## Makefile Questions

1. What is the difference between shell scripts and Makefile scripts? Can shell commands be used in a Makefile?
2. What are the actual relative paths for the SRC\_VERILOG files after substituting the variable names?
3. What happens when the user runs make design? Which program will be launched?
4. What happens when the user runs make clean, and why is this useful?

## OPTIONAL (But good to do)

1. What is the purpose of a phony target, and how do you define one?
2. Explain how dependency checking works in a Makefile.
3. What happens if a target file exists but is older than one of its dependencies?
4. Write a new Makefile target called test. This target should depend on prerequisite files from the Python tutorial. When invoked, it should check whether the folder Makefile\_tutorial/test\_completed exists. If it does, run your Python script; if not, create the folder first before running the script.

## Python Questions

1. How many paths have negative slack (i.e., violated timing constraints)?
2. What is the average depth (or number of gates) of paths with the smallest positive slack?
3. What is this design's operating frequency? From a timing perspective, is there potential to achieve higher performance?

(If you are familiar with setup/hold time, and static timing analysis please read these articles:

Setup/Hold: <https://nandland.com/lesson-12-setup-and-hold-time/>

STA: <https://anysilicon.com/the-ultimate-guide-to-static-timing-analysis-sta/> )

Advanced Questions - please try to answer as much as you know without the use of the internet

1. This timing report comes from the signoff stage of design (after routing). What is the first issue you notice in this report?
2. How would you address the issue identified in question 4? Additionally, reconsider your answer to question 3 based on this observation.

## Openlane Colab Questions

1. Open this link to the openlane2 open-source electronic design automation flow: [link](https://colab.research.google.com/github/efabless/openlane2/blob/main/notebook.ipynb) , and here is the [Openlane reference manual](https://openlane.readthedocs.io/en/latest/reference/index.html)
2. Run and read all steps in the openlan2 google colab notebook   
     
   Note: These flow steps are standardized, and the same steps are performed in our production flow. Reading and understanding the blurbs for these steps will help clarify what production the flow is doing.
3. Describe in your own words what the following flow steps do
   1. Floorplanning
   2. Clock Tree Synthesis
   3. Global Route
   4. Signoff Static Timing Analysis (Post-PnR Static Timing analysis)
4. Submit screenshots of the Floorplan step, PDN generation step, and GDS streamout step (final product) with the other portions of your onboarding project submission.

## Procedure (qor.html file is good for this section)

## Part 1: Timing Analysis

1. Read the timing summary of the flow after running the design.
2. Identify the Worst Negative Slack (WNS) from the timing summary (postroute).
3. Using the WNS and the clock frequency set in the SDC file, calculate how much faster the design can theoretically run (Fmax = 1/clock period):

## Part 2: Floorplan Area Calculation

1. Run the design through synthesis (flowtool –to synthesis) and note down the design area reported at the optimized synthesis stage (the area the design uses at the syn\_opt flow step)
2. Calculate how large the floorplan needs to be for logic to cover:  
   (Here it is easiest to assume the floorplan is a square, so both the width and height are the same dimensions) ie the square root of the new calculated area.
   * 60% utilization: *Floorplan Area=Design Area/0.6*
   * 90% utilization: *Floorplan Area=Design Area/0.9*

Then we convert these dimensions to be multiples of the site grid, a site is the smallest unit of space available for design components (think lego blocks) and everything should be a multiple of this dimension.  
  
As we can see in the floorplan.tcl script (inside the /scripts directory), the minimum site size for this process is 0.46 units per um, and 4.14 units per um. So, to get the values we can use for the floorplan, we divide the x dimension by 0.46 and divide the y dimension by 4.14

Ie. If the x and y dimensions were specified to be 100x100, the dimensions design units would be 217.4 by 24.15

1. Run the flow with floorplan utilization percentage (60% and 90%, run to completion):
   * Track metrics like worst negative slack (WNS), total negative slack (TNS), routing congestion, and flow runtime.
   * Record observations for the utilization percentage.

## Data Collection

## Part 1: Timing Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Clock Frequency | Worst Negative Slack (WNS) | New Clock Period | Observations |
| Original Frequency |  |  |  |

## Part 2: Floorplan Area Calculation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Utilization (%) | Floorplan Area | Dimensions | WNS | TNS |
| 60% |  |  |  |  |
| 90% |  |  |  |  |

## Observations:

* Discuss which utilization percentage is optimal for this design.
* Note any issues encountered with routing congestion, timing closure, or power dissipation at higher utilizations.

## Analysis Questions (OPTIONAL :p)

1. How much faster can this design theoretically run based on its WNS?
2. What challenges arise when running at a higher clock frequency (e.g., 1 GHz)?
3. Which floorplan utilization percentage provides a balance between timing closure and routing congestion? Why?
4. What are your recommendations for improving timing or reducing area in future iterations of this design?