Department of Electrical Engineering, IIT Palakkad

Hands-On Workshop On VLSI Design

ASIC Synthesis:
Static Timing Analysis and GLS

November 16, 2025

INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

Outline

1. Introduction
2. Synthesis Flow in Synopsys Design Compiler
3. DEMO
4. Reports and Analysis

Introduction

What is Logic Synthesis?

- Converts a design description written in a HDL, such as Verilog or VHDL, into an optimized gate-level netlist mapped to a specific logic library.
- Ensures functional equivalence while optimizing for area, timing, and power.
- Uses standard cell libraries for mapping.

Inputs and Outputs

Inputs: RTL (Verilog/VHDL), constraints (SDC), libraries (.db)

Outputs: Optimized Gate-level Netlist (.v), reports, and logs

Phases of Logic Synthesis

1. **Translation:** Conversion of high-level HDL description into combinational Boolean equations and sequential memory elements. The tool generates a technology independent gate-level representation (GTECH format).
2. **Logic Optimization:** The tool optimizes Boolean expressions and sequential elements, selecting a combination of library cells that meet the functional, timing, area, and power requirements of the design. (Still technology-independent.)
3. **Gate Mapping:** The optimized equations and memory elements are mapped to technology-dependent cells from the target library for design implementation.

Synthesis Flow in Synopsys DC

1. **Develop HDL Files**
2. **Specify the Libraries:**
   .db (link and target libraries), .sdb (symbol library), and .sldb (synthetic library)
3. **Read Design**
4. **Set Design Constraints**
5. **Synthesize and Optimize the Design**
6. **Save the Design Database**

DC Command: analyze

- Reads HDL source files and performs syntax checks.
- Converts HDL into an **intermediate representation** (HDL-independent).
- Creates HDL library objects for use in the next synthesis stages.
- Stores the intermediate files in a user-specified working directory.

Example

```
analyze -format verilog {module_top.v}
```

Source: www.vlsiexpert.com

DC Command: `elaborate`

- Translates the intermediate design into a **technology-independent (GTECH)** netlist.
- Replaces HDL arithmetic operators with equivalent **DesignWare components**.
- Automatically executes the `link` command to resolve design references.
- Result: a complete elaborated design ready for synthesis and optimization.

Example

```
elaborate  module_name_top
```

Source: www.vlsiexpert.com

DC Command: `compile` / `compile_ultra`

- **Purpose:** Performs synthesis, optimization and mapping of the elaborated (GTECH) design to the target cell library.
- **Behavior:**
  - Optimizes Boolean logic and sequential elements (retiming, resource sharing, pipelining when applicable).
  - Maps technology-independent gates to technology-dependent cells from the `target_library`.
  - Attempts to meet timing, area and power constraints supplied earlier.
- **Two common variants:**
  - `compile` — standard/fast synthesis.
  - `compile_ultra` — high-effort optimization.

Report Generation

```
report_timing
report_area
report_power
```

- `report_timing`: Shows critical paths and slack.
- `report_area`: Displays total cell area.

DEMO

SDC Constraints Summary (Part 1)

- `create_clock -period 10 [get_ports clk]`
  → Defines a 10 ns clock (100 MHz).
- `set_input_delay -max 0.25 -clock clk [all_inputs]`
  → Models external delay (0.25 ns) from device to FPGA inputs.
- `set_input_transition 0.2 [all_inputs]`
  → Assumes 0.2 ns rise/fall time — affects gate delay estimation.

SDC Constraints Summary (Part 2)

- `set_output_delay -max 0.25 -clock clk [all_outputs]`
  → Ensures outputs are stable 0.25 ns before external sampling.
- `set_clock_uncertainty -setup 0.1 [get_clocks clk]`
  → Adds 0.1 ns margin for jitter/skew (setup analysis).
- `set_clock_uncertainty -hold 0.1 [get_clocks clk]`
  → Adds margin for hold-time safety.
- `set_max_transition 0.20 [current_design]`
  → Restricts net transition to 0.20 ns for signal integrity.
- `set_max_transition -clock_path 0.150 [get_clocks clk]`
  → Tighter 0.15 ns limit for clock tree nets.

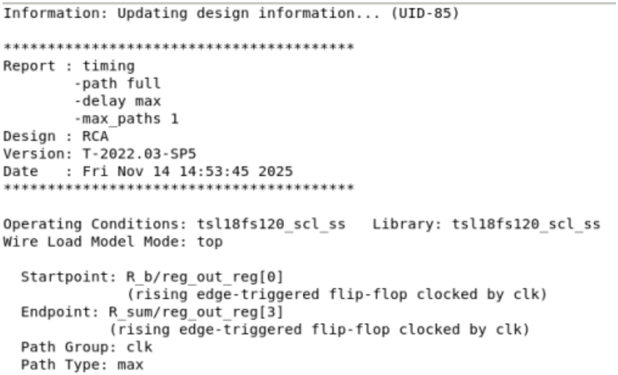
Interpreting Timing Reports

- **Slack:** difference between required arrival time and actual arrival time. Positive slack = timing met.
- **Worst negative slack (WNS):** the most negative slack value; primary metric for timing failures.
- **Total negative slack (TNS):** sum of negative slacks across failing paths.
- Use `report_timing -paths full -from ... -to ...` to inspect detailed path elements (cells, nets, delays).

Static Timing Signoff Checklist

- Accurate clock definitions (including generated clocks).
- Correct I/O delays matching board-level measurements or vendor specs.
- Proper path exceptions (false paths, multicycle paths).
- Run STA with worst-case and best-case corners (PVT variants).
- Back-annotate SDF for post-synthesis/post-layout timing checks.
- Track WNS/TNS across iterations until signoff targets are met.

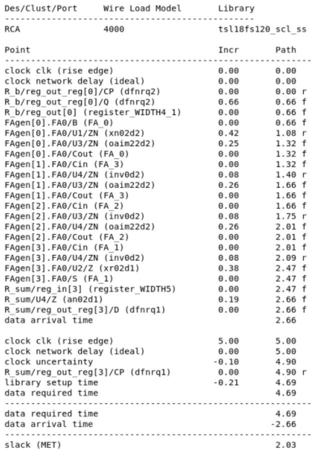
Timing Report Summary : RCA Module



```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : RCA
Version: T-2022.03-SP5
Date   : Fri Nov 14 14:53:45 2025
****************************************

Operating Conditions: tsl18fs120_scl_ss   Library: tsl18fs120_scl_ss
Wire Load Model Mode: top

  Startpoint: R_b/reg_out_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: R_sum/reg_out_reg[3]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max
```

Figure 1: Timing Report 1

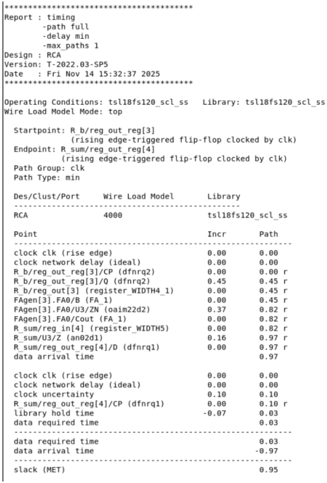
Timing Report Summary : RCA Module



```
Des/Clust/Port     Wire Load Model       Library
------------------------------------------------
RCA                4000                  tsl18fs120_scl_ss

Point                                    Incr       Path
-----------------------------------------------------------
clock clk (rise edge)                    0.00       0.00
clock network delay (ideal)              0.00       0.00
R_b/reg_out_reg[0]/CP (dfnrq2)           0.00       0.00 r
R_b/reg_out_reg[0]/Q (dfnrq2)            0.66       0.66 f
R_b/reg_out[0] (register_WIDTH4_1)       0.00       0.66 f
FAgen[0].FA0/B (FA_0)                    0.00       0.66 f
FAgen[0].FA0/U1/ZN (xn02d2)              0.42       1.08 r
FAgen[0].FA0/U3/ZN (oaim22d2)            0.25       1.32 f
FAgen[0].FA0/Cout (FA_0)                 0.00       1.32 f
FAgen[1].FA0/Cin (FA_3)                  0.00       1.32 f
FAgen[1].FA0/U4/ZN (inv0d2)              0.08       1.40 r
FAgen[1].FA0/U3/ZN (oaim22d2)            0.26       1.66 f
FAgen[1].FA0/Cout (FA_3)                 0.00       1.66 f
FAgen[2].FA0/Cin (FA_2)                  0.00       1.66 f
FAgen[2].FA0/U3/ZN (inv0d2)              0.08       1.75 r
FAgen[2].FA0/U4/ZN (oaim22d2)            0.26       2.01 f
FAgen[2].FA0/Cout (FA_2)                 0.00       2.01 f
FAgen[3].FA0/Cin (FA_1)                  0.00       2.01 f
FAgen[3].FA0/U4/ZN (inv0d2)              0.08       2.09 r
FAgen[3].FA0/U2/Z (xr02d1)               0.38       2.47 f
FAgen[3].FA0/S (FA_1)                    0.00       2.47 f
R_sum/reg_in[3] (register_WIDTH5)        0.00       2.47 f
R_sum/U4/Z (an02d1)                      0.19       2.66 f
R_sum/reg_out_reg[3]/D (dfnrq1)          0.00       2.66 f
data arrival time                                   2.66

clock clk (rise edge)                    5.00       5.00
clock network delay (ideal)              0.00       5.00
clock uncertainty                       -0.10       4.90
R_sum/reg_out_reg[3]/CP (dfnrq1)         0.00       4.90 r
library setup time                      -0.21       4.69
data required time                                  4.69
-----------------------------------------------------------
data required time                                  4.69
data arrival time                                  -2.66
-----------------------------------------------------------
slack (MET)                                         2.03
```

Figure 2: Timing Report 2

Timing Report Summary : RCA Module



```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : RCA
Version: T-2022.03-SP5
Date   : Fri Nov 14 15:32:37 2025
****************************************

Operating Conditions: tsl18fs120_scl_ss   Library: tsl18fs120_scl_ss
Wire Load Model Mode: top

  Startpoint: R_b/reg_out_reg[3]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: R_sum/reg_out_reg[4]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: min

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  RCA                4000                  tsl18fs120_scl_ss

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  R_b/reg_out_reg[3]/CP (dfnrq2)           0.00       0.00 r
  R_b/reg_out_reg[3]/Q (dfnrq2)            0.45       0.45 r
  R_b/reg_out[3] (register_WIDTH4_1)       0.00       0.45 r
  FAgen[3].FA0/B (FA_1)                    0.00       0.45 r
  FAgen[3].FA0/U3/ZN (oaim22d2)            0.37       0.82 r
  FAgen[3].FA0/Cout (FA_1)                 0.00       0.82 r
  R_sum/reg_in[4] (register_WIDTH5)        0.00       0.82 r
  R_sum/U3/Z (an02d1)                      0.16       0.97 r
  R_sum/reg_out_reg[4]/D (dfnrq1)          0.00       0.97 r
  data arrival time                                   0.97

  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  clock uncertainty                        0.10       0.10
  R_sum/reg_out_reg[4]/CP (dfnrq1)         0.00       0.10 r
  library hold time                       -0.07       0.03
  data required time                                  0.03
  -----------------------------------------------------------
  data required time                                  0.03
  data arrival time                                  -0.97
  -----------------------------------------------------------
  slack (MET)                                         0.95
```

Figure 3: Timing Report 3

RTL vs Gate-Level Simulation (GLS)

RTL Simulation	**Gate-Level Simulation (GLS)**
Fast (behavioral)	Slower (timing-accurate)
Works at register-transfer abstraction	Includes mapped cells and back-annotated delays (SDF)
Used for functional verification and testbench development	Used to validate timing, glitches, and post-mapping behavior
No SDF annotation	Requires SDF annotation for timing
Ideal for early development	Essential before tape-out or silicon sign-off

Gate-Level Simulation

Objective: After verifying timing closure, perform power estimation using realistic switching activity.

- **Functional GLS:** Simulate the synthesized gate-level netlist (`RCA.mapped.v`) to confirm logic correctness.
- **Timing GLS (SDF back-annotation):** Include gate and interconnect delays for accurate transition activity.
- **Waveform Dump:** Capture signal activity using `.vpd` or `.vcd` files during simulation.
- These simulations are run in `VCS` using standard cell library models (`tsl18fs120_scl.v`).

Switching Activity Capture and SAIF Import

Goal: Convert simulation waveforms into switching activity for realistic power estimation.

- Convert the simulation dump to SAIF:
  - `vpd2vcd inter.vpd inter.vcd`
  - `vcd2saif -input inter.vcd -output inter.saif`
- The SAIF file records toggle counts and signal durations for every net.
- In Design Compiler:
  - `report_power > init_power.rpt` (baseline without switching)
  - `read_saif -input ../SIM/inter.saif -instance_name rca_tb -verbose`
  - `compile -exact_map`
  - `report_power > gls_power.rpt`
- DC now recomputes power using the imported toggle data.

Interpretation

Comparison of Power Reports:

- `init_power.rpt` — assumes default activity factors (typically 0.5).
- `gls_power.rpt` — uses real switching information from simulation.

Reported components:

- **Internal Power:** Energy dissipated inside standard cells.
- **Switching Power:** Net-capacitance charging and discharging.
- **Leakage Power:** Static current under no switching.
- **Total Power:** Sum of all three components.

Thank You!