Productivity Tools for IC Design (EE2510). Oct-Nov 2025. Exercise Set 2.

Here is the dataset you can use in the following exercises. This contains some files generated in the IC design work flow for a RISC-V microprocessor. These contain design output as well as reports (ending with .rpt).
You also need to clone the git repository in your laptop. Use:
$ git clone https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts
[If git is not installed on your system, use
$ sudo apt install git
Which should install it.]

Section 1 Bash Shell

Preliminary exercises, try to do it on a single line using pipes.
1. In the current folder list the top 3 largest files.
2. From a file like this:

```
Name,Subject,Score
Alice,Math,92
Bob,Physics,87
Carol,Math,95
```

Create several files each corresponding to a particular subject which list the marks scored by the students in those subjects (e.g. math.txt should have two lines: Alice 92 [newline] Carol 95).
3. Convert all dates in format YYYY-MM-DD to DD/MM/YYYY in a text file.
4. From the output of the command
$ ps aex
Select only the PID and process name to be displayed.
5. Given a text file (e.g., article.txt), print the top 10 most frequent words, ignoring case and punctuation. Use a pipeline that can be done on a single line.
6. From the file, which lists the time taken for 3 builds:

```
build1: 3.2s
build2: 5.1s
build3: 2.9s
```

Calculate the total time taken by all the builds.

Section 2 OpenRoad related

These questions are related to the test files provided above:
1. Write a shell pipeline to find all the slacks in the “6_finish.rpt” file and display them in sorted order. Write a script to find out start and end points of all the slacks which were violated in 6_finish.rpt.

2. From the 6_final.def file, list all the ROUTED clauses and find out the total length of wiring used (Manhattan distance). Figure out what the numbers in brackets mean.

3. Find out the total area, power and worst slack for the design corresponding to the files provided in the .zipped folder.
4. (i) From the file 6_final.v, which contains the final netlist with cells from the PDK (physical design kit) of SKY130_HD, find all those cells. Count how many cells of each types were used. Now using the PDK file kept in “OpenROAD-flow-scripts/flow/platforms/sky130hd/lib” find the total leakage power and capacitance for all the cells.
The output data should be presented in this format:
Cell_Name	Total_Capacitance	Total_leakage
[the column entries are separated by tab character]
There seem to be 16872 cell entries in that .v file, many of which refer to the same cell used in different locations. You have to report one type of cell only once in the final output.
You have to use bash scripting. You may need to use grep, cut, uniq, sort, tr etc.

Section 3 Regular Expressions

Use grep -E for testing your regular expressions. (grep extended)
1. (i) Use regular expression to select cat and rat from this file:

```
cat
bat
Rat is bad bad bad sobad sobad
mat 12.45.124.0
Dog  12/3/2025 90.02.30.50000
apple
Orange is good 14-5-2021  10000.43.200000.100.12
egg
```

(ii) write regex to match lines which start and end with a vowel.
(iii) select all the words which start with capital letters.
(iv) select the number displays which follow dotted notation, e.g. xx.xx.xx.x……
(v) select dates.
(vi) find duplicate words.
(vii) find all IP addresses (note that each part of the IP address must be less than 256).
(viii) find duplicate words which are only 3 characters long.

Section 4 Make

1. Create a Makefile from scratch, by pasting the following into a new file. Name the file “Makefile”.

```
out:
      echo “hi”
```

Run this file using
$ make -f Makefile
It produces an error. This is due to spaces used in front of “echo”. Replace all those spaces with a single tab character and run it again.

2. Use the following in Makefile

Shell
```
all: synth place route

SCRIPT = ./flow.tcl
DESIGN ?= gcd
.PHONY: run
run:
    $(SCRIPT) -design $(DESIGN) > $(DESIGN)_flow.log
synth:
       echo "Running Yosys synthesis" > synth.log
place: synth
       echo "Running placement" > place.log
route: place
       echo "Running routing" > route.log
reports: synth.log place.log route.log
       grep "Running" *.log > summary.txt
.PHONY: clean
clean:
       rm -f ./synth.log ./place.log ./route.log
```
And try to run all different rules, explain what is happening.

3. (i) Write a makefile which takes a user specified cell and lists all its variants available in the library file (OpenROAD-flow-scripts/flow/platforms/sky130hd/lib/sky130_fd_sc_hd__tt_025C_1v80.lib) .
E.g.
$ make cell inv
This should list:

```
sky130_fd_sc_hd__inv_1
sky130_fd_sc_hd__inv_12
sky130_fd_sc_hd__inv_16
sky130_fd_sc_hd__inv_2
sky130_fd_sc_hd__inv_4
sky130_fd_sc_hd__inv_6
sky130_fd_sc_hd__inv_8
```

(ii) Modify this makefile to number these results and select one as an option (see the included Makefile as an example which was generated with help of Chatgpt).
(iii) Once you have the selected cell, modify the makefile to add a rule to extract its relevant information from the library file.
E.g. run "make do_info <selected-cell>"
(iv) Add another rule to extract the layout information of the selected cell from the LEF file in "OpenROAD-flow-scripts/flow/platforms/sky130hd/lef", "make do_layout <selected-cell>"
(v) [optional] Add another rule to generate a svg representation of the selected cell using the layout information. SVG is the text based (XML, a cousin of HTML) vector graphics format,

which can be displayed in your browser. You may need to read about the rectangles, paths and layers in SVG format. See https://svg-tutorial.com/.
E.g. make do_svg sky130_fd_sc_hd__inv_1
Should produce a .svg file for that inverter. Keep the colors of the layers consistent across different cells. Also implement the correct z-order of the layers.