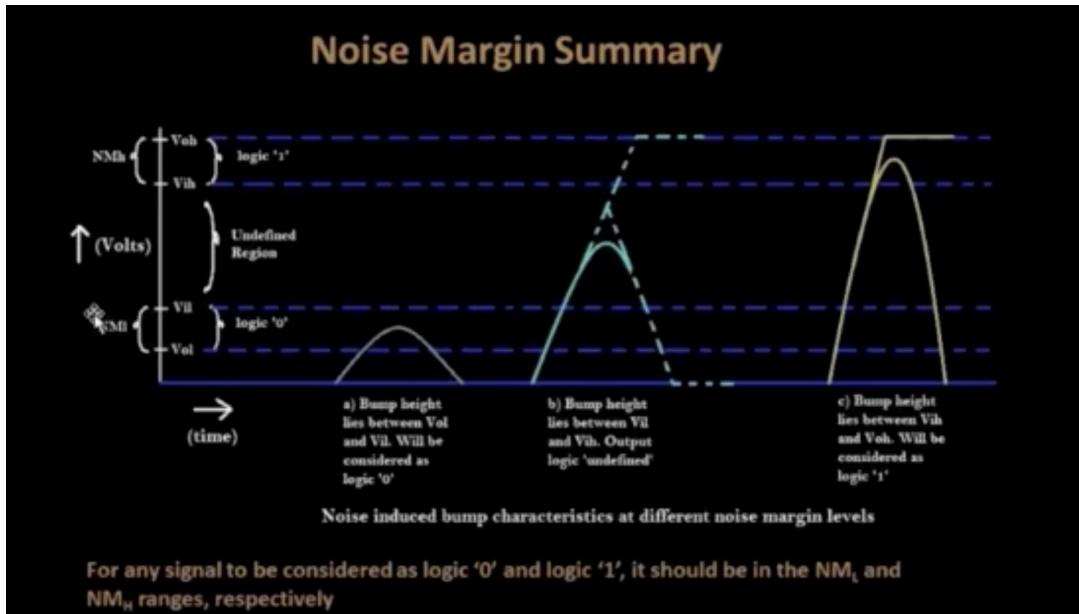
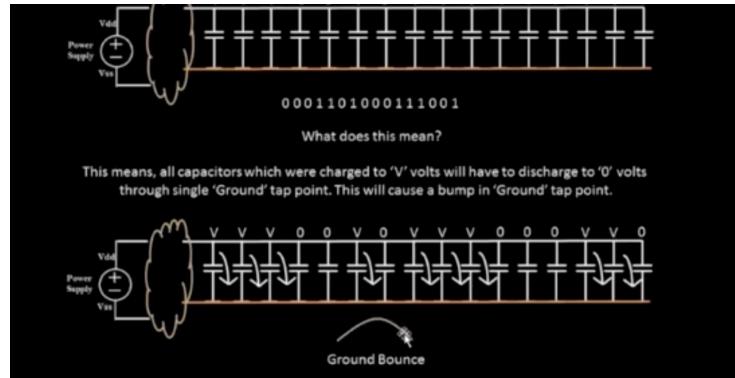


Define width + height of die:

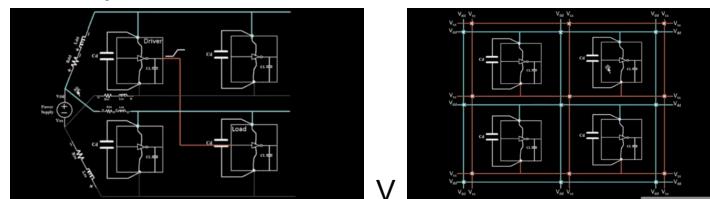
1. Begin with netlist
  - Add up widths + heights of all standard cells (wires later) (area of this = min area)
  - Utilization factor - area used by netlist/total core area
  - Aspect ratio = height/width
2. Define Locations of Preplaced cells
  - Combinational logic does some function
  - Separate into blocks to implement separately (modulation advantage of reuse - like functions)
  - Blackbox (only see IO)
  - Put before routing, not touched (implement once, use many times)
  - Place manually based on specs
3. Physical Spacing
  - Physical drop = resistance, so other parts of circuit won't reach power rail
  - Noise margin summary = range to still be considered logic 1 (this is problem with too much distance from power supply, solved by decoupling capacitors)



4. Power Cell placement
  - Load receives same signal as driver
  - Ground bounce due to inverter causing all capacitors to discharge, not '0', danger of undefined zone



- Can also get ground droop when all try to get energy from one line
- Solve by more power supplies, distributes strain



- (more power supply lines  
= less strain, less ground fluctuation)

## 5. Pin placement

- Interclock timing - units driven by separate clocks
- Verilog defines netlist (connectivity formation between gates)
- Front end defines netlist, backend decides pin placement
- Pick most essential connections - i.e. clocks bc flip flops - to go closest to pins

## README - vars

```
# Variables information
This page describes configuration variables and their default values.

# Required variables
variable | description
-----|-----
DESIGN_NAME | The name of the top level module of the design
CLK_PERIOD | The clock period for the design in ns
CLK_SOURCE | The source of the clock for the design
CLK_PORT | The name of the design's clock port

# Optional variables
These variables are optional that can be specified in the design configuration file.

# Synthesis
variable | description
-----|-----
LIB_SYNTH | The library used for synthesis by ares_vark (default: ./pkgs/ref_skywater180-EPBAC/lib.ref/library/ef04n/ef04n.tl_1.00v_25C.lib)
SYNTH_DRIVING_CELL_PDN | The name of the output driving cell output pin _drvndr (default: 'Y')
SYNTH_DRIVING_CELL_PDN_N | The name of the output driving cell output pin _drvndrn (default: 'Z')
SYNTH_MAX_FANOUT | The max load that the output ports can drive (ars) (default: 5 cells)
SYNTH_MAX_FANIN | The max load that the input ports can drive (ars) (default: 5 cells)
SYNTH_STRAATEGY | Strategies for ars logic synthesis and technology mapping doc Possible values are 0, 1 (delay), 2, and 3 (area) (default: 2)
SYNTH_SIZING | Enables ars cell sizing (instead of buffering) doc Enabled = 1. Disabled = 0 (ars) (default: '0')
SYNTH_VT | To preserve gate instances in the vts of the design, vts enabled = 1. Disabled = 0 (ars) (default: '1')
LIB_DRC | Library used for max delay calculation during STA. Docs /skylane/EPBAC/lib.ref/library/ef04n/ef04n.tl_1.00v_25C.lib
LIB_HDL | Library used for max delay calculation during STA. Docs /skylane/EPBAC/lib.ref/library/ef04n/ef04n.tl_1.00v_25C.lib
LIB_PHYSICAL | Library used for physical delay calculation during STA. Docs /skylane/EPBAC/lib.ref/library/ef04n/ef04n.tl_1.00v_25C.lib
CLK_BUFFER | Most close surface of the clock tree. ars (default: arsnd_clock 10 )
CLK_BUFFER_INPUT | Input pin of the clock tree. ars (default: arsnd_clock 4 )
CLK_BUFFER_OUTPUT | Output pin of the clock tree buffer. ars (default: x )

# Floorplanning
Variables | description
-----|-----
PC_CORE_UTIL | The core utilization percentage. ars (default: 10 percent)
PC_MARGIN_WAIST | The margin around the core area. ars (default: 0.05 micrometers)
PC_CORE_MARGIN | The length of the margin surrounding the core area. ars (default: 0.05 micrometers)
PC_IN_WAIST | The metal layer on which to place the pins vertically inside of the core. ars (default: '3')
PC_IN_VT | The metal layer on which to place the pins vertically outside of the core. ars (default: '2')
PC_OUT_WAIST | The metal layer on which to place the pins vertically inside of the core. ars (default: '3')
PC_OUT_VT | The metal layer on which to place the pins vertically outside of the core. ars (default: '2')
PC_M4_VT | The pitch of the vertical power stripes on the metal layer 4 in the power distribution network ars (default: '153.8 ')
PC_M5_VT | The pitch of the vertical power stripes on the metal layer 5 in the power distribution network ars (default: '153.8 ')
PC_M6_VT | The pitch of the horizontal power stripes on the metal layer 5 in the power distribution network ars (default: '153.8 ')
PC_IN_VTENDO | Extends the vertical in pins outside of the die by the specified units ars (default: 1) Disabled = 0
PC_IN_VTHICKNESS | The thickness of the pins layer underneath the metal layer ars (default: 1 micrometers)
PC_IN_VTHICKNESS_MULT | A multiplier for vertical pin thickness. Base thickness is the pins layer underneath ars (Default: 3 )
PC_IN_VTHICKNESS_MULT_X | A multiplier for horizontal pin thickness. Base thickness is the pins layer underneath ars (Default: 1 )
PC_PLACEMENT | Placement strategy for the pins. ars (default: 'center')

# Placement
PL_SPARSE | Description
-----|-----
PL_SPARSE | The target density of the placement. ars (default: 0.05)
PL_TIME_DRIVEN | Specifies whether the placer would use time driven placement. 0 = False, 1 = true ars (default: '0')
PL_LIB | Specifies the library for time driven placement ars (default: 'LIB_TYPICAL')

# CTS
Variables | description
-----|-----
CTS_BUFFER_SWF | The target clock skew in picoseconds. ars (default: 20 ps)
CTS_WAIT_SWF | The time of cell inserted at the root of the clock tree. ars (default: arsbuf 10 )

# H
```

```

# Copyright 2020 Efabless Corporation
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Floorplan defaults
set ::env(FP_SIZING) relative
set ::env(FP_CORE_UTIL) 50
# PL_TARGET_DENSITY default value set in all.tcl because of the order of sourcing.
set ::env(FP_ASPECT_RATIO) 1
set ::env(FP_PDN_SKIPTRIM) 0

```

inside floorplan.tcl, with all defaults: [floorplan.tcl](#)

```

design
set ::env(DESIGN_NAME) "picorv32a"
set ::env(VERILOG_FILES) ".:/designs/picorv32a/src/picorv32a.v"
set ::env(SOC_FILE) ".:/designs/picorv32a/src/picorv32a.soc"
set ::env(CLOCK_PERIOD) "10.000"
set ::env(CLOCK_PORT) "clk"

set ::env(CLOCK_NET) $::env(CLOCK_PORT)
set ::env(FP_CORE_UTIL) 65
set ::env(FP_PLACEMENT_ORDER) 1
set ::env(FP_IO_BIAS) 3
set ::env(FP_IO_VTOL) 1

set filename $::env(OPENLANE_ROOT)/designs/$::env(DESIGN_NAME)/$::env(PDK_VARIANT)_config.tcl
if { [file exists $filename] == 1 } {
    source $filename
}
info::tcl::info

```

Check all info in config.tcl in picorv32a:

Horizontal + vertical method one more than specified

Run floorplan:

```

% run_floorplan
[STEP 5]
[INFO]: Running Initial Floorplanning (log: designs/ci/picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan/5-initial_fp.log)...
[INFO]: Floorplanned with width 530.84 and height 530.4.
[STEP 6]
[INFO]: Running IO Placement (log: designs/ci/picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan/6-io.log)...
[STEP 7]
[INFO]: Running Tap/Decap Insertion (log: designs/ci/picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan/7-tap.log)...
[INFO]: Power planning with power {VPWR} and ground {VGND}...
[STEP 8]
[INFO]: Generating PDN (log: designs/ci/picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan/8-pdn.log)...
%
```

```

picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan$ ls -ltr
total 16
-rw-r--r-- 1 root root 2206 Jul 17 06:36 5-initial_fp.log
-rw-r--r-- 1 root root     0 Jul 17 06:36 5-initial_fp.warnings
-rw-r--r-- 1 root root     0 Jul 17 06:36 5-initial_fp.errors
-rw-r--r-- 1 root root 1739 Jul 17 06:36 6-io.log
-rw-r--r-- 1 root root     0 Jul 17 06:36 6-io.warnings
-rw-r--r-- 1 root root     0 Jul 17 06:36 6-io.errors
-rw-r--r-- 1 root root 1372 Jul 17 06:36 7-tap.log
-rw-r--r-- 1 root root     0 Jul 17 06:36 7-tap.warnings
-rw-r--r-- 1 root root     0 Jul 17 06:36 7-tap.errors
-rw-r--r-- 1 root root 1510 Jul 17 06:36 8-pdn.log
-rw-r--r-- 1 root root     0 Jul 17 06:36 8-pdn.warnings
-rw-r--r-- 1 root root     0 Jul 17 06:36 8-pdn.errors
beaver@openlanevm:~/Desktop/work/tools/openlane_working_dir/openlane/designs/ci/picorv32a/runs/RUN_2025.07.17_12.41.15/logs/floorplan$ 

```

Cd logs/floorplan/placement:

Config.tcl overrides system default

Config.tcl in run shows used info in run

Design config.tcl not auto override

Config.tcl in picorv32a ultimate override

After running: bash <(curl -s

<https://raw.githubusercontent.com/ZimengXiong/bASICs-openlane-apple-silicon-nvm/refs/heads/main/patch-1.sh>

In terminal, launched openlane. In designs/ci:

```
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 aes_core
drwxrwxr-x 9 openlane-user 1000 1000 4096 Jul 16 05:49 aes_user_project_wrapper
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 blabla
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 inverter
drwxrwxr-x 2 openlane-user 1000 1000 4096 Jul 16 05:49 gcd
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 caravel_upw
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 io_placer
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 picorv32a
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 mem_1r1w
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 manual_macro_placement_te
st
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 regfile_2rlw
drwxrwxr-x 5 openlane-user 1000 1000 4096 Jul 16 05:49 reproducibles
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 salsa20
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 s44
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 test_sram_macro
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 wbqsipflash
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 usb_cdc_core
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 usb
drwxrwxr-x 2 openlane-user 1000 1000 4096 Jul 16 05:49 tt05_i2c_bert
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 zipdiv
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 y_huff
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 xtea
OpenLane Container:/openlane/designs/ci%
```

Cd into picorv32a, in which there are a few items. Config.tcl bypasses configurations (one step above defaults –overrides)

```
# Design
set ::env(DESIGN_NAME) "picorv32a"
set ::env(VERILOG_FILES) "./designs/picorv32a/src/picorv32a.v"
set ::env(SDC_FILE) "./designs/picorv32a/src/picorv32a.sdc"

set ::env(CLOCK_PERIOD) "20.000"
set ::env(CLOCK_PORT) "clk"

set ::env(CLOCK_NET) $::env(CLOCK_PORT)

set filename $::env(OPENLANE_ROOT)/designs/$::env(DESIGN_NAME)/$::env(PDK)_$::env(PDK_VARIANT)_config.tcl
if { [ file exists $filename ] == 1 } {
    source $filename
}
config.tcl (END)
```

./flow.tcl -interactive

package require openlane

prep -design designs/ci/picorv32a

```
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 io_placer
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 mem_1r1w
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 manual_macro_placement_te
st
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 05:49 regfile_2rlw
drwxrwxr-x 5 openlane-user 1000 1000 4096 Jul 16 05:49 reproducibles
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 salsa20
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 s44
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 test_sram_macro
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 wbqsipflash
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 usb_cdc_core
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 usb
drwxrwxr-x 2 openlane-user 1000 1000 4096 Jul 16 05:49 tt05_i2c_bert
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 zipdiv
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 y_huff
drwxrwxr-x 3 openlane-user 1000 1000 4096 Jul 16 05:49 xtea
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 06:24 picorv32a
OpenLane Container:/openlane/designs/ci% cd picorv32a
OpenLane Container:/openlane/designs/ci/picorv32a% ls -ltd
drwxrwxr-x 4 openlane-user 1000 1000 4096 Jul 16 06:24 .
OpenLane Container:/openlane/designs/ci/picorv32a%
```

Successful item creation :

Runs folder, showing dates of days prep -design, with one day opened:

```
% pwd
/openlane/designs/ci/picorv32a/runs
% ls -ltr
total 0
drwxr-xr-x 6 root root 4096 Jul 16 06:26 RUN_2025.07.16_06.24.58
drwxr-xr-x 6 root root 4096 Jul 16 22:42 RUN_2025.07.16_22.41.58
% cd RUN_2025.07.16_22.41.58
% ls -ltr
total 0
drwxr-xr-x 1 root root 51 Jul 14 14:39 PDK_SOURCES
drwxr-xr-x 1 root root 674 Jul 16 22:41 config_in.tcl
drwxr-xr-x 8 root root 4096 Jul 16 22:41 reports
drwxr-xr-x 8 root root 4096 Jul 16 22:41 logs
drwxr-xr-x 8 root root 4096 Jul 16 22:41 results
drwxr-xr-x 8 root root 4096 Jul 16 22:41 tmp
drwxr-xr-x 1 root root 18685 Jul 16 22:42 config.tcl
drwxr-xr-x 1 root root 3783 Jul 16 22:42 cmd.log
drwxr-xr-x 1 root root 8 Jul 16 22:42 OPENLANE_VERSION
drwxr-xr-x 1 root root 525 Jul 16 22:42 openlane.log
drwxr-xr-x 1 root root 254 Jul 16 22:42 warnings.log
drwxr-xr-x 1 root root 79 Jul 16 22:42 runtime.yaml
%
```

Nothing in other files bc nothing run yet

Tmp directory (w/ merge ... .lef files created:

```
% cd tmp
% ls -ltr
total 4224
drwxr-xr-x 2 root root 4096 Jul 16 22:41 placement
drwxr-xr-x 2 root root 4096 Jul 16 22:41 floorplan
drwxr-xr-x 2 root root 4096 Jul 16 22:41 signoff
drwxr-xr-x 1 root root 28 Jul 16 22:41 layers.list
drwxr-xr-x 1 root root 1429507 Jul 16 22:41 merged.non.lef
drwxr-xr-x 1 root root 1429503 Jul 16 22:41 merged.min.lef
drwxr-xr-x 1 root root 1429506 Jul 16 22:41 merged.max.lef
drwxr-xr-x 2 root root 4096 Jul 16 22:41 synthesis
drwxr-xr-x 2 root root 4096 Jul 16 22:42 cts
drwxr-xr-x 2 root root 4096 Jul 16 22:42 routing
%
```

Config\_in.tcl shows all parameters for run - updates after run floorplan (good

to check modifications, flow correctly represented):

Cmds.log (record all commands):

Flop ratio: # D flip flops/cells

```
/* Generated by Yosys 0.38 (git sha1 543faed9c8c, clang++ 16.0.6 -fPIC -Os)
module picorv32(clk, resetn, trap, mem_valid, mem_instr, mem_ready, mem_addr,
    _m_wdata, mem_wstrb, mem_rdata, mem_la_read, mem_la_write, mem_la_addr, mem_la,
    _ata, mem_la_wstrb, pcpi_valid, pcpi_insn, pcpi_rsl, pcpi_rsz, pcpi_wr, pcpi,
    , pcpi_wait, pcpi_ready, irq, eoi, trace_valid, trace_data);
    wire _00000;
    wire _00001;
    wire _00002;
    wire _00003;
    wire _00004;
    wire _00005;
    wire _00006;
    wire _00007;
    wire _00008;
    wire _00009;
    wire _00010;
    wire _00011;
    wire _00012;
    wire _00013;
    wire _00014;
    wire _00015;
    wire _00016;
```

Proof of synthesis running:

status report, with all info for flop ratios:

```
63. Printing statistics.
== plicrv32 ==
Number of wires: 8989
Number of wire bits: 9371
Number of public wires: 1512
Number of public wire bits: 1894
Number of memory: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 9269
sky130_fd_sc_hd_a2110_2 8
sky130_fd_sc_hd_a2110_3 66
sky130_fd_sc_hd_a2110_2 8
sky130_fd_sc_hd_a2100_2 17
sky130_fd_sc_hd_a2100_2 5
sky130_fd_sc_hd_a2101_2 240
sky130_fd_sc_hd_a2101_2 144
sky130_fd_sc_hd_a2210_2 104
sky130_fd_sc_hd_a2210_2 2
sky130_fd_sc_hd_a2210_2 182
```

```
beaver@openlanenvm:~/Desktop/work/tools/openlane_working_dir/openlane/designs/ci
/picorv32a/runs/RUN_2025.07.17_12.41.15/results/floorplan$ ----- has
picorv32.def file (design exchange format):
```

```
VERSION 5.8 ;
DIVIDERCHAR "/";
BUSBITCHARS "[]";
DESIGN picorv32;
UNITS DISTANCE MICRONS 1000 ;
DIAREA ( 0 0 ) ( 542325 553045 ) ;
ROW ROW_0 unithd 5520 10880 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_1 unithd 5520 13600 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_2 unithd 5520 16320 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_3 unithd 5520 19040 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_4 unithd 5520 21760 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_5 unithd 5520 24480 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_6 unithd 5520 27200 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_7 unithd 5520 29920 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_8 unithd 5520 32640 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_9 unithd 5520 35360 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_10 unithd 5520 38080 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_11 unithd 5520 40800 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_12 unithd 5520 43520 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_13 unithd 5520 46240 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_14 unithd 5520 48960 N DO 1154 BY 1 STEP 460 0 ;
ROW ROW_15 unithd 5520 51680 FS DO 1154 BY 1 STEP 460 0 ;
ROW ROW_16 unithd 5520 54400 N DO 1154 BY 1 STEP 460 0 ;
picorv32.def
```

DIAREA = area of counted die ()(upperrightx, upperrighty)

Pay attention to units (microns)

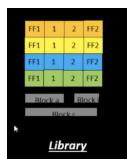
Floorplanning doesn't do power distribution, but does to + and - rails

Tap cells: (connect substrate to ground, README shows distances between taps)



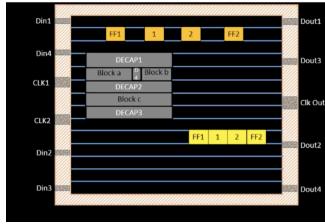
Make changes in design related config file

All symbols in logic gate have physical dimension as boxes (groups called libraries have info, sublibraries (ex. Shape(can change dimension, bigger=faster, bc less delay), dimensions, delay, required conditions of cell



for output)):

For placement, place gates close to IO pins they need to access to limit delay:



Optimize placement:

With distance : Capacitance (ability to hold charge) increases, resistance increases

Solution: signal integrating (middleman) - repeaters/buffers(replicate + send signal)

Slew - higher capacitance = high slew (needs more to charge)

Decide based on distance if needs buffer

Placed right next to each other = negligible time delay (basically none)

Clump if particular segment must work all together in time perfectly (ex high frequency)

Wires can go in different layers, enabling criss-cross

Ideal clocks + timing analysis (know if placement is reasonable) - Want clocks to reach flipflops as quickly as possible

Library characterization and modelling

Logic synthesis

1. gate representation doing functionality
2. Floor planning - decide size core + die (dependent on gates number, size, shape)
3. Placement - place logic cells so timing is optimized
4. Clock resynthesis - want clock signal to reach gates at same time (buffers ensure equal rise + fall time)
5. Routing - flow between cells based on cell characteristics
6. Static timing analysis - setup time (track via capture flop + launch



flop), max frequency

Need to design so EDA understands cells + functions

Condition strained placement before timing comes into play

2 types placement:

- Global - optimal wirelength focus (if overflow value decreases, will converge
- Detailed (legalization - no overlap, in rows. Timing considerations)

Run\_placement (in interactive mode of launcher)

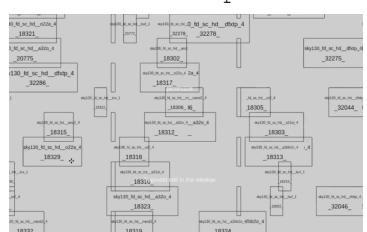
To view w/ magic: (in launcher, non-interactive mode, openlane)

```
% magic -T ../pdks/sky130A/libs.tech/magic/sky130A.tech lef read
designs/ci/picorv32a/runs/RUN_2025.07.17_16.08.33/tmp/merged.nom.lef def read
```

designs/ci/picorv32a/runs/RUN\_2025.07.17\_16.08.33/results/placement/picorv32.def &

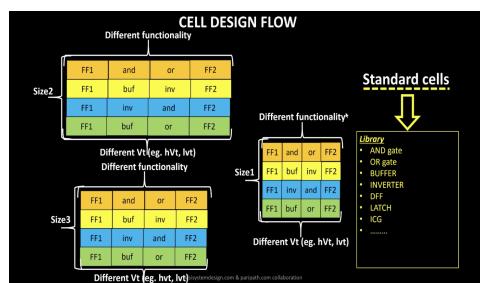
A screenshot of the openLane tool interface. The main window displays a dense, multi-layered circuit diagram with various components and connections. A legend on the right side lists numerous component types, each associated with a small colored square: errors, labels, subcell, dwells, nwell, pwell, nwells, pwells, oswells, pseeds, nbase, mtransistor, scratransistor, srammet, pmtransistor, nmtransistor, scratransistor, sramlet, nmtransistor, mntransistor, mptransistor, mntransistor, mvtransistor, mvtransistor, mvtransistor, varactor, and mvvaractor. The top status bar shows the command line: "beaver@openlanevm:~/Desktop/work/tools/openlane\_working\_dir/openlane\$ cd designs/c1/picrv32a/runs/RUN\_2025\_07.17\_15.09.05" and the current file path: "openLane.log OPENLANE\_VERSION PDK\_SOURCES reports results runtime.yaml tmp warn logs.log". The bottom status bar shows the IP address "7.15.09.05\$ cd ../../" and the terminal prompt "beaver@openlanevm:~\$".

(s) select  
(g) center  
(z) zoom in  
(left click + right click) define box of selection  
(shift + z) zoom out  
(s object + type in console  
what) - shows layer  
(No DRC yet (checks if can be manufactured correctly))  
Standard cell placement:



## Cell Designs Flow:

Img showing aspect library stores:



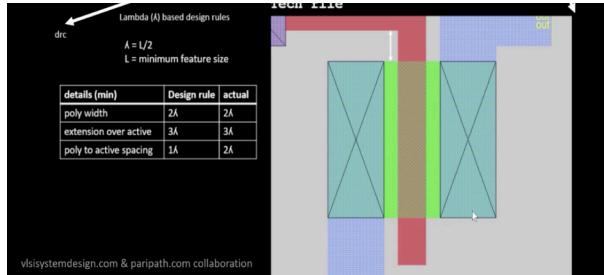
Libraries store:

- Size (high drive strength buffer (library classifies))
  - Threshold voltage (lower = faster)
  - functionality

Even small gates have to undergo cell design flow (IO, design steps)

## Inputs:

- PDKs (from foundry)
- DRC + LVS rules (tech file rules for foundry fabrication - poly width (design width), extension over active (extension over connection zone), poly to active spacing)



- SPICE models (threshold voltage equation - standard values from foundry, ex  $V_{to}$  = threshold voltage in absence of bias,

SPICE model parameters

Threshold Voltage Equation:

$$V_t = \frac{V_{ds} \cdot r_s \cdot \sqrt{1 - 2 \cdot \phi_f}}{\phi_m \cdot C_{ox}}$$

Linear region:

$$I_d = k_n \cdot (V_{gs} - V_t) \cdot V_{ds} / 2$$

Saturation region:

$$I_d = \frac{k_n \cdot W}{L} \cdot (V_{gs} - V_t)^2 / 2$$

capacitances)

- Library(ex cell-height (separation power +ground), drive strength (higher = longer wires allowed)
- user-defined specs (supply voltage (take care noise range, metal layers(what level), pin location, drawn gate-length)

Must design cell that aligns w/ inputs

Design steps:

- Circuit design
  - Implement function
  - Model pmos and nmos meet specs



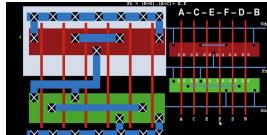
image of circuit design flow: [Screenshot of a circuit simulation software showing a complex network graph of nodes and connections, representing the implementation of a function through pmos and nmos components.] (things like  $V_m$  (switching threshold value), required current) (outputs CDL circuit description language)

- Layout design (implement circuit design) (best performance + area)
  - Function implemented through pmos + nmos
  - Make pmos and nmos network graph (get Euler's path - path that traverses every edge exactly once, then get stick diagram)
  - Output
    - CDL
    - gdsII (typical layout file)
    - Extracted spice netlist(.cir) (capacitances, resistances, my\_inv.sub- subcircuit(description, spice list, nmos model, pmos model info))

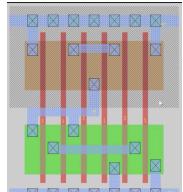


Euler's path diagram + path:

Finished layout diagram w/ specs/Stick diagram (connect sources + drains of



transistors):

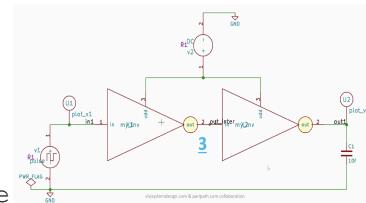


Cell layout: (all info known, area etc.)

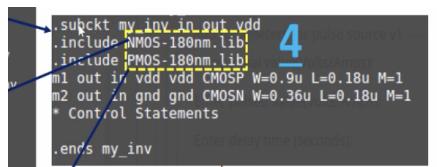
- Characterization (timing, noise, power info)



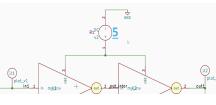
- Read in models + tech file
- Read extracted spicenet list



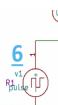
- Autorecognize behavior of piece



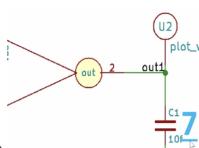
- Read subcircuits of piece



- Attach power sources



- Apply stimulus



- Provide output capacitance

- Provide necessary simulation command (.what kind of simulation)

```

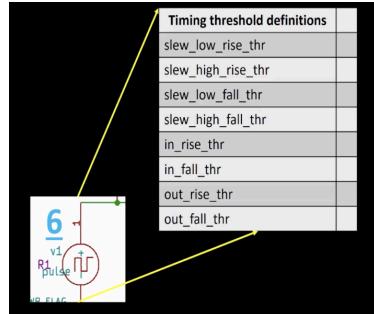
x1 in1 net- xl-pad2_ net- xl-pad3_ my_inv
c1 out1 gnd 10f
x2 net- xl-pad2_ out1 net- xl-pad3_ my_inv
.tran 10e-12 4e-09 0e+00
* Control Statements
.control
run

```

8                    2

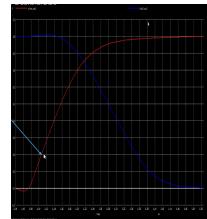
Feed all 8 steps in config file to characterization app GUNA (generates power noise timing characterization)

Timing characterization:

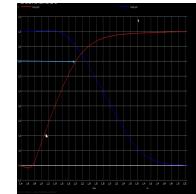


Vars related to threshold:

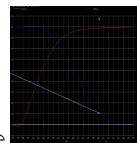
Slew\_low\_rise\_thr = output (blue) first inverter input (red) to second (slew =



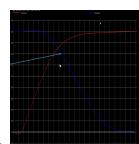
input (20 % from bottom) ~20-30%):



Slew\_high\_rise\_thr = (20 % to top timing diff from 20% bottom)

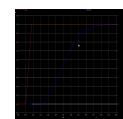


Slew\_low\_fall\_thr = same but for blue



Slew\_high\_fall\_thr = same but for blue

in/out\_rise/fall\_thr = (red = input, blue = output buffer) :  
- Ir = 50% point red



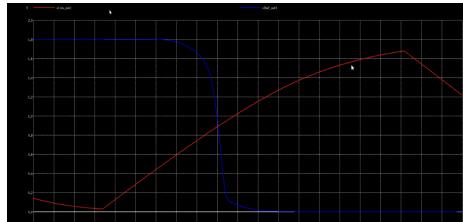
- Or = 50% blue
- (now can calculate delay between these points)
- (can do same for fall)

| Timing threshold definitions |     |
|------------------------------|-----|
| slew_low_rise_thr            | 20% |
| slew_high_rise_thr           | 80% |
| slew_low_fall_thr            | 20% |
| slew_high_fall_thr           | 80% |
| in_rise_thr                  | 50% |
| in_fall_thr                  | 50% |
| out_rise_thr                 | 50% |
| out_fall_thr                 | 50% |

Typical values of timing threshold definitions:

Propagation Delay:

- $\text{time}(\text{in\_}_*\text{thr}) - \text{time}(\text{out\_}_*\text{thr}) = \text{delay}$
- (can be -)



- Long wires = huge slew
- Even with proper thresholds can be -, bc spacing off
- Transition time:
  - $\text{time}(\text{slew\_high\_rise\_thr}) - \text{time}(\text{slew\_low\_rise\_thr})$
  - $\text{time}(\text{slew\_high\_fall\_thr}) - \text{time}(\text{slew\_low\_fall\_thr})$

Slew = time between low and high rise