PHYSICAL DESIGN PROJECT

WEEK -1

Author : Preethi grace Manthena
Date : 04-03-2025
Batch : G3_Integrated VLSI Design
Project Mentor: Sri. Veeramani sir, (Staff Engineer - Synopsys)

20-02-2025

Tool installation - Installed openlane

https://openlane.readthedocs.io/en/latest/getting_started/installation/installation_ubuntu.html

I have installed the openlane tool by taking this as reference.

23-02-2025

Linux command practice

1. Basic commands
   - ls list : ls -ltr
     - > l - long list
     - > t - timestamp
     - > r - reverse order
   - rm : rm <filename>
   - pwd : pwd - prints current working directory location
   - mkdir : mkdir <filename> -make directory
   - rmdir : rmdir <filename> -remove directory
   - cd : cd <filename> -change directory
   - gedit : gedit <filename> -text editor
   - vi : vi <filename> -text editor
   - cp : copy -cp <file1> <file2>
   - vidiff : vidiff <file1> <file2> -difference between the files
   - man : man <cmd name> - manual page

2. Vi editor

   Vi <filename> - opening
   i inserting
   Esc -escape
   :u -undo

:w - saving the file
:wq - save and quit
:q - quit
:wq! - save and quit forcefully
:gg - go to top of the file
:shift + gg - go to bottom of the file
/<search word pattern>
:%s/<word1>/<word2> - replacing word 1 with word 2
:g/<word1>/g -delete the line which contains word1

3. Chmod

chmod 756 <filename>
Digit 1 -permissions for owner
Digit 2 - permission for group
Digit 3 - permission for others

Read - 'r' : 4
Write - 'w' : 2
Execute - 'x' : 1

The digits represent the sum of these values of r,w,x.

4. Grep

grep <pattern> < filename> : prints the lines which contains the pattern in file
grep -i <pattern> <filename> : ignores the case of pattern
grep -v <pattern> <filename> : prints non matching lines
grep -n <pattern> <filename> : prints line number before ines printed
grep -w <pattern> <filename> : prints only exact matching pattern
grep -c <pattern> <filename> : counts the number of occurrences of the provided pattern

5. awk

This cmd read the file line by line
Each line is called as record
Each line can be divided into fields

Some built in
NR- number of records
NF - Number of field
FS - Field separator, default space or tab
RS - Record separator, default is new line

awk '{print $1 $2}' <file_name> #prints 2 columns
awk '{print $1 "and" $2}' <filename> #prints 2 column with separating by and
awk '{print $NF}'<filename> #prints the number of field

24-02-2025

Openlane environment learning :

- Directory structure explored

The files after the run of the design will saves in reports folder and the log files,

- We can avail these files in the results folder and reports folders.

TCL task:

1. Wire length calculation

In VLSI design, **wire length** refers to the total distance covered by interconnections between different components of a circuit. These wires are responsible for carrying **signals and power** across the chip.

- **Clock Wire Length**: The total length of interconnections dedicated to distributing the **clock signal** across the design. A **longer clock wire length** can lead to **clock skew** and increased **power consumption**.
- **Data Wire Length**: The total length of interconnections used for **transmitting data** between circuit components. Longer data wires can **increase signal delay and power consumption**, impacting overall circuit performance.

Manual Calculations

Wire length is measured in **micrometers (µm)**.

Found in: ~/OpenLane/designs/spm/runs/openlane_test/reports/routing

- Total wire length of design = 6188.845 µm
- Clock wire length = 767.72 µm
- Data wire length = 5421.125 µm

```
preethi-grace@HP-245-G7-Notebook:~/OpenLane/designs/spm/runs/openlane_test/reports$ cd routing
preethi-grace@HP-245-G7-Notebook:~/OpenLane/designs/spm/runs/openlane_test/reports/routing$ ls -ltr
total 1252
-rw-r--r-- 1 preethi-grace preethi-grace      0 Feb 24 15:36 drt.drc
-rw-r--r-- 1 preethi-grace preethi-grace    864 Feb 24 15:36 23-drt_metrics.json
-rw-r--r-- 1 preethi-grace preethi-grace    176 Feb 24 15:36 drt.klayout.xml
-rw-r--r-- 1 preethi-grace preethi-grace   5426 Feb 24 15:36 24-wire_lengths.csv
-rw-r--r-- 1 preethi-grace preethi-grace   5652 Feb 24 15:36 16-rsz_design_sta.checks.rpt
-rw-r--r-- 1 preethi-grace preethi-grace 186791 Feb 24 15:36 16-rsz_design_sta.min.rpt
-rw-r--r-- 1 preethi-grace preethi-grace 206392 Feb 24 15:36 16-rsz_design_sta.max.rpt
-rw-r--r-- 1 preethi-grace preethi-grace    307 Feb 24 15:36 16-rsz_design_sta.skew.rpt
-rw-r--r-- 1 preethi-grace preethi-grace    753 Feb 24 15:36 16-rsz_design_sta.summary.rpt
```

Using TCL :-

```
set filename "24-wire_lengths.csv"
set file_handle [open "$filename" r]
set clk_sum 0.0 ;
set data_sum 0.0 ;
set total_sum 0.0  ;

gets $file_handle ;
while { [gets $file_handle data ] >=0} {
        set net [lindex [split $data ","] 0];
        set length_um [lindex [split $data ","] 1];
        set total_sum [expr {$total_sum + $length_um}];
        if {[string match "*clk" $net]} {
                set clk_sum [expr {$clk_sum + $length_um}];
        }
        set data_sum [expr {$total_sum - $clk_sum}];
}
close $file_handle
puts "Total clock wire length : $clk_sum um "
puts "Total data wire length : $data_sum um"
puts "Total wire length in the design: $total_sum um"
```

We run tcl files using tclsh cmd.
The `tclsh` (TCL Shell) command is used to **execute TCL scripts** in a command-line environment.

Output:-

```
preethi-grace@HP-245-G7-Notebook:~/Downloads/SURE TRUST/project$ tclsh clk_length.tcl
Total clock wire length : 767.72 um
Total data wire length : 5421.125000000008 um
Total wire length in the design: 6188.845000000008 um
```

2) Find the worst IR drop and the % degradation

IR drop (voltage drop) occurs due to the resistance and current flow in power and ground interconnects, leading to a reduction in supply voltage at different points in the design. It can affect circuit performance, timing.

Manual calculation:
This file is available in reports folder:
/home/preethi-grace/OpenLane/designs/spm/runs/openlane_test/reports/signoff

For VPWR
Worst Case IR drop = 1.62* 10^-4 V

For VGND
Worst Case IR drop = 1.09*10^-4 V

$$\% \text{ degradation} = \frac{worst\ case\ IR\ drop}{Voltage\ supplied} \text{ X } 100$$

$$= \frac{1.62*10^{-4}}{1.8} \text{ X } 100$$

$$= 0.009\ \%$$

This means that **only 0.009% of the supplied voltage is lost** due to IR drop across wire, which is acceptable.

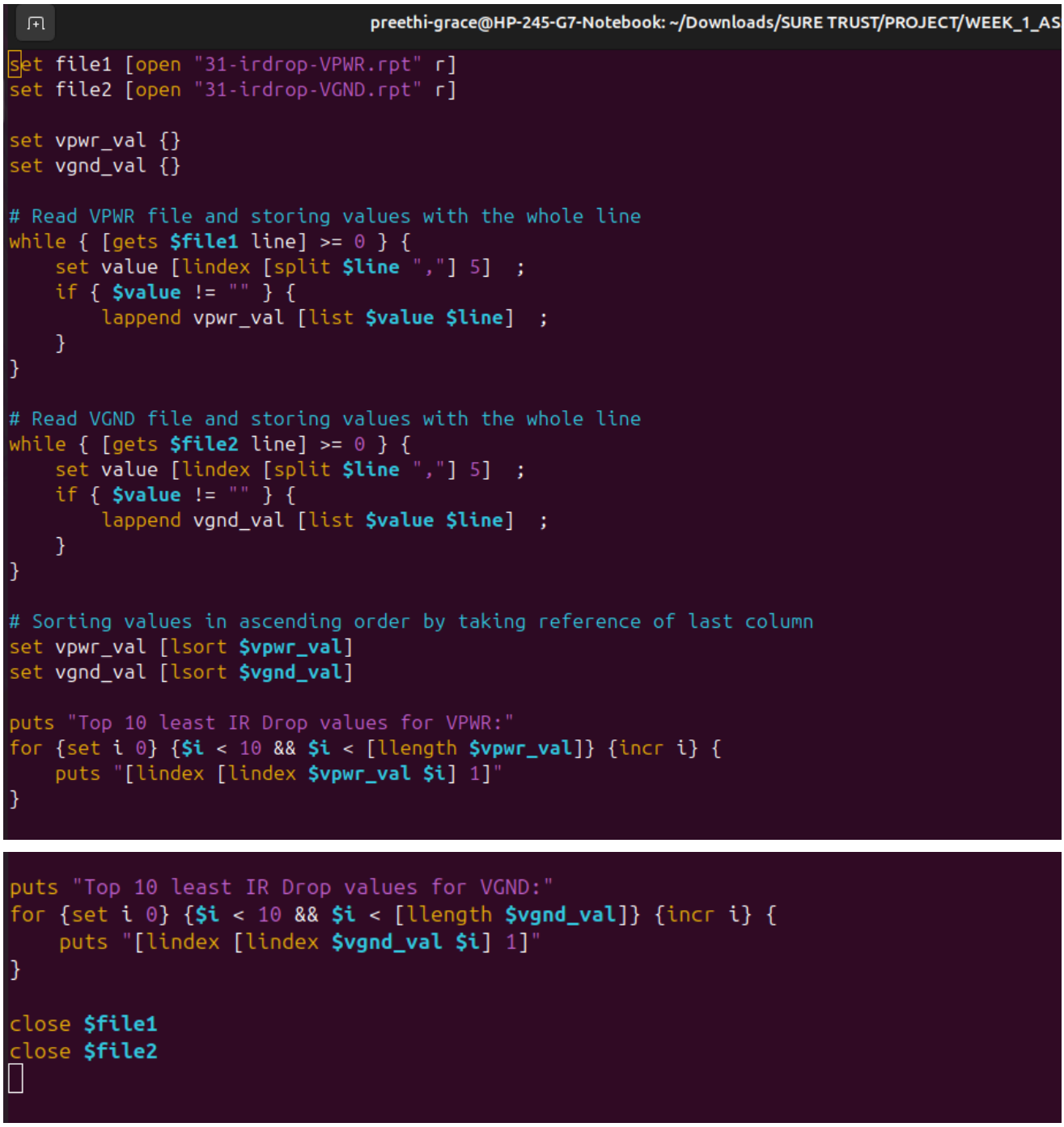
```
preethi-grace@HP-245-G7-Notebook: ~/OpenLane/designs/spm/runs/openlane_test/logs
Using 1e-09 for time...
Using 1e+00 for voltage...
Using 1e-03 for current...
Using 1e-09 for power...
Using 1e-06 for distance...
[INFO]: Reading ODB at '/openlane/designs/spm/runs/openlane_test/results/routing/spm.odb'…
Reading design constraints file at '/openlane/designs/spm/src/spm.sdc'…
[INFO]: Setting RC values...
[INFO PSM-0040] All shapes on net VPWR are connected.
########## IR report #################
Net              : VPWR
Corner           : Typical
Supply voltage   : 1.80e+00 V
Worstcase voltage: 1.80e+00 V
Average voltage  : 1.80e+00 V
Average IR drop  : 2.54e-05 V
Worstcase IR drop: 1.62e-04 V
Percentage drop  : 0.01 %
######################################
[INFO PSM-0040] All shapes on net VGND are connected.
########## IR report #################
Net              : VGND
Corner           : Typical
Supply voltage   : 0.00e+00 V
Worstcase voltage: 1.09e-04 V
Average voltage  : 2.43e-05 V
Average IR drop  : 2.43e-05 V
Worstcase IR drop: 1.09e-04 V
Percentage drop  : 0.01 %
######################################
(END)
```

3) Find which cell receives less voltage?

Some cells may receive less voltage due to IR drop, affecting their performance. Identifying these cells helps in power integrity analysis and optimization.

These cells can be modified by adding buffers in the path or decoupling capacitors for improving power delivery to circuit components.

TCL script for identifying the cells which are receiving less voltage.



Output of this script :-

```
preethi-grace@HP-245-G7-Notebook:~/Downloads/SURE TRUST/project$ tclsh leastirdrop.tcl
Top 10 least IR Drop values for VPWR:
FILLER_0_22_181,VPWR,li1,89.2400,73.4400,1.799838
FILLER_0_23_169,VPWR,li1,86.0200,73.4400,1.799838
FILLER_0_23_181,VPWR,li1,89.2400,73.4400,1.799838
PHY_EDGE_ROW_22_Right_22,VPWR,li1,90.3900,73.4400,1.799838
PHY_EDGE_ROW_23_Right_23,VPWR,li1,90.3900,73.4400,1.799838
clkbuf_3_7__f_clk,VPWR,li1,84.1800,73.4400,1.799838
FILLER_0_23_162,VPWR,li1,81.4200,73.4400,1.799849
TAP_TAPCELL_ROW_23_136,VPWR,li1,83.0300,73.7460,1.799849
_129_,VPWR,li1,79.1200,73.4400,1.799858
_128_,VPWR,li1,76.8200,73.4400,1.799868
Top 10 least IR Drop values for VGND:
FILLER_0_0_109,VGND,li1,56.3500,10.8800,0.000000
FILLER_0_0_113,VGND,li1,60.2600,10.8800,0.000000
FILLER_0_0_125,VGND,li1,65.7800,10.8800,0.000000
FILLER_0_0_137,VGND,li1,69.2300,10.8800,0.000000
FILLER_0_0_141,VGND,li1,71.3000,10.8800,0.000000
FILLER_0_0_145,VGND,li1,72.4500,10.8800,0.000000
FILLER_0_0_15,VGND,li1,15.1800,10.8800,0.000000
FILLER_0_0_162,VGND,li1,81.4200,10.8800,0.000000
FILLER_0_0_169,VGND,li1,86.0200,10.8800,0.000000
FILLER_0_0_181,VGND,li1,89.2400,10.8800,0.000000
```

4. Levels of Logic

Levels of logic represent the number of sequential and combinational logic stages between a start point and an endpoint in a design. More logic levels can introduce higher delays, affecting timing closure and performance.

The number of combinational circuits in between the launch flip flop and capture flipflop

Simply, Logic levels means the levels of combinational logic between two timing end points. E.g. a LUT is one level of logic. If there are 3 LUTs between two FFs, we say there are three logic levels.
LUT- (lookup table): a table that determines what the output is for any given input(s)
We mention it as a truth table in terms of VLSI.

```
# Opening the report file
set file_name [open "18-rsz_timing_sta.max.rpt" r]

# Printing table header row
puts "Startpoint\tEndpoint\tLevels_of_logics\tSlack"

# Initializing variables
set startpoint ""
set endpoint ""
set slack 0
set levels 0
set sum 0
set printed_paths [dict create]
# Reading file line by line
while {[gets $file_name line] != -1} {
    if {[regexp {Startpoint:\s+(\S+)} $line _ startpoint]} continue
    if {[regexp {Endpoint:\s+(\S+)} $line _ endpoint]} continue
    if {[regexp {(.*slack\s+\(\S+\))} $line _ slack]} continue
    # Detect logic level count
   if {[string match "*sky130_fd_sc_*" $line] && ![string match "*clkbuf*" $line] && ![string match "*dfrtp*" $line]} {
        incr levels
    }

        #as we have the levels counted twice, so dividing them by 2
        if {[regexp {data arrival time} $line]} {
        set key "$startpoint-$endpoint"
        set levels [expr {$levels / 2}]  ;
        # Printing only if not already printed
        if {![dict exists $printed_paths $key]} {
        puts [format "%-17s %-17s %-17s %s" $startpoint $endpoint $levels [string trim $slack]]
        }
        set sum [expr {$sum + $levels}]
        set levels 0

}}
puts "The total number of logic levels : $sum"
close $file_name
```

Output : -

```
preethi-grace@HP-245-G7-Notebook:~/Downloads/SURE TRUST/PROJECT/WEEK_1_ASSIGNMENT$ vi level.tcl
preethi-grace@HP-245-G7-Notebook:~/Downloads/SURE TRUST/PROJECT/WEEK_1_ASSIGNMENT$ tclsh level.tcl
Startpoint      Endpoint        Levels_of_logics        Slack
_300_             _300_             2                0
_274_             _274_             2                9.15   slack (MET)
_314_             _314_             2                9.15   slack (MET)
_270_             _270_             2                9.16   slack (MET)
_294_             _294_             2                9.16   slack (MET)
_326_             _326_             2                9.16   slack (MET)
_302_             _302_             2                9.16   slack (MET)
_324_             _324_             2                9.16   slack (MET)
_300_             _301_             3                9.16   slack (MET)
_304_             _304_             2                9.16   slack (MET)
_312_             _312_             2                9.16   slack (MET)
_328_             _328_             2                9.16   slack (MET)
_282_             _282_             2                9.16   slack (MET)
_316_             _316_             2                9.17   slack (MET)
_284_             _284_             2                9.17   slack (MET)
_308_             _308_             2                9.17   slack (MET)
_306_             _306_             2                9.17   slack (MET)
_318_             _318_             2                9.17   slack (MET)
_320_             _320_             2                9.17   slack (MET)
_330_             _330_             2                9.17   slack (MET)
_272_             _272_             2                9.17   slack (MET)
_290_             _290_             2                9.17   slack (MET)
_310_             _310_             2                9.17   slack (MET)
_286_             _286_             2                9.17   slack (MET)
_294_             _295_             3                9.17   slack (MET)
_288_             _288_             2                9.17   slack (MET)
```

```
_278_           _278_           2               9.18   slack (MET)
_304_           _305_           3               9.18   slack (MET)
_284_           _285_           3               9.18   slack (MET)
_274_           _275_           3               9.19   slack (MET)
_272_           _273_           3               9.19   slack (MET)
_282_           _283_           3               9.19   slack (MET)
_270_           _271_           3               9.19   slack (MET)
_308_           _309_           3               9.19   slack (MET)
_318_           _319_           3               9.20   slack (MET)
_328_           _329_           3               9.20   slack (MET)
_326_           _327_           3               9.20   slack (MET)
_310_           _311_           3               9.20   slack (MET)
_324_           _325_           3               9.20   slack (MET)
_320_           _321_           3               9.20   slack (MET)
_286_           _287_           3               9.20   slack (MET)
_296_           _297_           3               9.20   slack (MET)
_316_           _317_           3               9.20   slack (MET)
_312_           _313_           3               9.20   slack (MET)
_298_           _299_           3               9.20   slack (MET)
_276_           _277_           3               9.20   slack (MET)
_280_           _281_           3               9.21   slack (MET)
_278_           _279_           3               9.21   slack (MET)
_330_           _331_           3               9.21   slack (MET)
_322_           _323_           3               9.21   slack (MET)
_268_           _269_           2               9.21   slack (MET)
_292_           _293_           3               9.21   slack (MET)
_288_           _289_           3               9.22   slack (MET)
_268_           _268_           1               9.22   slack (MET)
The total number of logic levels : 158
```