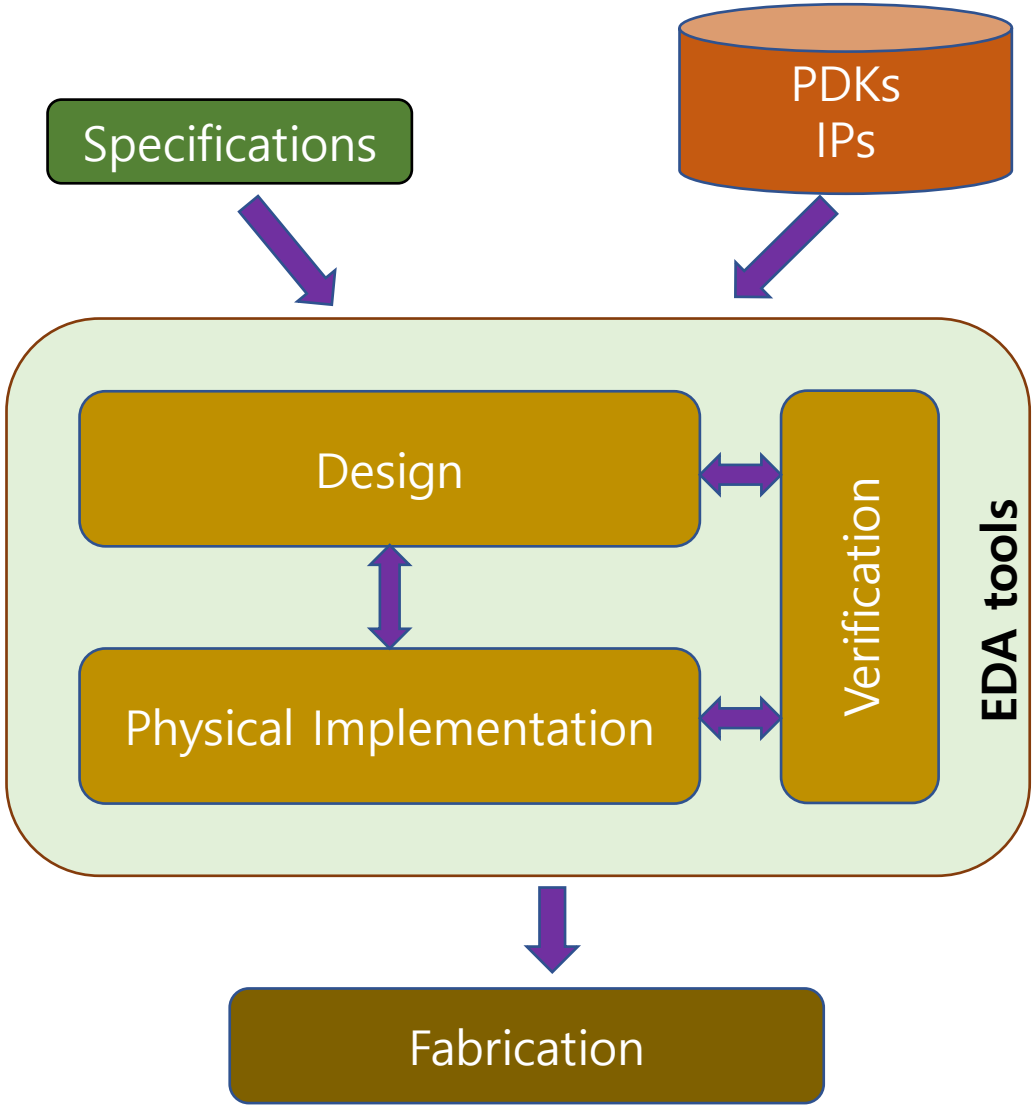
Custom ASIC/eFPGA Design Using Opensource tools

Nguyen Dao
nguyenvt11@gmail.com

FPGA Ignite Summer School 2023 – Heidelberg University Jul. 31st – Aug. 4th

ASIC Design Flow



Opensource EDA tools and PDKs

Opensource tools

- Xschem – Analog/mixed signal design
- Yosys – RTL synthesis
- Icarus/Iverilog + GTKWave – synthesis/simulation and waveform viewer
- Openroad – Physical implementation
- OpenTimer/OpenSTA – Timing/Power analysis
- Magic/Klayout – Layout/DRC check
- Netgen – LVS check
- CVC – Circuit Validity checker

- https://xschem.sourceforge.io/stefan/xschem_man/xschem_man.html
- https://github.com/YosysHQ/yosys
- http://iverilog.icarus.com
- https://gtkwave.sourceforge.net
- https://github.com/The-OpenROAD-Project/OpenLane
- http://opencircuitdesign.com/magic/
- https://www.klayout.de
- https://github.com/hpretl/iic-osic-tools
- https://anysilicon.com/the-ultimate-guide-to-open-source-eda-tools/

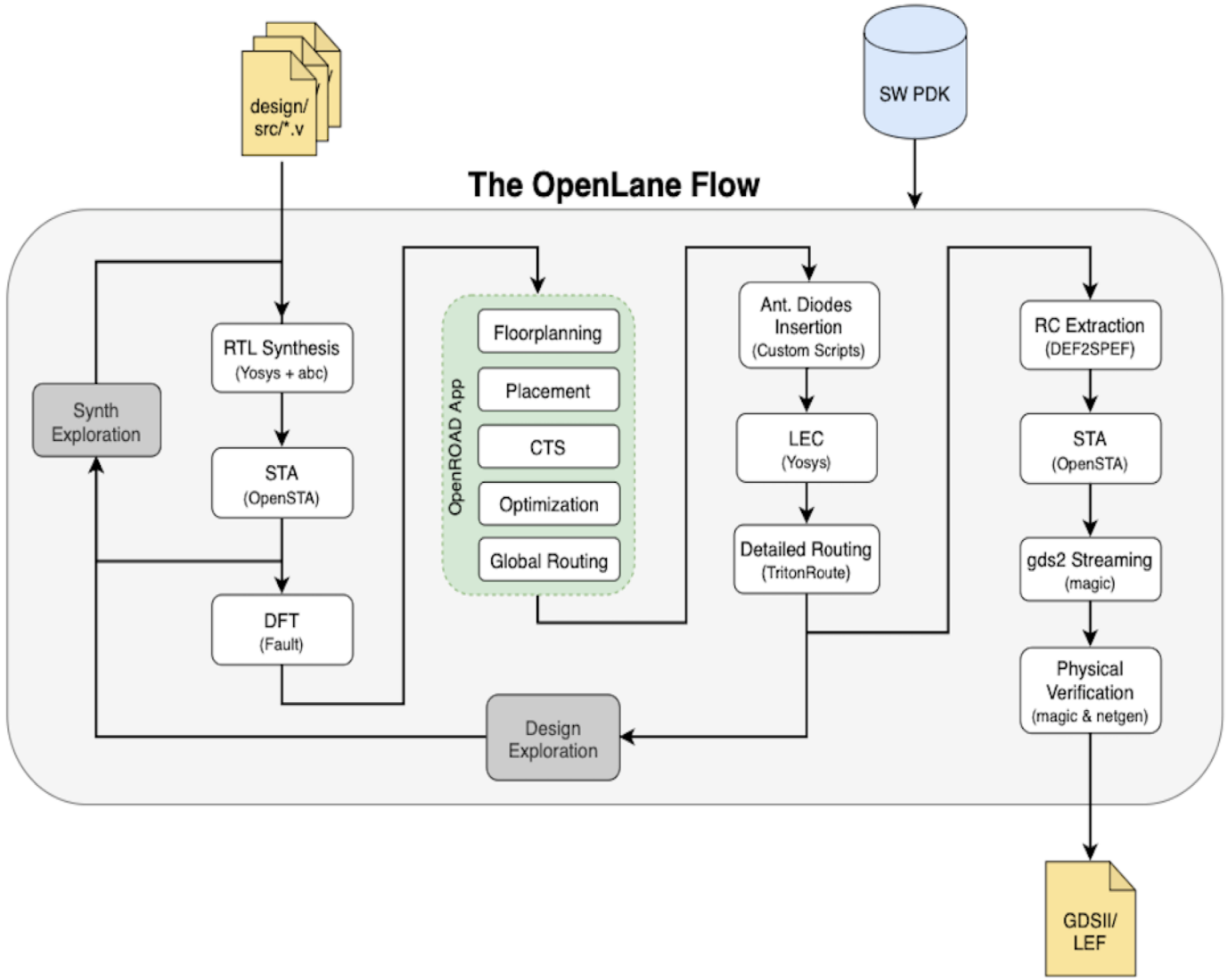
Opensource PDKs

- GF180MCU (GlobalFoundries 180nm)
- Sky130 – Sky90-FDSOI (Skywater 130nm – 90nm)
- FreePDK45 (45nm) - FreePDK15 (15nm)
- ASAP7 (Predictive 7nm Process)
- FreePDK3 (Predictive 3nm Process)

- https://skywater-pdk.readthedocs.io/en/main/
- https://opensource.googleblog.com/2022/07
- https://eda.ncsu.edu/
- https://asap.asu.edu

Openlane Design Flow

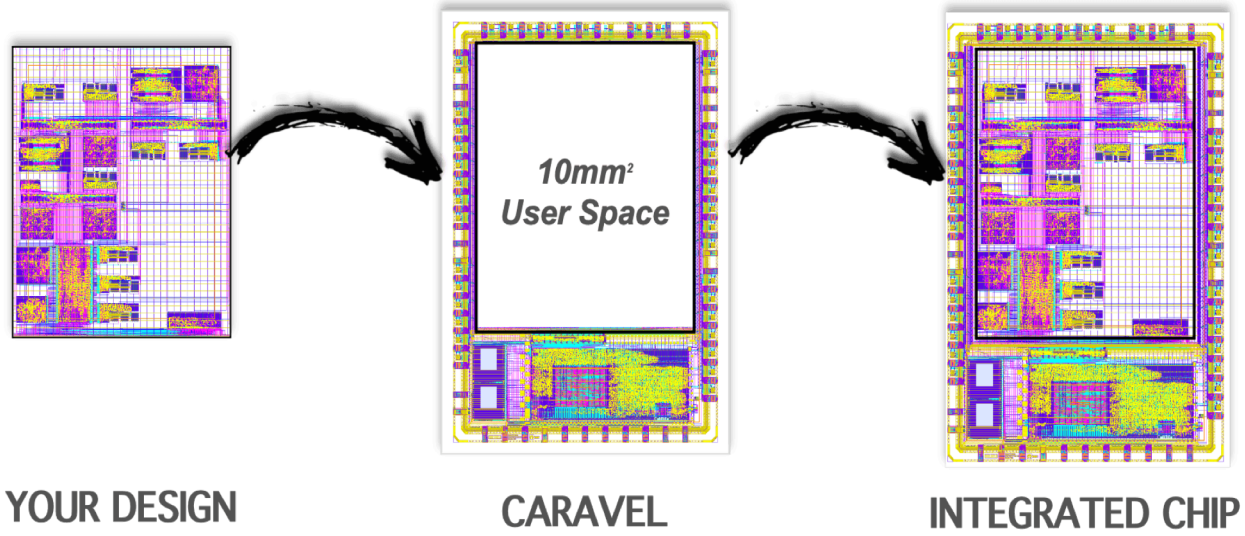
1. **Synthesis**
   - o Yosys – RTL synthesis
   - o ABC – technology mapping
   - o OpenSTA – static timing analysis
2. **Floorplan and PDN**
   - o Init_fp – core area planning
   - o Ioplacer – macros/Ios placement
   - o Pdn – implement power distribution network
   - o Tapcell – insert welltap/decap cells
3. **Placement**
   - o RePlace – perform global placement
   - o Resizer – optimize the design
   - o OpenDP – perform detailed placements
4. **CTS**
   - o TritionCTS – Clock Tree Synthesis
5. **Routing**
   - o FastRout/CU-GR – perform global routing
   - o TritonRoute – perform detailed routing
   - o SPEF-Extractor - perform parasitic extraction
6. **GDSII Generation**
   - o Magic/Klayout – stream out the final GDSII layout file



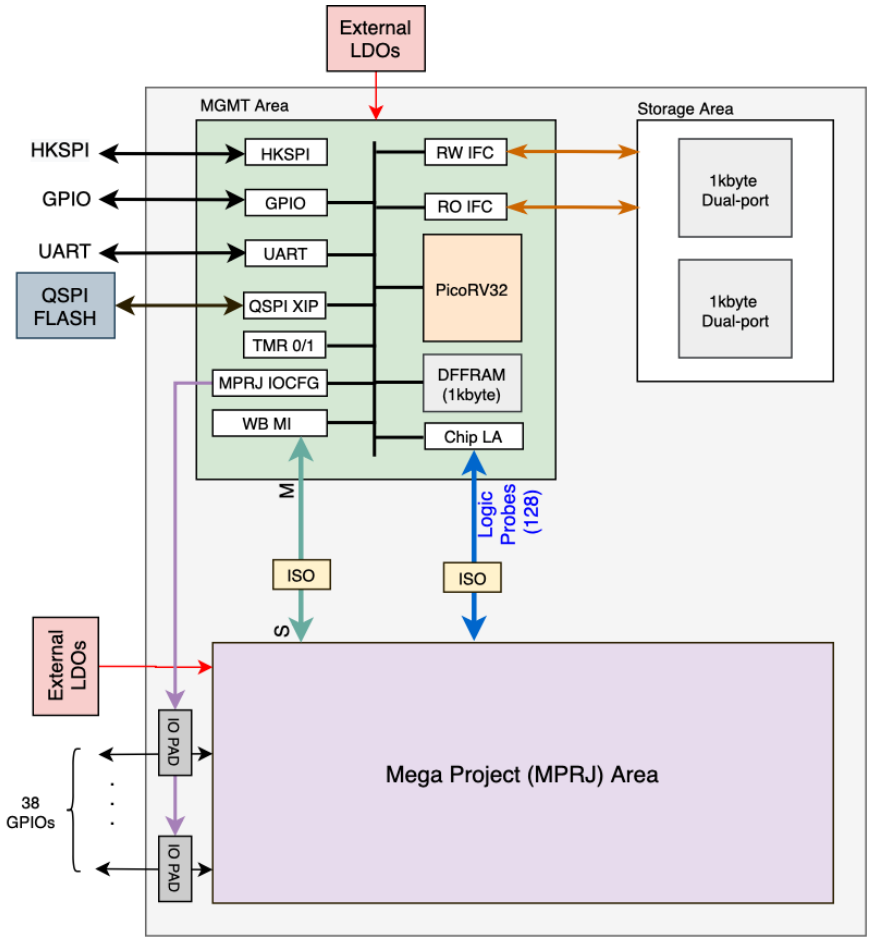
https://github.com/efabless/caravel_user_project/blob/main/docs/source/quickstart.rst

eFabless Caravel SoC

- **Caravel SoC** is composed of the harness frame, the **management area** and the **user project area**
- The management SoC is a RISC-V based SoC that includes several peripherals such as UART, GPIOs etc.
- The management SoC runs firmware that can be used to configure the IOs, control the power supply and observe/control signals to/from User project wrapper

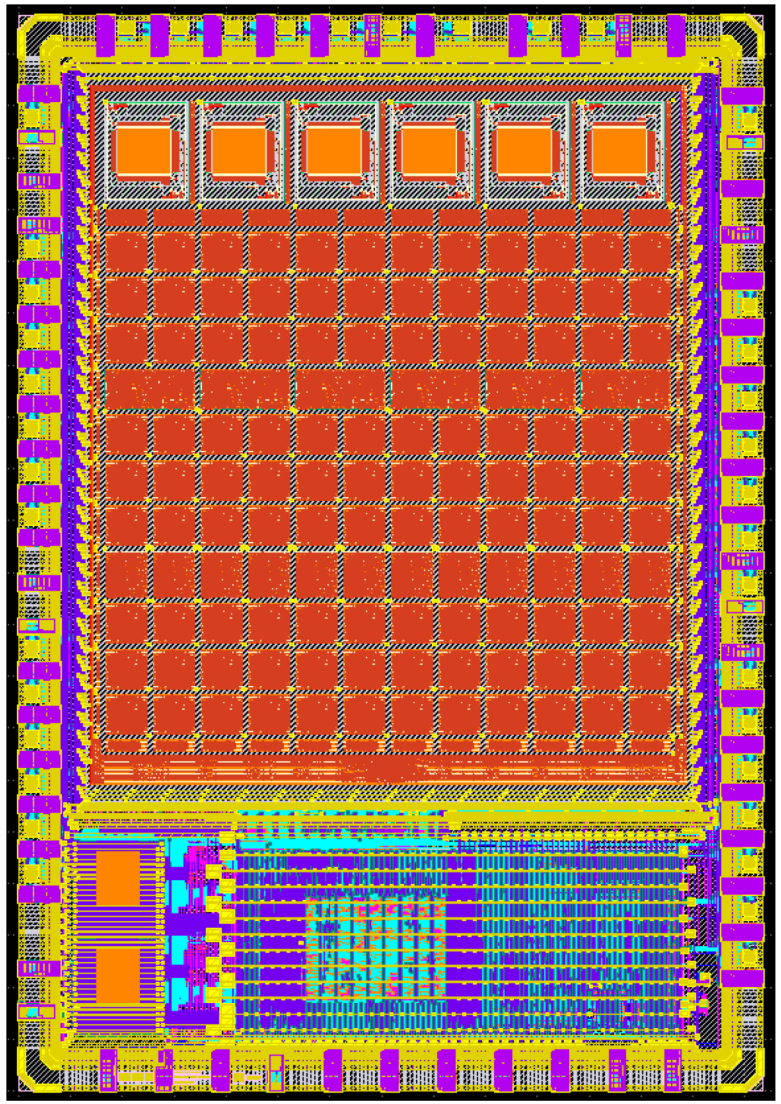


- User project area (2.92mm x 3.52 mm) has fixed 38 GPIOs, 128 Logic analyzer probes and Wishbone port connections to management SoC



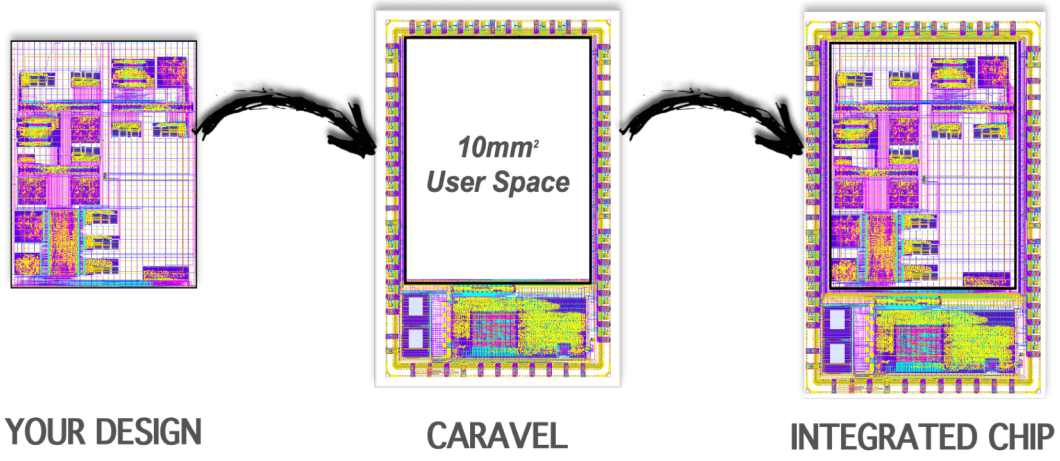
eFPGA design flow using Openlane

1. Create/Generate RTLs
2. Customize the cells (optional)
3. Hardening tiles (optional)
4. Hardening the fabric (eFPGA_top)/User project wrapper
5. Caravel Integration



eFabless Caravel workspace

```
<your_caravel_project>
├── caravel                 # Caravel SoC
├── def                     # Final DEF files
├── dependencies            # PDKs and OpenLane tools
├── docs
├── gds                     # Final GDS files
├── lef                     # Final LEF files
├── LICENSE
├── mag                     # Final magic layout files
├── maglef                  # Final MAGLEF files
├── Makefile
├── mgmt_core_wrapper
├── openlane                # Design configurations
├── precheck_results        # Local precheck
├── README.md
├── sdc                     # Timing constraints
├── sdf                     # Delay timing
├── signoff                 # Signoff results
├── spef                    # Parasitic RC extractions
├── spi                     # Spice/LVS
└── Verilog                 # Gate netlist/RTL sources
```



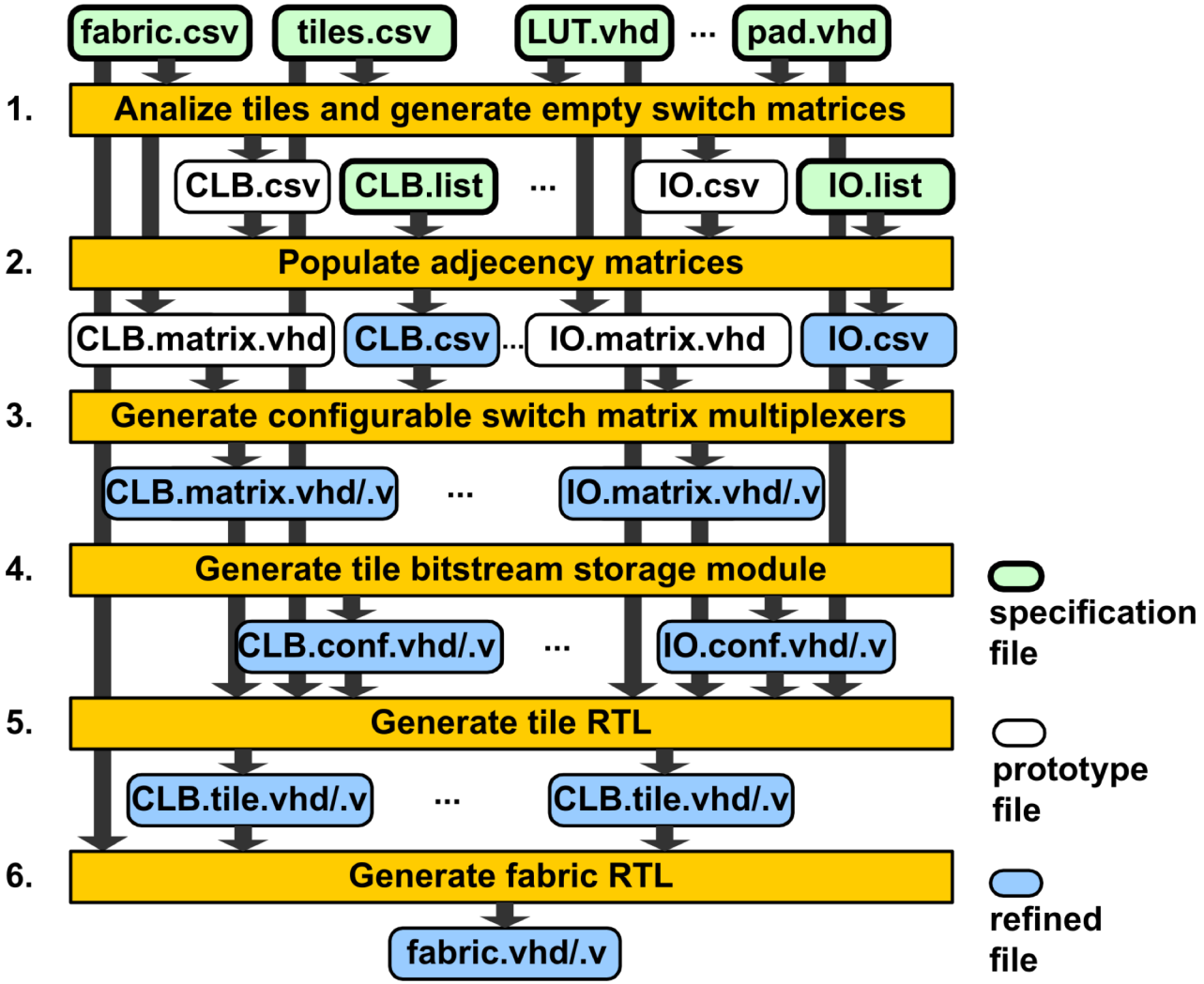

```
openlane/
├── Makefile
├── user_project_wrapper        # Top module (fixed name)
│   ├── config.tcl              # Flow/Design constraints
│   ├── fixed_dont_change       # Default setups
│   ├── macros                  # Macro lib/placements
│   └── runs                    # Implementation results
└── user_proj_example           # Your submodules/designs
    ├── config.tcl
    ├── pin_order.cfg
    └── runs
```

```
$make user_proj_example      # implement your submodule/design
$make user_project_wrapper   # implement the top user_project_wrapper
```

eFPGA design flow using Openlane

1. Generate the fabrics (RTLs)

More details at
https://fabulous.readthedocs.io/en/latest/Building%20fabric.html



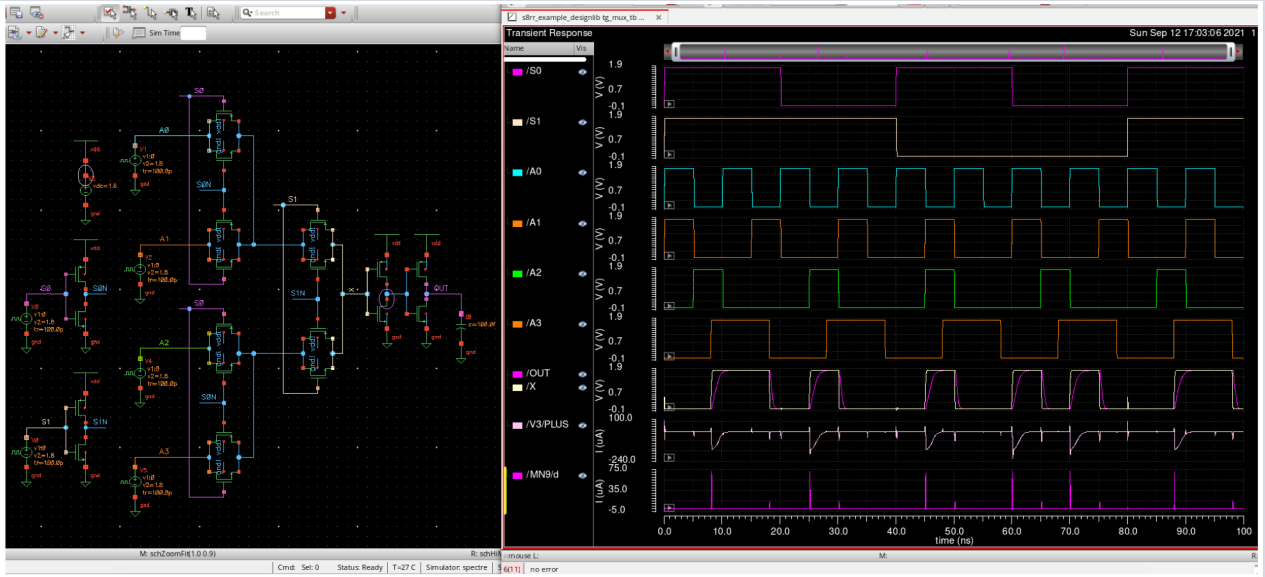
eFPGA design flow using Openlane

2. Customize the cells (using LibreCell - optional)

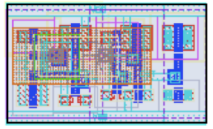
- 2.1. Create schematic
- 2.2. Simulation
- 2.3. Layout
- 2.4. Extract and do post-layout simulation
- 2.5. Export LEF/lib/GDS

```
VERSION 5.7 ;
  NOWIREEXTENSIONATPIN ON ;
  DIVIDERCHAR "/" ;
  BUSBITCHARS "[]" ;
MACRO cus_tg_mux41_buf
  CLASS CORE ;
  FOREIGN cus_tg_mux41_buf ;
  ORIGIN 0.000 0.000 ;
  SIZE 6.440 BY 2.720 ;
  SYMMETRY X Y R90 ;
  SITE unithd ;
  PIN S0
    ANTENNAGATEAREA 0.216000 ;
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER met2 ;
        RECT 1.980 1.310 2.300 1.570 ;
        RECT 2.035 0.800 2.245 1.310 ;
        RECT 1.995 0.480 2.255 0.800 ;
    END
    PORT
      LAYER met1 ;
        RECT 0.145 1.550 0.375 1.700 ;
        RECT 1.980 1.550 2.300 1.570 ;
        RECT 0.145 1.410 2.300 1.550 ;
        RECT 1.980 1.310 2.300 1.410 ;
      LAYER via ;
        RECT 2.010 1.310 2.270 1.570 ;
    END
  END S0
...
  END
END cus_tg_mux41_buf
```

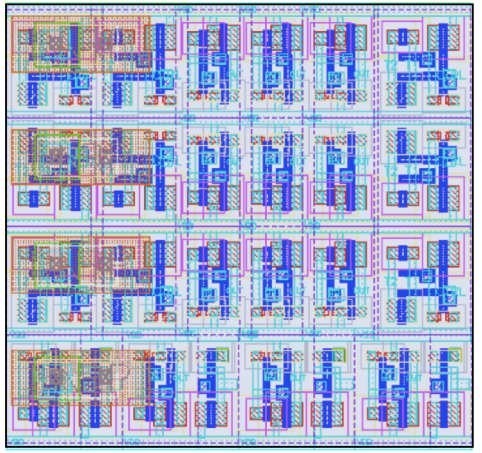
```
library ("custom_mux") {
    define(def_sim_opt,library,string);
    ...
    time_unit : "1ns";
    voltage_unit : "1V";
    leakage_power_unit : "1nW";
    current_unit : "1mA";
    pulling_resistance_unit : "1kohm";
    capacitive_load_unit(1.0000000000, "pf"); ...
    default_arc_mode : "worst_edges";
    default_constraint_arc_mode : "worst";
    default_leakage_power_density : 0.0000000000;
    default_operating_conditions : "tt_025C_1v80";
    operating_conditions ("tt_025C_1v80") {
        voltage : 1.8000000000;
        process : 1.0000000000;
        temperature : 25.000000000;
        tree_type : "balanced_tree";
    }
    power_lut_template ("power_inputs_1") {
        variable_1 : "input_transition_time";
        index_1("1, 2, 3, 4, 5, 6, 7");
cell ("cus_tg_mux41_buf") {
        leakage_power () {
            value : 0.0137458000;
            when : "!A0&!A1&!A2&!A3&!S0&S1";
        }
                related_pin : "S1";
                rise_transition ("del_1_7_7") {

        ...
                timing_sense : "negative_unate";
                timing_type : "combinational";
            }
}
```

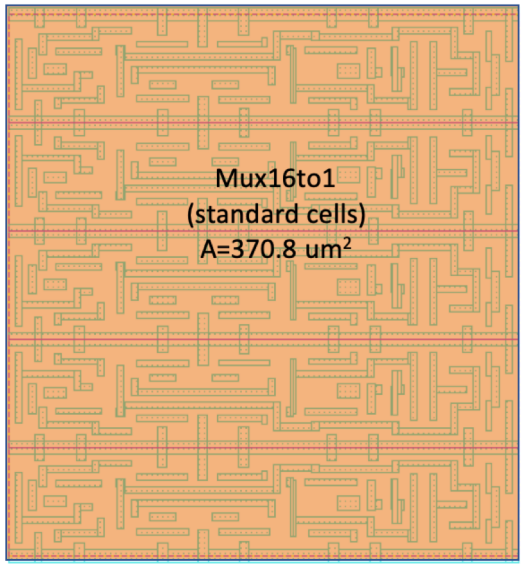


Mux2to1
A=27.3 um²



Mux16to1
A =272.45 um²

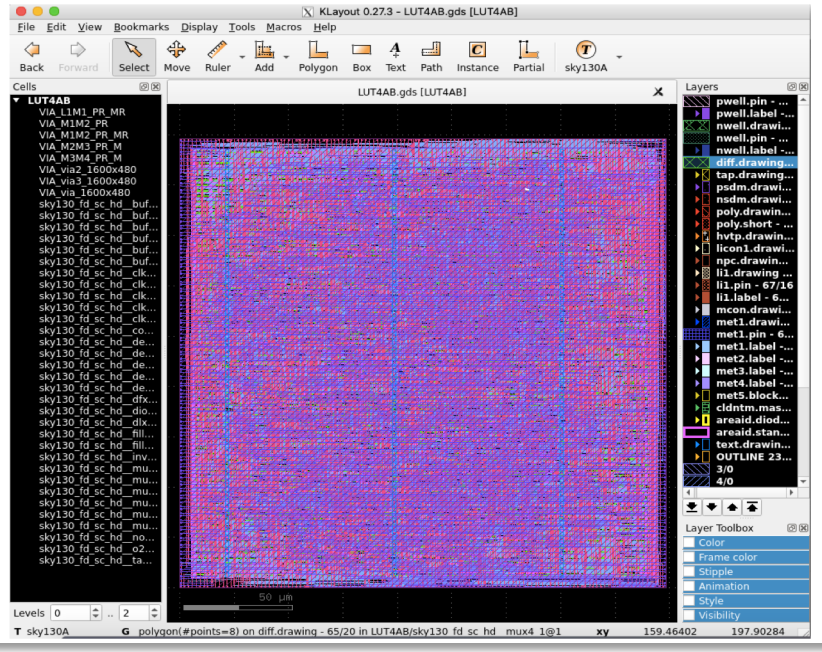




eFPGA design flow using Openlane

3. Hardening tiles

- Configure the flow and design constraints (config.tcl)
- Initial/Set Technology/Lib and Top design
- Set area/density
- Set clock constraints
- Set technology/custom gates mapping
- Set IO pins arrangement
- Set Timing constraints (disable timing loops)
- Set routing constraints (layers/halos)



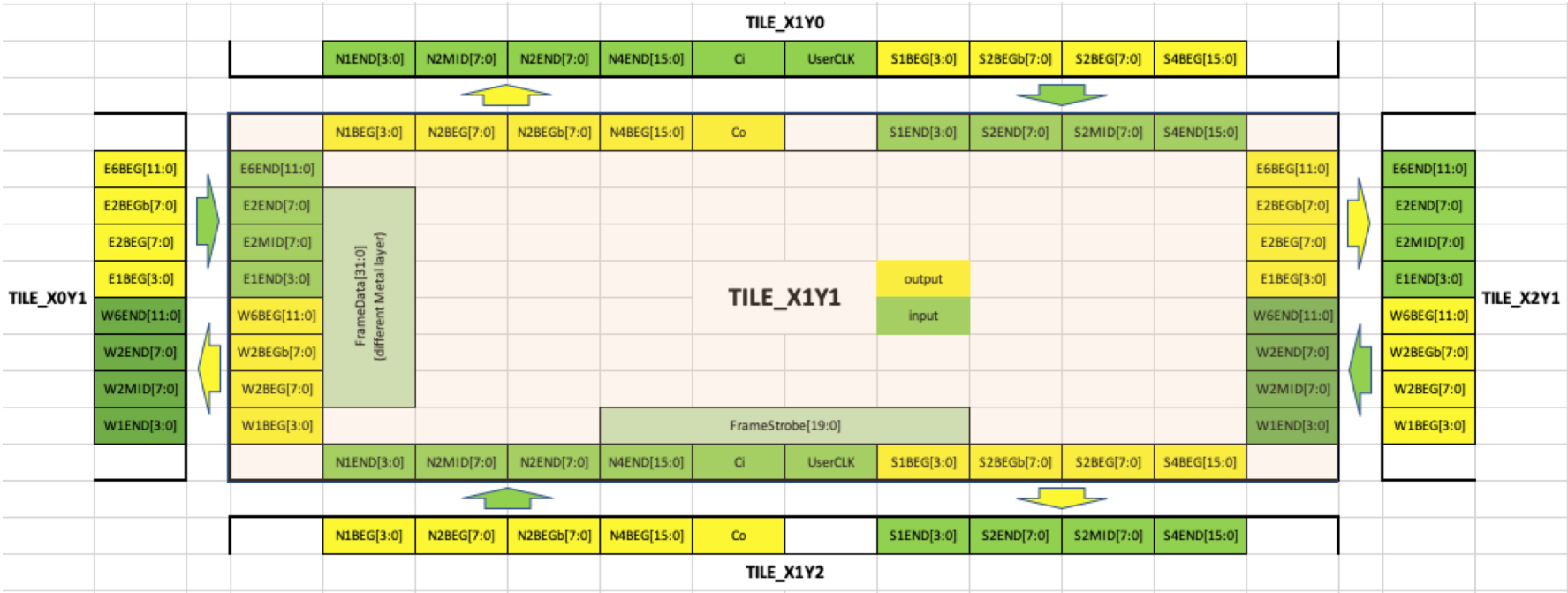

```
# User config
set ::env(DESIGN_NAME) LUT4AB

# Change if needed
set ::env(VERILOG_FILES) [glob $::env(DESIGN_DIR)/src/*.v]

# Use FP_CORE_UTIL to experiment the tile's size
#set ::env(FP_CORE_UTIL) 55

# Use FP_SIZING to fix the tile's area
set ::env(FP_SIZING) "absolute"
set ::env(DIE_AREA) "0 0 223.275 223.115"; #baseline LUT4AB(223.275 223.115)

#set ::env(PL_TARGET_DENSITY) [ expr ($::env(FP_CORE_UTIL)+5) / 100.0 ]
set ::env(PL_TARGET_DENSITY) 0.6

# Clock config
set ::env(CLOCK_PERIOD) "40"
set ::env(CLOCK_PORT) "UserCLK"
set ::env(CLOCK_TREE_SYNTH) 1

# Synthesis mode - should disable flattening the hierarchy that helps setting timing contraints later
# It also requires remove "-flatten" option at line 353 in scripts/yosys/synth.tcl
set ::env(SYNTH_NO_FLAT) 1

# DESIGN_IS_CORE 1 default, 0 is a macro
set ::env(DESIGN_IS_CORE) 0
set ::env(FP_PDN_CORE_RING) 0
set ::env(GLB_RT_MAXLAYER) 5

# Specify latch gate for mapping
set ::env(SYNTH_LATCH_MAP) $::env(DESIGN_DIR)/gate_map.v

# Change sdc file
set ::env(BASE_SDC_FILE) $::env(DESIGN_DIR)/LUT4AB.sdc

# Change the size and arrange the IO pins
set ::env(FP_IO_VLENGTH) 0.8
set ::env(FP_IO_HLENGTH) 0.8
set ::env(FP_IO_HTHICKNESS_MULT) 2
set ::env(FP_IO_VTHICKNESS_MULT) 2
set ::env(FP_IO_MODE) 0
set ::env(FP_PIN_ORDER_CFG) $::env(DESIGN_DIR)/pin_order.cfg

# Adjust the floorplan
set ::env(TOP_MARGIN_MULT) 2
set ::env(BOTTOM_MARGIN_MULT) 2

# Set the power pins
set ::env(VDD_PINS) "vccd1"
set ::env(GND_PINS) "vssd1"

set ::env(ROUTING_CORES) 12
```

eFPGA design flow using Openlane

3. Hardening tiles (Cont.)

- Set IO pins – pin_order.cfg

```
#BUS_SORT
#NR
N1BEG*
N2BEG*
...
S4END*

#S
N1END*
N2MID*
...
S4BEG*
FrameStrobe*

#E
E6END*
...
W1BEG*
FrameData*

#WR
E6BEG*
...
W1END*
```

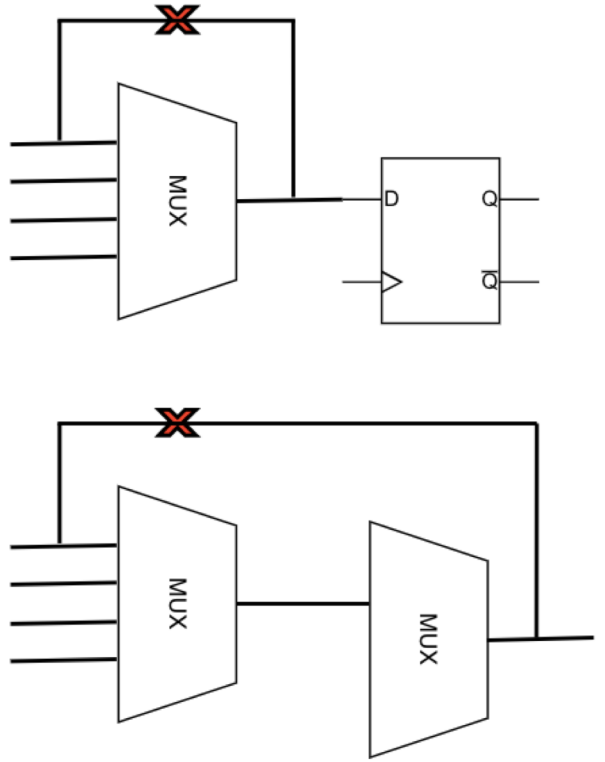


eFPGA design flow using Openlane

3. Hardening tiles (optional) (Cont.)

- Set timing constraints (e.g., LUT4AB.sdc)

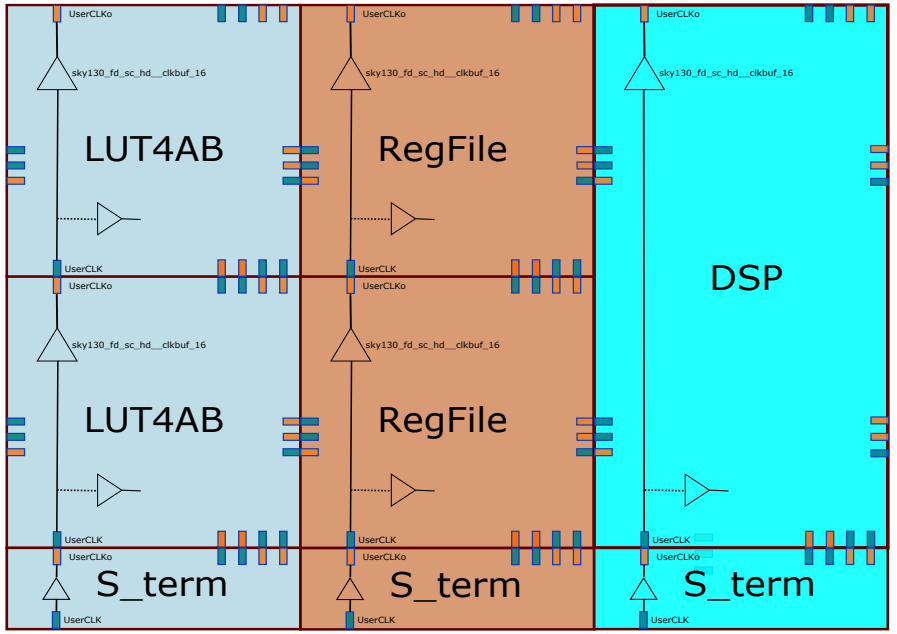
```
create_clock [get_ports $::env(CLOCK_PORT)]  -name $::env(CLOCK_PORT)  -period $::env(CLOCK_PERIOD)
set input_delay_value [expr $::env(CLOCK_PERIOD) * $::env(IO_PCT)]
set output_delay_value [expr $::env(CLOCK_PERIOD) * $::env(IO_PCT)]

set clk_indx [lsearch [all_inputs] [get_port $::env(CLOCK_PORT)]]
#set rst_indx [lsearch [all_inputs] [get_port resetn]]
set all_inputs_wo_clk [lreplace [all_inputs] $clk_indx $clk_indx]
#set all_inputs_wo_clk_rst [lreplace $all_inputs_wo_clk $rst_indx $rst_indx]
set all_inputs_wo_clk_rst $all_inputs_wo_clk

# correct resetn
set_input_delay $input_delay_value  -clock [get_clocks $::env(CLOCK_PORT)] $all_inputs_wo_clk_rst
#set_input_delay 0.0 -clock [get_clocks $::env(CLOCK_PORT)] {resetn}
set_output_delay $output_delay_value  -clock [get_clocks $::env(CLOCK_PORT)] [all_outputs]

# TODO set this as parameter
set_driving_cell -lib_cell $::env(SYNTH_DRIVING_CELL) -pin $::env(SYNTH_DRIVING_CELL_PIN) [all_inputs]
set cap_load [expr $::env(SYNTH_CAP_LOAD) / 1000.0]
set_load  $cap_load [all_outputs]

#set_disable_timing for timing loops
#Tile_X0Y*_W_IO
set_disable_timing [get_pins Tile_X0Y*_W_IO/Inst_W_IO_switch_matrix/E1BEG*]
set_disable_timing [get_pins Tile_X0Y*_W_IO/Inst_W_IO_switch_matrix/E2BEG*]
set_disable_timing [get_pins Tile_X0Y*_W_IO/Inst_W_IO_switch_matrix/E2BEGb*]
set_disable_timing [get_pins Tile_X0Y*_W_IO/Inst_W_IO_switch_matrix/E6BEGb*]

#Tile_X*Y*_LUT4AB
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J2MID_ABa_BEG*]
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J2MID_CDa_BEG*]
…
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J_l_AB_BEG*]
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J_l_CD_BEG*]
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J_l_EF_BEG*]
set_disable_timing [get_pins Tile_X*Y*_LUT4AB/Inst_LUT4AB_switch_matrix/J_l_GH_BEG*]
…
```



eFPGA design flow using Openlane

3. Hardening tiles – Notes (Cont.)

- Need to add the lib/lef of the custom cells to the sky130 tech files
- Enable latch mapping – need to specify the latch used for configurations and specify the custom cells be used (gate_map.v)
- Enable hierarchical synthesis (SYNTH_NO_FLAT=1) to resist changing the module name during yosys synthesis
- Set Mux4 as preferable for better density (Yosys uses Mux2 as the default)
- RTL syntax, limited support for SystemVerilog e.g., global param/inherited param (#)
- Disable combinational loops by replacing the default *base.sdc*
- IO pins placement is limited (e.g., single metal layer only)
- Clock tree synthesis is limited (cannot handle a very large number of connections)



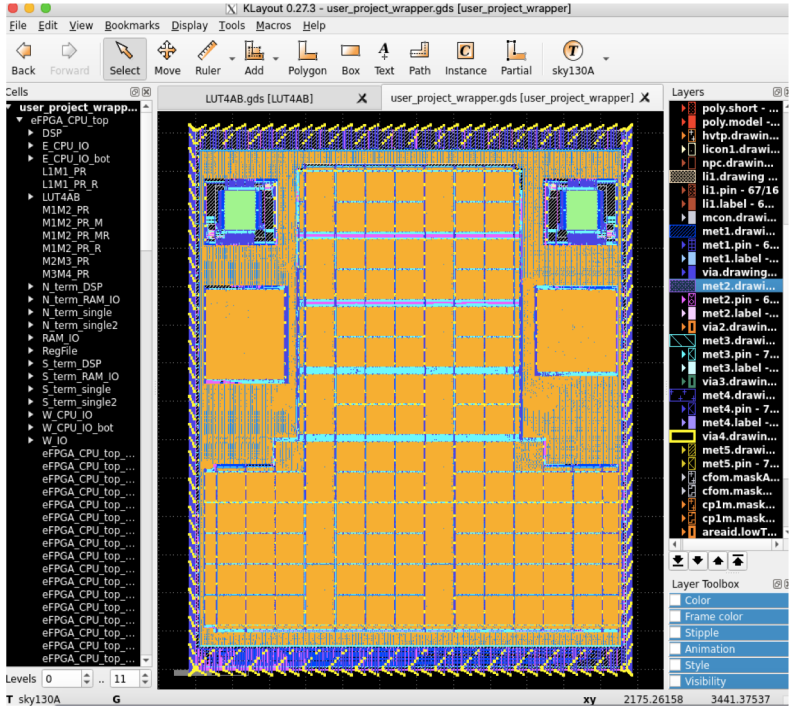
https://github.com/nguyendao-uom/open_eFPGA

eFPGA design flow using Openlane

4. Hardening the User project (eFPGA_top fabric)

- Instantiate and connect the fabric (eFPGA_top) in User_project_wrapper.v
- Floorplan and Placement constraints
- Configure the Openlane flow and the design constraints
- Hardening the User_project_wrapper

(Note: the power rails can be unconnected to some tiles/macros if they are not in the range of PDN pitch)

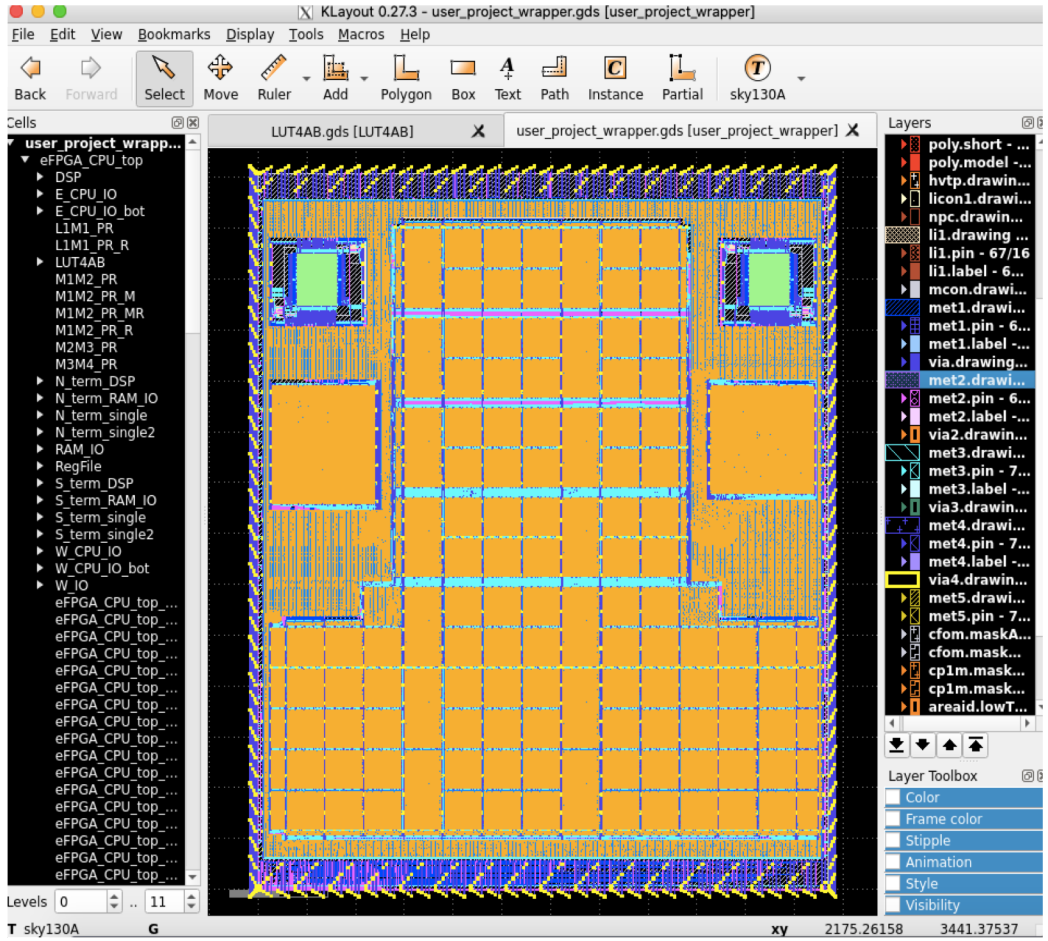



```
set script_dir [file dirname [file normalize [info script]]]
source $::env(CARAVEL_ROOT)/openlane/user_project_wrapper/fixed_wrapper_cfgs.tcl
source $::env(CARAVEL_ROOT)/openlane/user_project_wrapper/default_wrapper_cfgs.tcl

set ::env(DESIGN_NAME) user_project_wrapper
set ::env(FP_PDN_ENABLE_RAILS) 1
set ::env(GLB_RT_OBS) "met1 0 0 $::env(DIE_AREA),\
                       met2 0 0 $::env(DIE_AREA),\
                       met3 0 0 $::env(DIE_AREA),\
                       met4 0 0 $::env(DIE_AREA),\
                       met5 0 0 $::env(DIE_AREA)"

set ::env(CLOCK_PORT) "user_clock2"
set ::env(CLOCK_NET) "inst_eFPGA_top.user_clock2"
set ::env(CLOCK_PERIOD) "40"

set ::env(PL_OPENPHYSYN_OPTIMIZATIONS) 0
set ::env(DIODE_INSERTION_STRATEGY) 5
set ::env(MAGIC_WRITE_FULL_LEF) 0

set ::env(SYNTH_FLAT_TOP) 1
set ::env(CLOCK_TREE_SYNTH) 1
set ::env(DESIGN_IS_CORE) 1
set ::env(STA_REPORT_POWER) 0
set ::env(SYNTH_USE_PG_PINS_DEFINES) "USE_POWER_PINS"
set ::env(VDD_NETS) {vccd1 vdda1 vdda2 vccd2}
set ::env(GND_NETS) {vssd1 vssa1 vssa2 vssd2}
set ::env(VDD_PIN) "vccd1"
set ::env(GND_PIN) "vssd1"
set ::env(PL_TARGET_DENSITY) 0.45
set ::env(CTS_TARGET_SKEW) 200
set ::env(CTS_SINK_CLUSTERING_SIZE) 100
set ::env(CTS_SINK_CLUSTERING_MAX_DIAMETER) 1000
set ::env(ROUTING_CORES) 12
set ::env(GLB_RT_MAXLAYER) 5
set ::env(FP_PDN_CHECK_NODES) 0
set ::env(FP_PDN_IRDROP) 0

set ::env(FP_TAP_HORIZONTAL_HALO) 20
set ::env(FP_TAP_VERTICAL_HALO) 20
set ::env(FP_PDN_HORIZONTAL_HALO) 30
set ::env(FP_PDN_VERTICAL_HALO) 30
set ::env(FP_PDN_VWIDTH) 1.6
set ::env(FP_PDN_VPITCH) 2800

set ::env(SYNTH_READ_BLACKBOX_LIB) 1
set ::env(VERILOG_FILES) [glob $script_dir/../../verilog/rtl/defines.v $script_dir/../../../verilog/rtl/*.v ]
## Internal Macros
### Macro Placement
set ::env(MACRO_PLACEMENT_CFG) "$script_dir/../../../openlane/user_project_wrapper/macros/placements/macro_placement.cfg"
### Black-box verilog and views
set ::env(VERILOG_FILES_BLACKBOX) [glob $script_dir/../../../verilog/rtl/BB/*.v]
set ::env(EXTRA_LEFS) [glob $script_dir/../../../openlane/user_project_wrapper/macros/lef/*.lef]
set ::env(EXTRA_GDS_FILES) [glob $script_dir/../../../openlane/user_project_wrapper/macros/gds/*.gds]
### Macro PDN Connections
set ::env(FP_PDN_MACRO_HOOKS) "\
        inst_eFPGA_top.Inst_eFPGA.Tile_X0Y1_W_IO vccd1 vssd1 \
        inst_eFPGA_top.Inst_eFPGA.Tile_X0Y2_W_IO vccd1 vssd1 \
        inst_eFPGA_top.Inst_eFPGA.Tile_X0Y3_W_IO vccd1 vssd1 \
                                                              51,1          9%
```

eFPGA design flow using Openlane

4. Hardening the User project (eFPGA_top fabric)

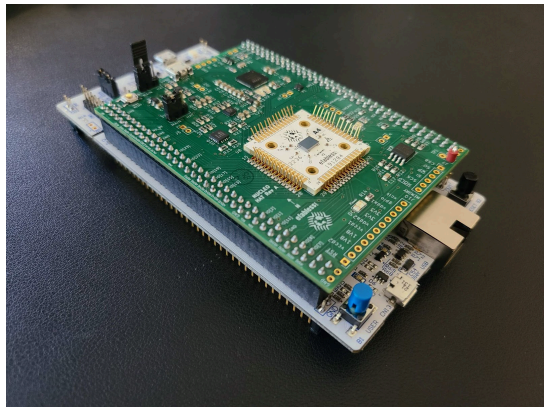
- Blocks/Macros placements

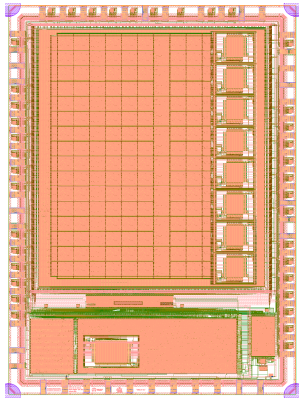



```
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y1_W_IO 150 2723.12 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y2_W_IO 150 2502.8599999999997 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y3_W_IO 150 2282.5999999999995 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y4_W_IO 150 2062.3399999999992 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y5_W_IO 150 1842.0799999999992 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y6_W_IO 150 1621.8199999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y7_W_IO 150 1401.5599999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y8_W_IO 150 1181.2999999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y9_W_IO 150 961.0399999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y10_W_IO 150 740.7799999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y11_W_IO 150 520.5199999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X0Y12_W_IO 150 300.2599999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y0_N_term_single 250.04000000000002 2943.38 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y1_LUT4AB 250.04000000000002 2723.12 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y2_LUT4AB 250.04000000000002 2502.8599999999997 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y3_LUT4AB 250.04000000000002 2282.5999999999995 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y4_LUT4AB 250.04000000000002 2062.3399999999992 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y5_LUT4AB 250.04000000000002 1842.0799999999992 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y6_LUT4AB 250.04000000000002 1621.8199999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y7_LUT4AB 250.04000000000002 1401.5599999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y8_LUT4AB 250.04000000000002 1181.2999999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y9_LUT4AB 250.04000000000002 961.0399999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y10_LUT4AB 250.04000000000002 740.7799999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y11_LUT4AB 250.04000000000002 520.5199999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y12_LUT4AB 250.04000000000002 300.2599999999993 N
inst_eFPGA_top.Inst_eFPGA.Tile_X1Y13_S_term_single 250.04000000000002 249.99999999999932 N
inst_eFPGA_top.Inst_eFPGA.Tile_X2Y0_N_term_single 470.14 2943.38 N
inst_eFPGA_top.Inst_eFPGA.Tile_X2Y1_LUT4AB 470.14 2723.12 N
inst_eFPGA_top.Inst_eFPGA.Tile_X2Y2_LUT4AB 470.14 2502.8599999999997 N
inst_eFPGA_top.Inst_eFPGA.Tile_X2Y3_LUT4AB 470.14 2282.5999999999995 N
inst_eFPGA_top.Inst_eFPGA.Tile_X2Y4_LUT4AB 470.14 2062.3399999999992 N
```

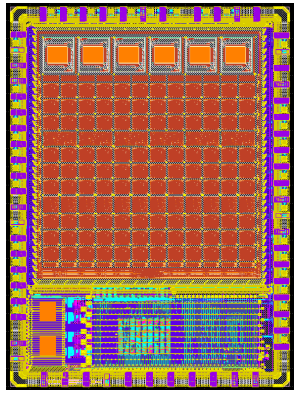
eFPGA design flow using Openlane

5. Caravel Integration
6. Run local precheck
7. Submit the GDS job on eFabless portal
   >>> DONE !!!

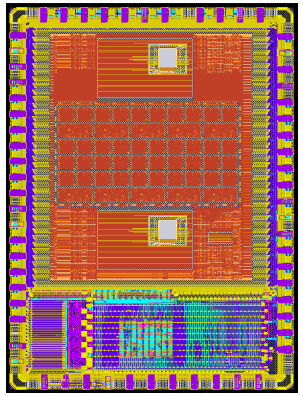




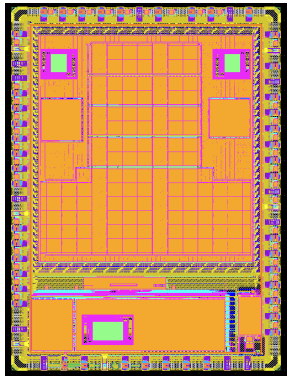
eFPGA_caravel_sky130
CLBs, DSPs, RegFiles, BBRAMs
Google Shuttle - MPW-2
https://github.com/nguyendao-uom/eFPGA_v3_caravel



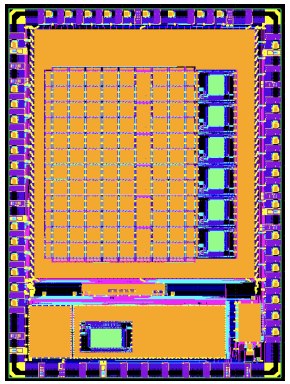
eFPGA_caravel_sky130
CLBs, DSPs, RegFiles, BBRAMs with custom cells
Google Shuttle - MPW-3
https://github.com/FPGA-Research-Manchester/FABulous-Sky---a-heterogeneous-FPGA-fabric-in-Skywater130



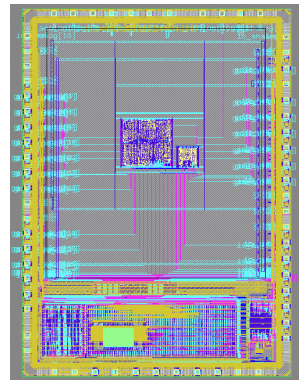
eFPGA_RISCV_sky130
RISCV with eFPGA for tensorflow micro applications
Google Shuttle - MPW-3
https://github.com/nguyendao-uom/fuserisc_ver2



ICESOC_caravel
Ibex-Crypto-eFPGA for cryptography
Google Shuttle - MPW-4
https://github.com/nguyendao-uom/ICESOC



Open_eFPGA
Full-opensource eFPGA with OpenLane and SKY130
Google Shuttle - MPW5
https://github.com/nguyendao-uom/open_eFPGA



Open_ReRAM_eFPGA
Full-opensource Reram_based eFPGA SKY130
Google Shuttle - MPW4
https://github.com/nguyendao-uom/rram_testchip

Acknowledgement

This work is supported by EPSRC Program Grant FORTE (EP/R024642/1)

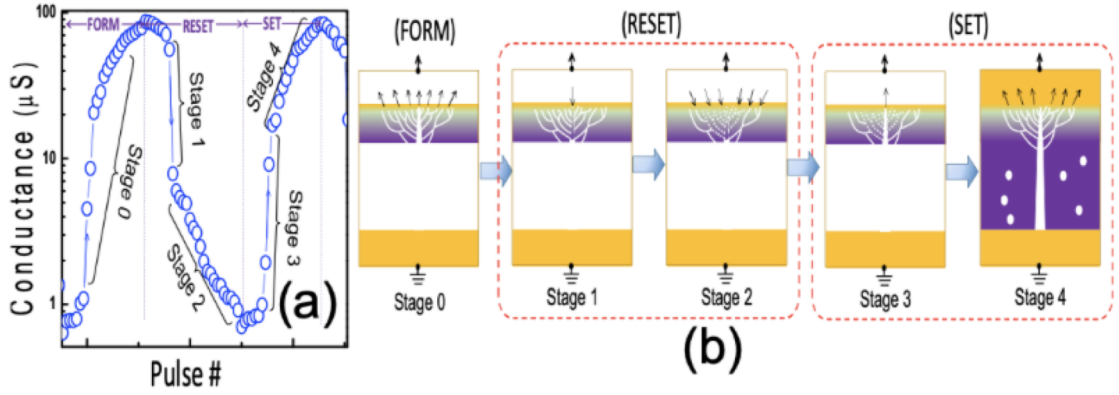
Thank you!

ReRAM/Memristor SKY130

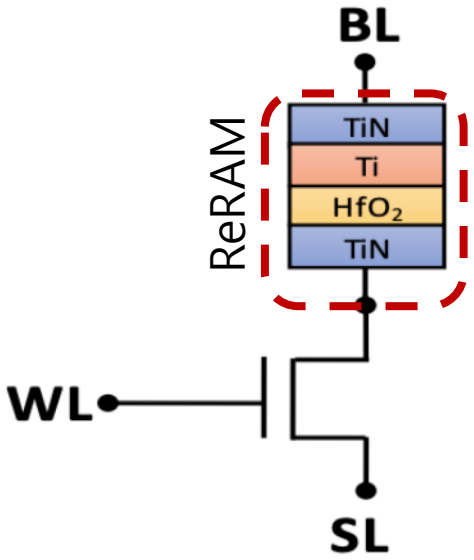
Resistive RAM (ReRAM): consists of a metal-oxide switching layer sandwiched by top and bottom metal electrodes.

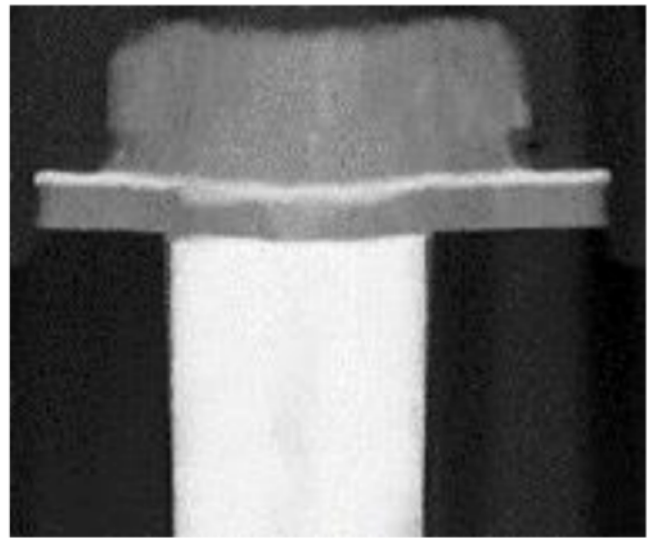
Operation:

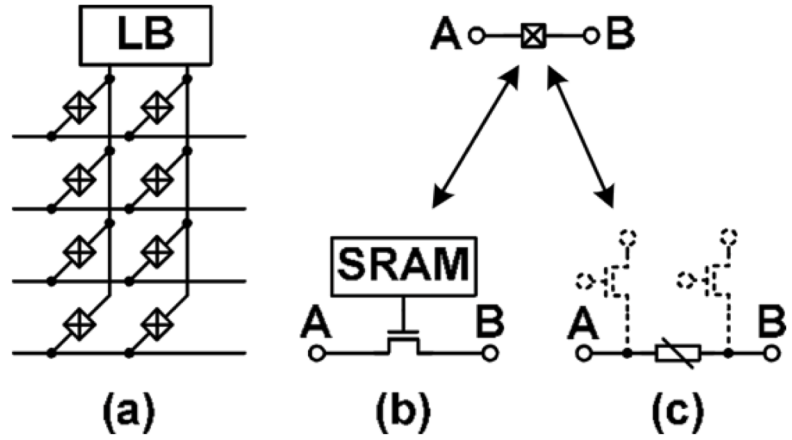
1. Forming
2. Programming (set/reset)
3. Reading



https://sky130-fd-pr-reram.readthedocs.io/en/latest/
https://github.com/barakhoffer/sky130_xyce_reram

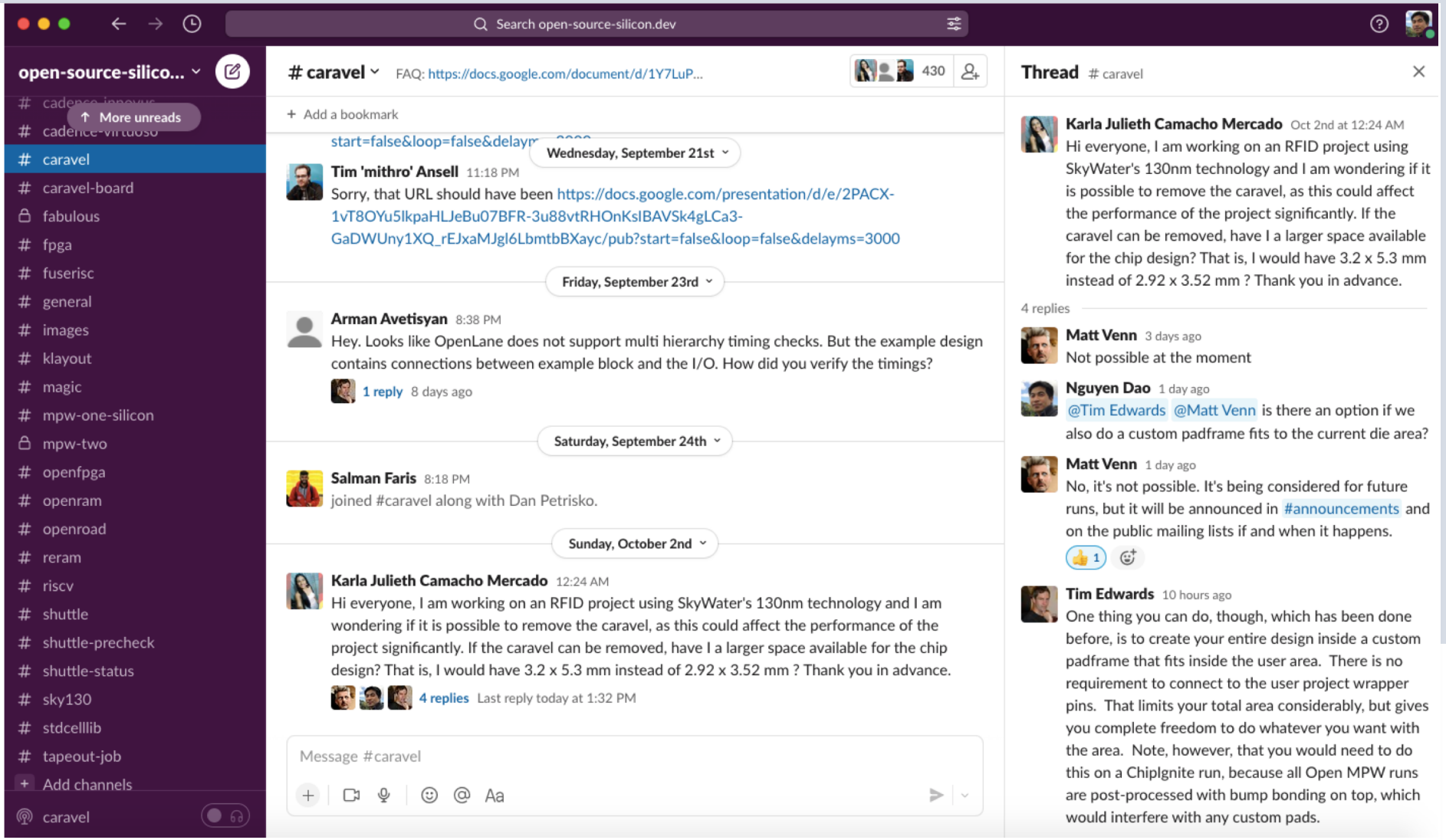






Replacing SRAM cell in FPGA routing

Questions?



https://join.slack.com/t/open-source-silicon/shared_invite/zt-1hb6gydjo-C2NCyrjGtkAwWcaaRTSbNQ