WEEK 3: PART 1

BY: SAKSHAM RAO

EXPLORE THE FOLDERS:

```
❯ cd VSDBabySoC

VSDBabySoC on  main
❯ ls -l
total 84
drwxrwxr-x  2 srao srao  4096 Sep 28 23:22 images
-rw-rw-r--  1 srao srao 11357 Sep 28 23:22 LICENSE
-rw-rw-r--  1 srao srao  6583 Sep 28 23:22 Makefile
drwxrwxr-x  5 srao srao  4096 Sep 28 23:48 output
-rw-rw-r--  1 srao srao 49350 Sep 28 23:22 README.md
drwxrwxr-x 11 srao srao  4096 Sep 28 23:22 src

VSDBabySoC on  main
❯ cd src

VSDBabySoC/src on  main
❯ ls -l
total 36
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 gds
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 gls_model
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 include
drwxrwxr-x 4 srao srao 4096 Sep 28 23:22 layout_conf
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 lef
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 lib
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 module
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 script
drwxrwxr-x 2 srao srao 4096 Sep 28 23:22 sdc

VSDBabySoC/src on  main
❯ cd module

VSDBabySoC/src/module on  main via V
❯ ls -l
total 44
-rw-rw-r-- 1 srao srao  1122 Sep 28 23:22 avsddac.v
-rw-rw-r-- 1 srao srao   947 Sep 28 23:22 avsdpll.v
-rw-rw-r-- 1 srao srao  1680 Sep 28 23:22 clk_gate.v
-rw-rw-r-- 1 srao srao   908 Sep 28 23:22 pseudo_rand_gen.sv
-rw-rw-r-- 1 srao srao  3108 Sep 28 23:22 pseudo_rand.sv
-rw-rw-r-- 1 srao srao 10492 Sep 28 23:22 rvmyth.tlv
-rw-rw-r-- 1 srao srao   603 Sep 28 23:22 testbench.rvmyth.post-routing.v
-rwxrwxr-x 1 srao srao  1256 Sep 28 23:22 testbench.v
-rw-rw-r-- 1 srao srao   590 Sep 28 23:22 vsdbabysoc.v
```

COMMANDS TO RUN in VSDBabySOC/src:

synth -top vsdbabysoc

dfflibmap -liberty ./lib/sky130_fd_sc_hd__tt_025C_1v80.lib

opt

abc -liberty ./lib/sky130_fd_sc_hd__tt_025C_1v80.lib -script +strash;scorr;ifraig;retime;{D};strash;dch,-f;map,-M,1,{D}

flatten

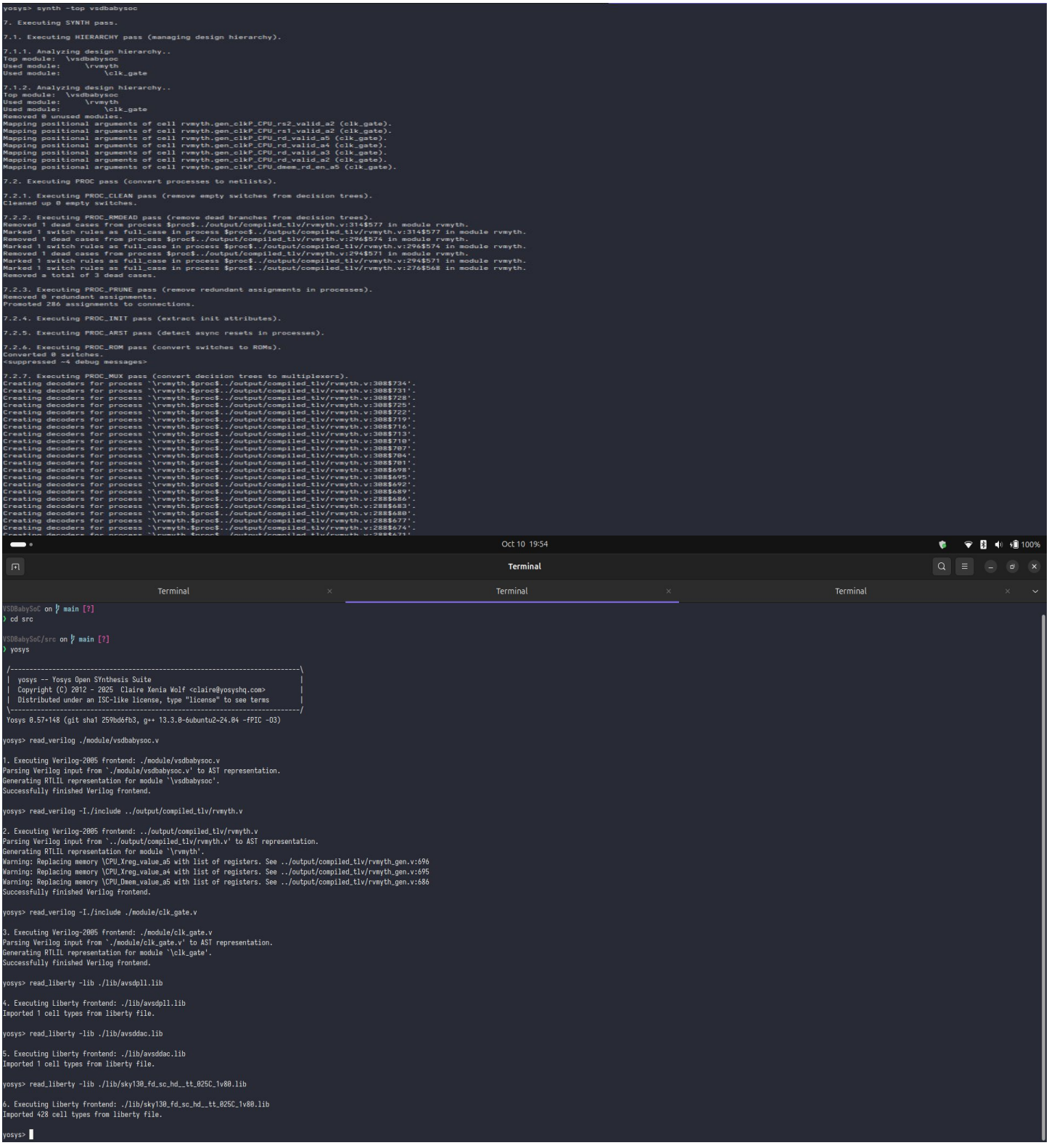
setundef -zero

clean -purge

rename -enumerate

stat

write_verilog -noattr ../output/synth/vsdbabysoc.synth.v



```
        1   avsddac
        1   avsdpll
        1 submodules
        1   rvmyth

7.26. Executing CHECK pass (checking for obvious problems).
Checking module clk_gate...
Checking module rvmyth...
Checking module vsdbabysoc...
Found and reported 0 problems.

yosys> dfflibmap -liberty ./lib/sky130_fd_sc_hd__tt_025C_1v80.lib

8. Executing DFFLIBMAP pass (mapping DFF cells to sequential cells from liberty file).
  cell sky130_fd_sc_hd__dfxtp_1 (noninv, pins=3, area=20.02) is a direct match for cell type $_DFF_P_.
  cell sky130_fd_sc_hd__dfrtn_1 (noninv, pins=4, area=25.02) is a direct match for cell type $_DFF_NN0_.
  cell sky130_fd_sc_hd__dfrtp_1 (noninv, pins=4, area=25.02) is a direct match for cell type $_DFF_PN0_.
  cell sky130_fd_sc_hd__dfstp_2 (noninv, pins=4, area=26.28) is a direct match for cell type $_DFF_PN1_.
  cell sky130_fd_sc_hd__edfxtp_1 (noninv, pins=4, area=30.03) is a direct match for cell type $_DFFE_PP_.
  cell sky130_fd_sc_hd__dfbbn_1 (noninv, pins=6, area=32.53) is a direct match for cell type $_DFFSR_NNN_.
  cell sky130_fd_sc_hd__dfbbp_1 (noninv, pins=6, area=32.53) is a direct match for cell type $_DFFSR_PNN_.
  final dff cell mappings:
    unmapped dff cell: $_DFF_N_
    \sky130_fd_sc_hd__dfxtp_1 _DFF_P_ (.CLK( C), .D( D), .Q( Q));
    \sky130_fd_sc_hd__dfrtn_1 _DFF_NN0_ (.CLK_N( C), .D( D), .Q( Q), .RESET_B( R));
    unmapped dff cell: $_DFF_NN1_
    unmapped dff cell: $_DFF_NP0_
    unmapped dff cell: $_DFF_NP1_
    \sky130_fd_sc_hd__dfrtp_1 _DFF_PN0_ (.CLK( C), .D( D), .Q( Q), .RESET_B( R));
    \sky130_fd_sc_hd__dfstp_2 _DFF_PN1_ (.CLK( C), .D( D), .Q( Q), .SET_B( R));
    unmapped dff cell: $_DFF_PP0_
    unmapped dff cell: $_DFF_PP1_
    unmapped dff cell: $_DFFE_NN_
    unmapped dff cell: $_DFFE_NP_
    unmapped dff cell: $_DFFE_PN_
    \sky130_fd_sc_hd__edfxtp_1 _DFFE_PP_ (.CLK( C), .D( D), .DE( E), .Q( Q));
    \sky130_fd_sc_hd__dfbbn_1 _DFFSR_NNN_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q), .RESET_B( R), .SET_B( S));
    unmapped dff cell: $_DFFSR_NNP_
    unmapped dff cell: $_DFFSR_NPN_
    unmapped dff cell: $_DFFSR_NPP_
    \sky130_fd_sc_hd__dfbbp_1 _DFFSR_PNN_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .RESET_B( R), .SET_B( S));
    unmapped dff cell: $_DFFSR_PNP_
    unmapped dff cell: $_DFFSR_PPN_
    unmapped dff cell: $_DFFSR_PPP_

8.1. Executing DFFLEGALIZE pass (convert FFs to types supported by the target).
<suppressed ~24 debug messages>
Mapping DFF cells in module `\clk_gate':
Mapping DFF cells in module `\rvmyth':
  mapped 1273 $_DFF_P_ cells to \sky130_fd_sc_hd__dfxtp_1 cells.
Mapping DFF cells in module `\vsdbabysoc':

yosys>
```

```
yosys> opt

9. Executing OPT pass (performing simple optimizations).

9.1. Executing OPT_EXPR pass (perform const folding).
Optimizing module clk_gate.
Optimizing module rvmyth.
Optimizing module vsdbabysoc.

9.2. Executing OPT_MERGE pass (detect identical cells).
Finding identical cells in module `\clk_gate'.
Finding identical cells in module `\rvmyth'.
Finding identical cells in module `\vsdbabysoc'.
Removed a total of 0 cells.

9.3. Executing OPT_MUXTREE pass (detect dead branches in mux trees).
Running muxtree optimizer on module \clk_gate..
  Creating internal representation of mux trees.
  No muxes found in this module.
Running muxtree optimizer on module \rvmyth..
  Creating internal representation of mux trees.
  No muxes found in this module.
Running muxtree optimizer on module \vsdbabysoc..
  Creating internal representation of mux trees.
  No muxes found in this module.
Removed 0 multiplexer ports.

9.4. Executing OPT_REDUCE pass (consolidate $*mux and $reduce_* inputs).
  Optimizing cells in module \clk_gate.
  Optimizing cells in module \rvmyth.
  Optimizing cells in module \vsdbabysoc.
Performed a total of 0 changes.

9.5. Executing OPT_MERGE pass (detect identical cells).
Finding identical cells in module `\clk_gate'.
Finding identical cells in module `\rvmyth'.
Finding identical cells in module `\vsdbabysoc'.
Removed a total of 0 cells.

9.6. Executing OPT_DFF pass (perform DFF optimizations).

9.7. Executing OPT_CLEAN pass (remove unused cells and wires).
Finding unused cells or wires in module \clk_gate..
Finding unused cells or wires in module \rvmyth..
Finding unused cells or wires in module \vsdbabysoc..

9.8. Executing OPT_EXPR pass (perform const folding).
Optimizing module clk_gate.
Optimizing module rvmyth.
Optimizing module vsdbabysoc.

9.9. Finished fast OPT passes. (There is nothing left to do.)
```

```
yosys> abc -liberty ./lib/sky130_fd_sc_hd__tt_025C_1v80.lib -script +strash;scorr;ifraig;retime;{D};strash;dch,-f;map,-M,1,{D}

10. Executing ABC pass (technology mapping using ABC).

10.1. Extracting gate netlist of module `\clk_gate' to `<abc-temp-dir>/input.blif'..
Don't call ABC as there is nothing to map.

10.1.1. Executed ABC.
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Removing temp directory.

10.2. Extracting gate netlist of module `\rvmyth' to `<abc-temp-dir>/input.blif'..

10.2.1. Executed ABC.
Extracted 5942 gates and 7169 wires to a netlist network with 1225 inputs and 1173 outputs.
Running ABC script: <abc-temp-dir>/abc.script
ABC: UC Berkeley, ABC 1.01 (compiled Sep 20 2025 00:00:59)
ABC: abc 01> set abcout /dev/stdout
ABC: abc 01> empty
ABC: abc 01> source <abc-temp-dir>/abc.script
ABC: + read_blif <abc-temp-dir>/input.blif
ABC: + read_lib -w /home/srao/VSDBabySoC/src/./lib/sky130_fd_sc_hd__tt_025C_1v80.lib
ABC: Parsing finished successfully.  Parsing time =     0.05 sec
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__decap_12" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__decap_3" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__decap_4" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__decap_6" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__decap_8" without logic function.
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfbbn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfbbn_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfbbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrbp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrtn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrtp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrtp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfrtp_4".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfsbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfsbp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfstp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfstp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfstp_4".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfxbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfxbp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfxtp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfxtp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dfxtp_4".
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__diode_2" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__dlclkp_1" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__dlclkp_2" without logic function.
ABC: Scl_LibertyReadGenlib() skipped cell "sky130_fd_sc_hd__dlclkp_4" without logic function.
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrbn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrbn_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrbp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtn_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtn_4".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtp_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlrtp_4".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxbn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxbn_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxbp_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxtn_1".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxtn_2".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxtn_4".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "sky130_fd_sc_hd__dlxtp_1".
```

```
yosys> flatten

11. Executing FLATTEN pass (flatten design).
Deleting now unused module clk_gate.
Deleting now unused module rvmyth.
<suppressed ~8 debug messages>

yosys> setundef -zero

12. Executing SETUNDEF pass (replace undef values with defined constants).

yosys> clean -purge
Removed 396 unused cells and 7576 unused wires.

yosys> rename -enumerate

yosys> stat

13. Printing statistics.

=== vsdbabysoc ===

        +----------Local Count, excluding submodules.
        |
     4736 wires
     6210 wire bits
     4736 public wires
     6210 public wire bits
        7 ports
        7 port bits
     5920 cells
        8   $scopeinfo
        1   avsddac
        1   avsdpll
       10   sky130_fd_sc_hd__a2111oi_0
        1   sky130_fd_sc_hd__a211o_2
       26   sky130_fd_sc_hd__a211oi_1
        4   sky130_fd_sc_hd__a21boi_0
        1   sky130_fd_sc_hd__a21o_2
      672   sky130_fd_sc_hd__a21oi_1
        1   sky130_fd_sc_hd__a221o_2
      163   sky130_fd_sc_hd__a221oi_1
        4   sky130_fd_sc_hd__a22o_2
      123   sky130_fd_sc_hd__a22oi_1
        4   sky130_fd_sc_hd__a311oi_1
        1   sky130_fd_sc_hd__a31o_2
      344   sky130_fd_sc_hd__a31oi_1
        2   sky130_fd_sc_hd__a32oi_1
       26   sky130_fd_sc_hd__a41oi_1
       12   sky130_fd_sc_hd__and2_2
        1   sky130_fd_sc_hd__and3_2
      597   sky130_fd_sc_hd__clkinv_1
     1144   sky130_fd_sc_hd__dfxtp_1
        1   sky130_fd_sc_hd__lpflow_inputiso0p_1
       12   sky130_fd_sc_hd__mux2i_1
      839   sky130_fd_sc_hd__nand2_1
      249   sky130_fd_sc_hd__nand3_1
        1   sky130_fd_sc_hd__nand3b_1
       41   sky130_fd_sc_hd__nand4_1
      403   sky130_fd_sc_hd__nor2_1
       35   sky130_fd_sc_hd__nor3_1
        2   sky130_fd_sc_hd__nor4_1
       20   sky130_fd_sc_hd__o2111ai_1
        1   sky130_fd_sc_hd__o211a_1
       49   sky130_fd_sc_hd__o211ai_1
        6   sky130_fd_sc_hd__o21a_1
      866   sky130_fd_sc_hd__o21ai_0
        1   sky130_fd_sc_hd__o21ba_2
       18   sky130_fd_sc_hd__o21bai_1
        1   sky130_fd_sc_hd__o221a_2
        7   sky130_fd_sc_hd__o221ai_1
      155   sky130_fd_sc_hd__o22ai_1
        1   sky130_fd_sc_hd__o2bb2ai_1
```

```
yosys> write_verilog -noattr ../output/synth/vsdbabysoc.synth.v
14. Executing Verilog backend.

14.1. Executing BMUXMAP pass.

14.2. Executing DEMUXMAP pass.
Dumping module `\vsdbabysoc'.

yosys>
```

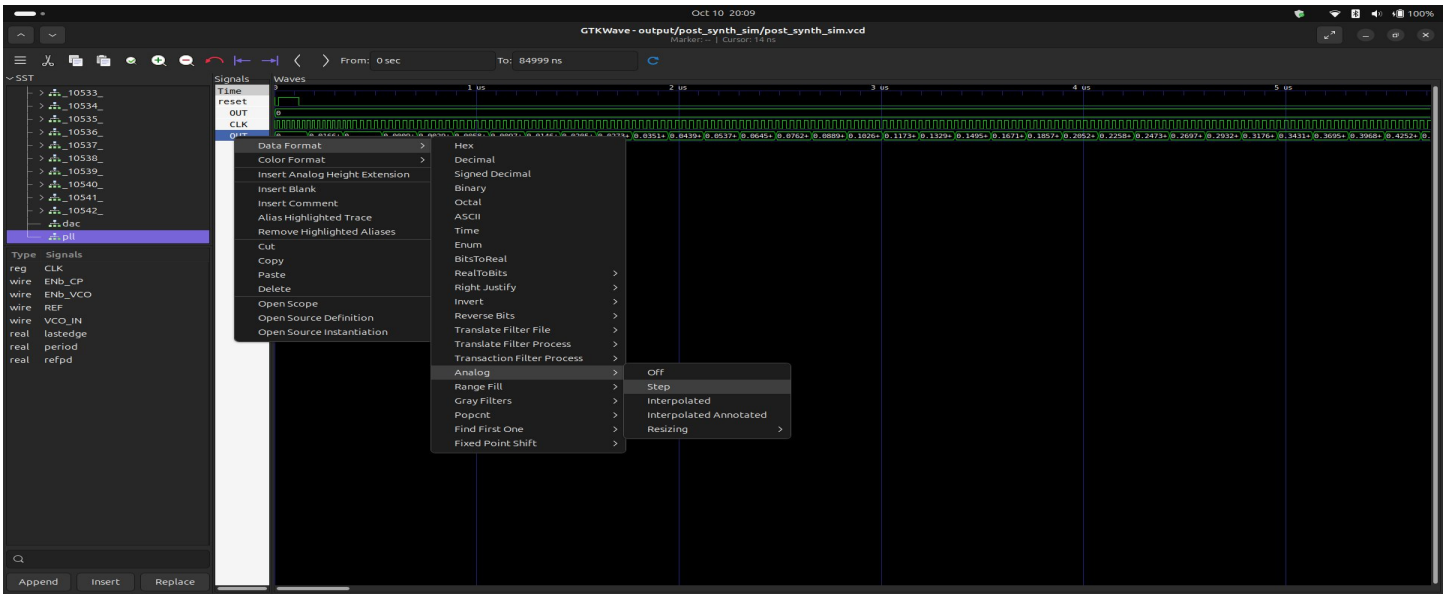
```
VSDBabySoC on  main [?]
❯ mkdir -p output/post_synth_sim

VSDBabySoC on  main [?]
❯ iverilog -o output/post_synth_sim/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUNIT_DELAY=#1 src/module/testbench.v -I src/include -I src/module -I src/gls_model -I output/synth;

VSDBabySoC on  main [?]
❯ cd output/post_synth_sim

VSDBabySoC/output/post_synth_sim on  main [?]
❯ ./post_synth_sim.out
VCD info: dumpfile post_synth_sim.vcd opened for output.
src/module/testbench.v:63: $finish called at 84999000 (1ps)

VSDBabySoC/output/post_synth_sim on  main [?] took 2s
❯
```



THE RESULT FROM WEEK3 part A and WEEK 2 are exactly the same, hence GLS= FUNCTIONAL O/Ps.

WEEK 3: PART 2

BY:SAKSHAM RAO

Static Timing Analysis (STA) – Key Notes

Definition:
STA verifies timing performance of a digital circuit *without simulation* by checking all paths for setup and hold violations using defined clock constraints.

1. Core Concepts

- **Clock Period (Tclk):** Time between active clock edges.
- **Launch Edge:** Clock edge that launches data.
- **Capture Edge:** Clock edge that captures data.
- **Clock Skew:** Difference in clock arrival times.
- **Clock Jitter:** Clock edge variation due to noise.
- **Data Path Delay:** Delay through combinational logic.

2. Setup Check

- **Setup Time (Tsetup):** Data must be stable *before* capture edge.
- **Condition:**
  `Data Arrival ≤ Capture Clock - Tsetup`
- **Setup Slack:**
  `(Clock Period + Capture Clock - Launch Clock) - (Data Delay + Tsetup)`
- **Violation:** Slack < 0 → path too slow.

3. Hold Check

- **Hold Time (Thold):** Data must stay stable *after* capture edge.
- **Condition:**
  `Data Arrival ≥ Capture Clock + Thold`
- **Hold Slack:**
  `(Data Delay + Launch Clock) - (Capture Clock + Thold)`
- **Violation:** Slack < 0 → path too fast; fix by adding delay.

4. Slack

- **Slack = Required Time – Arrival Time**
  - Positive → meets timing
  - Zero → boundary condition
  - Negative → violation

5. Clock Definitions

- Defined using constraints (`create_clock`).
- Includes name, period, waveform, and derived clocks.
- Proper clock definition is crucial for accurate STA.

6. Path-Based Analysis

- **Timing Path:** Startpoint → Combinational Logic → Endpoint
  - Startpoint: Flip-flop/Q or primary input

    - Endpoint: Flip-flop/D or primary output
- Each path is checked for setup and hold independently.
- **Critical Path:** Path with least (most negative) slack; limits clock frequency.

7. Fixing Violations

- **Setup Violation:** Optimize logic, add pipeline, or slow clock.
- **Hold Violation:** Add buffers or delay cells.

Summary:

STA ensures reliable operation at target frequency by mathematically verifying that all data paths meet setup and hold timing across all clocks.

WEEK 3: PART 3

BY:SAKSHAM RAO

```
mkdir -p output/synth; \
        docker run -it --rm \
                -v /home/manili/OpenLane:/openLANE_flow \
                -v /home/manili/OpenLane/pdks:/openLANE_flow/pdks \
                -v /home/srao/VSDBabySoC:/VSDBabySoC \
                -e PDK_ROOT=/openLANE_flow/pdks \
                -u 1000:1000 \
                efabless/openlane:2021.09.09_03.00.48 \
                bash -c "cd /VSDBabySoC/src; yosys -s /VSDBabySoC/src/script/yosys.ys | tee ../output/synth/synth.log"; \

mkdir -p output/sta; \
        docker run -it --rm \
                -v /home/manili/OpenLane:/openLANE_flow \
                -v /home/manili/OpenLane/pdks:/openLANE_flow/pdks \
                -v /home/srao/VSDBabySoC:/VSDBabySoC \
                -e PDK_ROOT=/openLANE_flow/pdks \
                -u 1000:1000 \
                efabless/openlane:2021.09.09_03.00.48 \
                bash -c "cd /VSDBabySoC/src; sta -exit -threads max /VSDBabySoC/src/script/sta.conf | tee ../output/sta/sta.log"; \
```

```
VSDBabySoC on  main [!?] took 4s
❯ mkdir -p output/sta; \
       docker run -it --rm \
               -v /home/srao/OpenLane:/openLANE_flow \
               -v /home/srao/OpenLane/pdks:/openLANE_flow/pdks \
               -v /home/srao/VSDBabySoC:/VSDBabySoC \
               -e PDK_ROOT=/openLANE_flow/pdks \
               -u 1000:1000 \
               efabless/openlane:2021.09.09_03.00.48 \
               bash -c "cd /VSDBabySoC/src; sta -exit -threads max /VSDBabySoC/src/script/sta.conf | tee ../output/sta/sta.log"; \

OpenSTA 2.3.0 21ec025dab Copyright (c) 2019, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type `show_copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
Warning: ./lib/sky130_fd_sc_hd__tt_025C_1v80.lib line 1, library sky130_fd_sc_hd__tt_025C_1v80 already exists.
Warning: ./lib/avsdpll.lib line 1, library avsdpll already exists.
Warning: ./lib/avsddac.lib line 1, library avsddac already exists.
Startpoint: _9532_ (rising edge-triggered flip-flop clocked by clk)
Endpoint: _10034_ (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

  Delay    Time   Description
---------------------------------------------------------
   0.00    0.00   clock clk (rise edge)
   0.00    0.00   clock network delay (ideal)
   0.00    0.00 ^ _9532_/CLK (sky130_fd_sc_hd__dfxtp_1)
   4.40    4.40 ^ _9532_/Q (sky130_fd_sc_hd__dfxtp_1)
   5.06    9.47 v _8103_/Y (sky130_fd_sc_hd__clkinv_1)
   0.54   10.01 ^ _8106_/Y (sky130_fd_sc_hd__o211ai_1)
   0.00   10.01 ^ _10034_/D (sky130_fd_sc_hd__dfxtp_1)
          10.01   data arrival time

  11.00   11.00   clock clk (rise edge)
   0.00   11.00   clock network delay (ideal)
   0.00   11.00   clock reconvergence pessimism
          11.00 ^ _10034_/CLK (sky130_fd_sc_hd__dfxtp_1)
  -0.13   10.87   library setup time
          10.87   data required time
---------------------------------------------------------
          10.87   data required time
         -10.01   data arrival time
---------------------------------------------------------
           0.86   slack (MET)
```