Power Planning and IR Drop Analysis

Dr.-Ing. Amir Najafi, Abdul Aziz

February 7, 2024

Contents

---

1 Goal

This document contains information on power planning and IR drop analysis. Specifically, it cointains:

- Theory and math related to power and rail analysis.
- Practical know-how on doing rail and power analysis using EDA tools.
- New technologies for IR drop reduction.
- State of the art approaches for IR drop estimation using AI.

2 Theory

`https://www.ansys.com/blog/minimizing-ir-drop-in-integrated-circuit-design`
`https://www.vlsi-expert.com/2017/12/vias.html`

- liberty file explanation(.lib) or timing files

`https://www.vlsisystemdesign.com/worried-about-liberty-basics-lets-start-from-ground-zero`

- STA

`https://github.com/brabect1/sta_basics_course/blob/master/doc/sta_basics_course.rst#static-timing-analysis`

- File

`https://eternallearning.github.io/phys-synthesis-and-optimization/`

- Path delay

It is refer to the time, it takes for a signal to travel from one point to another point (destination).

- Mapping

The goal of mapping is to transform a technology-independent description of a logic circuit into a technology specific reprentation

2.1 What is IR Drop?

The potential difference,or voltage drop, between two ends of a conducting wire during current flow is called IR drop (from Ohm's law: V=IR). In chips, power and ground are distributed through metal networks constituting the power delivery networks (PDNs). When current flows through the PDN, part of the voltage is dropped across the network, as per Ohm's law. This phenomenon is called IR drop.

This voltage drop in the power delivery network before the voltage reaches standard cell power pins significantly affects the chip's switching speed and performance. The same effect on the ground net leads to ground bounce, in which the nominally zero voltage rises above zero.

2.2 Why IR drop analysis is important?

- Excessive IR drops degrades the circuit performence.
- Leads to functional failure
- Some ATPG test vectors causes excessive IR drop that results yield loss.

2.3 Types of IR drop analysis

- Static IR drop
  - Based on leackage power of transistor or standard cell.
- dynamic IR drop
  - vectorless
  - vectorbased

---

3 Literature Review

A summary of books, papers, or online articles;

3.1 paper 1: IncPIRD. Fast learning-based prediction of Incremental IR drop.

- denser power grid. for ir requirements and spaser power grid. to give more routing resources
- by using superposition and partitioning techniques, it is possible to extract electrical features- given soc floorplan and PDN.
- using a mchine learning model to predict the updated static IR drop for each power node -having tap current source attached
- Traditional way of ir violation fixing:
- macro/cell block movement
- macro/cell block change
- modification of PDN to have a denser grid
- adding/moving power pads
- reduce overdesigned PDN resources by replacing with sparser PDN grids
- removing power pads
- input/given: LEF, DEF current distribution, technology file, IR drop file, power pad file of floorplan after modifications.

3.2 paper 2: ML based Dyn IR drop pridiction for ECO*

- Why ECO:
- 1.Dsign corrections and modifications
- 2.Cost and time efficiency
- 3.iterative design process
- 4.Changing requirements
- 5.Compatibility issues
- 6.Regulatory Compliance
- 7.Bug Fixes
- 8.Optimizations
- Eco plays a crucial role on physical design.This helps manage costs, save time and adapt to evolving project requirements.

In the time of ECO design sign off, many iteration needed this means long time required. Waste of reresources. Repeated dynamic IR drop simulations.

- W= maximum distance that a cell instance may move during ECO.
- cell instance/power node should be stayed in the small region every IR drop violation will be calculated in (WXW)

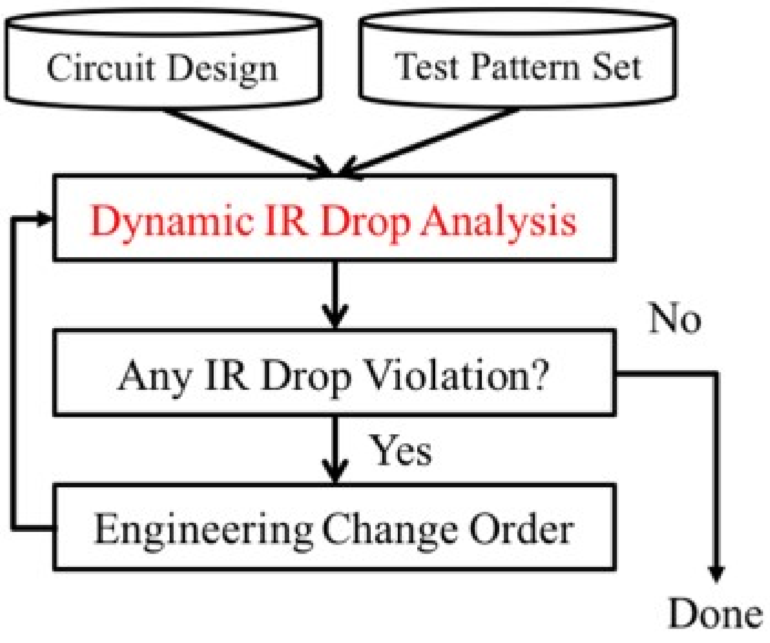


Figure 1: Traditional IR Drop signoff flow

3.3 paper 3: Vector-based Dynamic IR-drop prediction Using ML

- Motivation: Long simulation time of vectorbased dynamic IR drop analysis No such good methods to identify IR-drop risky vectors
- Goals: Predict vector-based* dynamic IR drop for all cells Identify IR-drop risky vectors quickly
- Inputs:
- Outputs:
- Technical terms IR drop risky vector:
- MIMO Chart

Input	+ output
GDSII	+ vectorprofile.rpt
tech lib .cl	+ toggle of ip
stdcell lib.cl	+ toggle of op
macros lib .cl	+ toggle of internal connection
VCD file	+ minimum arrival time
verilog file .v.gz	
def file .def	
power format .cpf	
spef file .spef	

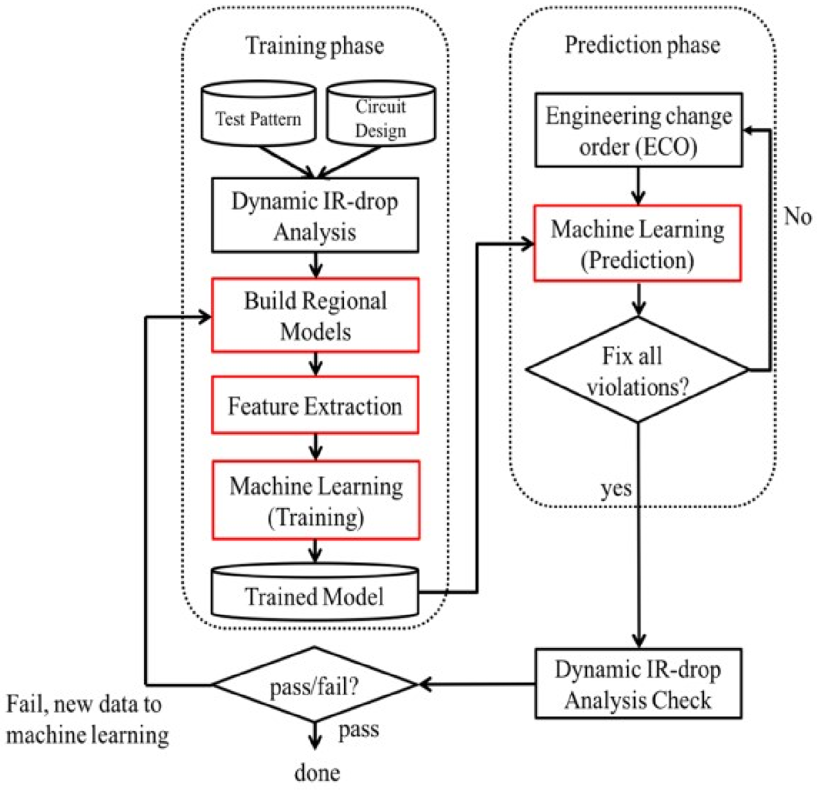
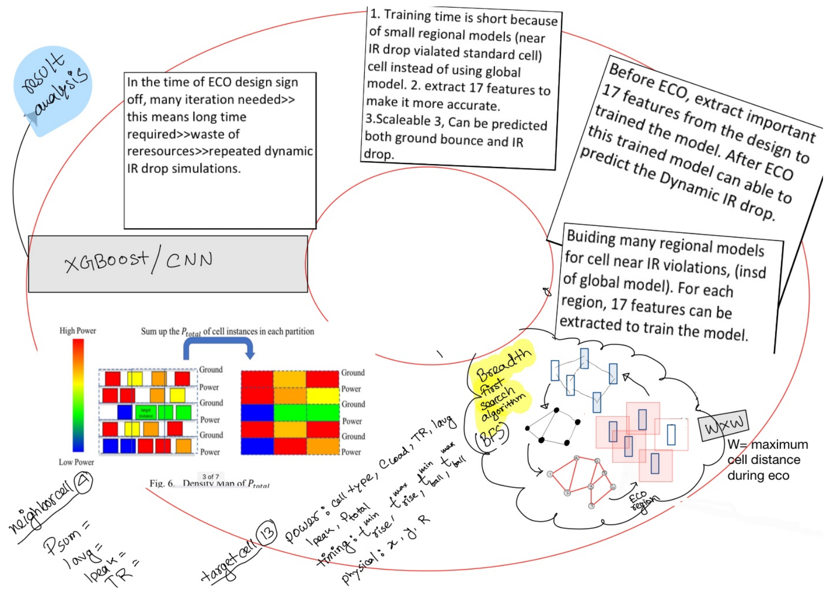


Figure 2: Model training(left) and prediction flow


In the time of ECO design sign off, many iteration needed>> this means long time required>>waste of reresources>>repeated dynamic IR drop simulations.

1. Training time is short because of small regional models (near IR drop vialated standard cell) cell instead of using global model. 2. extract 17 features to make it more accurate. 3.Scaleable 3, Can be predicted both ground bounce and IR drop.

Before ECO, extract important 17 features from the design to trained the model. After ECO this trained model can able to predict the Dynamic IR drop.

Buiding many regional models for cell near IR violations, (insd of global model). For each region, 17 features can be extracted to train the model.

result analysis
XGBoost/CNN
High Power
Low Power
Sum up the P_{total} of cell instances in each partition
Ground
Power
Fig. 6. Density Map of P_{total}
neighborcell (4)
P_{sum} =
I_{avg} =
I_{peak} =
TR =
targetcell (13)
power: cell type, C_{load}, TR, I_{avg}, I_{peak}, P_{total}
timing: t^{min}_{rise}, t^{max}_{rise}, t^{min}_{fall}, t^{max}_{fall}
physical: x, y, R
Breadth first search algorithm (BFS)
ECO region
WxW
W= maximum cell distance during eco

Figure 3: Model training(left) and prediction flow

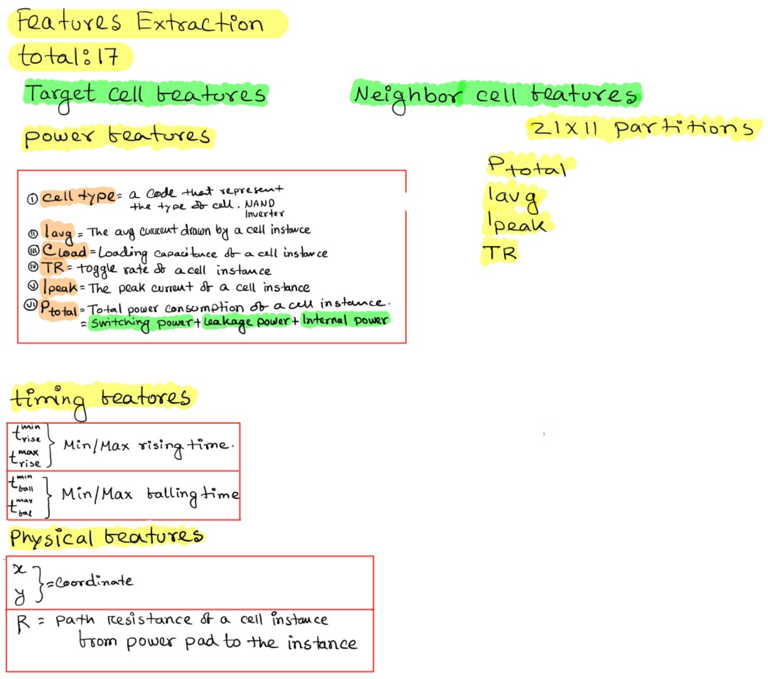
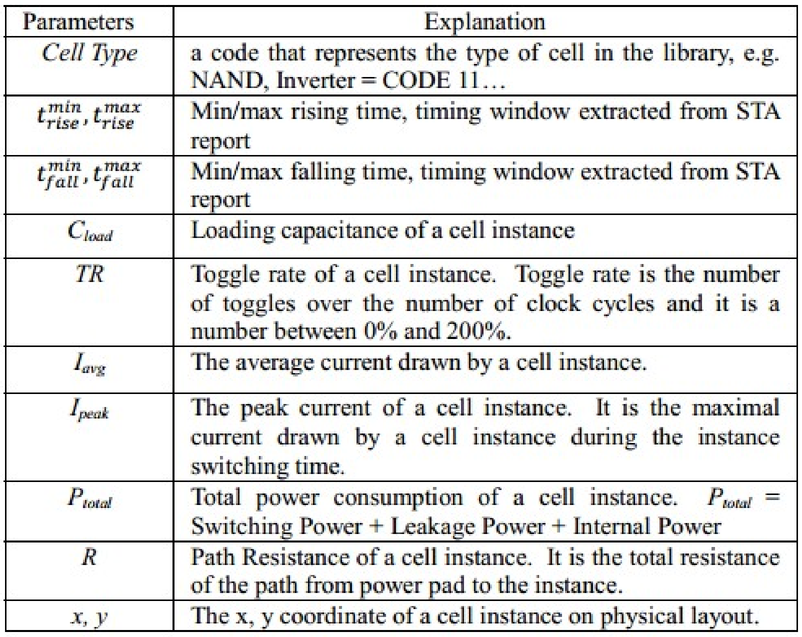

Features Extraction
total: 17
Target cell features
Neighbor cell features
21x11 partitions
P_{total}
I_{avg}
I_{peak}
TR
power features
(i) cell type = a code that represent the type of cell. NAND Inverter
(ii) I_{avg} = The avg current drawn by a cell instance
(iii) C_{load} = Loading capacitance of a cell instance
(iv) TR = toggle rate of a cell instance
(v) I_{peak} = The peak current of a cell instance
(vi) P_{total} = Total power consumption of a cell instance. = Switching power + Leakage power + Internal power
timing features
$\left.\begin{matrix} t^{min}_{rise} \\ t^{max}_{rise} \end{matrix}\right\}$ Min/Max rising time.
$\left.\begin{matrix} t^{min}_{fall} \\ t^{max}_{fall} \end{matrix}\right\}$ Min/Max falling time
Physical features
$\left.\begin{matrix} x \\ y \end{matrix}\right\}$ = Coordinate
R = path resistance of a cell instance from power pad to the instance

Figure 4: Features extraction



Parameters	Explanation
Cell Type	a code that represents the type of cell in the library, e.g. NAND, Inverter = CODE 11…
$t^{min}_{rise}, t^{max}_{rise}$	Min/max rising time, timing window extracted from STA report
$t^{min}_{fall}, t^{max}_{fall}$	Min/max falling time, timing window extracted from STA report
C_{load}	Loading capacitance of a cell instance
TR	Toggle rate of a cell instance. Toggle rate is the number of toggles over the number of clock cycles and it is a number between 0% and 200%.
I_{avg}	The average current drawn by a cell instance.
I_{peak}	The peak current of a cell instance. It is the maximal current drawn by a cell instance during the instance switching time.
P_{total}	Total power consumption of a cell instance. P_{total} = Switching Power + Leakage Power + Internal Power
R	Path Resistance of a cell instance. It is the total resistance of the path from power pad to the instance.
x, y	The x, y coordinate of a cell instance on physical layout.

Figure 5: This is the caption for the next figure link (or table)

4 Design flow

4.1 Synthesis flow

- Writing behavioral verilog code
- selection of technology and libraries or node
- setting operating environment
- setting design constraints
- how fast synthesesis circuit to run
- how big the circuit should be and other contraints
- setup speed
- setup area
- how hard compiler tries to optimize the behavairal synthesis
- commands are: create_clock: s for synthesis, et speed goal, set_clock_latancy, set_propagated_clock, set_clock_uncertainty, set_clock_transition, set_input_delay. set_output_delay, set_max_area

4.2 .lib file structure

The timing library (.lib) is an ASCII representation of the Timing, Power and Area associated with the standard cells. Characterization of cells under different PVT conditions results in the timing library (.lib). The delay calculation happens based on input transition (Slew) and the output capacitance (Load). Nowadays, CCS and ECSM models are used to characterize the library, where the calculations are based on current models which is more accurate. (In earlier days, it was NLDM model which was based on voltage calculation.) There are basically three major parts in the .lib file: Global definition Cell definition Pin definition

```
library(pso_ring_wc) {

    delay_model : table_lookup;
    in_place_swap_mode : match_footprint;

    * unit attributes *
    time_unit : "1ns";
    voltage_unit : "1V";
    current_unit : "1uA";
    pulling_resistance_unit : "1kohm";
    leakage_power_unit : "1nW";
    capacitive_load_unit (1,pf);

    slew_upper_threshold_pct_rise : 70;
    slew_lower_threshold_pct_rise : 30;
    slew_upper_threshold_pct_fall : 70;
    slew_lower_threshold_pct_fall : 30;
    slew_derate_from_library : 0.50;
    input_threshold_pct_rise : 50;
    input_threshold_pct_fall : 50;
    output_threshold_pct_rise : 50;
    output_threshold_pct_fall : 50;
    nom_process : 1;
    nom_voltage : 1.08;
    nom_temperature : 125;
    operating_conditions ( WCCOM ) {
            process : 1;
            voltage : 1.08;
            temperature : 125;
    }
    default_operating_conditions : WCCOM;

    lu_table_template(delay_template_7x7) {
          variable_1 : input_net_transition;
          variable_2 : total_output_net_capacitance;
          index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
          index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
    }
    power_lut_template(energy_template_7x7) {
          variable_1 : input_transition_time;
          variable_2 : total_output_net_capacitance;
          index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
          index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0, 1005.0,
1006.0");
    }
```

4.2.1 output files:

- .v
- .sdc
- .rep
- Gate level netlist .v or .vhd

4.3 Floorplanning

- I/O contraint file, Aspect ratio, I/O to core clearence, Flip, Abut,Double Back.

4.4 Partitioning

- Logical Groups, Clock Groups

5 Power integrity tool: Voltus

IR drop analysis using EDA tools | Practical

Voltus IC power integrity solution tool: it perform gate level power grid analysis on ASIC to determine whether the power grid will be adequate.

we can Two voltus features from INNOVUS without voltus license i. static power analysis ii. ERA with static power

but we need license(VTS-XL) for ERA with dynamic power.

Understanding License:

5.1 Goal of Voltus

- verious cell-level power
- rail analysis flows

5.1.1 Data requirements for Power and IRDrop Analysis in VOLTUS

+ library ('_') timing library
Common Timing Libraries*:
worst timing libraries*:
best timing libraries*:
worst noise libraries:
best noise libraries:
noise libraries:

+ Design ('_')
verilog netlist*:
top level netlist*:Could be anyname
timing constraint*: .sdc file
spef*:
sdf delay:

+ Physical ('_')

lef*:
def*: Could be many options like
specific def or special Net and
component def and so on.
floorplan file:
placement file:
routing file:

Low power:
soce msmv file:
power net/s: its just name of power e.g VDD or AVDD
voltage/s: 0.9V or 1.8V
Ground net/s: its just namae of ground e.g VSS or
era_vss

+ MMMC
view definition file:

A file system that organizes data and program files in a top-to-bottom structure. All modern operating systems use hierarchical file systems

5.2 votlus console

- opening and operating voltus

$voltus -no_gui

- if you want gui

$start_gui
+suspend voltus to use another console $Control -z #voltus prompt is no longer displayed

- tp return voltus session

$fg #foreground

- help command

$help read_lib #seeking help about read_lib command

- To see the entire help system

$help

- Filer hierarchy for VOLTUS tool

```
Primary Lab Data Directory Structure
+--voltus_labs
+-- design
| +-- super_filter.cpf
| +-- super_filter.def.gz
| +-- postRouteOpt_RC_wc_0.spef.gz
| +-- postRouteOpt_RC_wc_125.spef.gz
| +-- postRouteOpt_RC_bc_0.spef.gz
| +-- postRouteOpt_RC_bc_125.spef.gz
| +-- super_filter_VDD_AO.pp
| +-- super_filter_VDD_external.pp
| +-- super_filter_VSS.pp
| +-- base.sdc
| +-- postRouteOpt.design
| +-- postRouteOpt.design.dat/
| | +-- viewDefinition.tcl
| | +-- super_filter.v.gz
| | +-- super_filter.fp.gz
+-- data
| +-- gds
| | +-- pll.gds
| +-- lef
| | +-- <manyLefs>.lef
| +-- libs
| | +-- <manyLibs>.lib
| +-- netlists
| | +-- pso_ring.spi
| | +-- pso_header.spi
| | +-- pll.sp
| | +-- gsclib090.sp
| +-- qrc
| | +-- tech file
| | +-- CapTbl
| +-- models
| | +--spectre
| +-- pgv_dir
| | +-- power grid view libraries
| +-- voltus
| | +-- layermap files
+-- tcl
| +-- Tcl commands
+-- lab
+-- era
```

Lab work: Simulation practice

5.2.1 Module 3_1 Design Data importing and sanity checks

- To ensure the design is clean before running power and rail analysis.
- Different methods to importing data.
- importing innovus data into voltus
- importing 3rd party data into voltus
- Run data
- sanity checks

5.2.2 Module 4_1 Early Rail Analysis

5.2.3 Goal

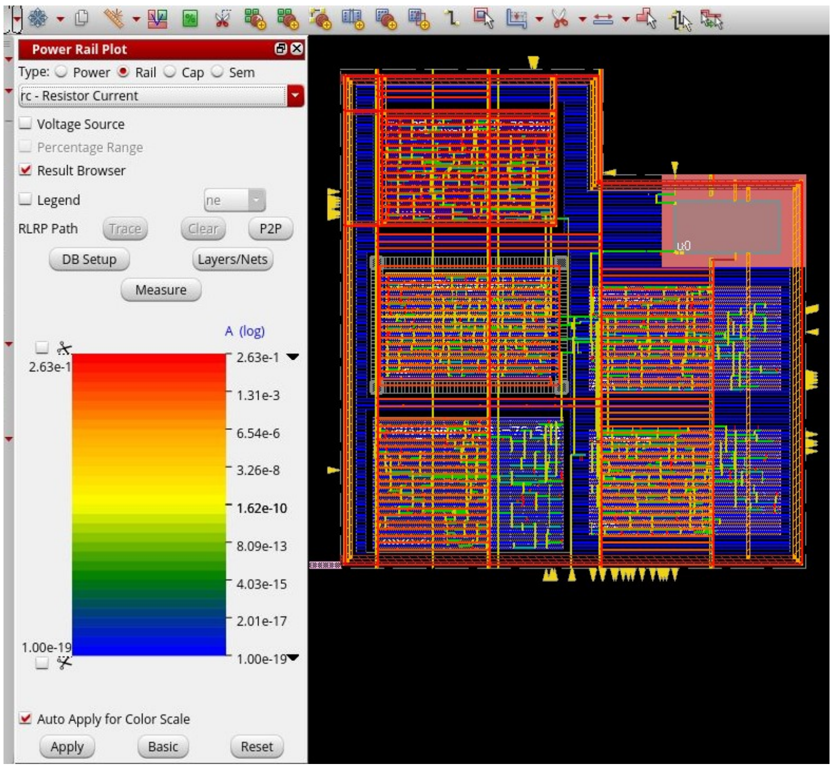
- Power grid analysis to determine the maximum current handling capacity

- Can make some asumption whether the power pad layout is sufficient or not.
- Usuaully have done before placement and routing.

5.2.4 Design details

It is a FIR filter with 8653 instances, The only macro is PLL ,PDK:cadence 90 nm.

- FIRSTLY, I configured the rail analysis from (Setup Rail Analysis) TAB
- SECONDLY, I Ran the rail analysis from (Run Rail Analysis)
- THIRDLY, Report checking using Power Rail Result



Power Rail Plot

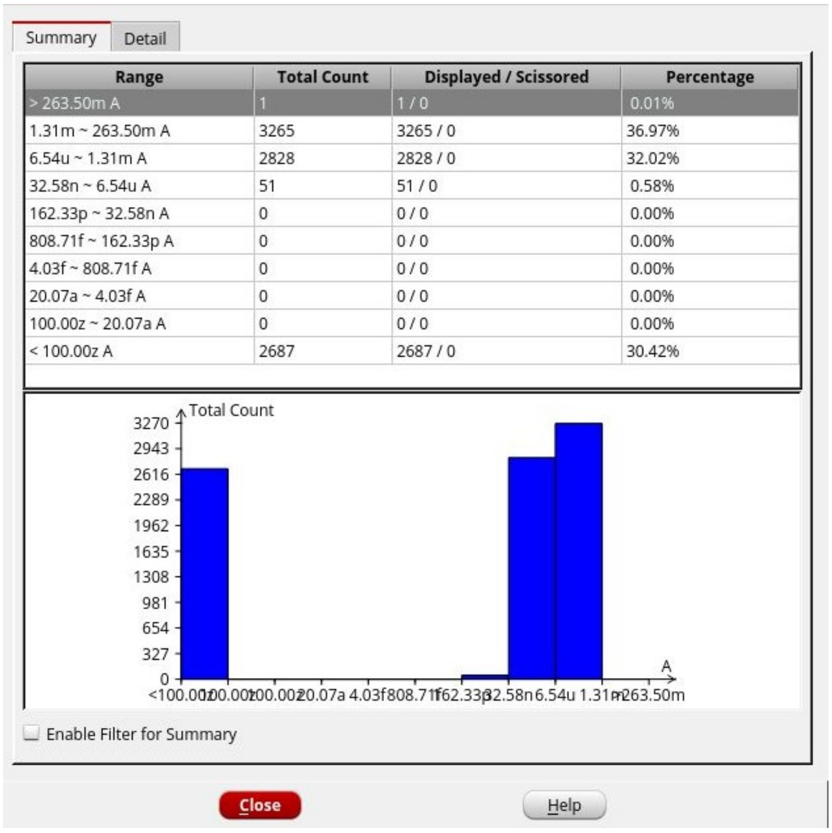
Type: Power [✓ Rail] Cap Sem

rc - Resistor Current

- [ ] Voltage Source
- [ ] Percentage Range
- [x] Result Browser
- [ ] Legend
- [x] Auto Apply for Color Scale

Figure 6: 500mA on M5: Cell instances versus current consumption plot (Resistor current)

```
set_rail_analysis_mode \ #explaining analysis mode
        -method era_static -accuracy xd \ #analysis method static
        -extraction_tech_file ../data/qrc/gpdk090_91.tch / #tech-
    nology file 90nm
        -temperature 125 -analysis_view AV_wc_on /
```



Range	Total Count	Displayed / Scissored	Percentage
> 263.50m A	1	1 / 0	0.01%
1.31m ~ 263.50m A	3265	3265 / 0	36.97%
6.54u ~ 1.31m A	2828	2828 / 0	32.02%
32.58n ~ 6.54u A	51	51 / 0	0.58%
162.33p ~ 32.58n A	0	0 / 0	0.00%
808.71f ~ 162.33p A	0	0 / 0	0.00%
4.03f ~ 808.71f A	0	0 / 0	0.00%
20.07a ~ 4.03f A	0	0 / 0	0.00%
100.00z ~ 20.07a A	0	0 / 0	0.00%
< 100.00z A	2687	2687 / 0	30.42%

Figure 7: 500mA on M5: Cell instances versus current consumption plot (Resistor current)

```
        -vsrc_search_distance 50
        -era_current_region_file VSS.curRegion
set_pg_nets \

           -net VSS -voltage 0 -threshold 0.05


set_power_data -reset
set_power_data -format ascii -scale 1 -bias_voltage 0.05 VSS.curRegion

set_power_pads -reset
set_power_pads -format xy -file ../design/super_filter_VSS.pp -
    net VSS

analyze_rail \
        -type net -output ./era_vss VSS
```

5.2.5 method

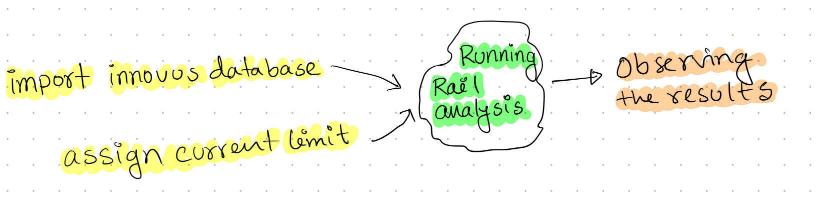


Figure 8: This is the caption for the next figure link (or table)

6 Benchmark circuit

I was trying to compare my design with benchmark circuit from Literature review. git-hub site file list.

- b19.bench
- b19.blif
- b19.edf
- b19.fau
- b19.vhd
- b19_C.bench
- b19_C.blif
- b19_C.edf
- b19_C.fau

- b19 benchmark circuits(Viper and 80386 microprocessor)

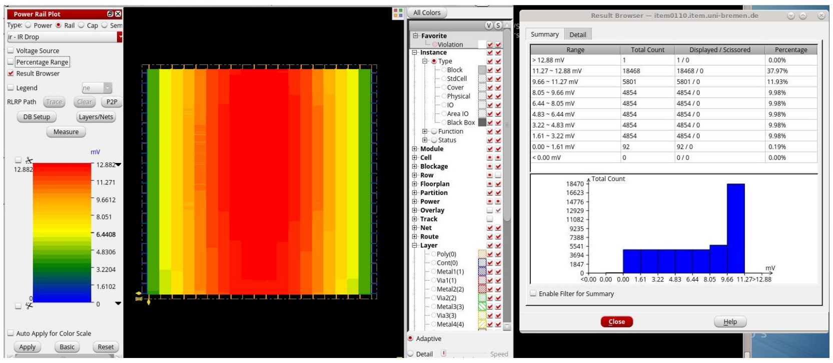
`https://www.cerc.utexas.edu/itc99-benchmarks/polibench.pdf`

7 Simulation: RISCV & DNN Accelerator

7.1 DNN Accelerator GEMMINI IR DROP ANALYSIS.

Steps

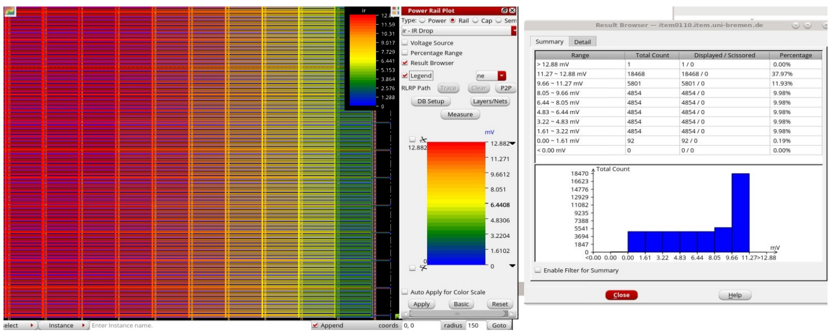
- Synthesize the RTL using PDK45
- Used PDK45 lib and lef file for flooePlan
- Extract tech library view and std library view files using QRC tech file (which I found in PDK45)
- In floorPlan, I used extend to boundary for both vss and vdd stripe that will automatically implement VDD and VSS physical pad cell by default.
- Early Rail Analysis (ERA) of GEMMINI tcl script
- Here is the report and IR drop distributions for VSS/ground bounce
- **1. Early Rail Analysis_VSS_rail**



Result Browser — item0110.item.uni-bremen.de

Range	Total Count	Displayed / Scissored	Percentage
> 12.88 mV	1	1 / 0	0.00%
11.27 ~ 12.88 mV	18468	18468 / 0	37.97%
9.66 ~ 11.27 mV	5801	5801 / 0	11.93%
8.05 ~ 9.66 mV	4854	4854 / 0	9.98%
6.44 ~ 8.05 mV	4854	4854 / 0	9.98%
4.83 ~ 6.44 mV	4854	4854 / 0	9.98%
3.22 ~ 4.83 mV	4854	4854 / 0	9.98%
1.61 ~ 3.22 mV	4854	4854 / 0	9.98%
0.00 ~ 1.61 mV	92	92 / 0	0.19%
< 0.00 mV	0	0 / 0	0.00%

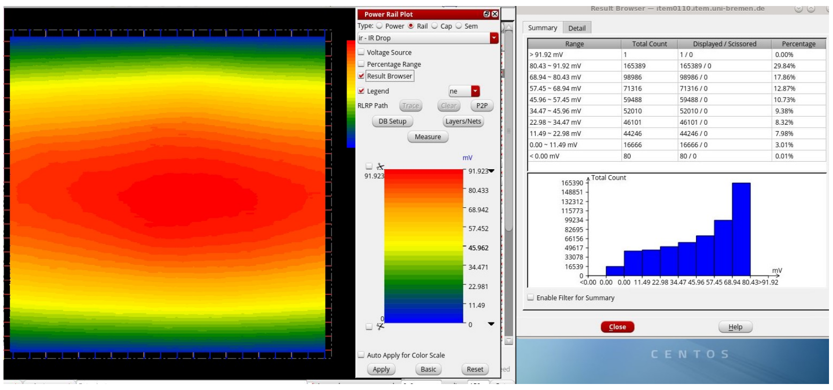
- Report: IR drop for VSS Report



Result Browser — item0110.item.uni-bremen.de

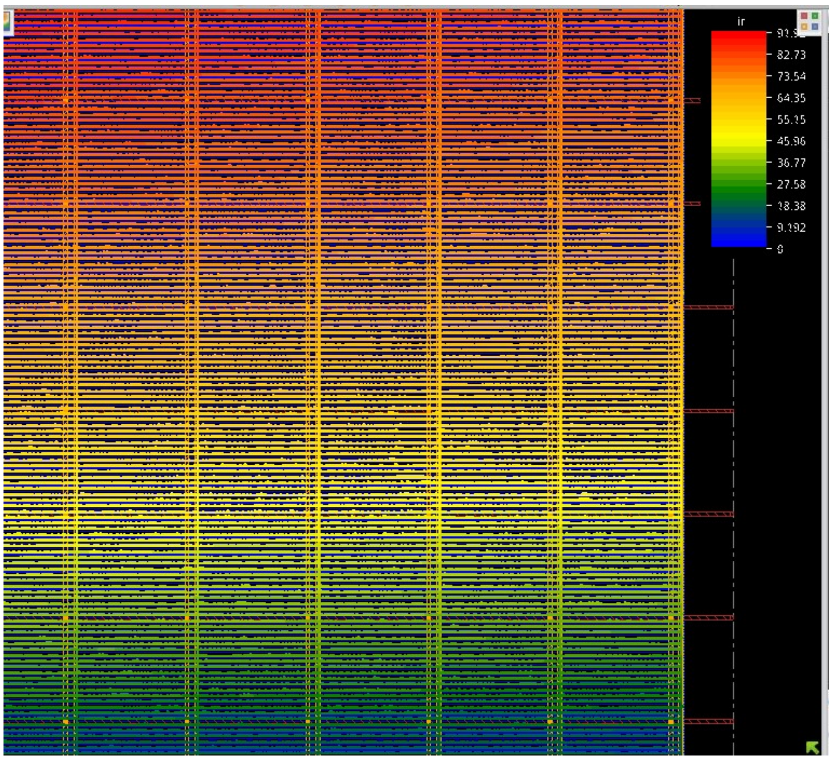
Range	Total Count	Displayed / Scissored	Percentage
> 12.88 mV	1	1 / 0	0.00%
11.27 ~ 12.88 mV	18468	18468 / 0	37.97%
9.66 ~ 11.27 mV	5801	5801 / 0	11.93%
8.05 ~ 9.66 mV	4854	4854 / 0	9.98%
6.44 ~ 8.05 mV	4854	4854 / 0	9.98%
4.83 ~ 6.44 mV	4854	4854 / 0	9.98%
3.22 ~ 4.83 mV	4854	4854 / 0	9.98%
1.61 ~ 3.22 mV	4854	4854 / 0	9.98%
0.00 ~ 1.61 mV	92	92 / 0	0.19%
< 0.00 mV	0	0 / 0	0.00%

- **2. Early Rail Analysis_VDD_rail**



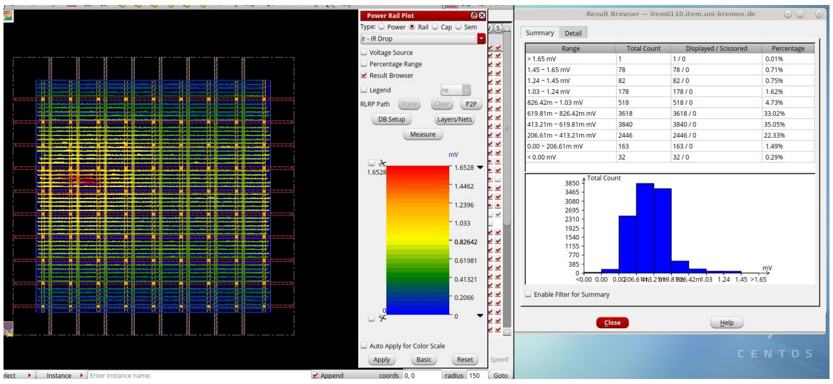
Range	Total Count	Displayed / Scissored	Percentage
> 91.92 mV	1	1 / 0	0.00%
80.43 ~ 91.92 mV	165389	165389 / 0	29.84%
68.94 ~ 80.43 mV	98986	98986 / 0	17.86%
57.45 ~ 68.94 mV	71316	71316 / 0	12.87%
45.96 ~ 57.45 mV	59488	59488 / 0	10.73%
34.47 ~ 45.96 mV	52010	52010 / 0	9.38%
22.98 ~ 34.47 mV	46101	46101 / 0	8.32%
11.49 ~ 22.98 mV	44246	44246 / 0	7.98%
0.00 ~ 11.49 mV	16666	16666 / 0	3.01%
< 0.00 mV	80	80 / 0	0.01%

- Report: IR drop for VDD Report



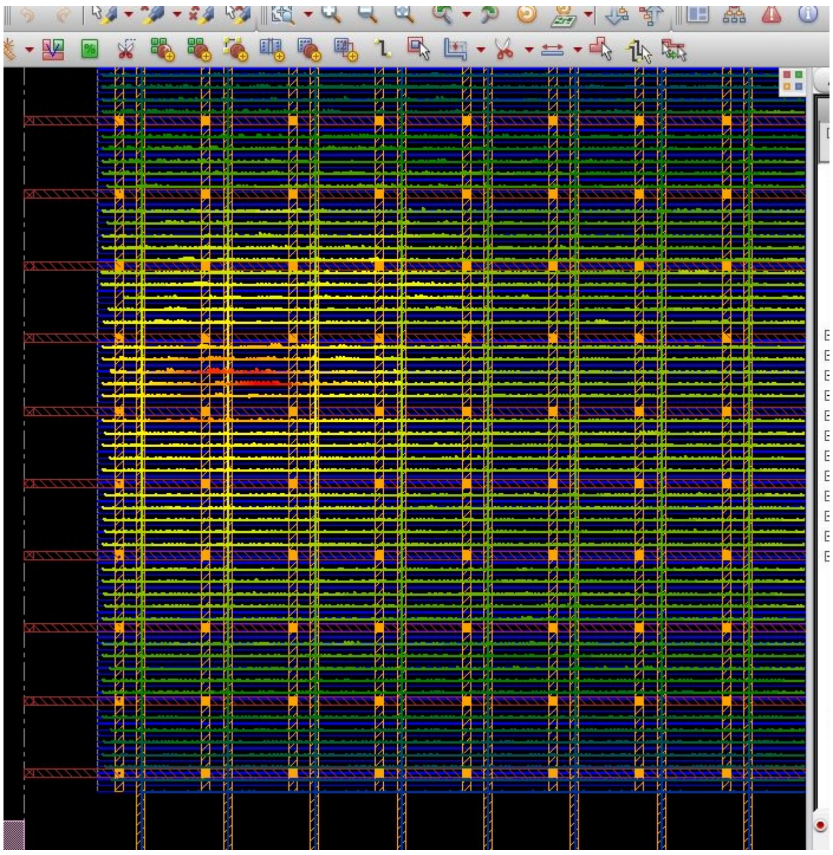
7.1.1 Early Rail Analysis (ERA) of rocket core

- **1. Early Rail Analysis_VDD_rail**



Range	Total Count	Displayed / Scissored	Percentage
> 1.65 mV	1	1 / 0	0.01%
1.45 ~ 1.65 mV	78	78 / 0	0.71%
1.24 ~ 1.45 mV	82	82 / 0	0.75%
1.03 ~ 1.24 mV	178	178 / 0	1.62%
826.42m ~ 1.03 mV	518	518 / 0	4.73%
619.81m ~ 826.42m mV	3618	3618 / 0	33.02%
413.21m ~ 619.81m mV	3840	3840 / 0	35.05%
206.61m ~ 413.21m mV	2446	2446 / 0	22.33%
0.00 ~ 206.61m mV	163	163 / 0	1.49%
< 0.00 mV	32	32 / 0	0.29%

- Report: IR drop for VDD Report



8 Personal Notes

8.1 Innovus Guide

- CPF sample
- PnR master script sample

- What is follow pin in VLSI physical design? In VLSI physical design, a follow pin is a special type of pin used to specify the routing direction of a net or signal. The follow pin is used to guide the routing of a net, making sure that it follows a specific direction or path.

Follow pins are often used in high-speed digital circuits, where the routing of signals can significantly affect the performance of the circuit. Designers may ensure that the signal takes the best path by setting the routing direction of a net using a follow pin, lowering the chance of crosstalk and other problems.
• innovus.cmd – Contains list of commands executed during the session. This file can be used to create scripts to automate the execution of the commands and learn what text commands correspond to commands executed through the GUI. • innovus.log – Contains basic information output from the executed commands. The commands in the file are preceded with in the file. • innovus.logv – Similar to innovus.log but contains a more verbose amount of output. Useful for debugging

8.2 Notes

- CPU Benchmarks
  - Geekbench
  - Cinebench
- GPU Benchmarks
  - 3DMark
  - Unigine Heaven and Valley Benchmarks
- Web Browser Benchmarks
  - Octane and Kraken
- There are two types of library file in VLSI
  - Technology library (e.g
  - Cell library (eg nand, inverter)
- EDIF file is a file format for transferring design information between EDA vendors and EDA vendors and IC vendors
- Berkeley Logic Interchange Format (BLIF)design information
- Before starting the main design work check "Your INNOVUS have the required license for node tech,

maximum cell instance number and so on, Because innovus has a lot of different versions.

- definition file
- design exchange file
- chache parameter file

- gz > GNU zip file
- standard parasitic exchange format spef file
- power net power file .ppfile
- sdc » synopsys design constraints file

9 Question

- Which pdk I should use in b19?
- not finding GPDK file TSMC 40nm and 65nm
- need synthesis tool , design vision
- Could you share me the lab file
- UPF file is same as CPF file? both of the files are providing power details of chip.
- definition.tcl or definition.view both are can be use as mmmc_file for view analysis
- Why we need to set init_gnd and init_pwr VDD and init_gnd VSS in the time of design importing since we will provide cpf file (UPF file format can also be converted into cpf file, so chill both are kind of same) during imporing design.

cpf: It is an optional file for importing power domain configuration; Define different kind of power cell as like isolation cell and level shifter cell and different rule, differnt power mode, always on cell, power clocking gate and so on. and init power and gnd in the time of importing design is necessary this is how we initiate create power ring and stripes.

9.1 Task

9.1.1 TODO [#05] [DONE] Genus synthesis -> .v and .sdc file is ready

9.1.2 TODO [#07] [DONE] Get for importing design

9.1.3 TODO [#10] [DONE] Write floorplan script in .tcl

9.1.4 TODO [#20] [DONE] Write placement script in -tcl

9.1.5 TODO [#30] [DONE] Get ready for CTS script

9.1.6 TODO [#40] [DONE] script for Routing

9.1.7 TODO [#55] [DONE] script for Optimization

9.1.8 TODO [#65] [DONE] Def exporting script

9.1.9 TODO [#80] [DONE] save design in .enc format

Abdul Aziz$_{P}rojectSoSe23$