

# 반도체 토탈설계 프로젝트

개인 칩 설계 : UART 송신기

신재호 / 32217401 / 융합반도체공학과

김종남 교수님 분반



# 목차



목표



설계 계획



진행 과정 및 결과

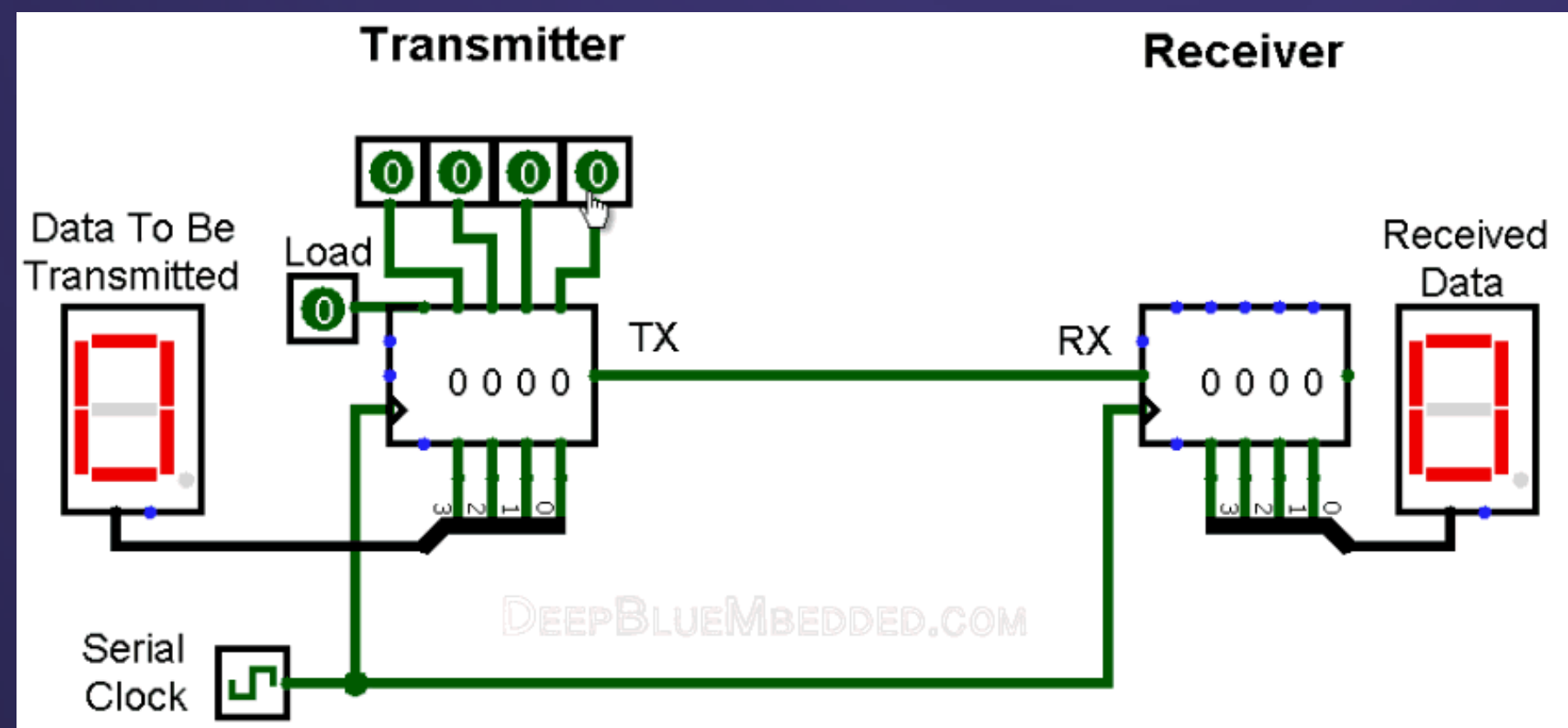
01

# 목표



회로설계 8단계의 공정을 OpenLane + Sky130 pdk 활용 위주로  
UART 송신기 완성하기 (ASIC으로 만들기)

115200 bps, Even Parity 조건으로  
Asic 구현하고 STA/DRC/LVS를 통과하는 GDSII 결과물 생성하기





# 01 주제 선정 이유

01

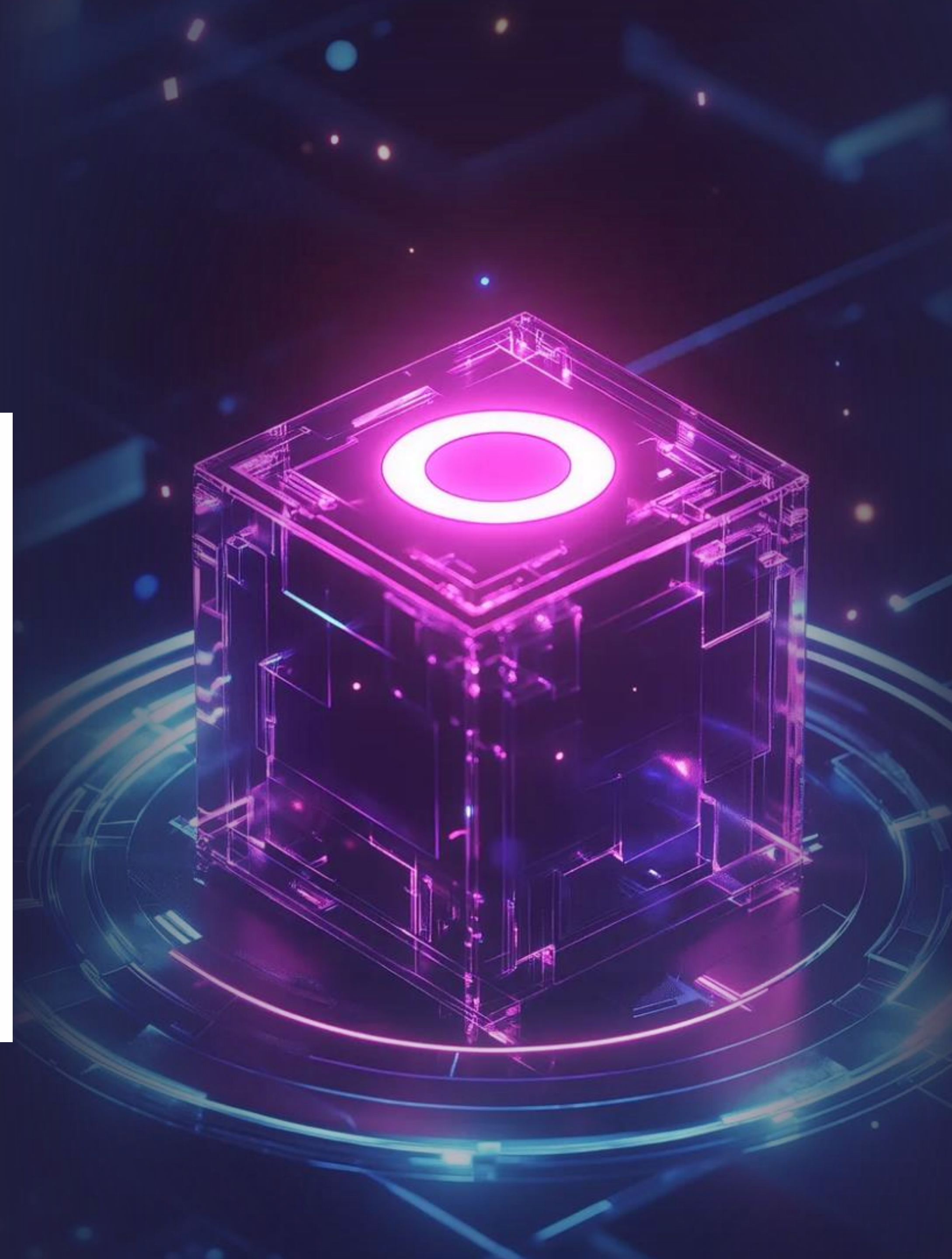
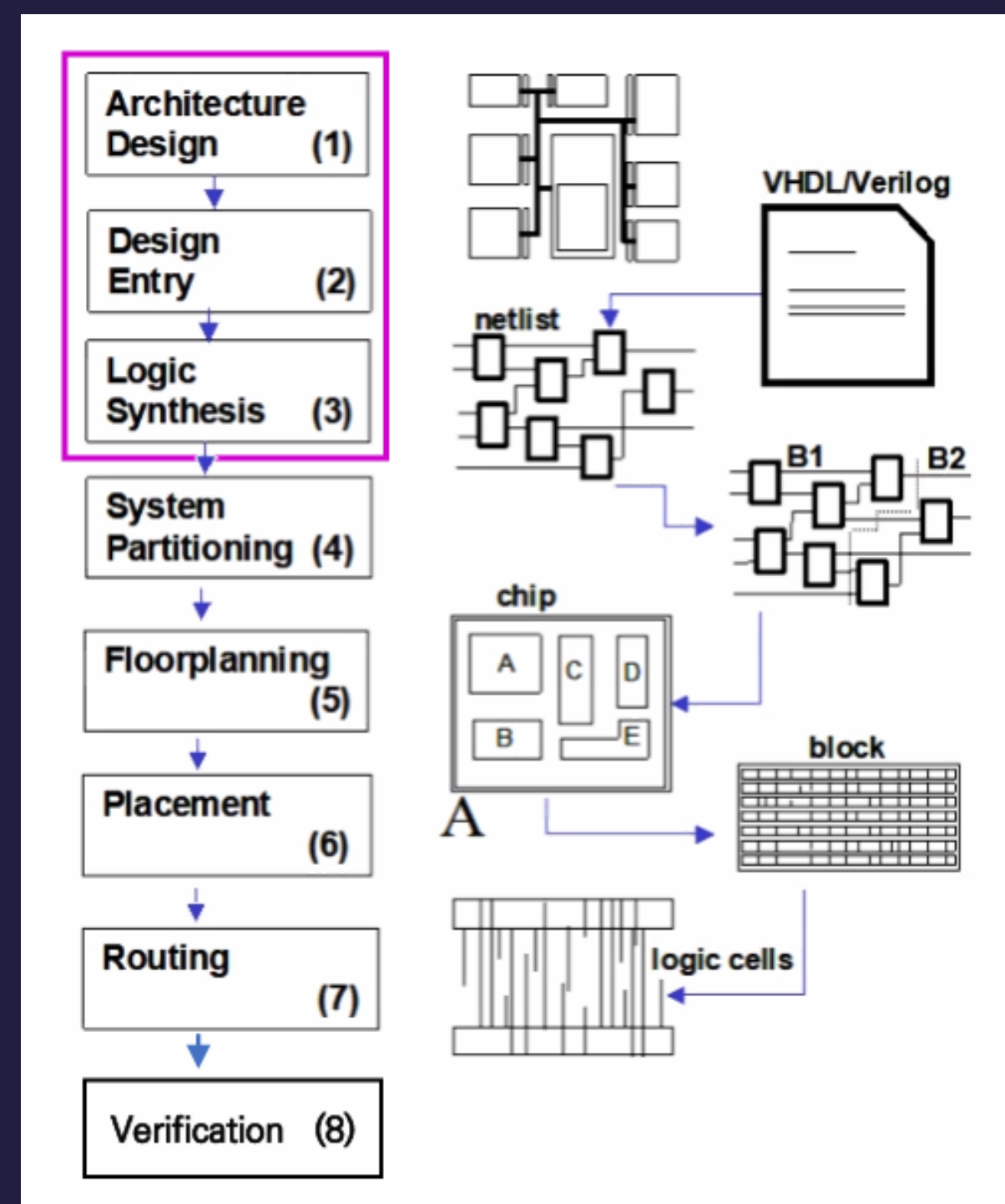
UART 송신기는  
실무표준 IP이고, SoC에서 가장  
보편적인 통신 모듈이다.

02

수업시간에 다른  
디지털 설계 프로세스를  
거의 그대로 적용가능

03

모두 오픈 소스 기반 설계  
**OpenLane + Sky130** 조합으로  
**Tape-Out** 수준까지 완성도를  
높일수 있을 가능성이 높음





02

# 설계 계획



## Design Entry

VS code + Verilog

RTL Cdoe 작성



## Logic Synthesis

Yosys

RTL -> Netlist 변환



## System Partitioning

VS Code + Yosys

계층화 및 모듈 분할

02

# 설계 계획



## Floorplanning

OpenLane + OpenROAD

Die/Core 영역 설정



## Placement

OpenROAD

셀 배치 최적화과정



## Routing

OpenROAD

배선



## Verification

OpenSTA + Icarus Verilog +

Klayout

STA/DRC/LVS

## 03

## 진행 과정

## 1. Github Uart Source 받기

```
sjh@DESKTOP-KDK01K9:~$ cd ~
sjh@DESKTOP-KDK01K9:~$ mkdir projects
sjh@DESKTOP-KDK01K9:~$ cd projects
sjh@DESKTOP-KDK01K9:~/projects$ git clone https://github.com/TinyTapeout/ttsky-verilog-template.git uart-project
Cloning into 'uart-project'...
remote: Enumerating objects: 287, done.
remote: Counting objects: 100% (129/129), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 287 (delta 107), reused 91 (delta 91), pack-reused 158 (from 2)
Receiving objects: 100% (287/287), 63.83 KiB | 1.03 MiB/s, done.
Resolving deltas: 100% (144/144), done.
sjh@DESKTOP-KDK01K9:~/projects$ cd uart-project
sjh@DESKTOP-KDK01K9:~/projects/uart-project$ python3 -m venv venv
sjh@DESKTOP-KDK01K9:~/projects/uart-project$ source venv/bin/activate
(venv) sjh@DESKTOP-KDK01K9:~/projects/uart-project$ pip install -r test/requirements.txt
Collecting pytest==8.4.2 (from -r test/requirements.txt (line 1))
  Downloading pytest-8.4.2-py3-none-any.whl.metadata (7.7 kB)
Collecting cocotb==2.0.1 (from -r test/requirements.txt (line 2))
  Downloading cocotb-2.0.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.8 kB)
Collecting iniconfig>=1 (from pytest==8.4.2->-r test/requirements.txt (line 1))
  Downloading iniconfig-2.3.0-py3-none-any.whl.metadata (2.5 kB)
Collecting packaging>=20 (from pytest==8.4.2->-r test/requirements.txt (line 1))
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting pluggy<2,>=1.5 (from pytest==8.4.2->-r test/requirements.txt (line 1))
  Downloading pluggy-1.6.0-py3-none-any.whl.metadata (4.8 kB)
Collecting pygments>=2.7.2 (from pytest==8.4.2->-r test/requirements.txt (line 1))
  Downloading pygments-2.19.2-py3-none-any.whl.metadata (2.5 kB)
Collecting find_libpython (from cocotb==2.0.1->-r test/requirements.txt (line 2))
  Downloading find_libpython-0.5.0-py3-none-any.whl.metadata (2.8 kB)
Downloading pytest-8.4.2-py3-none-any.whl (365 kB)
   365.8/365.8 kB 4.0 MB/s eta 0:00:00
Downloading cocotb-2.0.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.6 MB)
   4.6/4.6 MB 7.5 MB/s eta 0:00:00
Downloading iniconfig-2.3.0-py3-none-any.whl (7.5 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
   66.5/66.5 kB 3.4 MB/s eta 0:00:00
Downloading pluggy-1.6.0-py3-none-any.whl (20 kB)
Downloading pygments-2.19.2-py3-none-any.whl (1.2 MB)
   1.2/1.2 MB 5.3 MB/s eta 0:00:00
Downloading find_libpython-0.5.0-py3-none-any.whl (9.2 kB)
Installing collected packages: pygments, pluggy, packaging, iniconfig, find_libpython, pytest, cocotb
Successfully installed cocotb-2.0.1 find_libpython-0.5.0 iniconfig-2.3.0 packaging-25.0 pluggy-1.6.0 pygments-2.19.2
pytest-8.4.2
(venv) sjh@DESKTOP-KDK01K9:~/projects/uart-project$
```

→ Github에서 uart source를 받아온다.

→ Python3 가상환경과 필요 라이브러리를 세팅한다.

## 03

## 진행 과정

## 2. Hardening

```
(venv) sjh@DESKTOP-KDK01K9:~/projects/OpenLane$ make
make[1]: Entering directory '/home/sjh/projects/OpenLane'
ff5509f65b17bfa4068d5336495ab1718987ff69: Pulling from the-openroad-project/openlane
Digest: sha256:d7e2bdd21e757e3f0b15d6ea08c36d9a96653c31b4fb2fafd4de88c9b6a3f3fa
Status: Image is up to date for ghcr.io/the-openroad-project/openlane:ff5509f65b17bfa4068d5336495ab1718987ff69
ghcr.io/the-openroad-project/openlane:ff5509f65b17bfa4068d5336495ab1718987ff69
make[1]: Leaving directory '/home/sjh/projects/OpenLane'
./venv/bin/ciel enable --pdk sky130A
Version 0fe599b2afb6708d281543108caf8310912f54af not found locally, attempting to download...
Downloading common.tar.zst... 100% 0:00:00
Downloading sky130_fd_io.tar.zst... 100% 0:00:00
Downloading sky130_fd_pr.tar.zst... 100% 0:00:00
Downloading sky130_fd_sc_hd.tar.zst... 100% 0:00:00
Downloading sky130_fd_sc_hvl.tar.zst... 100% 0:00:00
Downloading sky130_ml_xx_hd.tar.zst... 100% 0:00:00
Downloading sky130_sram_macros.tar.zst... 100% 0:00:00
Version 0fe599b2afb6708d281543108caf8310912f54af enabled for the sky130 PDK.
```

→ Sky130 PDK를 활용하여 make (hardening)



## 03

## 진행 과정

## 3. RTL Code 코드 설계 (by using VS Code)

```
module uart_tx #(
    parameter CLK_FREQ = 50000000, // 50MHz 클럭 기준
    parameter BAUD_RATE = 115200 // 목표 통신 속도
)(
    input wire clk,
    input wire rst_n,
    input wire tx_start, // 전송 시작 신호
    input wire [7:0] data_in, // 보낼 데이터 (8비트)
    output reg tx_out, // 출력 핀 (TX)
    output reg tx_busy // 전송 중 상태 알림
);
```

## // 상태 머신 정의 (FSM)

```
localparam IDLE = 3'b000;
localparam START = 3'b001;
localparam DATA = 3'b010;
localparam PARITY = 3'b011;
localparam STOP = 3'b100;

reg [2:0] state;
localparam CLKS_PER_BIT = CLK_FREQ / BAUD_RATE;
reg [15:0] clk_cnt;
reg [2:0] bit_idx;
reg [7:0] data_reg;
reg parity_bit;
```

```
always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        state <= IDLE;
        tx_out <= 1'b1; // 유힬상태일때 High
        tx_busy <= 1'b0;
        clk_cnt <= 0;
        bit_idx <= 0;
        data_reg <= 0;
        parity_bit <= 0;
    end else begin
        case (state)
            IDLE: begin
                tx_out <= 1'b1;
                tx_busy <= 1'b0;
                clk_cnt <= 0;
                bit_idx <= 0;
                if (tx_start) begin
                    state <= START;
                    tx_busy <= 1'b1;
                    data_reg <= data_in;
                    parity_bit <= ^data_in; // 1의 개수가 홀수면 1 (Even Parity)
                end
            end
            START: begin // 시작 비트 (0)
                tx_out <= 1'b0;
                if (clk_cnt < CLKS_PER_BIT - 1) clk_cnt <= clk_cnt + 1;
            end else begin
                clk_cnt <= 0;
                state <= DATA;
            end
        endcase
    end
end
```

```
DATA: begin // 데이터 8비트
    tx_out <= data_reg[bit_idx];
    if (clk_cnt < CLKS_PER_BIT - 1) clk_cnt <= clk_cnt + 1;
    else begin
        clk_cnt <= 0;
        if (bit_idx < 7) bit_idx <= bit_idx + 1;
        else begin
            bit_idx <= 0;
            state <= PARITY;
        end
    end
end
PARITY: begin // 패리티 비트
    tx_out <= parity_bit;
    if (clk_cnt < CLKS_PER_BIT - 1) clk_cnt <= clk_cnt + 1;
    else begin
        clk_cnt <= 0;
        state <= STOP;
    end
end
STOP: begin // 정지 비트 (1)
    tx_out <= 1'b1;
    if (clk_cnt < CLKS_PER_BIT - 1) clk_cnt <= clk_cnt + 1;
    else begin
        clk_cnt <= 0;
        state <= IDLE;
    end
end

default: state <= IDLE;
endcase
end
endmodule
```



03

# 진행 과정

4. json Code 작성 (by using VS Code)

```
{
  "DESIGN_NAME": "uart_tx",
  "VERILOG_FILES": "dir::src/uart_tx.v",
  "CLOCK_PORT": "clk",
  "CLOCK_NET": "clk",
  "CLOCK_PERIOD": 20.0,

  "FP_SIZING": "absolute",
  "DIE_AREA": "0 0 150 150",
  "FP_CORE_UTIL": 50,

  "PL_TARGET_DENSITY": 0.75,
  "PL_BASIC_PLACEMENT": 0,

  "RUN_KLAYOUT_DRC": 1,
  "RUN_KLAYOUT_XOR": 1,
  "LINTER": 1,

  "pdk::sky130*": {
    "FP_CORE_UTIL": 45,
    "scl::sky130_fd_sc_hd": {
      "CLOCK_PERIOD": 20.0
    }
  }
}
```

회로 제약조건 작성

150x150 그리드조건



03

# 진행 과정

## 6. OpenLane

```
(venv) sjh@DESKTOP-KDK01K9:~/projects/OpenLane$ make mount
cd /home/sjh/projects/OpenLane && #
    docker run --rm -v /home/sjh/projects/OpenLane:/openlane -v /home/sjh/projects/OpenLane/designs:/openlane/install -v /home/sjh:/home/sjh -v /home/sjh/.ciel:/home/sjh/.ciel -e PDK_ROOT=/home/sjh/.ciel -e PDK=sky130A --user 1000:1000 -e DISPLAY=:0 -v /tmp/.X11-unix:/tmp/.X11-unix --network host --security-opt seccomp=unconfined -ti ghcr.io/the-openroad-p
```

```
OpenLane Container (ff5509f):/openlane$ ./flow.tcl -design uart_tx
OpenLane v1.0.2 (ff5509f6b17bf40b8d5336495ab17189b71f69)
All rights reserved. (c) 2020-2025 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO] : Using configuration in 'designs/uart_tx/config.json'...
[INFO] : PDK Root: /home/sjh/.ciel
[INFO] : Process Design Kit: sky130A
[INFO] : Standard Cell Library: sky130_fd_sc_hd
[INFO] : Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO] : Run Directory: /openlane/designs/uart_tx/runs/RUN_2025.12.11_19.43.52
[INFO] : Saving runtime environment...
[INFO] : Preparing LEF files for the nom corner...
[INFO] : Preparing LEF files for the min corner...
[INFO] : Preparing LEF files for the max corner...
[INFO] : Running linter (Verilator) (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/synthesis/linter.log)...
[INFO] : 0 errors found by linter
[WARNING] : 4 warnings found by linter
[STEP 1]
[INFO] : Running Synthesis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/synthesis/2-sta.log)...
[STEP 3]
[INFO] : Running Initial Floorplanning (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/floorplan/3-initial_fp.log)...
[WARNING] : Current core area is too small for the power grid settings chosen. The power grid will be scaled down.
[INFO] : Floorplanned with width 138.92 and height 127.84.
[STEP 4]
[INFO] : Running IO Placement...
[STEP 5]
[INFO] : Running Tap/Decap Insertion (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/floorplan/5-tap.log)...
[INFO] : Power planning with power {VPWR} and ground {VGND}...
[STEP 6]
[INFO] : Generating PDN (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/floorplan/6-pdn.log)...
[STEP 7]
[INFO] : Running Global Placement (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/placement/7-global.log)...
[STEP 8]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/placement/8-gpl_sta.log)...
[STEP 9]
[INFO] : Running Placement Resizer Design Optimizations (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/placement/9-resizer.log)...
[STEP 10]
[INFO] : Running Detailed Placement (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/placement/10-detailed.log)... [STEP 11]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/placement/11-dpl_sta.log)...
[STEP 12]
[INFO] : Running Clock Tree Synthesis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/cts/12-cts.log)...
[STEP 13]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/cts/13-cts_sta.log)...
[STEP 14]
[INFO] : Running Placement Resizer Timing Optimizations (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/cts/14-resizer.log)...
[STEP 15]
[INFO] : Running Global Routing Resizer Design Optimizations (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/15-resizer_design.log)...
[STEP 16]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/16-rsz_design_sta.log)...
[STEP 17]
[INFO] : Running Global Routing Resizer Timing Optimizations (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/17-resizer_timing.log)...
[STEP 18]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/18-rsz_timing_sta.log)...
[STEP 19]
[INFO] : Running Global Routing (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/19-global.log)...
[INFO] : Starting OpenROAD Antenna Repair Iterations...
[STEP 20]
[INFO] : Writing Verilog (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/19-global_write_netlist.log)...
[STEP 21]
[INFO] : Running Single-Corner Static Timing Analysis (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/21-qrt_sta.log)...
[STEP 22]
[INFO] : Running Fill Insertion (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/22-fill.log)...
[STEP 23]
[INFO] : Running Detailed Routing (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/23-detailed.log)...
[INFO] : No DRC violations after detailed routing.
[STEP 24]
[INFO] : Checking Wire Lengths (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/routing/24-wire_lengths.log)...
[STEP 25]
[INFO] : Running SPEF Extraction at the min process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/25-parasitics_extraction.min.log)...
[STEP 26]
[INFO] : Running Multi-Corner Static Timing Analysis at the min process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/26-rcx_mcsta.min.log)...
[STEP 27]
[INFO] : Running SPEF Extraction at the max process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/27-parasitics_extraction.max.log)...
[STEP 28]
[INFO] : Running Multi-Corner Static Timing Analysis at the max process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/28-rcx_mcsta.max.log)...
[STEP 29]
[INFO] : Running SPEF Extraction at the nom process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/29-parasitics_extraction.nom.log)...
[STEP 30]
[INFO] : Running Multi-Corner Static Timing Analysis at the nom process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/30-rcx_mcsta.nom.log)...
[STEP 31]
[INFO] : Running Single-Corner Static Timing Analysis at the nom process corner (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/31-rcx_sta.log)...
[WARNING] : Module sky130_fd_sc_hd__tapvprvrgnd_l1 blackboxed during sta
[WARNING] : Module sky130_ef_sc_hd__decap_l2 blackboxed during sta
[WARNING] : Module sky130_fd_sc_hd__fill_l2 blackboxed during sta
[WARNING] : Module sky130_fd_sc_hd__fill_l1 blackboxed during sta
[STEP 32]
[INFO] : Creating IR Drop Report (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/32-irdrop.log)...
[WARNING] : VSRC_LOC_FILES is not defined. The IR drop analysis will run, but the values may be inaccurate.
[STEP 33]
[INFO] : Running Magic to generate various views...
[INFO] : Streaming out GDSII with Magic (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/33-gdsii.log)...
[INFO] : Generating MAGLEF views...
[INFO] : Generating lef with Magic (/openlane/designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/33-lef.log)...
```

Flow complete가 뜬 모습

RTL코드와 json코드를 활용하여 make mount함

이 과정에서 Synthesis ~ GDSII 까지 모두 진행됨

```
[INFO] : Generating net with magic (/openlane/designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/33-lef.log)...
[STEP 34]
[INFO] : Streaming out GDSII with KLayout (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/34-gdsii-klayout.log)...
[STEP 35]
[INFO] : Running XOR on the layouts using KLayout (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/35-xor.log)...
[INFO] : No XOR differences between KLayout and Magic gds.
[STEP 36]
[INFO] : Running Magic Spice Export from LEF (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/36-spice.log)...
[STEP 37]
[INFO] : Writing Powered Verilog (logs: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/37-write_powered_def.log, designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/37-write_powered_verilog.log)...
[STEP 38]
[INFO] : Writing Verilog (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/37-write_powered_verilog.log)... [STEP 39]
[INFO] : Running LVS (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/39-lvs.lef.log)...
[STEP 40]
[INFO] : Running Magic DRC (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/40-drc.log)...
[INFO] : Converting Magic DRC database to various tool-readable formats...
[INFO] : No DRC violations after GDS streaming out.
[WARNING] : ::env(KLAYOUT_DRC_TECH_SCRIPT) is not defined or doesn't exist for the current PDK. So, GDSII streaming out using KLayout will be skipped.
[WARNING] : This warning can be turned off by setting ::env(RUN_KLAYOUT_DRC) to 0, or designating a tech file.
[STEP 41]
[INFO] : Running OpenROAD Antenna Rule Checker (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/41-arc.log)...
[STEP 42]
[INFO] : Running Circuit Validity Checker ERC (log: designs/uart_tx/runs/RUN_2025.12.11_19.43.52/logs/signoff/42-erc_screen.log)...
[INFO] : Saving current set of views in 'designs/uart_tx/runs/RUN_2025.12.11_19.43.52/results/final'...
[INFO] : Saving runtime environment...
[INFO] : Generating final set of reports...
[INFO] : Created manufacturability report at 'designs/uart_tx/runs/RUN_2025.12.11_19.43.52/reports/manufacturability.rpt'.
[INFO] : Created metrics report at 'designs/uart_tx/runs/RUN_2025.12.11_19.43.52/reports/metrics.csv'.
[INFO] : There are no max slew, max fanout or max capacitance violations in the design at the typical corner.
[INFO] : There are no hold violations in the design at the typical corner.
[INFO] : There are no setup violations in the design at the typical corner.
[SUCCESS] : Flow complete.
[INFO] : Note that the following warnings have been generated:
[WARNING] : 4 warnings found by linter
[WARNING] : Current core area is too small for the power grid settings chosen. The power grid will be scaled down.
[WARNING] : Module sky130_fd_sc_hd__tapvprvrgnd_l1 blackboxed during sta
[WARNING] : Module sky130_ef_sc_hd__decap_l2 blackboxed during sta
[WARNING] : Module sky130_fd_sc_hd__fill_l2 blackboxed during sta
[WARNING] : Module sky130_fd_sc_hd__fill_l1 blackboxed during sta
[WARNING] : VSRC_LOC_FILES is not defined. The IR drop analysis will run, but the values may be inaccurate.
[WARNING] : ::env(KLAYOUT_DRC_TECH_SCRIPT) is not defined or doesn't exist for the current PDK. So, GDSII streaming out using KLayout will be skipped.
[WARNING] : This warning can be turned off by setting ::env(RUN_KLAYOUT_DRC) to 0, or designating a tech file.
```

OpenLane Container (ff5509f):/openlane\$



03

진행 과정

6. OpenLane 각Step별 보강해석

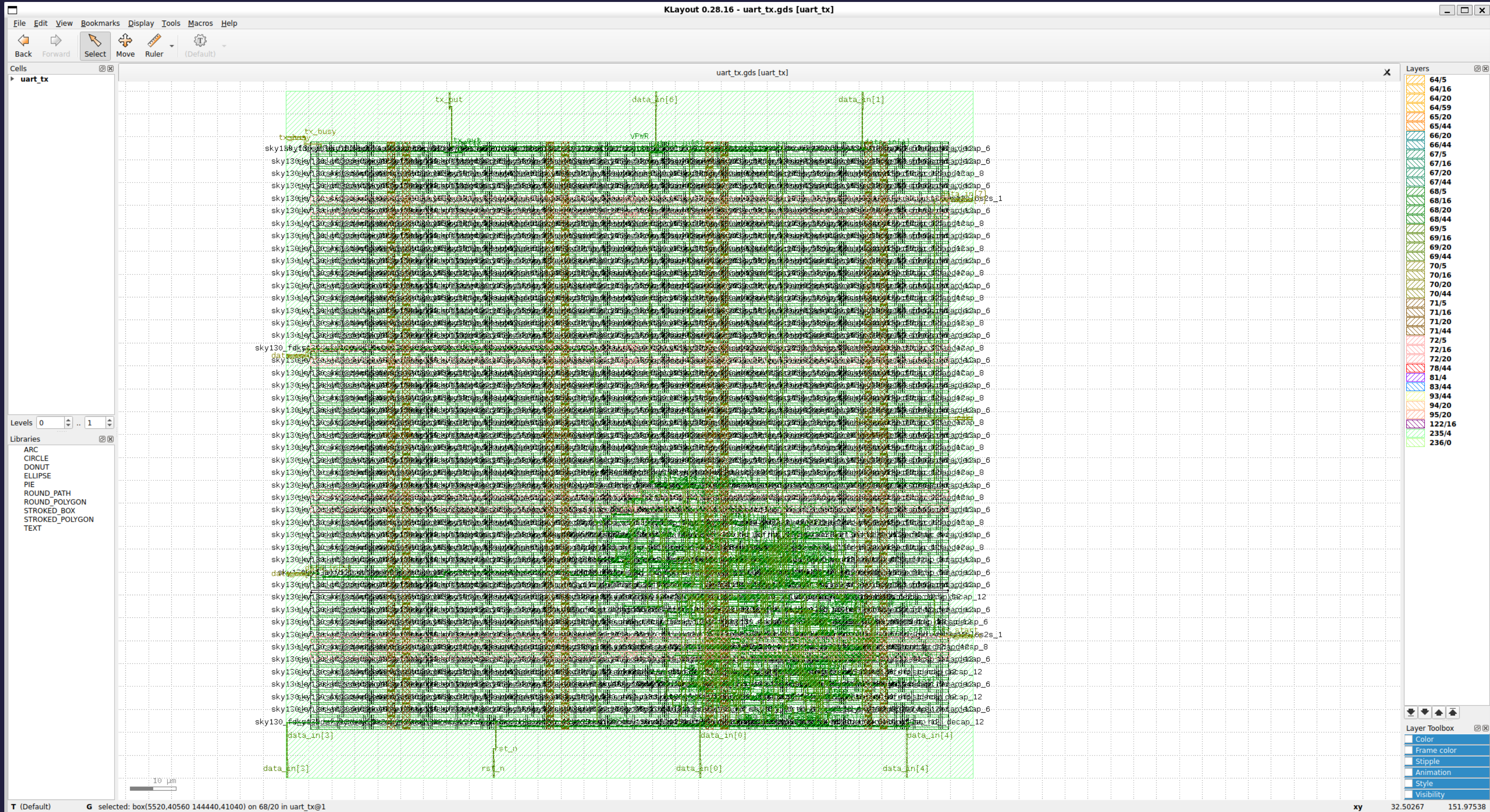
단계 (Phase)	Step	주요 프로세스	과정 설명	주목할 메세지
1. 합성 (Synthesis)	Step 1 ~ 2	RTL Synthesis Initial STA	작성한 Verilog 코드를 Netlist로 변환하고, 1차 타이밍을 점검함.	Running Synthesis Running Single-Corner STA
2. 플로어플랜 (Floorplan)	Step 3 ~ 6	Floorplanning IO Place PDN Gen	Json code에 따라 칩의 전체 크기를 잡고, 입출력 핀을 배치하며, 전원망을 설계함.	Floorplanned with width... Generating PDN
3. 배치 (Placement)	Step 7 ~ 11	Global/Detailed Placement	논리 게이트(Standard Cell)들을 칩 내부 공간에 최적의 위치로 배치함.	Running Global Placement Running Detailed Placement
4. 클럭 트리 (CTS)	Step 12 ~ 14	Clock Tree Synthesis	모든 소자에 클럭 신호가 동시에 도달하도록 클럭 전용 배선망을 구축함.	Running Clock Tree Synthesis Running CTS STA
5. 배선 (Routing)	Step 15 ~ 24	Global/Detailed Routing	배치된 소자들을 실제 금속선(Metal Wire)으로 연결하고 합선(DRC)을 검사함.	Running Detailed Routing No DRC violations after detailed routing
6. 최종 검증 (Signoff)	Step 25 ~ 32	Parasitics Extraction Final STA	배선 후 기생 성분(R, C)을 추출하여 최종적인 타이밍(Setup/Hold) 위반 여부를 확인함.	Running Multi-Corner STA There are no setup/hold violations
7. 출력 (GDSII)	Step 33 ~ 42	GDSII DRC / LVS	최종 레이아웃 파일(.gds)을 생성하고, 설계 도면과 레이아웃의 일치 여부를 최종 확인.	Streaming out GDSII SUCCESS: Flow complete



03

# 진행 과정

7. Layout (by Using KLayout)

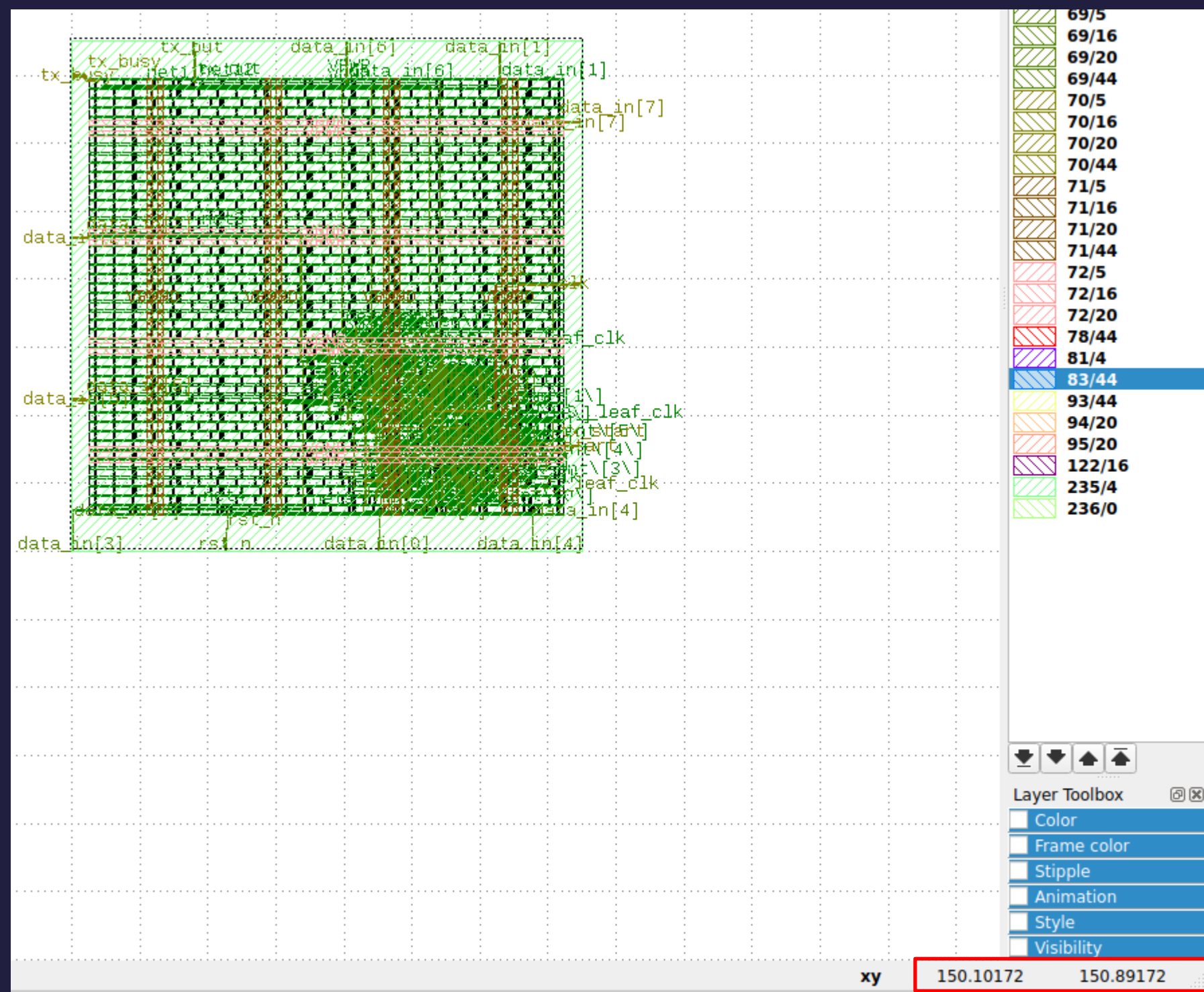




03

# 진행 과정

7. Layout (by Using KLayout)



Layout이 형성된 Grid영역의 길이를 재보면 초기 config.json 설계대로 **150x150 Grid** 형성



03

진행 과정

8. Report (STA, GDS)

STA ( by using OpenSTA )

선택 sjh@DESKTOP-KDK01K9: ~/projects/OpenLane/designs/uart\_tx/runs/RUN\_2025.12.11\_20.28.22/reports/synthesis

Startpoint: \_227\_ (rising edge-triggered flip-flop clocked by clk)  
Endpoint: \_244\_ (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path Type: max

Fanout	Cap	Slew	Delay	Time	Description
7	0.02	0.15	0.00	0.00	clock clk (rise edge)
		0.00	0.00	0.00	clock network delay (ideal)
		0.15	0.00	0.00	^ _227_/CLK (sky130_fd_sc_hd__dfrtp_2)
		0.07	0.49	0.49	v _227_/D (sky130_fd_sc_hd__dfrtp_2)
6	0.01	0.07	0.00	0.49	v _140_/A1 (sky130_fd_sc_hd__o21a_2)
		0.05	0.25	0.73	v _140_/X (sky130_fd_sc_hd__o21a_2)
		0.05	0.00	0.73	v _156_/A (sky130_fd_sc_hd__buf_1)
		0.10	0.15	0.89	v _156_/X (sky130_fd_sc_hd__buf_1)
10	0.02	0.10	0.00	0.89	v _193_/B (sky130_fd_sc_hd__and2_2)
		0.03	0.21	1.10	v _193_/X (sky130_fd_sc_hd__and2_2)
		0.10	0.00	0.89	v _195_/B (sky130_fd_sc_hd__and2_2)
		0.03	0.21	1.10	v _195_/X (sky130_fd_sc_hd__and2_2)
1	0.00	0.03	0.00	1.10	v _194_/A (sky130_fd_sc_hd__buf_1)
		0.02	0.08	1.18	v _194_/X (sky130_fd_sc_hd__buf_1)
		0.02	0.00	1.18	v _244_/D (sky130_fd_sc_hd__dfrtp_2)
		0.02	0.00	1.18	data arrival time
1	0.00	0.15	20.00	20.00	clock clk (rise edge)
		0.00	20.00	20.00	clock network delay (ideal)
		-0.25	19.75	19.75	clock uncertainty
		0.00	19.75	19.75	clock reconvergence pessimism
		0.00	19.75	19.75	^ _244_/CLK (sky130_fd_sc_hd__dfrtp_2)
		-0.08	19.67	19.67	library setup time
		-0.08	19.67	19.67	data required time
		-0.08	19.67	19.67	data required time
				19.67	data required time
				-1.18	data arrival time
					18.49 slack (MET)

Startpoint: \_227\_ (rising edge-triggered flip-flop clocked by clk)  
Endpoint: \_245\_ (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path Type: max

Fanout	Cap	Slew	Delay	Time	Description
7	0.02	0.15	0.00	0.00	clock clk (rise edge)
		0.00	0.00	0.00	clock network delay (ideal)
		0.15	0.00	0.00	^ _227_/CLK (sky130_fd_sc_hd__dfrtp_2)
		0.07	0.49	0.49	v _227_/D (sky130_fd_sc_hd__dfrtp_2)
6	0.01	0.07	0.00	0.49	v _140_/A1 (sky130_fd_sc_hd__o21a_2)
		0.05	0.25	0.73	v _140_/X (sky130_fd_sc_hd__o21a_2)
		0.05	0.00	0.73	v _156_/A (sky130_fd_sc_hd__buf_1)
		0.10	0.15	0.89	v _156_/X (sky130_fd_sc_hd__buf_1)
10	0.02	0.10	0.00	0.89	v _195_/B (sky130_fd_sc_hd__and2_2)
		0.03	0.21	1.10	v _195_/X (sky130_fd_sc_hd__and2_2)
		0.10	0.00	0.89	v _196_/A (sky130_fd_sc_hd__buf_1)
		0.03	0.21	1.10	v _196_/X (sky130_fd_sc_hd__buf_1)
1	0.00	0.03	0.00	1.10	v _196_/A (sky130_fd_sc_hd__buf_1)
		0.02	0.08	1.18	v _196_/X (sky130_fd_sc_hd__buf_1)
		0.02	0.00	1.18	v _245_/D (sky130_fd_sc_hd__dfrtp_2)
		0.02	0.00	1.18	data arrival time
1	0.00	0.15	20.00	20.00	clock clk (rise edge)
		0.00	20.00	20.00	clock network delay (ideal)
		-0.25	19.75	19.75	clock uncertainty
		0.00	19.75	19.75	clock reconvergence pessimism
		0.00	19.75	19.75	^ _245_/CLK (sky130_fd_sc_hd__dfrtp_2)
		-0.08	19.67	19.67	library setup time
		-0.08	19.67	19.67	data required time
		-0.08	19.67	19.67	data required time
				19.67	data required time
				-1.18	data arrival time
					18.49 slack (MET)

STA Report상에서 Met 확인

```
sjh@DESKTOP-KDK01K9: ~/projects/OpenLane/designs/uart_tx/runs/RUN_2025.12.11_20.28.22/reports/synthesis$ cat 2-syn_sta.summary.rpt

=====
report_tns
=====
tns 0.00

=====
report_wns
=====
wns 0.00

=====
report_worst_slack -max (Setup)
=====
worst slack 14.48

=====
report_worst_slack -min (Hold)
=====
worst slack 0.25
```

STA Report summary상에서  
Setup time 과 Hold time 의 Slack이 모두 양수임을 확인 -> Violation 아님 = 정상동작

STA ➡ 에러없음

```
(venv) sjh@DESKTOP-KDK01K9:~/projects/OpenLane/designs/uart_tx/runs/RUN_2025.12.11_19.43.52/results/final/
gds$ ls -l
total 1020
-rw-r--r-- 1 sjh sjh 1041150 Dec 12 04:44 uart_tx.gds
```

Result에 gds파일이 생성됨

## 03

## 진행 과정

9. DRC(by using Magic) ,LVS(by using Netgen in OpenSTA)

```
(venv) sjh@DESKTOP-KDK01K9:~/projects/OpenLane/designs/uart_tx/runs/RUN_2025.12.11_19.43.52/reports$ python3 -c "import csv; f=open('metrics.csv'); r=list(csv.DictReader(f))[0]; print(f'Magic DRC: {r.get(\"Magic_violations\")}\nWNS: {r.get(\"wns\")}\nTNS: {r.get(\"tns\")}')"
Magic DRC: 0
WNS: 0.0
TNS: 0.0
```

DRC = 0 → 에러없음

```
sjh@DESKTOP-KDK01K9:~$ grep "Total errors" 39-lvs.lef.log
grep: 39-lvs.lef.log: No such file or directory
```

LVS → 에러없음

이상으로 STA,DRC,LVS를 모두 통과함



03

진행 과정

10. Github Upload & Test

9 workflow runs		Event ▾	Status ▾	Branch ▾	Actor ▾
✔ Final Submission: UART Transmitter Design				main	
Verilog Simulation #4: Commit <a href="#">ba766f0</a> pushed by <a href="#">sjh1v</a>			Dec 12, 6:47 AM GMT+9 ⌚ 23s		...
✔ Add files via upload				main	
docs #3: Commit <a href="#">5ce3776</a> pushed by <a href="#">sjh1v</a>			Oct 31, 11:08 PM GMT+9 ⌚ 2m 19s		...
✔ Add files via upload				main	
gds #5: Commit <a href="#">5ce3776</a> pushed by <a href="#">sjh1v</a>			Oct 31, 11:08 PM GMT+9 ⌚ 6m 10s		...
✔ Add files via upload				main	
Verilog Simulation #3: Commit <a href="#">5ce3776</a> pushed by <a href="#">sjh1v</a>			Oct 31, 11:08 PM GMT+9 ⌚ 1m 54s		...

Github 상에서 gds, docs, test 모두 통과



감사합니다.

