Steps for the Week 3 Assignment of Gate-Level-Synthesis and Functional Simulation

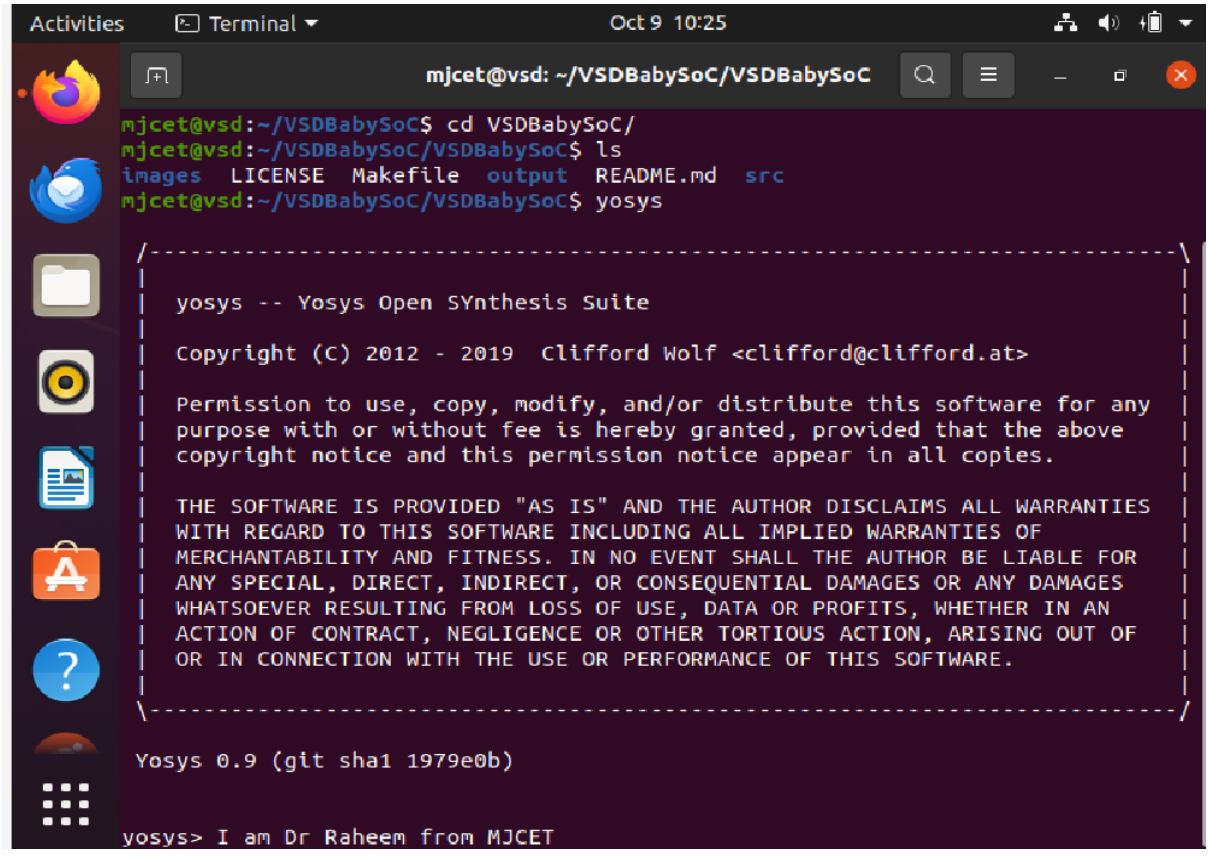
Directory Structure

Make sure the following project structure is set up:

/home/mjcet/VSDBabySoC/VSDBabySoC/

```
├── src/
│   ├── include/        # Verilog include files
│   ├── module/         # Verilog modules (vsdbabysoc.v, clkgate.v, rvmyth.v, etc)
│   └── lib/            # Liberty files (sky130fdschdtt025C1v80.lib, avsddac.lib, avsdpll.lib)
```

Step1 Open the yosys



Load the Top-Level and other Verilog files

Indicate SV (SystemVerilog) support and all include/module paths. Avoid typographical errors in commands or filenames:

Step 2 : Read_verilog

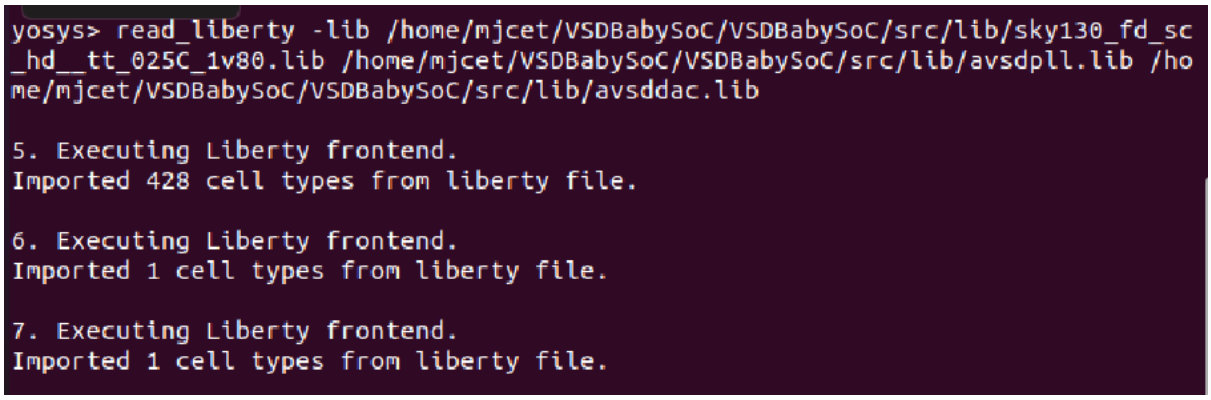
read_verilog -sv /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/vsdbabysoc.v /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth.v

/home/mjcet/VSDBabySoC/VSDBabySoC/src/module/clk_gate.v -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include

Step 3. Load Liberty Cell Libraries

Load the technology cell library (example for Sky130):

read_liberty -lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avdspll.lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avsddac.lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/sky



Step 4 :Set the top-level module for synthesis

Synth -top vsdbabysoc.v

```
=== clk_gate ===

   Number of wires:                 5
   Number of wire bits:             5
   Number of public wires:          5
   Number of public wire bits:      5
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                 0
```

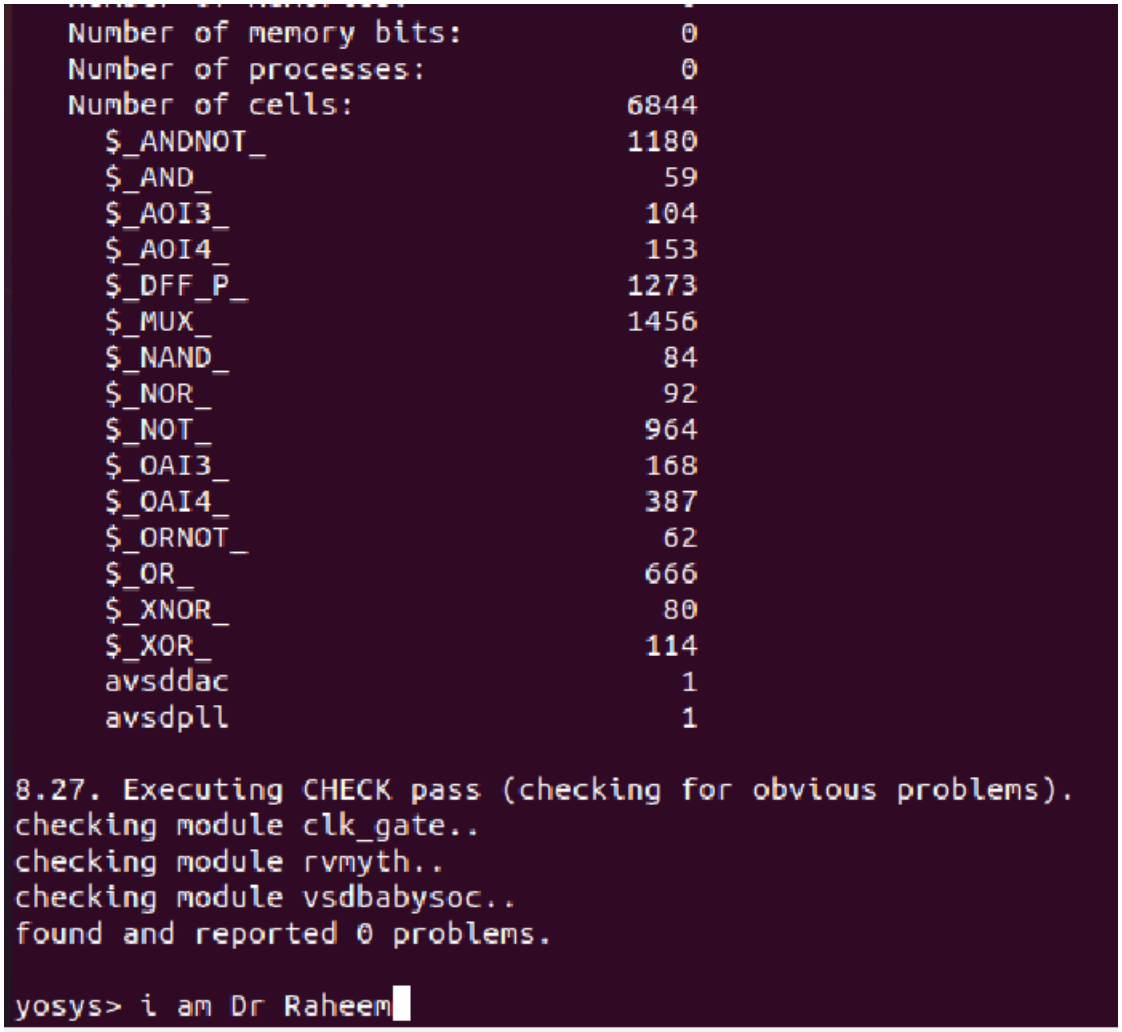
```
=== vsdbabysoc ===

   Number of wires:                 9
   Number of wire bits:            18
   Number of public wires:          9
   Number of public wire bits:     18
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                 3
     avsddac                        1
     avsdpll                        1
     rvmyth                         1
```

```
=== design hierarchy ===

   vsdbabysoc                       1
     rvmyth                         1
       clk_gate                     7

   Number of wires:              4707
   Number of wire bits:          8343
   Number of public wires:        311
   Number of public wire bits:   3947
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:              6844
     $_ANDNOT_                   1180
```



Step 5 Technology Mapping

dfflibmap -liberty
/home/mjcet/VSDBabySoC/VSDBabySoC/src/lib//sky130_fd_sc_hd__tt_025C_1v80.lib

```
   sky130_fd_sc_hd__dfstp_2 _DFF_PP1_ (.CLK( C), .D( D), .Q( Q), .SET_B(~R));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NNN_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B( R), .SET_B( S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NNP_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B(~R), .SET_B( S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NPN_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B( R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NPP_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B(~R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PNN_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B( R), .SET_B( S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PNP_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B(~R), .SET_B( S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PPN_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B( R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PPP_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B(~R), .SET_B(~S));
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':
```

Step 6 Write the Synthesized Netlist

write_verilog vsdbabysoc_synth_net.v

abc -liberty /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib//sky130_fd_sc_hd__tt_025C_1v80.lib

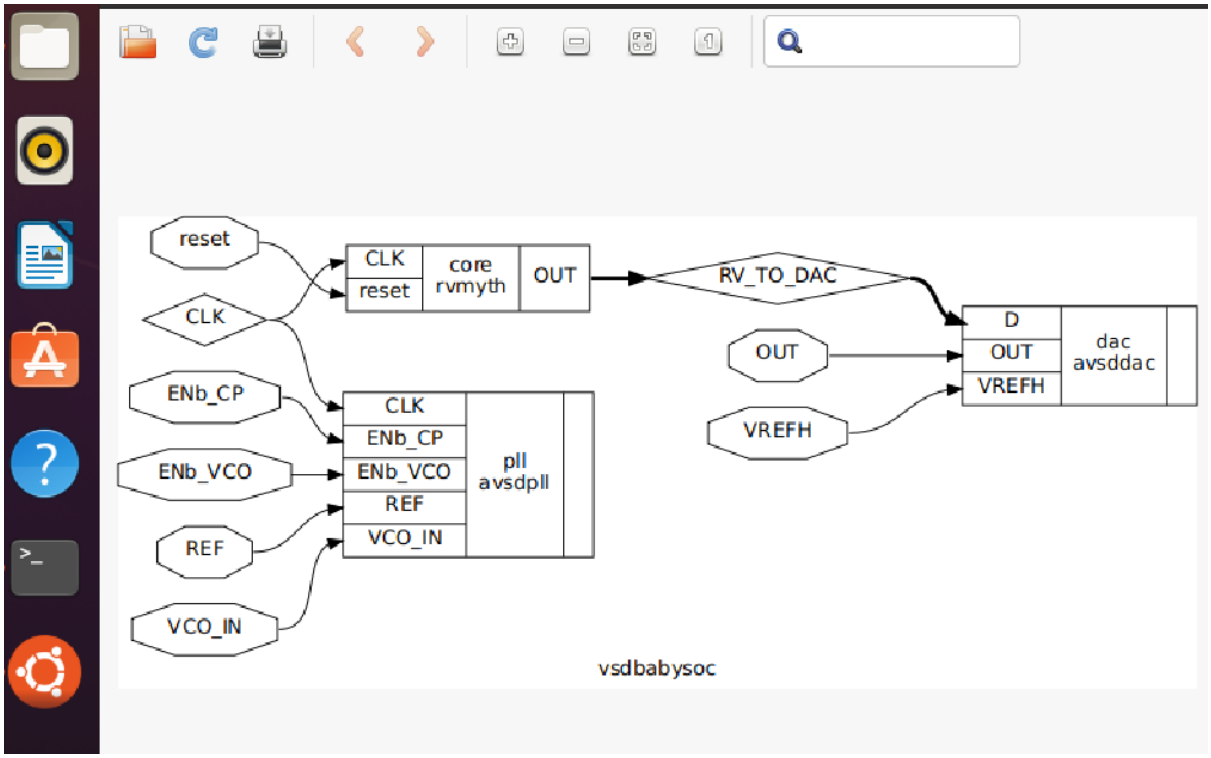
```
ABC RESULTS:    sky130_fd_sc_hd__o211a_1 cells:        1
ABC RESULTS:    sky130_fd_sc_hd__o211ai_1 cells:       25
ABC RESULTS:    sky130_fd_sc_hd__o21a_1 cells:         13
ABC RESULTS:    sky130_fd_sc_hd__o21ai_0 cells:      1104
ABC RESULTS:    sky130_fd_sc_hd__o21bai_1 cells:       17
ABC RESULTS:    sky130_fd_sc_hd__o221a_2 cells:        2
ABC RESULTS:    sky130_fd_sc_hd__o221ai_1 cells:       6
ABC RESULTS:    sky130_fd_sc_hd__o22ai_1 cells:      117
ABC RESULTS:    sky130_fd_sc_hd__o2bb2ai_1 cells:      2
ABC RESULTS:    sky130_fd_sc_hd__o311ai_0 cells:       3
ABC RESULTS:    sky130_fd_sc_hd__o31ai_1 cells:        3
ABC RESULTS:    sky130_fd_sc_hd__o32ai_1 cells:        1
ABC RESULTS:    sky130_fd_sc_hd__o41ai_1 cells:        4
ABC RESULTS:    sky130_fd_sc_hd__or2_2 cells:         15
ABC RESULTS:    sky130_fd_sc_hd__xnor2_1 cells:        9
ABC RESULTS:    sky130_fd_sc_hd__xor2_1 cells:        38
ABC RESULTS:        internal signals:     4396
ABC RESULTS:           input signals:     1225
ABC RESULTS:          output signals:     1173
Removing temp directory.

10.3. Extracting gate netlist of module `\vsdbabysoc' to `<abc-temp-dir>/input.
blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> i am Dr Raheem
```

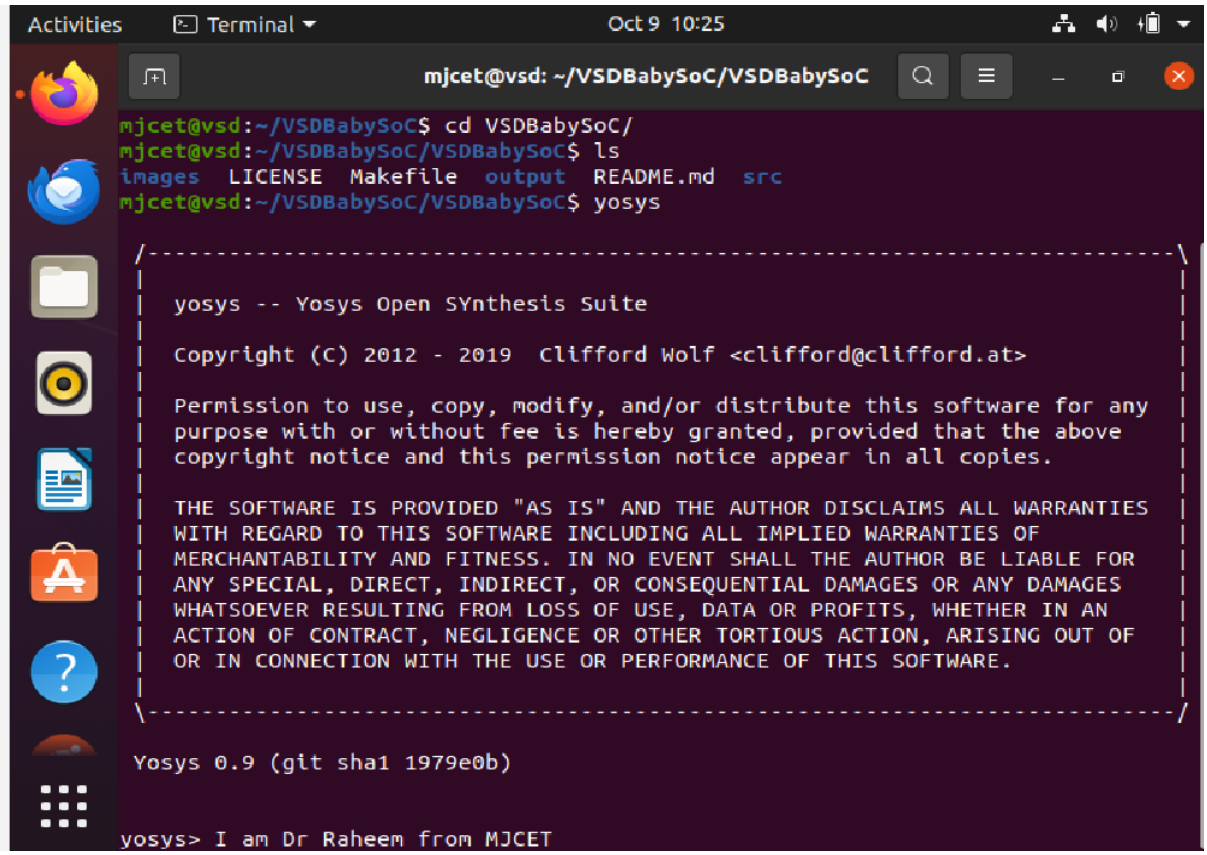
Step 7 Synthesis visualize

show vsdbabysoc



POST Synthesis and Simulation Waveforms of vsdbabysoc.v

Step1 Open the yosys



Load the Top-Level and other Verilog files R

Indicate SV (SystemVerilog) support and all include/module paths. Avoid typographical errors in commands or filenames:

Step 2 : Read_verilog

read_verilog -sv /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/vsdbabysoc.v /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth.v

/home/mjcet/VSDBabySoC/VSDBabySoC/src/module/clk_gate.v -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include

```
yosys> read_verilog  /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/vsdbabysoc.v
1. Executing Verilog-2005 frontend: /home/mjcet/VSDBabySoC/VSDBabySoC/src/modul
e/vsdbabysoc.v
Parsing Verilog input from `/home/mjcet/VSDBabySoC/VSDBabySoC/src/module/vsdbab
ysoc.v' to AST representation.
Generating RTLIL representation for module `\vsdbabysoc'.
Successfully finished Verilog frontend.

yosys> read_verilog -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include /home/mjce
t/VSDBabySoC/VSDBabySoC/src/module/rvmyth.v

2. Executing Verilog-2005 frontend: /home/mjcet/VSDBabySoC/VSDBabySoC/src/modul
e/rvmyth.v
Parsing Verilog input from `/home/mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth
.v' to AST representation.
Generating RTLIL representation for module `\rvmyth'.
Warning: Replacing memory \CPU_Xreg_value_a5 with list of registers. See /home/
mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth_gen.v:696
Warning: Replacing memory \CPU_Xreg_value_a4 with list of registers. See /home/
mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth_gen.v:695
Warning: Replacing memory \CPU_Dmem_value_a5 with list of registers. See /home/
mjcet/VSDBabySoC/VSDBabySoC/src/module/rvmyth_gen.v:686
Successfully finished Verilog frontend.

yosys> read_verilog -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include /home/mjce
t/VSDBabySoC/VSDBabySoC/src/module/clk_gate.v
```

Step 3. Load Liberty Cell Libraries

Load the technology cell library (example for Sky130):

read_liberty -lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avdspll.lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avsddac.lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/sky Specify the Top Module

```
yosys> read_liberty -lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/sky130_fd_sc
_hd__tt_025C_1v80.lib /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avsdpll.lib /ho
me/mjcet/VSDBabySoC/VSDBabySoC/src/lib/avsddac.lib

5. Executing Liberty frontend.
Imported 428 cell types from liberty file.

6. Executing Liberty frontend.
Imported 1 cell types from liberty file.

7. Executing Liberty frontend.
Imported 1 cell types from liberty file.
```

Step 4 :Set the top-level module for synthesis

Synth -top vsdbabysoc.v

```
=== clk_gate ===

   Number of wires:                 5
   Number of wire bits:             5
   Number of public wires:          5
   Number of public wire bits:      5
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                 0
```

```
=== vsdbabysoc ===

   Number of wires:                 9
   Number of wire bits:            18
   Number of public wires:          9
   Number of public wire bits:     18
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                 3
     avsddac                        1
     avsdpll                        1
     rvmyth                         1
```

```
=== vsdbabysoc ===

   Number of wires:                  9
   Number of wire bits:             18
   Number of public wires:           9
   Number of public wire bits:      18
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                  3
     avsddac                         1
     avsdpll                         1
     rvmyth                          1
```

```
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:               6844
     $_ANDNOT_                    1180
     $_AND_                         59
     $_AOI3_                       104
     $_AOI4_                       153
     $_DFF_P_                     1273
     $_MUX_                       1456
     $_NAND_                        84
     $_NOR_                         92
     $_NOT_                        964
     $_OAI3_                       168
     $_OAI4_                       387
     $_ORNOT_                       62
     $_OR_                         666
     $_XNOR_                        80
     $_XOR_                        114
     avsddac                         1
     avsdpll                         1

8.27. Executing CHECK pass (checking for obvious problems).
checking module clk_gate..
checking module rvmyth..
checking module vsdbabysoc..
found and reported 0 problems.

yosys> i am Dr Raheem
```

Step 5 Technology Mapping

dfflibmap -liberty /home/mjcet/VSDBabySoC/VSDBabySoC/src/lib//sky130_fd_sc_hd__tt_025C_1v80.lib

```
   sky130_fd_sc_hd__dfstp_2 _DFF_PP1_ (.CLK( C), .D( D), .Q( Q), .SET_B(~R));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NNN_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B( R), .SET_B( S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NNP_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B(~R), .SET_B( S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NPN_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B( R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbn_1 _DFFSR_NPP_ (.CLK_N( C), .D( D), .Q( Q), .Q_N(~Q),
.RESET_B(~R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PNN_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B( R), .SET_B( S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PNP_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B(~R), .SET_B( S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PPN_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B( R), .SET_B(~S));
   sky130_fd_sc_hd__dfbbp_1 _DFFSR_PPP_ (.CLK( C), .D( D), .Q( Q), .Q_N(~Q), .
RESET_B(~R), .SET_B(~S));
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':
```

Step 6 Write the Synthesized Netlist

write_verilog vsdbabysoc_synth_net.v

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

```
ABC RESULTS:   sky130_fd_sc_hd__o211a_1 cells:        1
ABC RESULTS:   sky130_fd_sc_hd__o211ai_1 cells:       25
ABC RESULTS:   sky130_fd_sc_hd__o21a_1 cells:        13
ABC RESULTS:   sky130_fd_sc_hd__o21ai_0 cells:     1104
ABC RESULTS:   sky130_fd_sc_hd__o21bai_1 cells:       17
ABC RESULTS:   sky130_fd_sc_hd__o221a_2 cells:        2
ABC RESULTS:   sky130_fd_sc_hd__o221ai_1 cells:       6
ABC RESULTS:   sky130_fd_sc_hd__o22ai_1 cells:      117
ABC RESULTS:   sky130_fd_sc_hd__o2bb2ai_1 cells:      2
ABC RESULTS:   sky130_fd_sc_hd__o311ai_0 cells:       3
ABC RESULTS:   sky130_fd_sc_hd__o31ai_1 cells:        3
ABC RESULTS:   sky130_fd_sc_hd__o32ai_1 cells:        1
ABC RESULTS:   sky130_fd_sc_hd__o41ai_1 cells:        4
ABC RESULTS:   sky130_fd_sc_hd__or2_2 cells:         15
ABC RESULTS:   sky130_fd_sc_hd__xnor2_1 cells:        9
ABC RESULTS:   sky130_fd_sc_hd__xor2_1 cells:        38
ABC RESULTS:        internal signals:     4396
ABC RESULTS:           input signals:     1225
ABC RESULTS:          output signals:     1173
Removing temp directory.

10.3. Extracting gate netlist of module `\vsdbabysoc' to `<abc-temp-dir>/input.
blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> i am Dr Raheem
```

command invokes the ABC tool integrated in Yosys for advanced logic optimization and technology mapping. The script applies a sequence of transformations:

+strash: Structural hashing to reduce logic.

scorr: Structural correlation.

ifraig: Iterative functional reduction.

retime{D}: Retiming to optimize registers.

dch,-f: Don't care optimization.

map,-M,1{D}: Map logic to standard cells with specific constraints.

Step 7 flatten

```
11. Executing FLATTEN pass (flatten design).
Using template rvmyth for cells of type rvmyth.
Using template clk_gate for cells of type clk_gate.
<suppressed ~8 debug messages>
No more expansions possible.
Deleting now unused module clk_gate.
Deleting now unused module rvmyth.

yosys> i am Dr Raheem
```

Step 8 setundef -zero

```
yosys> setundef -zero

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

yosys> i am Dr Raheem
```

Step 9 clean -purge (Clean Unused Wires and Cells)

```
yosys> setundef -zero

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

yosys> clean -purge
Removed 399 unused cells and 7113 unused wires.

yosys> rename -enumerate
```

Step 10 rename -enumerate (rename nets and cells)

Step 11 stats (print Synthesis)

```
13. Printing statistics.

=== vsdbabysoc ===

   Number of wires:                 3905
   Number of wire bits:             6303
   Number of public wires:          3905
   Number of public wire bits:      6303
   Number of memories:                 0
   Number of memory bits:              0
   Number of processes:                0
   Number of cells:                 5957
     avsddac                           1
     avsdpll                           1
     sky130_fd_sc_hd__a2111oi_0        7
```

Step 12

(This exports the final synthesized netlist to a Verilog file without any synthesis attributes (-noattr). This netlist is technology-mapped and optimized, ready for place-and-route or further analysis.)

write_verilog -noattr /home/mjcet/VSDBabySoC/VSDBabySoC/src/vsdbabysoc.synth.v

Post Synthesis Simulation and Waveforms

Compile the Testbench

Before running the iverilog command, copy the necessary standard cell and primitive models: These files must be present in the same directory as the testbench (src/module) to resolve all module references during compilation.

To ensure that the synthesized Verilog file (vsdbabysoc.synth.v) is available in the src/module directory for further processing or simulation, you can copy it from the output directory to the src/module directory. Here is the step to do that:

Run the following iverilog command to compile the testbench:

iverilog -o /home/mjcet/VSDBabySoC/VSDBabySoC /src/output/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUNIT_DELAY=#1 -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include/ -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/ /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/testbench.v

```
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# iverilog -o /home/mjcet/VSDBaby
SoC/VSDBabySoC/src/module/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUN
IT_DELAY=#1 -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include/ -I /home/mjcet/VS
DBabySoC/VSDBabySoC/src/module/ /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/te
stbench.v
```

Option / Argument	Purpose / Description
iverilog	Icarus Verilog compiler used to compile Verilog files into a simulation executable.
-o /home/mjcet/VSDBabySoC/VSDBabySoC/output/post_synth_sim.out	Specifies the output binary file for simulation.
-DPOST_SYNTH_SIM	Defines the macro POST_SYNTH_SIM (used in testbench to switch simulation modes).
-DFUNCTIONAL	Defines FUNCTIONAL to use behavioral models instead of detailed gate-level timing.
-DUNIT_DELAY=#1	Assigns a unit delay of #1 to all gates for post-synthesis simulation.
-I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include	Adds the include directory to the search path for \include` directives.
-I /home/mjcet/VSDBabySoC/VSDBabySoC/src/module	Adds the module directory to the include path for additional module references.

Option / Argument	Purpose / Description
/home/mjcet/VSDBabySoC/VSDBabySoC/src/module/testbench.v	Specifies the testbench file as the top-level design for simulation.

❗ **Note - You may encounter this error:**

$ iverilog -o /home/mjcet/VSDBabySoC/VSDBabySoC/output/post_synth_sim/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUNIT_DELAY=#1 -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/module /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/testbench.v

```
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# iverilog -o /home/mjcet/VSDBaby
SoC/VSDBabySoC/src/module/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUN
IT_DELAY=#1 -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include/ -I /home/mjcet/VS
DBabySoC/VSDBabySoC/src/module/ /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/te
stbench.v
```

Navigate to the Post-Synthesis Simulation Output Directory

cd output/

Step 3: Run the Simulation

./post_synth_sim.out

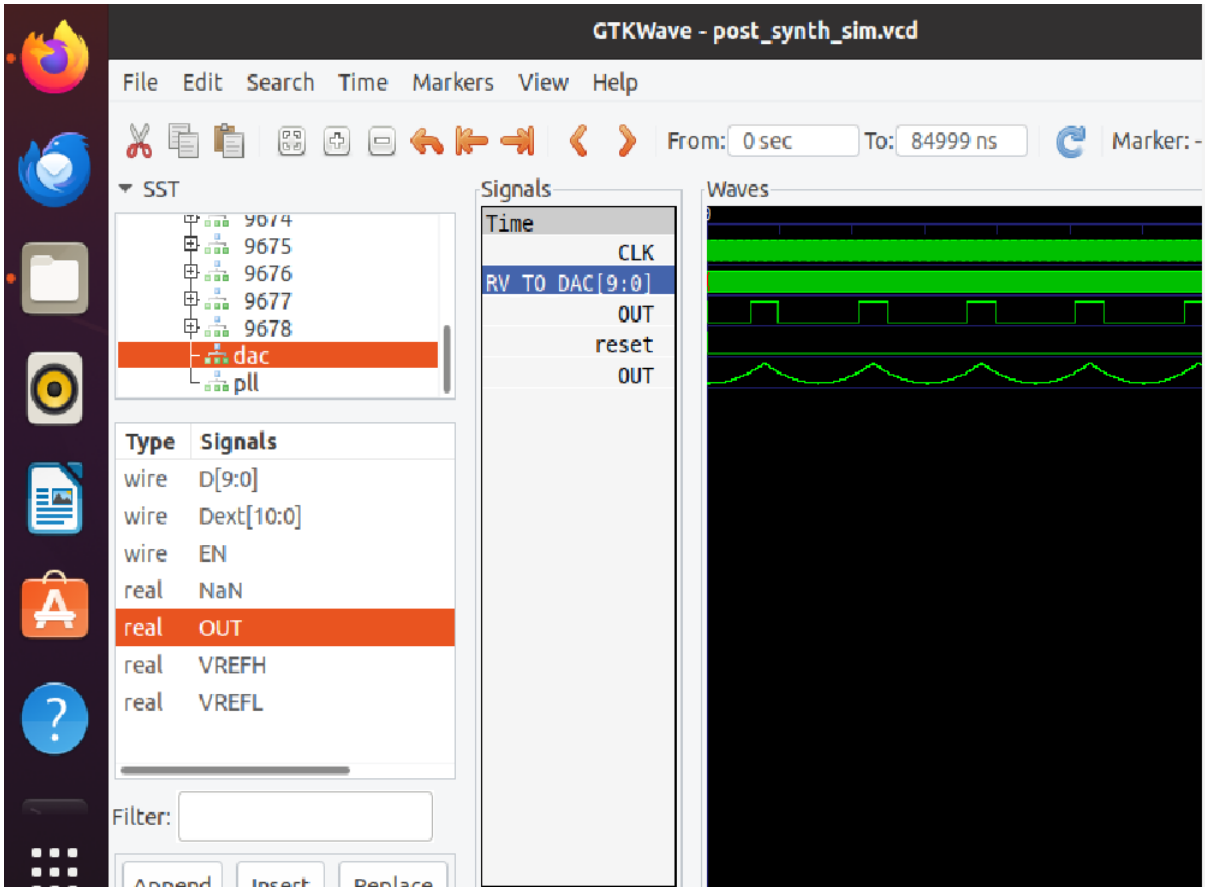
```
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# iverilog -o /home/mjcet/VSDBaby
SoC/VSDBabySoC/src/module/post_synth_sim.out -DPOST_SYNTH_SIM -DFUNCTIONAL -DUN
IT_DELAY=#1 -I /home/mjcet/VSDBabySoC/VSDBabySoC/src/include/ -I /home/mjcet/VS
DBabySoC/VSDBabySoC/src/module/ /home/mjcet/VSDBabySoC/VSDBabySoC/src/module/te
stbench.v
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# ./post_synth_sim.out
bash: ./post_synth_sim.out: No such file or directory
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# cd module/
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src/module# ./post_synth_sim.out
VCD info: dumpfile post_synth_sim.vcd opened for output.
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src/module# gtkwave post_synth_sim.v
cd
Gtk-Message: 16:06:07.124: Failed to load module "canberra-gtk-module"

GTKWave Analyzer v3.3.103 (w)1999-2019 BSI


(gtkwave:5358): dconf-WARNING **: 16:06:07.161: failed to commit changes to dco
nf: The connection is closed
[0] start time.
[84999000] end time.
```

Step 4: View the Waveforms in GTKWave

gtkwave post_synth_sim.vcd



Thank you for VSD for this program I learned a lot Thanks to VSD Team , Openlane, Skywater , github and IIT Gandhinagar for the RSIC V Program..... Dr Raheem from MJCET Hyderabad.