

Maximum flicker frequency = 2G and
Minimum Flicker Frequency = 10M.
By taking these noisy stimuli we get following RJ for Figure 4.7
RJ at net1 of Figure 4.7 = 14.811fs
Inverter block introduces 14.811fs RJ.
RJ at net2 of Figure 4.7 = 63.143fs
Inverter chain block introduces 48.332fs RJ.
RJ at out net of Figure 4.7 = 69.855fs
Buffer introduces 6.712fs RJ.
RavenSI calculates sigma value (RJ) and passes to SIGSIM and there we calculate the bit error rate from the eye opening created by the SIGSIM.
Conclusion:
Inverter chain block of Figure 4.7 is major source of the RJ (Random jitter).

4.5 References:

1. Presto user manual

2. White paper on Basic principles of signal integrity-Altera.

3. http://www.eetimes.com/

4. http://www.maximintegrated.com/