ece 260c.

Lab 0
Welcome & Tools Setup

Davit Markarian

Lab Logistics

- Labs release after the class ends
  - Check the Schedule to see their release dates
- Labs can be found in Canvas > Assignments
  - Also linked on the Schedule
- Labs are due at 11:59 PM on their listed due date
  - Lab 0 is due on April 8th
  - 3 AM late due time (no penalty)
- Start labs early

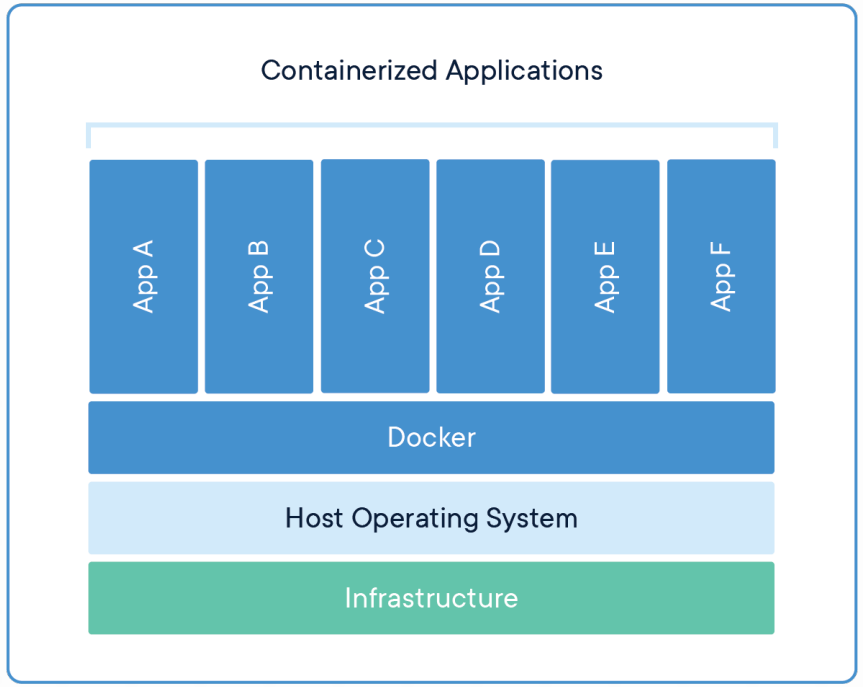
- Please review the Syllabus collaboration and GenAI policies

Lab 0 Rundown

- Software Setup Guide
  - Docker
  - Git/GitHub
- OpenROAD-flow-scripts
  - Running OpenROAD for the first time
  - Making changes to design parameters
  - Using the GUI to make analyses
- Submission with Google Docs/Gradescope and GitHub Classroom

Docker

- Docker is an app that lets you spin up "containers" – isolated copies of Linux-based operating systems prepackaged with apps
- For this course, we provide a downloadable container image for OpenROAD and OpenROAD-flow-scripts
  - You can run OpenROAD without a Linux machine and without spending time on software dependencies
- For this lab, you will install Docker and run our container
  - You will follow the Software Setup guide (also pinned on Canvas)



Realizing the Flow

- OpenROAD, like the commercial tools, is inert without scripting → we must write tcl or Python scripts to realize implementation

- In 260B, you achieved this by guided writing or copying of scripts
  - then, running them stage-by-stage.

- This is an error-prone process
  - Stages must be executed in the correct order
  - Tools are large and complex: many configurable variables with scattered documentation

- Need for a starting-point “reference flow”

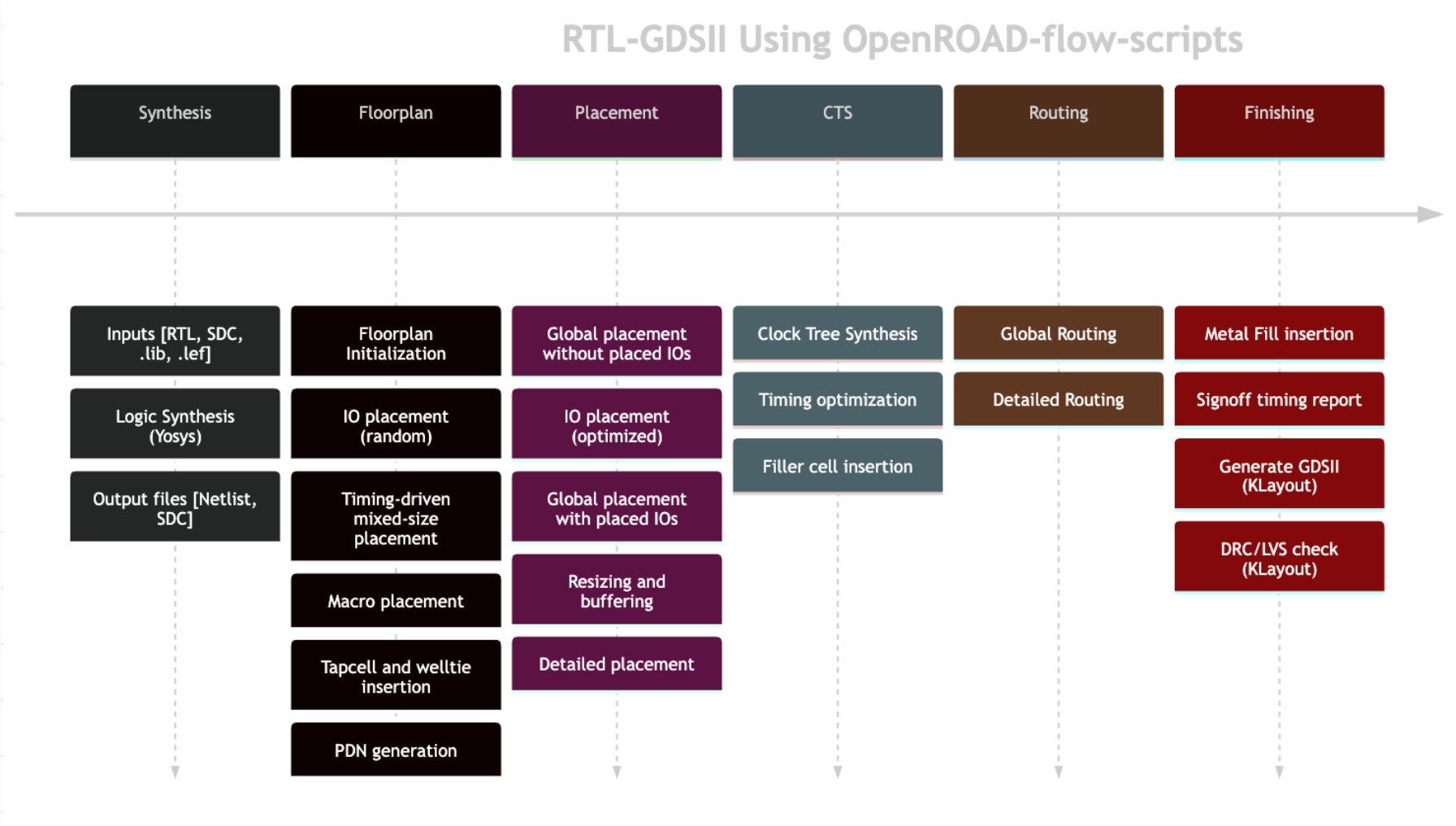
OpenROAD-flow-scripts

- ORFS is a batteries-included flow for OpenROAD
  - It includes highly configurable scripts that you can use to complete implementation
  - Driven by Makefiles with well-documented variables.

- It also includes test designs and PDKs
  - Great for testing, exploration, and research
  - For this course, we will use the included open IHP 130nm SG12G2 process

- Includes scripting hooks between flow stages
  - You can your custom implementation behavior on top of ORFS

OpenROAD-flow-scripts



config.mk

```
export DESIGN_NAME = spi
export PLATFORM    = ihp-sg13g2

export VERILOG_FILES = spi.v
export SDC_FILE      = constraint.sdc

export USE_FILL = 1

export PLACE_DENSITY ?= 0.88
export CORE_UTILIZATION = 20
export TNS_END_PERCENT = 100
```

- `config.mk` is what control an implementation in ORFS
  - Just set flow variables
- ORFS is driven by `flow/Makefile`, which pulls your design's `config.mk`

Building a design is as easy as:

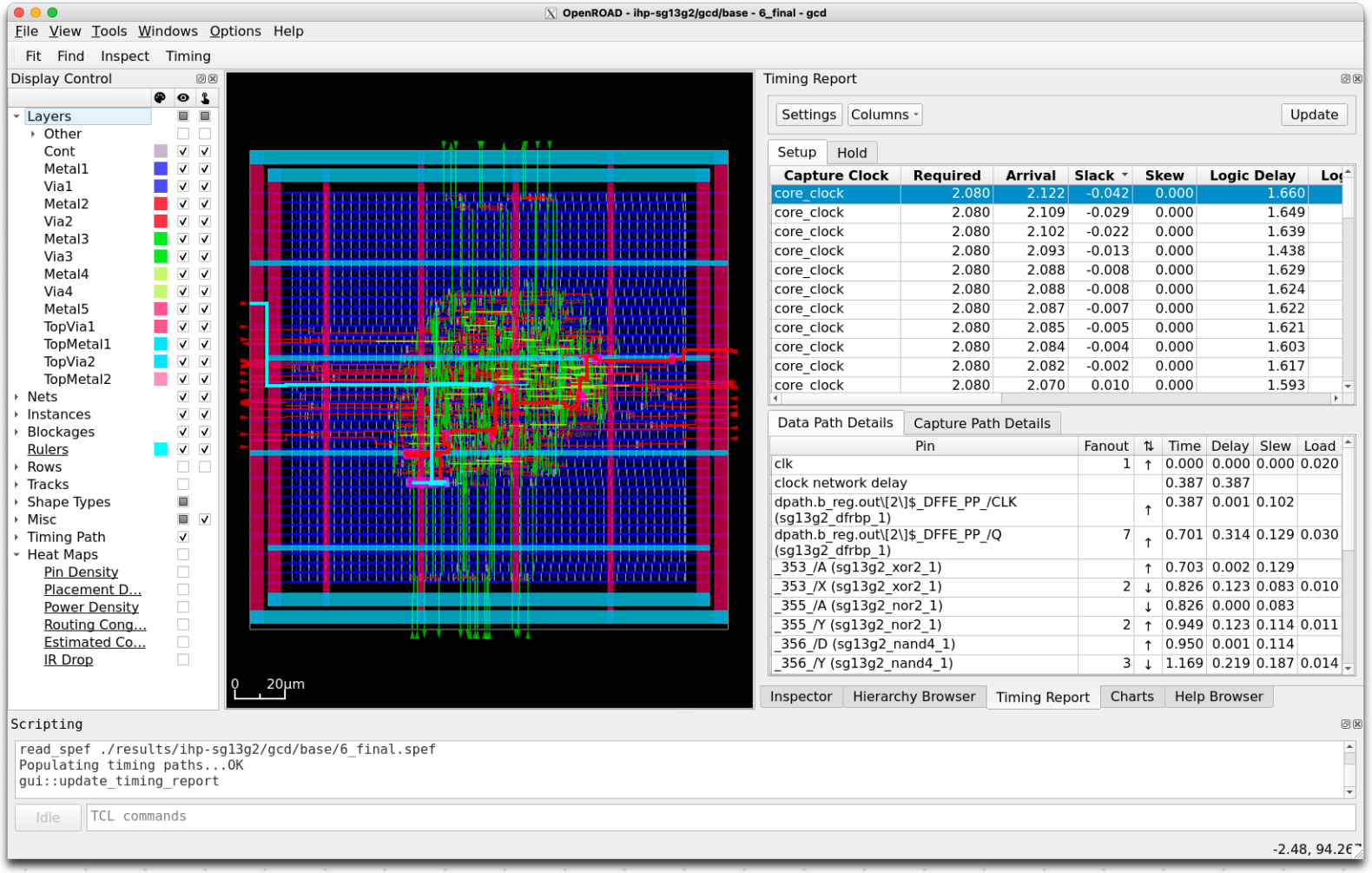
```
DESIGN_CONFIG=./designs/ihp-sg13g2/gcd/config.mk make
```

Outputs

- ORFS will output what you expect from a flow:
  - GDS, DEF, SPEF, Netlist
  - for every stage
- ORFS will also output reports
  - QoR as you expect
  - + stage-by-stage logs/renders
  - + machine-readable reports (JSON)
- You can open interactive sessions for any stage
  - `make open_final` / `make gui_final`


```
{
    ...
    "finish__design__instance__count": 2021,
    "finish__design__instance__area": 23939.2,
    "finish__timing__setup__tns": 0,
    "finish__timing__hold__tns": 0,
    "finish__timing__setup__ws": 0.0957634,
    "finish__timing__hold__ws": 0.369189,
    "finish__clock__skew__setup": 0.0141816,
    "finish__clock__skew__hold": 0.0141816,
    "finish__power__internal__total": 0.00121364,
    "finish__power__switching__total": 0.000514615,
    "finish__power__leakage__total": 1.17737e-06,
    "finish__power__total": 0.00172943,
    "finish__design__io": 54,
    "finish__design__die__area": 39917.8,
    "finish__design__core__area": 23939.2,
    ...
}
```

OpenROAD GUI



Git & GitHub

- Git is the most popular source version control system
  - If you are unfamiliar, please look at an intro guide.
- GitHub is an online platform hosting Git repositories
  - If you do not have an account, create one.
- GitHub Classroom lets you submit Git repos containing your work for this course
  - You will get invitation links to each lab submission within the lab report template
  - Some labs will have starter code too
  - For the first time, you must associate your GitHub account to your name on the class roster. **Please do this carefully.**
- The Git and GitHub command line tools are part of our container

Completion & Submission

- Go to Canvas > Assignments and follow the instructions to make a copy of the lab report template on Google Docs
- You will follow the lab instructions to
  - Perform the full software setup as noted in the Guide.
  - Run the container and experiment with ORFS and the OpenROAD GUI
  - Join the GitHub Classroom assignment, clone the repo to your container, and push your results into it
  - In your browser, verify the GitHub repo – your latest commit is your submission
  - Export your report to PDF and upload it to Gradescope

Final Notes

- With this lab, now is the time to iron out software issues
  → Use Piazza and Office Hours

- Skim both the Software Setup Guide and the Lab 0 template before getting started

- Start early
  - Leave time in case you need software support
  - This lab requires both a large software download (7 GB) and runs of ORFS, which both take time