

## VSDBabySoC basic timing analysis

### Prepare Required Files

To begin static timing analysis on the VSDBabySoC design, you must organize and prepare the required files in specific directories.

1. git clone <https://github.com/parallaxsw/OpenSTA.git>

```
  images  LICENSE  Makefile  output  README.md  SRC
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC# git clone https://github.com/parall
axsw/OpenSTA.git
Cloning into 'OpenSTA'...
remote: Enumerating objects: 18302, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 18302 (delta 15), reused 20 (delta 8), pack-reused 18261 (from 1)
Receiving objects: 100% (18302/18302), 49.99 MiB | 4.25 MiB/s, done.
Resolving deltas: 100% (14879/14879), done.
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC# cd OpenSTA
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/OpenSTA# docker build --file Dockerf
ile.ubuntu22.04 --tag opensta .
DEPRECATED: The legacy builder is deprecated and will be removed in a future re
lease.
      Install the buildx component to build images with BuildKit:
      https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 75.75MB
Step 1/11 : FROM ubuntu:22.04
22.04: Pulling from library/ubuntu
af6eca94c810: Pull complete
```

prerequisite

Timing Library

/home/mjcet/VSDBabySoC/VSDBabySoC/src/timing\_libs

read netlist

/home/mjcet/VSDBabySoC/VSDBabySoC/src/BabySoC/vsdbabysoc.synth.v

read sdc

/home/mjcet/VSDBabySoC/VSDBabySoC/src/BabySoC/vsdbabysoc\_synthesis.sdc

```
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src# cd BabySoC/
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src/BabySoC# ls
STA_OUTPUT  vsdbabysoc_synthesis.sdc  vsdbabysoc.synth.v
root@vsd:/home/mjcet/VSDBabySoC/VSDBabySoC/src/BabySoC#
```

```

# Create a directory to store Liberty timing libraries
home/mjchet/VSDBabySoC/VSDBabySoC/src$ mkdir -p examples/timing_libs/
/VSDBabySoC/src/ ls timing_libs/
avsdac.lib avsdpll.lib sky130_fd_sc_hd_tt_025C_1v80.lib

# Create a directory to store synthesized netlist and constraint files
/home/mjchet/VSDBabySoC/VSDBabySoC/src / mkdir -p /BabySoC
/VSDBabySoC/src/ ls BabySoC/
gcd_sky130hd.sdc vsdbabysoc_synthesis.sdc vsdbabysoc.synth.v

```

These files include:

- Standard cell library: sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib
- IP-specific Liberty libraries: avsdpll.lib, avsdac.lib
- Synthesized gate-level netlist: vsdbabysoc.synth.v
- Timing constraints: vsdbabysoc\_synthesis.sdc

Below is the TCL script to run complete min/max timing checks on the SoC:

### **vsdbabysoc\_min\_max\_delays.tcl**

```

# Load Liberty Libraries (standard cell + IPs)
read_liberty -min /data/timing_libs/sky130_fd_sc_hd_tt_025C_1v80.lib
read_liberty -max /data/timing_libs/sky130_fd_sc_hd_tt_025C_1v80.lib

read_liberty -min /data/timing_libs/avsdpll.lib
read_liberty -max /data/timing_libs/avsdpll.lib

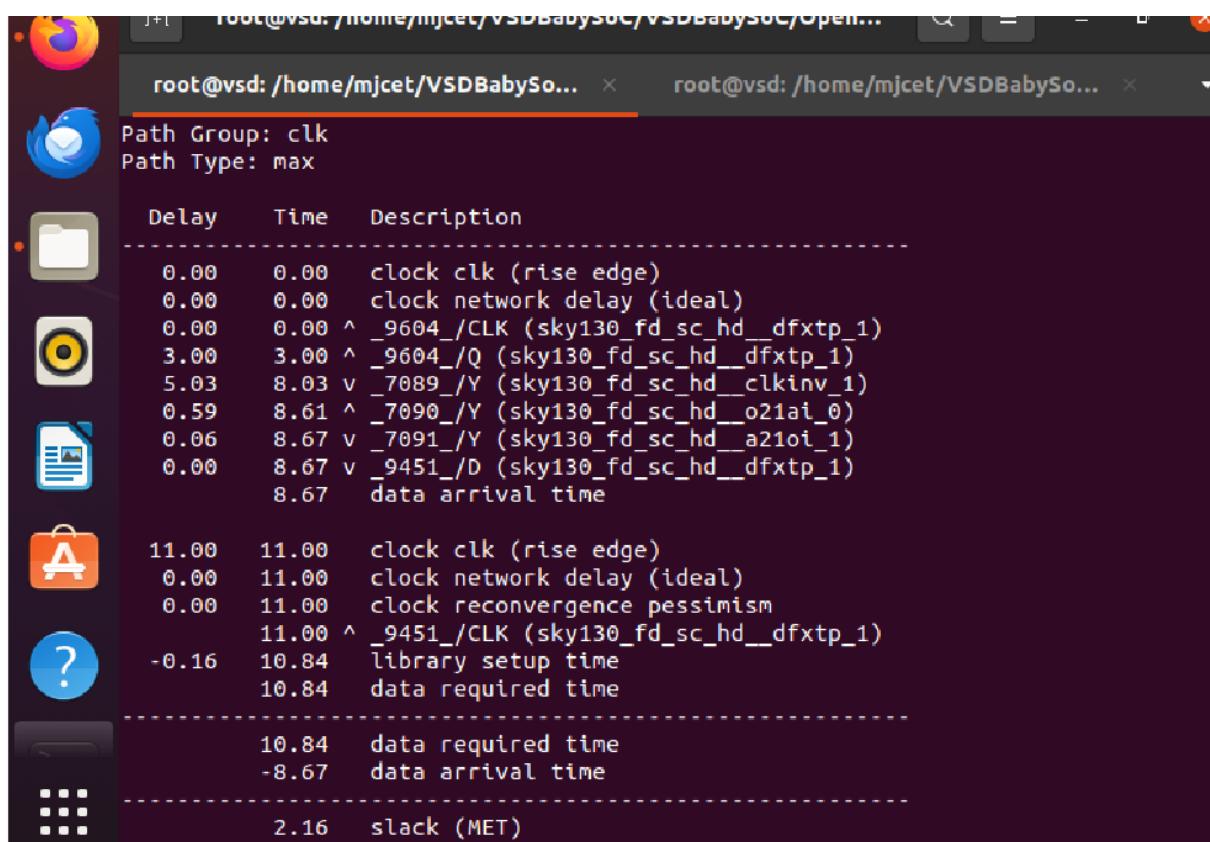
read_liberty -min /data/timing_libs/avsdac.lib
read_liberty -max /data/timing_libs/avsdac.lib

# Read Synthesized Netlist
read_verilog /data/BabySoC/vsdbabysoc.synth.v

```

```
# Link the Top-Level Design
```

```
link_design vsdbabysoc
```



```
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  ^_9604_/CLK (sky130_fd_sc_hd_dfxtpl_1)
          3.00  3.00  ^_9604_/Q (sky130_fd_sc_hd_dfxtpl_1)
          5.03  8.03  v_7089_/Y (sky130_fd_sc_hd_clkinv_1)
          0.59  8.61  ^_7090_/Y (sky130_fd_sc_hd_o21ai_0)
          0.06  8.67  v_7091_/Y (sky130_fd_sc_hd_a21oi_1)
          0.00  8.67  v_9451_/D (sky130_fd_sc_hd_dfxtpl_1)
          8.67  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 ^_9451_/CLK (sky130_fd_sc_hd_dfxtpl_1)
         -0.16 10.84  library setup time
          10.84  data required time
          10.84  data required time
          -8.67  data arrival time

          2.16  slack (MET)
```

```
# Apply SDC Constraints
```

```
read_sdc /data/BabySoC/vsdbabysoc_synthesis.sdc
```

```
# Generate Timing Report
```

```
report_checks
```

Line of Code	Purpose	Explanation
read_liberty -min ...sky130... & -max ...sky130...	Load standard cell library	Loads the <b>typical PVT corner</b> for both min (hold) and max (setup) timing analysis.

Line of Code	Purpose	Explanation
read_liberty -min/-max avsdpll.lib	Load PLL IP Liberty	Includes Liberty timing views of the <b>PLL IP</b> used in the design.
read_liberty -min/-max avsddac.lib	Load DAC IP Liberty	Includes Liberty timing views of the <b>DAC IP</b> used in the design.
read_verilog vsdbabysoc.synth.v	Load synthesized netlist	Loads the gate-level Verilog netlist of the <b>VSDBabySoC</b> design.
link_design vsdbabysoc	Link top- level module	Links the hierarchy using vsdbabysoc as the <b>top module</b> for timing analysis.
read_sdc vsdbabysoc_synthesis.sdc	Load constraints	Loads SDC file specifying <b>clock definitions, input/output delays, and false paths</b> .
report_checks	Run timing analysis	Generates a default <b>setup timing report</b> . Add -path_delay min_max to see both hold and setup.

execute it inside the Docker container:

```
docker run -it -v $HOME:/data opensta
/data/VLSI/VSDBabySoC/OpenSTA/examples/BabySoC/vsdbabysoc_min_max_delays.tcl
```

### Possible Error Alert

You may encounter the following error when running the script:

Warning: /data/timing\_libs/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib line 23,  
default\_fanout\_load is 0.0.

Warning: /data/timing\_libs/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib line 1, library  
sky130\_fd\_sc\_hd\_tt\_025C\_1v80 already exists.

Warning: /data/timing\_libs/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib line 23,  
default\_fanout\_load is 0.0.

Error: /data/timing\_libs/avsdpll.lib line 54, syntax error

 **Fix:**

This error occurs because Liberty syntax does not support // for single-line comments, and more importantly, the { character appearing after // confuses the Liberty parser. Specifically, check around *line 54 of avsdpll.lib* and correct any syntax issues such as:

```
//pin (GND#2){  
// direction : input;  
// max_transition : 2.5;  
// capacitance : 0.001;  
//}
```

 **Replace with:**

```
/*  
pin (GND#2){  
direction : input;  
max_transition : 2.5;  
capacitance : 0.001;  
}  
*/
```

This should allow OpenSTA to parse the Liberty file without throwing syntax errors.

After fixing the Liberty file comment syntax as shown above, you can rerun the script to perform complete timing analysis for VSDBabySoC:

avsdpll.lib

```

15     capacitance : 0.001;
16 }
17
18     pin (GND) {
19         direction : input;
20         max_transition : 2.5;
21         capacitance : 0.001;
22     }
23 /*
24     pin (GND#2) {
25         direction : input;
26         max_transition : 2.5;
27         capacitance : 0.001;
28     }*/
29
30     pin (VDD) {
31         direction : input;
32         max_transition : 2.5;
33         capacitance : 0.001;
34     }
35 /*
36     pin (VDD#2) {
37         direction : input;
38         max_transition : 2.5;
39         capacitance : 0.001;
40     }*/
41 /*
42 */
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71 */

```

spatha@spatha-VirtualBox:~/VLSI/VSDBabySoC\$ docker run -it -v \$HOME:/data opensta /data/VLSI/VSDBabySoC/OpenSTA/examples/BabySoC/vsdbabysoc\_min\_max\_delays.tcl

OpenSTA 2.7.0 0b59461bd Copyright (c) 2025, 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\_copyright' for details.

This program comes with ABSOLUTELY NO WARRANTY; for details type 'show\_warranty'.

Warning: /data/VLSI/VSDBabySoC/OpenSTA/examples/timing\_libs/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib line 1, library sky130\_fd\_sc\_hd\_tt\_025C\_1v80 already exists.

Warning: /data/VLSI/VSDBabySoC/OpenSTA/examples/timing\_libs/avsdpll.lib line 1, library avsdpll already exists.

Warning: /data/VLSI/VSDBabySoC/OpenSTA/examples/timing\_libs/avsdac.lib line 1, library avsdac already exists.

Startpoint: \_10446\_ (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	-10446./CLK (sky130_fd_sc_hd_dfxtpl_1)	
4.13	4.13 ^ _10446./Q (sky130_fd_sc_hd_dfxtpl_1)	
5.06	9.19 v _8121./Y (sky130_fd_sc_hd_clkinv_1)	
6.57	9.76 ^ _8884./Y (sky130_fd_sc_hd_o2l1a_1)	
0.00	9.76 ^ _10034./D (sky130_fd_sc_hd_dfxtpl_1)	
9.76	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_dfxtpl_1)	
-6.14	10.86	library setup time
	10.86	data required time
	10.86	data required time
	-9.76	data arrival time
	1.11	slack (MET)

## VSDBabySoC PVT Corner Analysis (Post-Synthesis Timing)

Static Timing Analysis (STA) is performed across various **PVT (Process-Voltage-Temperature)** corners to ensure the design meets timing requirements under different conditions.

 sdc	2 items	30 Sep
 timing_libs	24 items	16:37
 sta_across_pvt.tcl	2.0 kB	16:43

## Critical Timing Corners

### Worst Max Path (Setup-critical) Corners:

- ss\_LowTemp\_LowVolt
  - ss\_HighTemp\_LowVolt
- These represent the **slowest** operating conditions.*

### Worst Min Path (Hold-critical) Corners:

- ff\_LowTemp\_HighVolt
  - ff\_HighTemp\_HighVolt
- These represent the **fastest** operating conditions.*

**Timing libraries** required for this analysis can be downloaded from:

 [Skywater PDK - sky130\\_fd\\_sc\\_hd Timing Libraries](#)

Below is the script that can be used to perform STA across the PVT corners for which the Sky130 Liberty files are available.

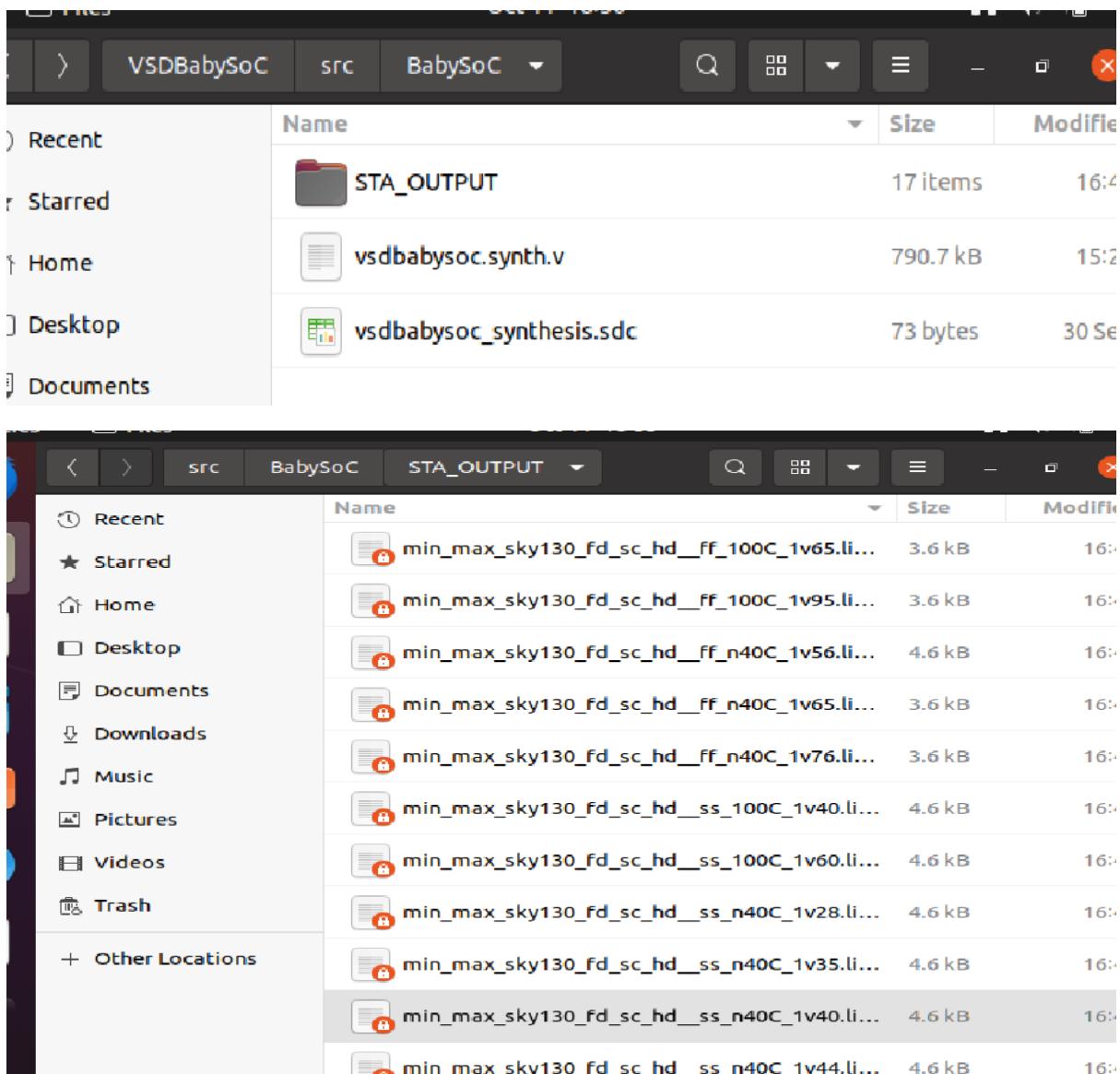
### sta\_across\_pvt.tcl

```
set list_of_lib_files(1) "sky130_fd_sc_hd_tt_025C_1v80.lib"
set list_of_lib_files(2) "sky130_fd_sc_hd_ff_100C_1v65.lib"
set list_of_lib_files(3) "sky130_fd_sc_hd_ff_100C_1v95.lib"
set list_of_lib_files(4) "sky130_fd_sc_hd_ff_n40C_1v56.lib"
set list_of_lib_files(5) "sky130_fd_sc_hd_ff_n40C_1v65.lib"
set list_of_lib_files(6) "sky130_fd_sc_hd_ff_n40C_1v76.lib"
set list_of_lib_files(7) "sky130_fd_sc_hd_ss_100C_1v40.lib"
set list_of_lib_files(8) "sky130_fd_sc_hd_ss_100C_1v60.lib"
set list_of_lib_files(9) "sky130_fd_sc_hd_ss_n40C_1v28.lib"
set list_of_lib_files(10) "sky130_fd_sc_hd_ss_n40C_1v35.lib"
set list_of_lib_files(11) "sky130_fd_sc_hd_ss_n40C_1v40.lib"
```

```
set list_of_lib_files(12) "sky130_fd_sc_hd_ss_n40C_1v44.lib"
set list_of_lib_files(13) "sky130_fd_sc_hd_ss_n40C_1v76.lib"
```

```
read_liberty /data/timing_libs/avsdpll.lib
```

```
read_liberty /data/timing_libs/avsddac.lib
```



```
for {set i 1} {$i <= [array size list_of_lib_files]} {incr i} {
    read_liberty /data/$list_of_lib_files($i)
    read_verilog /data/BabySoC/vsdbabysoc.synth.v
    link_design vsdbabysoc
```

```

current_design

read_sdc /data/BabySoC/vsdbabysoc_synthesis.sdc

check_setup -verbose

report_checks -path_delay min_max -fields {nets cap slew input_pins fanout} -digits {4}
> /data/BabySoC/STA_OUTPUT/min_max_$list_of_lib_files($i).txt

exec echo "$list_of_lib_files($i)" >>
/data/BabySoC/STA_OUTPUT/sta_worst_max_slack.txt

report_worst_slack -max -digits {4} >>
/data/BabySoC/STA_OUTPUT/sta_worst_max_slack.txt

exec echo "$list_of_lib_files($i)" >>
/data/BabySoC/STA_OUTPUT/sta_worst_min_slack.txt

report_worst_slack -min -digits {4} >>
/data/BabySoC/STA_OUTPUT/sta_worst_min_slack.txt

exec echo "$list_of_lib_files($i)" >> /data/BabySoC/STA_OUTPUT/sta_tns.txt

report_tns -digits {4} >> /data/BabySoC/STA_OUTPUT/sta_tns.txt

exec echo "$list_of_lib_files($i)" >> /data/BabySoC/STA_OUTPUT/sta_wns.txt

report_wns -digits {4} >> /data/BabySoC/STA_OUTPUT/sta_wns.txt

}

```

<b>Command</b>	<b>Purpose</b>	<b>Explanation</b>
report_worst_slack - max	Report Worst Setup Slack	Outputs the <b>most negative setup slack</b> (WNS) in the design for the current PVT corner.
report_worst_slack - min	Report Worst Hold Slack	Outputs the <b>most negative hold slack</b> in the design for the current PVT corner.

Command	Purpose	Explanation
report_tns	Report Total Negative Slack (TNS)	Prints the <b>sum of all negative slacks</b> (across all violating paths). Reflects how widespread timing violations are.
report_wns	Report Worst Negative Slack (WNS)	Prints the <b>single worst slack</b> (i.e., the most timing-violating path). Indicates severity of the critical path violation.

execute it inside the Docker container:

```
docker run -it -v $HOME:/data opensta /data/sta_across_pvt.tcl
```

After executing the above script, you can find the generated timing reports in the STA\_OUTPUT directory:

```
% source /data/sta_across_pvt.tcl
Warning: /data/timing_libs/avsdpll.lib line 1, library avsdpll already exists.
Warning: /data/timing_libs/avsddac.lib line 1, library avsddac already exists.
Warning: /data/timing_libs/sky130_fd_sc_hd_tt_025C_1v80.lib line 1, library sky130_fd_sc_hd_tt_025C_1v80 already exists.
Warning: There are 6 input ports missing set_input_delay.
    ENb_CP
    ENb_VCO
    REF
    VCO_IN
    VREFH
    reset
```

```
min_max_sky130_fd_sc_hd_ff_100C_1v65.lib.txt
min_max_sky130_fd_sc_hd_ss_100C_1v40.lib.txt
min_max_sky130_fd_sc_hd_ss_n40C_1v44.lib.txt sta_worst_max_slack.txt

min_max_sky130_fd_sc_hd_ff_100C_1v95.lib.txt
min_max_sky130_fd_sc_hd_ss_100C_1v60.lib.txt
min_max_sky130_fd_sc_hd_ss_n40C_1v76.lib.txt sta_worst_min_slack.txt

min_max_sky130_fd_sc_hd_ff_n40C_1v56.lib.txt
min_max_sky130_fd_sc_hd_ss_n40C_1v28.lib.txt
min_max_sky130_fd_sc_hd_tt_025C_1v80.lib.txt

min_max_sky130_fd_sc_hd_ff_n40C_1v65.lib.txt
min_max_sky130_fd_sc_hd_ss_n40C_1v35.lib.txt sta_tns.txt

min_max_sky130_fd_sc_hd_ff_n40C_1v76.lib.txt
min_max_sky130_fd_sc_hd_ss_n40C_1v40.lib.txt sta_wns.txt
```

<b>File</b>	<b>Description</b>
min_max_<lib>.txt	Detailed timing report for setup and hold paths for each PVT corner
sta_worst_max_slack.txt	Worst setup slack values across all corners
sta_worst_min_slack.txt	Worst hold slack values across all corners
sta_tns.txt	Total negative slack values across all corners
sta_wns.txt	Worst negative slack values across all corners

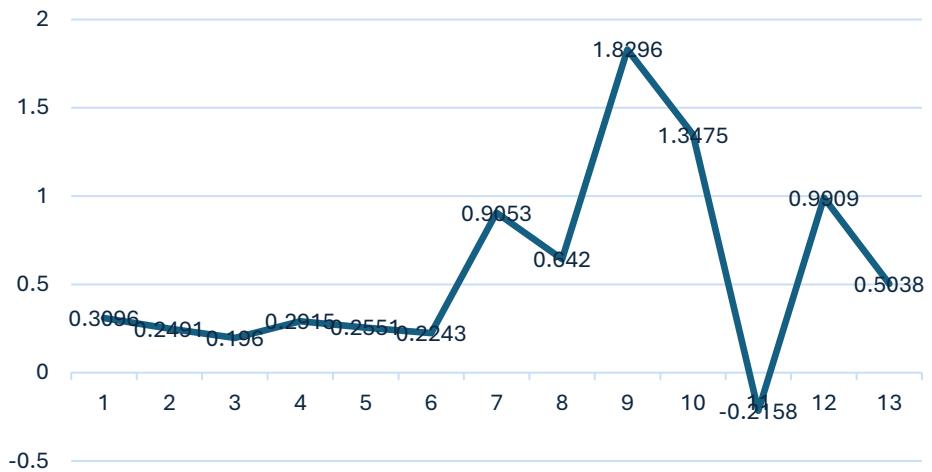
### **Timing Summary Across PVT Corners (Post-Synthesis STA Results)**

The following timing summary table was collected by running STA across 13 PVT corners using OpenSTA.

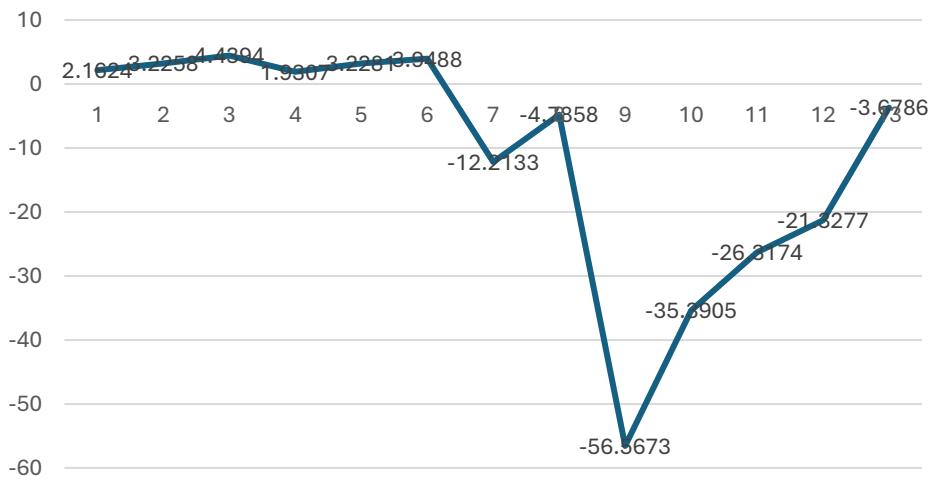
Metrics such as Worst Hold Slack, Worst Setup Slack, WNS, and TNS were extracted from the output reports.

PVT_Corner	Worst_Hold_Slack	Worst_Setup_Slack	WNS	TNS
tt_025_1v80	0.3096	2.1624	0	0
ff_100C_1v65	0.2491	3.2258	0	0
ff_100C_1v95	0.196	4.4394	0	0
ff_40C_1v56	0.2915	1.9307	0	0
ff_n40C_1v65	0.2551	3.2281	0	0
ff_n40C_1v76	0.2243	3.9488	0	0
ss_100C_1v40	0.9053	-12.2133	-12.2133	-7896.7788
ss_100C_1v60	0.642	-4.7858	-4.7858	-2900.1301
ss_40C_1v28	1.8296	-56.5673	-56.5673	-35352.5508
ss_40C_1v35	1.3475	-35.3905	-35.3905	-22393.0137
ss_40C_1v40	-0.2158	-26.3174	-26.3174	-16483.6797
ss_40C_1v44	0.9909	-21.3277	-21.3277	-13008.2969
ss_40C_1v76	0.5038	-3.6786	-3.6786	-1699.4681

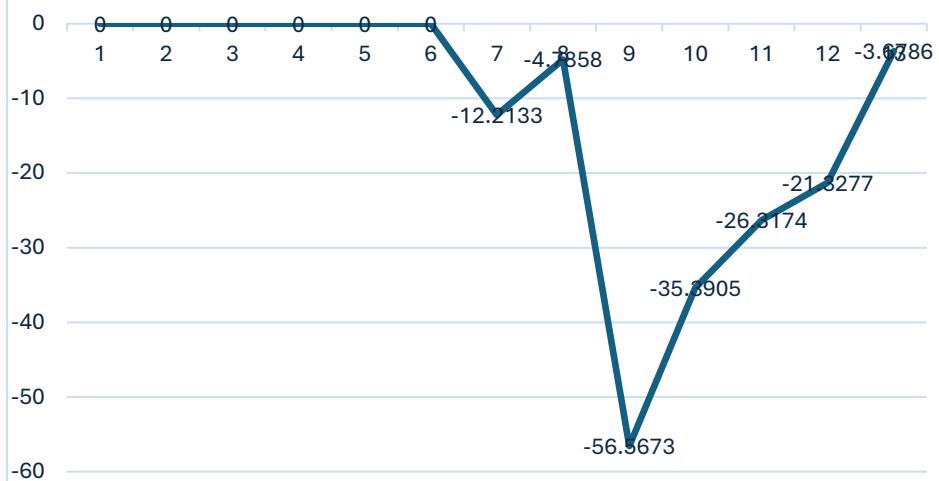
### Worst\_Hold\_Slack



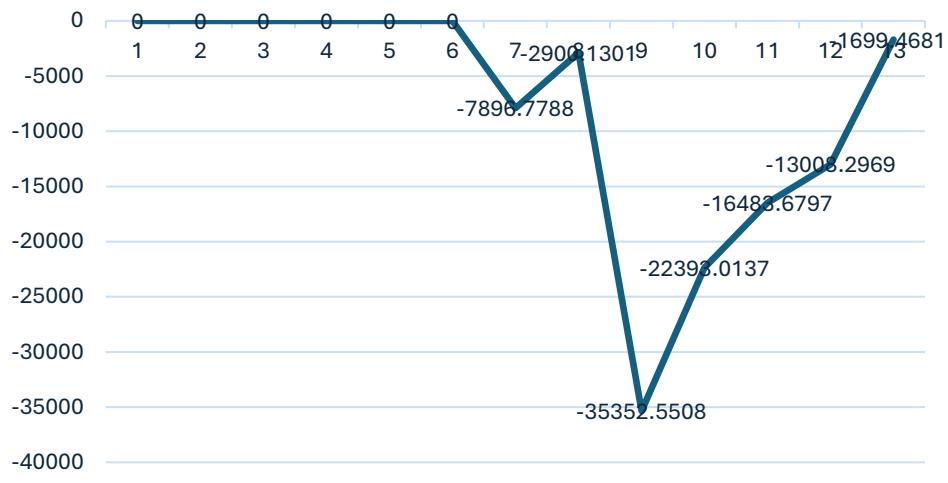
### Worst\_Setup\_Slack



### **WNS\_Raheem**



### **TNS\_Raheem**



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