

## Chapter 04 - OpenROAD first run

Course authors (Git file)



1 Doing this chapter

2 The Makefile

3 The designs to run



## Section 1

Doing this chapter



# Doing this chapter

- This chapter is mostly a training.
- We will try to get our first results (GDS files).
- There are example designs available.
- You can start your own design.



## Section 2

### The Makefile



# The Makefile

Let's have a look into the Makefile first.



## The flow steps

The Makfile (in the `/flow` directory) contains all the flow steps in the same order we already have seen:



Figure 1: The flow steps in the Makefile

# DESIGN\_CONFIG

The Makefile starts with the selection of the design to run.

Of interest for this course are the lines regarding to the IHP PDK:

```
1 #DESIGN_CONFIG=./designs/ihp-sg13g2/aes/config.mk
2 #DESIGN_CONFIG=./designs/ihp-sg13g2/ibex/config.mk
3 DESIGN_CONFIG=./designs/ihp-sg13g2/gcd/config.mk
4 #DESIGN_CONFIG=./designs/ihp-sg13g2/spi/config.mk
5 #DESIGN_CONFIG=./designs/ihp-sg13g2/riscv32i/config.mk
6 #DESIGN_CONFIG=./designs/ihp-sg13g2/masked_aes/config.mk
```

The gcd example is selected for the next run.



## Section 3

The designs to run



# gcd (greatest common denominator)

- The gcd design is included with the OpenROAD-flow-script examples.
- It consists of only a single Verilog file, easy to read.
- Should run on the course server in a few minutes.



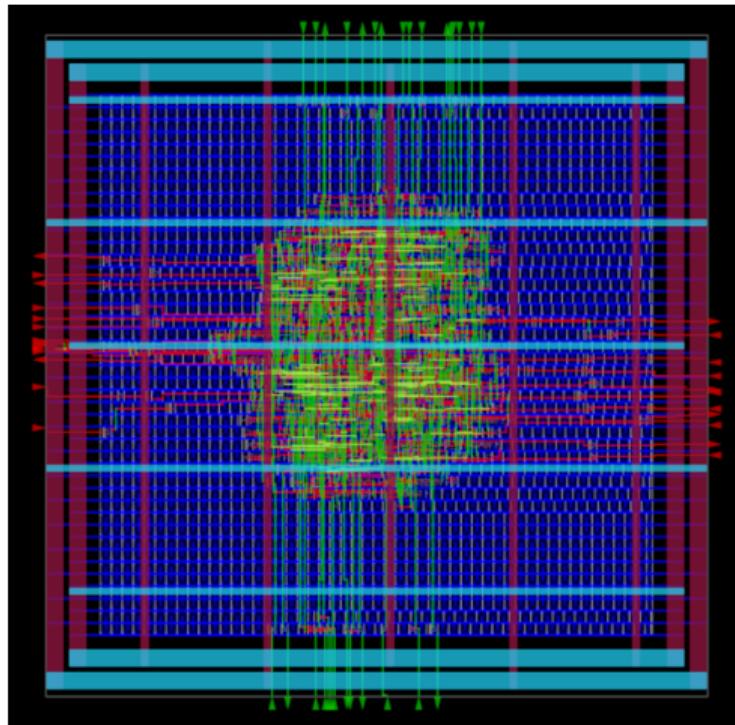


Figure 2: GDS gcd



## gcd files:

```
1 | /src/gcd$ ls  
2 | gcd.v  
3 | README.md
```

```
1 | /ihp-sg13g2/gcd$ ls  
2 | autotuner.json  
3 | config.mk          (important)  
4 | constraint.sdc    (important)  
5 | metadata-base-ok.json  
6 | rules-base.json
```



# ibex: RISC-V core

- The ibex design is included with the OpenROAD-flow-script examples.
- It consists of many Verilog files, not that easy to read.
- A single run might take more than 30 minutes on the course server.



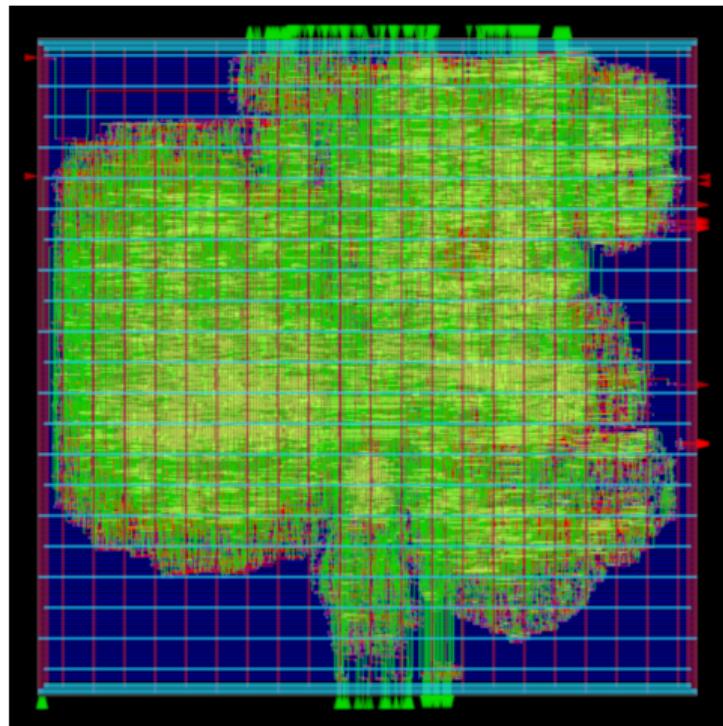


Figure 3: GDS ibex



## ibex files:

```
1 src/ibex$ ls
2 ibex_alu.v           ibex_ex_block.v      ibex_register_file_ff.v   prim_ram_1p.v
3 ibex_branch_predict.v ibex_fetch_fifo.v    ibex_register_file_fpga.v prim_secded_28_22_dec.v
4 ibex_compressed_decoder.v ibex_icache.v     ibex_register_file_latch.v prim_secded_28_22_enc.v
5 ibex_controller.v     ibex_id_stage.v     ibex_wb_stage.v        prim_secded_39_32_dec.v
6 ibex_core.v          ibex_if_stage.v     LICENSE                prim_secded_39_32_enc.v
7 ibex_counter.v        ibex_load_store_unit.v prim_badbit_ram_1p.v  prim_secded_72_64_dec.v
8 ibex_cs_registers.v  ibex_multdiv_fast.v  prim_clock_gating.v   prim_secded_72_64_enc.v
9 ibex_csr.v            ibex_multdiv_slow.v prim_generic_clock_gating.v prim_xilinx_clock_gating.v
10 ibex_decoder.v       ibex_pmp.v         prim_generic_ram_1p.v README.md
11 ibex_dummy_instr.v   ibex_prefetch_buffer.v prim_lfsr.v
```



```
1 ihp-sg13g2/ibex$ ls
2 autotuner.json
3 config.mk
4 constraint_doe.sdc
5 constraint.sdc
6 metadata-base-ok.json
7 rules-base.json
```



# masked\_aes

- The masked\_aes design is part of a research project and is available on Github:
  - HEP Alliance - Masked AES
- It consists of three Verilog files, one of them >2000 lines of code.
- Should run in a few minutes on the course server.

## Special Features:

- Contains I/O Pads and a Padring
- Has a README that contains the how-to of a sealring
- Has a README that links to the Metall filler script



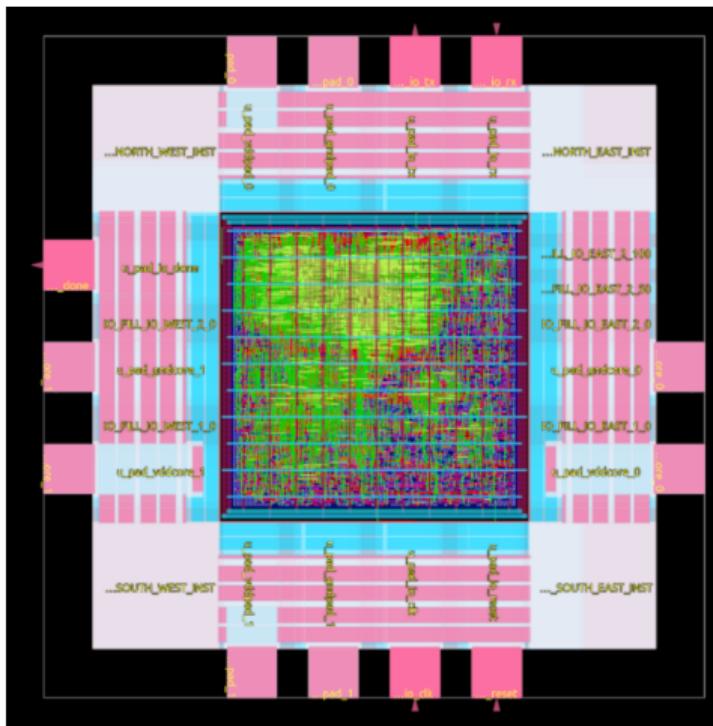


Figure 4: GDS masked\_aes



## masked\_aes files:

```
1 ihp-sg13g2/masked_aes$ ls
2 config.mk
3 constraint.sdc
4 footprint.tcl
5 LICENSE
6 README.md
7 sealring.gds
8 src          (src directory!)
```

```
1 ihp-sg13g2/masked_aes/src$ ls
2 AES_Masked.v
3 AesTb.v
4 MaskedAes.v
```



# Ifsr

- The Ifsr design example must be created from the scratch.
- The Verilog code is available in the lecture slides of chapter 3 and should become a single file.
- The structure of other examples must be copied for this.
- The configuration files must be copied and adapted for this.
- A single run should be very short.



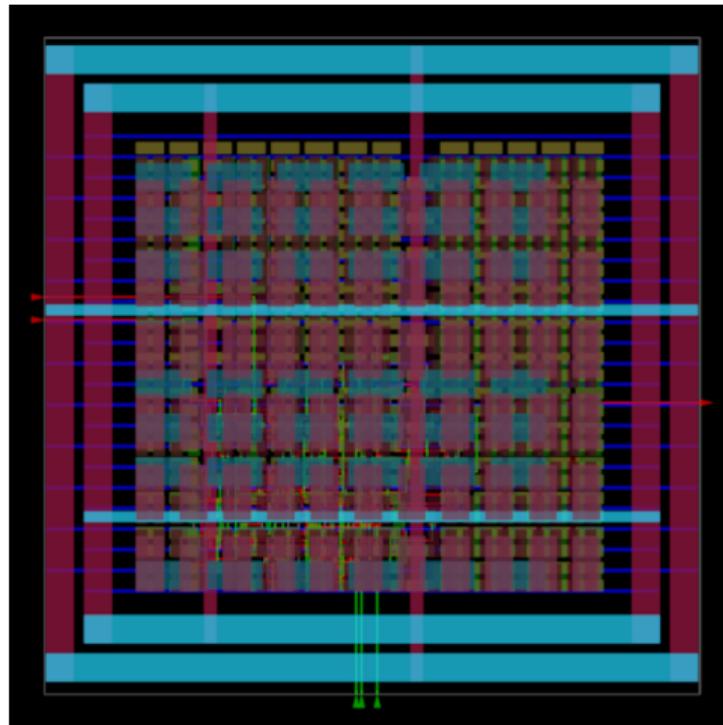


Figure 5: GDS Ifsr

# A TinyTapeout design?

- The design example must be created from the scratch.
- The Verilog code is available as open-source via TinyTapeout
- The structure of other examples must be copied for this.
- The configuration files must be copied and adapted for this.
- The run time is unpredicted.

Suggestion:

The VGA clock example from the pictures earlier:

[https://tinytapeout.com/runs/ttihp0p2/tt\\_um\\_vga\\_clock](https://tinytapeout.com/runs/ttihp0p2/tt_um_vga_clock)



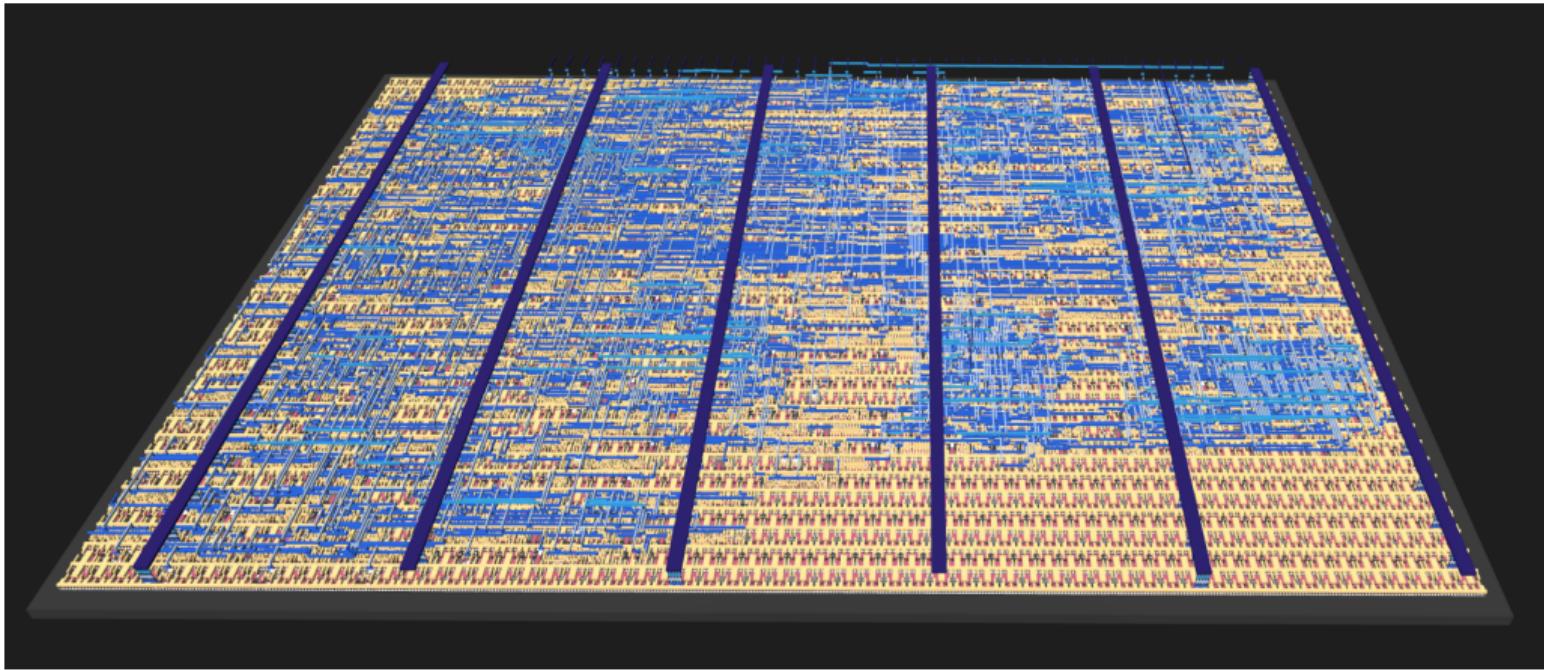


Figure 6: VGA Clock GDS render

