



# Cortex-M0 기반의 SoC 설계 프로젝트

# AAA 프로그램 소개

## Arm Academic Access

### What do you get?

<b>Processors</b> Cortex-A53 Cortex-A35 Cortex-A34 Cortex-A32 Cortex-A7 Cortex-A5 (UP and MP)	<b>Graphics Processors</b> Mali-G52 Mali-G31 Also includes Mali DDKs	<b>System Controllers</b> CoreLink GIC-500 CoreLink GIC-400 PL192 VIC CoreLink TZC-400 CoreLink L2C-310 CoreLink MMU-500 BP140 AXI BP141 TrustZone Mem. Adapter	<b>Peripheral Controllers</b> PL011 UART PL022 SPI PL031 RTC CoreLink DMA-230 CoreLink DMA-330
<b>CoreLink Interconnect</b> Cortex-R52 Cortex-R8 Cortex-R5 Cortex-M33 Cortex-M23 Cortex-M7 Cortex-M4 Cortex-M3 Cortex-M0+ Cortex-M0		<b>Security IP</b> True Random Number Generator	<b>CoreSight Debug &amp; Trace</b> CoreSight SoC-400 CoreSight SoC-600 CoreSight SDC-600 CoreSight STM-500 CoreSight System Trace Macrocell CoreSight Trace Memory Controller
		<b>Corstone Foundation IP</b> Corstone-101 Corstone-102 Corstone-201 Corstone-500	<b>Tools &amp; Models</b> Socrates IP Tooling Arm DS-Gold Virtual system models (fast & cycle accurate) MDK V8-M

### Relevant Research Areas

- System design to support specific research applications / topics**  
Accelerators and SW for specific use cases, peripherals, analog circuits
- System design to support architecture research**  
Heterogenous subsystems design, focus on accelerator topology, cache hierarchy, NoC technologies, system security features etc.
- System design to support design methodology research**  
New system design methodologies, tools and flows
- Physical Design**  
Physical IP used for efficient implementation of complex SoC designs including compatibility testing and optimisation .

Note: Arm IP is subject to applicable export control requirements. A list of export classifications for Arm products is available on the Arm Academic Access website.

7 Arm Limited © 2020

**arm**

# 디지털 라이브러리 테스트

# 실습 준비

# 디지털 라이브러리 테스트

## 실습준비

1. \$> cd SoC1 ↵

2. \$> mkdir GPDK045 ↵

```
[ex_poly1@npit ~]$ cd SoC1  
[ex_poly1@npit ~/SoC1]$ mkdir GPDK045
```

디렉토리  
생성

```
[ex_poly1@npit ~/SoC2]$ ls  
GPDK045
```

# 디지털 라이브러리 테스트

## 실습준비

1. \$> cp -a ~/digital.tar.gz . ↵

복사

```
[ex_poly1@npit GPDK045]$ cp -a ~/digital.tar.gz .
```

2. \$> tar xvzf digital.tar.gz ↵

압축해제

```
[ex_poly1@npit GPDK045]$ tar xvzf digital.tar.gz
```

```
[ex_poly1@npit GPDK045]$ ls  
digital digital.tar.gz
```

# 디지털 라이브러리 테스트

## 실습준비

1. \$> cd ↵

2. \$> source digital.cshrc ↵

```
[ex_poly1@npit GPDK045]$ cd  
[ex_poly1@npit ~]$ source digital.cshrc
```

\* \$> source ~/digital.cshrc

# 디지털 라이브러리 테스트

## 실습준비

1. cd SoC1 ↵
2. cd GPDK045 ↵
3. cd digital ↵
4. cd TEST ↵
5. cd std\_cell ↵

```
[ex_poly1@npit GPDK045]$ ls  
digital
```

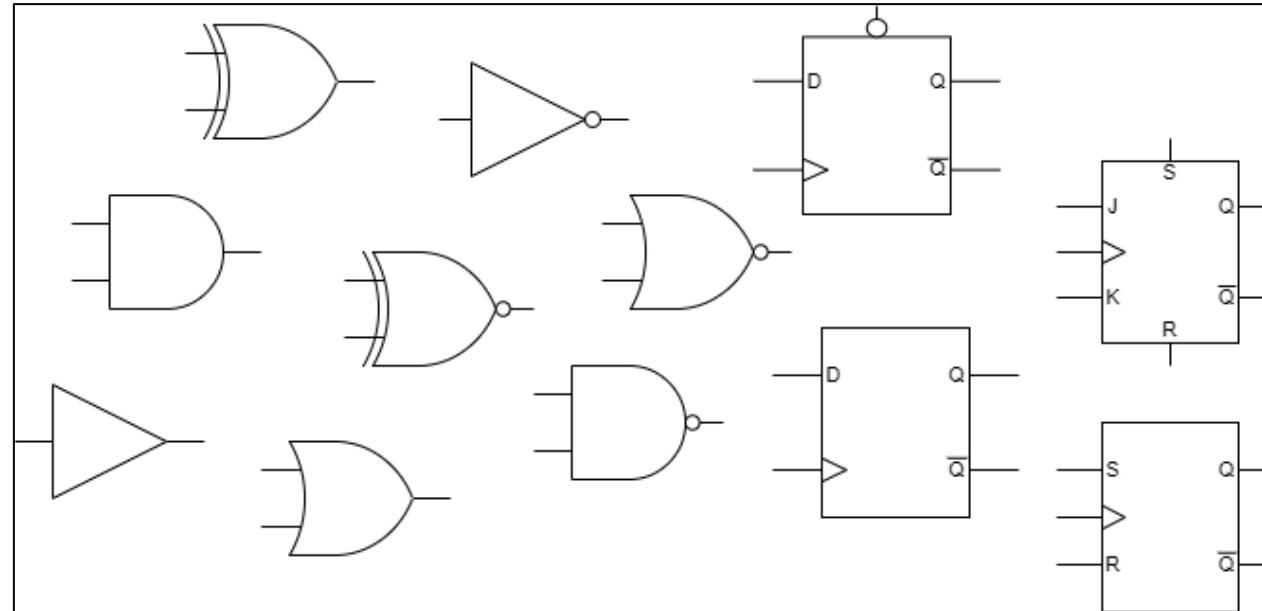
```
[ex_poly1@npit digital]$ cd TEST/  
[ex_poly1@npit TEST]$ ls  
io std_cell
```

# STD셀 라이브러리 시뮬레이션

# 디지털 라이브러리 테스트

## STD셀 이란?

1. 우리가 흔히 알고 있는 AND, OR, XOR, 플립플롭 등 기본 논리 게이트가 들어있는 셀



# LAB1

# 디지털 라이브러리 테스트

## LAB1의 목적

1. standard sell에서 몇가지 cell을 가져와 동작을 시뮬레이션
2. 게이트의 동작을 시뮬레이션 -gui옵션으로 확인
3. Dump 폴더에 파형이 자동 입력
4. 시뮬레이션 start하면 정해진 시간에 stop 확인

# 디지털 라이브러리 테스트

## LAB1 파일 확인

1. \$>cd LAB1 ↵

2. \$> ll ↵

```
[ex_poly1@npit LAB1]$ ll
total 108
-rwxr-xr-x 1 ex_poly1 rnd    34 Oct 14 00:19 clean.tcl
drwxr-xr-x 3 ex_poly1 rnd    33 Oct 14 00:44 dump
-rw-r--r-- 1 ex_poly1 rnd   319 Oct 27 15:48 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd 42418 Oct 27 15:49 func_sim.log
drwxr-xr-x 2 ex_poly1 rnd    37 Oct 14 00:45 log
-rw-r--r-- 1 ex_poly1 rnd   115 Oct 15 23:40 README
-rwxr-xr-x 1 ex_poly1 rnd   662 Oct 15 23:44 run_function.tcl
-rw-r--r-- 1 ex_poly1 rnd  7026 Oct 14 21:53 shm.prof
-rw-r--r-- 1 ex_poly1 rnd   859 Oct 15 23:44 TB_std_cell.v
drwxr-xr-x 6 ex_poly1 rnd   214 Oct 27 15:48 xcelium.d
-rw-r--r-- 1 ex_poly1 rnd 23170 Oct 27 15:49 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd   577 Oct 23 18:45 xrun.history
-rw-r--r-- 1 ex_poly1 rnd      5 Oct 27 15:49 xrun.key
-rw-r--r-- 1 ex_poly1 rnd  331 Oct 23 18:45 xrun.log
```

# 디지털 라이브러리 테스트

## README 파일 확인

### 1. \$> vi README ↵

- 시뮬레이션이 이런 식으로 진행되는 것을 알 수 있음

```
with option -gui  
simvision auto started  
dump folder is auto opened  
press start button and stop at $stop function
```

# 디지털 라이브러리 테스트

## clean.tcl 파일 확인

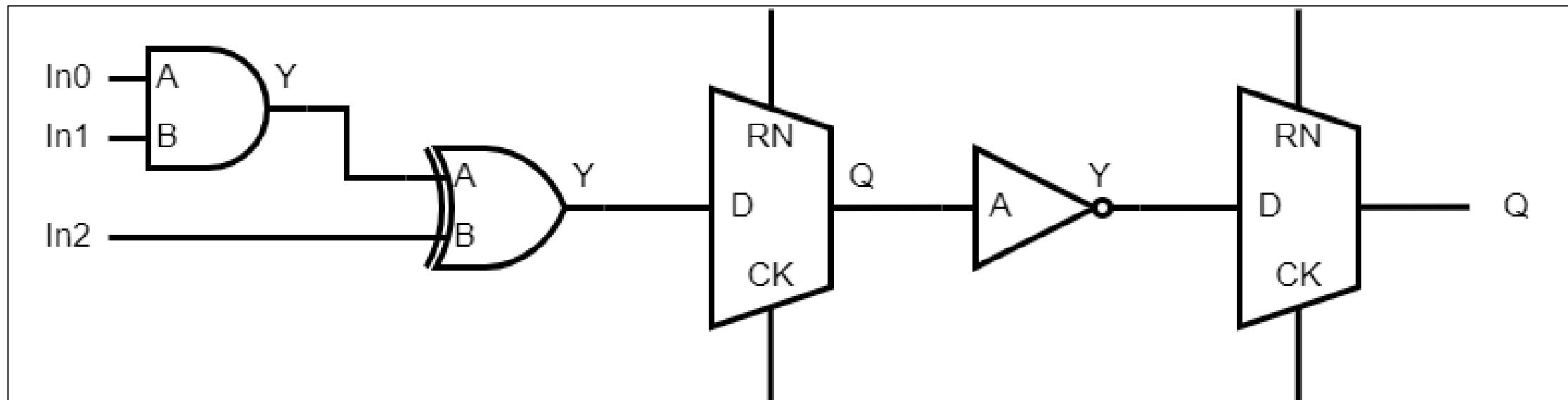
1. \$> vi clean.tcl ↵

- clean.tcl 을 실행할 시 일어나는 일들을 정리해둔 파일
- clean.tcl 실행 시 rm -rf이 자동으로 실행 됨

```
rm -rf work xcelium.d/ waves.shm
```

# 디지털 라이브러리 테스트

## TB\_std\_cell.v 구조



<그림1>

# 디지털 라이브러리 테스트

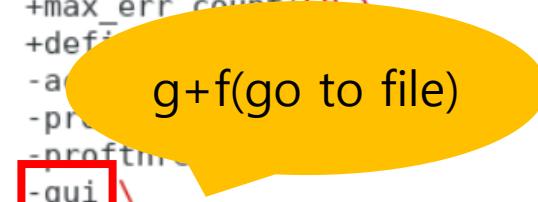
## run\_function.tcl 파일 확인

1. \$> vi run\_function.tcl ↵

- xrun 명령에 대한 여러 옵션 확인
- -gui 옵션을 통해 simvision이 자동으로 실행됨

```
#debug option is essential to view all design hierarchy
#vcs -full64 -j16 -fgp -debug_access+all -kdb -lca
#
#vcs -full64 -debug_access+all -kdb -lca \
#-debug \
#+define+function_sim \
#+v2k \
#+libext+.v+.vlib \
#+incdir+../../SAED32_EDK/lib/stdcell_rvt/verilog\
#./TB_std_cell.v \
#../../SAED32_EDK/lib/stdcell_rvt/verilog/saed32nm.v\
#-l func_sim.log

#      -libext+.v+.s
xrun -64bit \
+max_err_count=50 \
+defn \
-a \
-pri \
-prf \
-prfth \
-gui \
TB_std_cell.v \
../../../../gsclib045_all_v4.4/gsclib045_svt_v4.4/gsclib045/verilog/slow_v \
dd1v0_basicCells.v \
-l ./func_sim.log
```



g+f(go to file)

# 디지털 라이브러리 테스트

## TB\_std\_cell.v 구조

1. \$> vi TB\_std\_cell.v

:set nu

- 13행 부터 48행에서 TB의 구조 확인

```
13 AND2X1 UAND2X1 (
14     .A(IN[0]),
15     .B(IN[1]),
16     .Y(AND_RESULT)
17 );
18
19
20 XOR2X1 UXOR2X1 (
21     .A(AND_RESULT),
22     .B(IN[2]),
23     .Y(XOR_RESULT)
24 );
25
26
27
28 DFFRHQX1 UDFFRHQX1 (
29     .D(XOR_RESULT),
30     .CK(CK),
31     .RN(RN),
32     .Q(Q1)
33 );
34
35
36
37 INVX1 UINVX1 (
38     .A(Q1),
39     .Y(Q1_BAR)
40 );
41
42
43 DFFRHQX1 UDFFRHQX2 (
44     .D(Q1_BAR),
45     .CK(CK),
46     .RN(RN),
47     .Q(Q2)
48 );
```

# 디지털 라이브러리 테스트

검색하고 싶은 단어 Shift+3

## TB\_std\_cell.v 구조

- **Initial begin** : reg 타입으로 선언한 모든 것들에 대한 초기값 설정
- **always begin** : 반복적으로 일어나는 것에 대한 기술

```
50 initial begin
51     CK      <= 1'b0;
52     IN      <= 3'b000;
53     RN      <= 1'b1;
54
55 #199    RN      <= 1'b0;
56 #100    RN      <= 1'b1;
57
58 #500
59 $stop;
60 end
61
62 always begin
63 #2      CK      <= ~CK;
64 end
65
66 always begin
67 #2      IN      <= IN+1'b1;
68 end
```

# 디지털 라이브러리 테스트

## TB\_std\_cell.v 구조

- 70행 initial begin의 경우 특별히 ifdef기능을 이용하여 외부파형을 저장할 수 있게 하는 구문 사용함

```
70 initial begin
71     `ifdef function_sim
72         // $sdf_annotate (".../.../xx.sdf", TB_std_cell.core,,, "MAXIMUM");
73         $shm_open("./dump/TB_std_cell");
74         $shm_probe(TB_std_cell, "AC");
75         TEST <= $fopen("./log/tb_std_cell.rpt");
76     `endif
77 end
```

# LAB1

# 시뮬레이션

# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

1. ./clean.tcl ↵

2. ./run\_function.tcl ↵

- xrun 실행

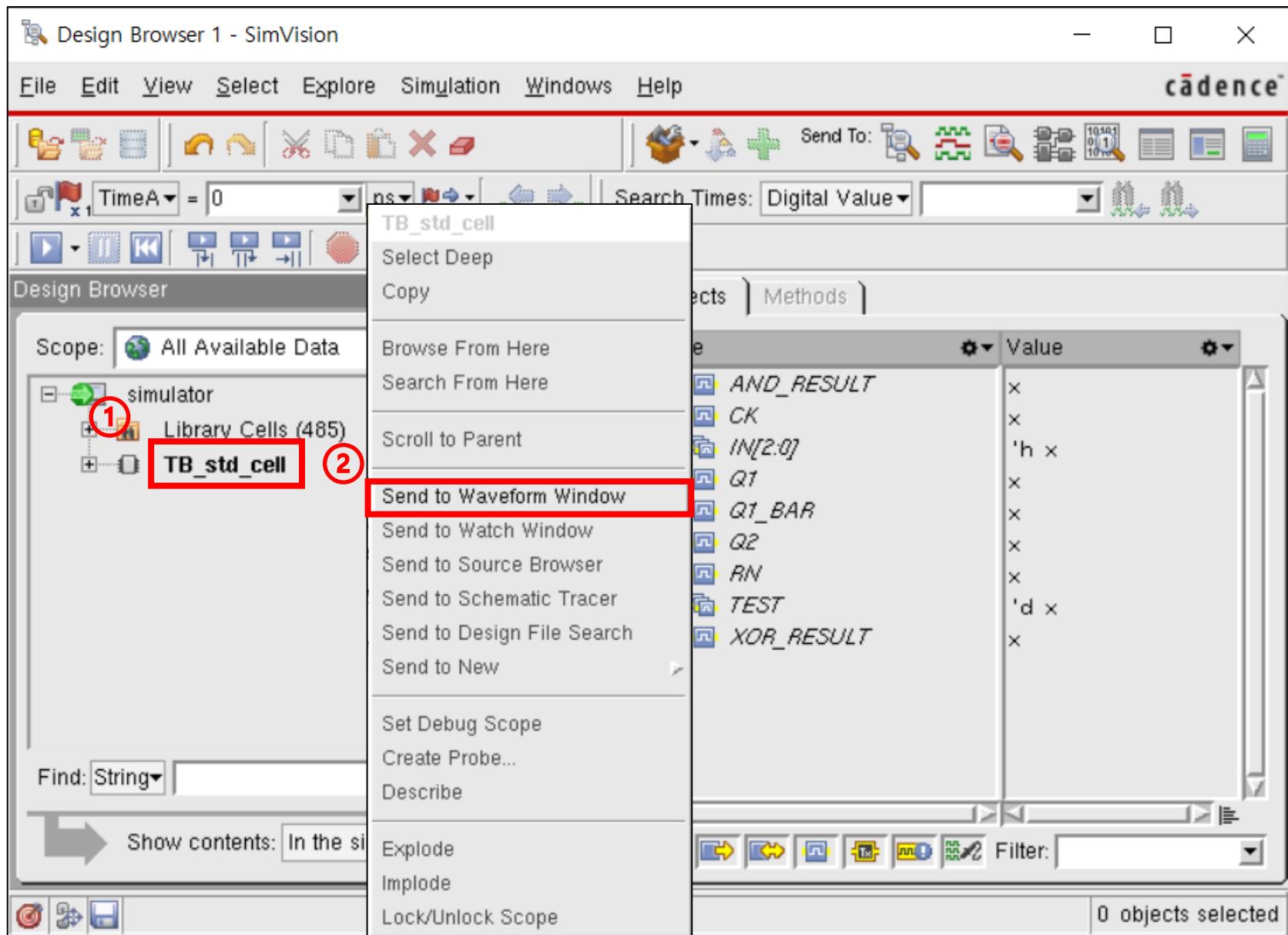
```
[ex_poly1@npit LAB1]$ ./clean.tcl
[ex_poly1@npit LAB1]$ ./run_function.tcl
```

# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- ① 마우스 우 클릭
- ② Send to... 클릭



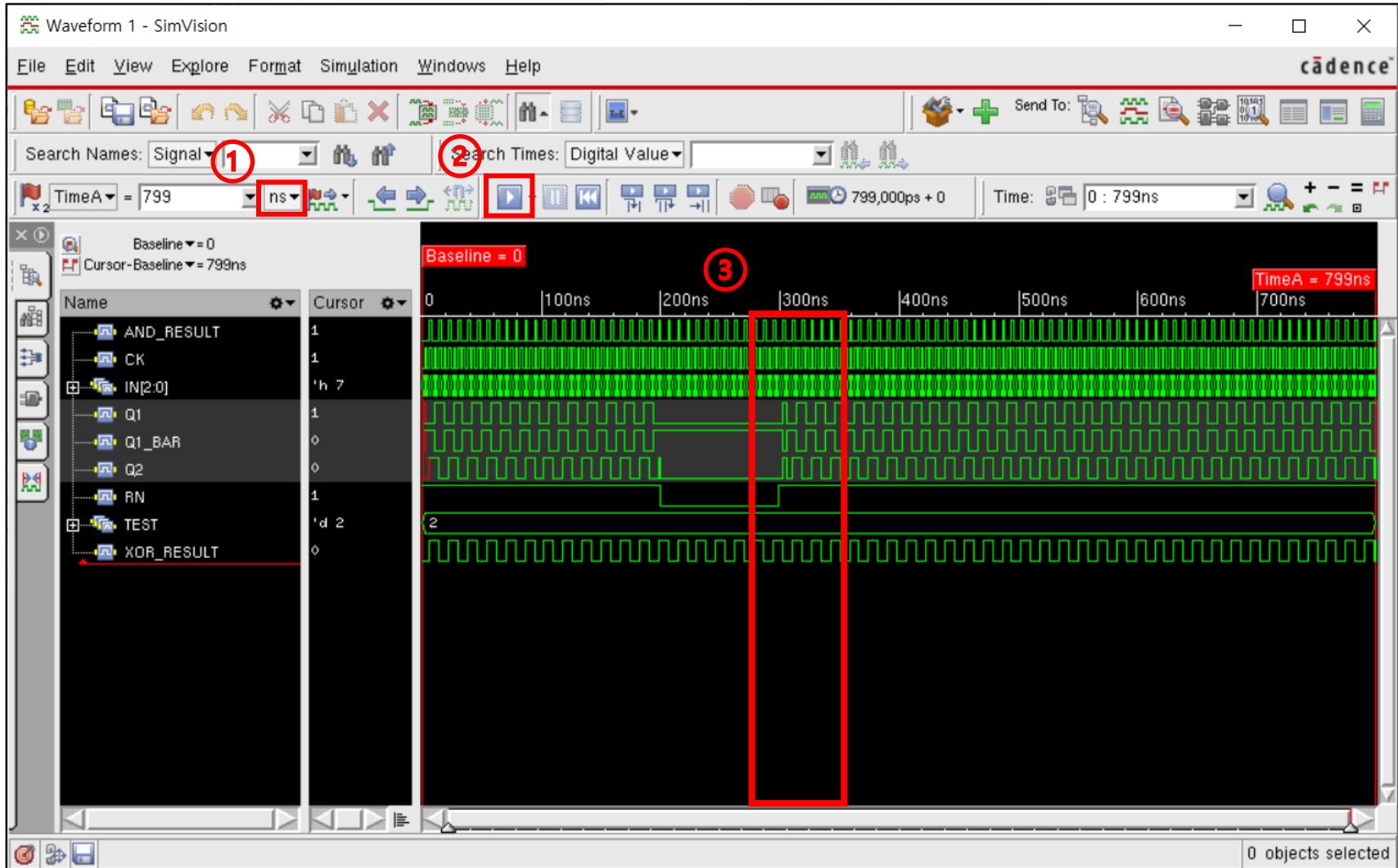
# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- ① ps → ns
- ② 실행 버튼 클릭
- ③ 드래그시 확대 가능

- Ctrl + 휼, 드래그로 범위 설정

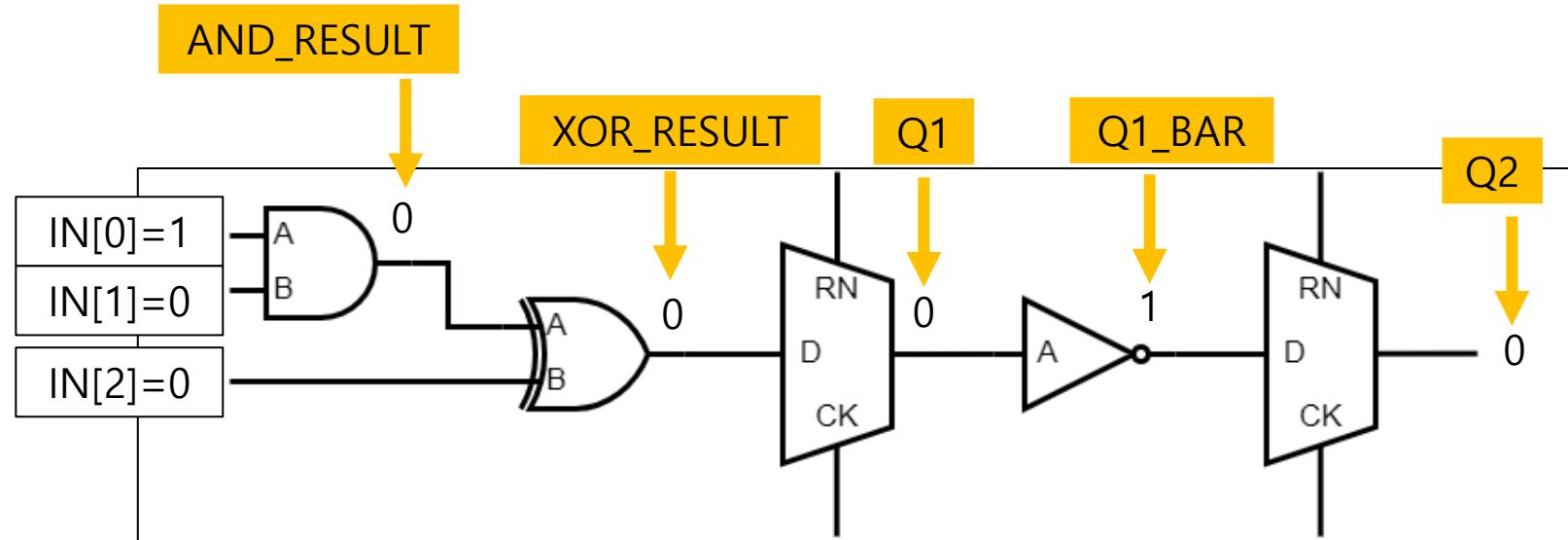


# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. 파형 결과 확인

- 리셋 직후 결과 확인



# LAB2

# 디지털 라이브러리 테스트

## LAB2의 목적

1. -gui 옵션 없으므로 simvision옵션을 사용하여 시뮬레이션 확인
2. 시뮬레이션 재시작 할 필요 없이 Dump 폴더 database로 파형 확인
3. TB\_std\_se1l.v내에 \$finish 명령 확인

# 디지털 라이브러리 테스트

## LAB2 파일 확인

1. \$>cd LAB2 ↵

2. \$> ll ↵

```
[ex_poly1@npit LAB2]$ ll
total 96
-rw-r--r-- 1 ex_poly1 rnd    135 Oct 15 23:46 README
-rw-r--r-- 1 ex_poly1 rnd   860 Oct 15 23:44 TB_std_cell.v
-rwxr-xr-x 1 ex_poly1 rnd     34 Oct 14 00:19 clean.tcl
drwxr-xr-x 3 ex_poly1 rnd    33 Oct 14 00:44 dump
-rw-r--r-- 1 ex_poly1 rnd   314 Oct 27 15:49 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd 42346 Oct 27 15:49 func_sim.log
drwxr-xr-x 2 ex_poly1 rnd    37 Oct 15 22:30 log
-rwxr-xr-x 1 ex_poly1 rnd   650 Oct 15 23:44 run_function.tcl
-rw-r--r-- 1 ex_poly1 rnd  7026 Oct 27 15:49 shm.prof
-rw-r--r-- 1 ex_poly1 rnd 11598 Oct 27 15:49 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd   177 Oct 14 00:04 xrun.history
-rw-r--r-- 1 ex_poly1 rnd      5 Oct 15 23:29 xrun.key
-rw-r--r-- 1 ex_poly1 rnd   432 Oct 14 00:04 xrun.log
```

# 디지털 라이브러리 테스트

## README 파일 확인

### 1. \$> vi README ↵

- 시뮬레이션이 이런 식으로 진행되는 것을 알 수 있음

```
$finish in testbench file  
1) just ./run_function.tcl  
2) $> simvision  
3) open database in dump folder  
4) no need to restart simulation
```

# 디지털 라이브러리 테스트

## TB\_std\_cell.v 구조

1. \$> vi TB\_std\_cell.v

- 59행 \$finish 명령어 확인
- 나머지 LAB1과 동일

```
50 initial begin
51     CK      <= 1'b0;
52     IN      <= 3'b000;
53     RN      <= 1'b1;
54
55 #199    RN      <= 1'b0;
56 #100    RN      <= 1'b1;
57
58 #500
59 $finish;
60 end
61
62 always begin
63 #2      CK      <= ~CK;
64 end
65
66 always begin
67 #2      IN      <= IN+1'b1;
68 end
```



시뮬레이션  
799ns 까지  
진행

# LAB2

# 시뮬레이션

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

1. \$> ./clean.tcl ↵

2. \$> ./run\_function.tcl ↵

- xrun 실행

-gui 옵션을 넣지 않아 799ns 시뮬레이션  
진행 후 아무런 창이 뜨지 않음

```
[ex_poly1@npit LAB2]$ ./clean.tcl
[ex_poly1@npit LAB2]$ ./run_function.tcl
```

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

1. \$> //

시뮬레이션 입력 시간 확인 가능						
[ex_poly1@npit LAB2]\$ ll						
total 96						
-rw-r--r-- 1 ex_poly1 rnd 135 Oct 27 23:46 README						
-rw-r--r-- 1 ex_poly1 rnd 860 Oct 27 23:44 TB_std_cell.v						
-rwxr-xr-x 1 ex_poly1 rnd 34 Oct 27 00:19 clean.tcl						
drwxr-xr-x 3 ex_poly1 rnd 33 Oct 27 00:44 dump						
-rw-r--r-- 1 ex_poly1 rnd 314 Oct 27 15:49 func_sim.history						
-rw-r--r-- 1 ex_poly1 rnd 42346 Oct 27 15:49 func_sim.log						
drwxr-xr-x 2 ex_poly1 rnd 37 Oct 15 22:30 log						
-rwxr-xr-x 1 ex_poly1 rnd 650 Oct 15 23:44 run_function.tcl						
-rw-r--r-- 1 ex_poly1 rnd 7026 Oct 27 15:49 shm.prof						
drwxr-xr-x 6 ex_poly1 rnd 214 Oct 27 15:49 xcelium.d						
-rw-r--r-- 1 ex_poly1 rnd 11598 Oct 27 15:49 xmprof.out						
-rw-r--r-- 1 ex_poly1 rnd 177 Oct 14 00:04 xrun.history						
-rw-r--r-- 1 ex_poly1 rnd 5 Oct 15 23:29 xrun.key						
-rw-r--r-- 1 ex_poly1 rnd 432 Oct 14 00:04 xrun.log						

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. \$> simvision

- simvision 명령으로 시뮬레이션 파형 결과 확인

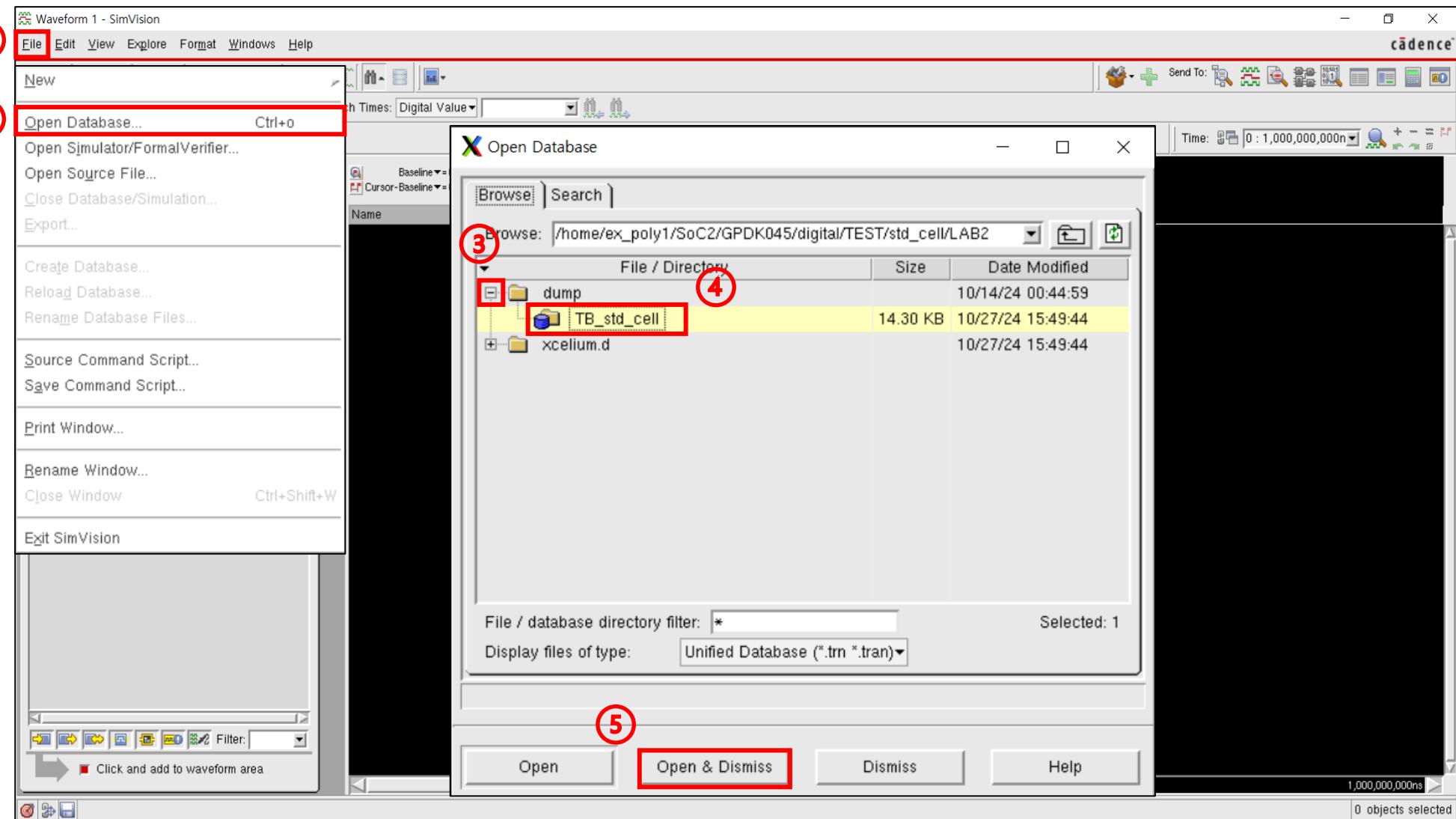
```
[ex_poly1@npit LAB2]$ simvision
simvision(64): 24.03-s005: (c) Copyright 1995-2024 Cadence Design Systems, Inc.
txe(64): 24.03-s005: (c) Copyright 1995-2024 Cadence Design Systems, Inc.
```

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. \$> simvision

- ① File 클릭
- ② Open Database 클릭
- ③ dump 폴더 오픈
- ④ TB\_std\_cell 클릭
- ⑤ Open&Dismiss 클릭

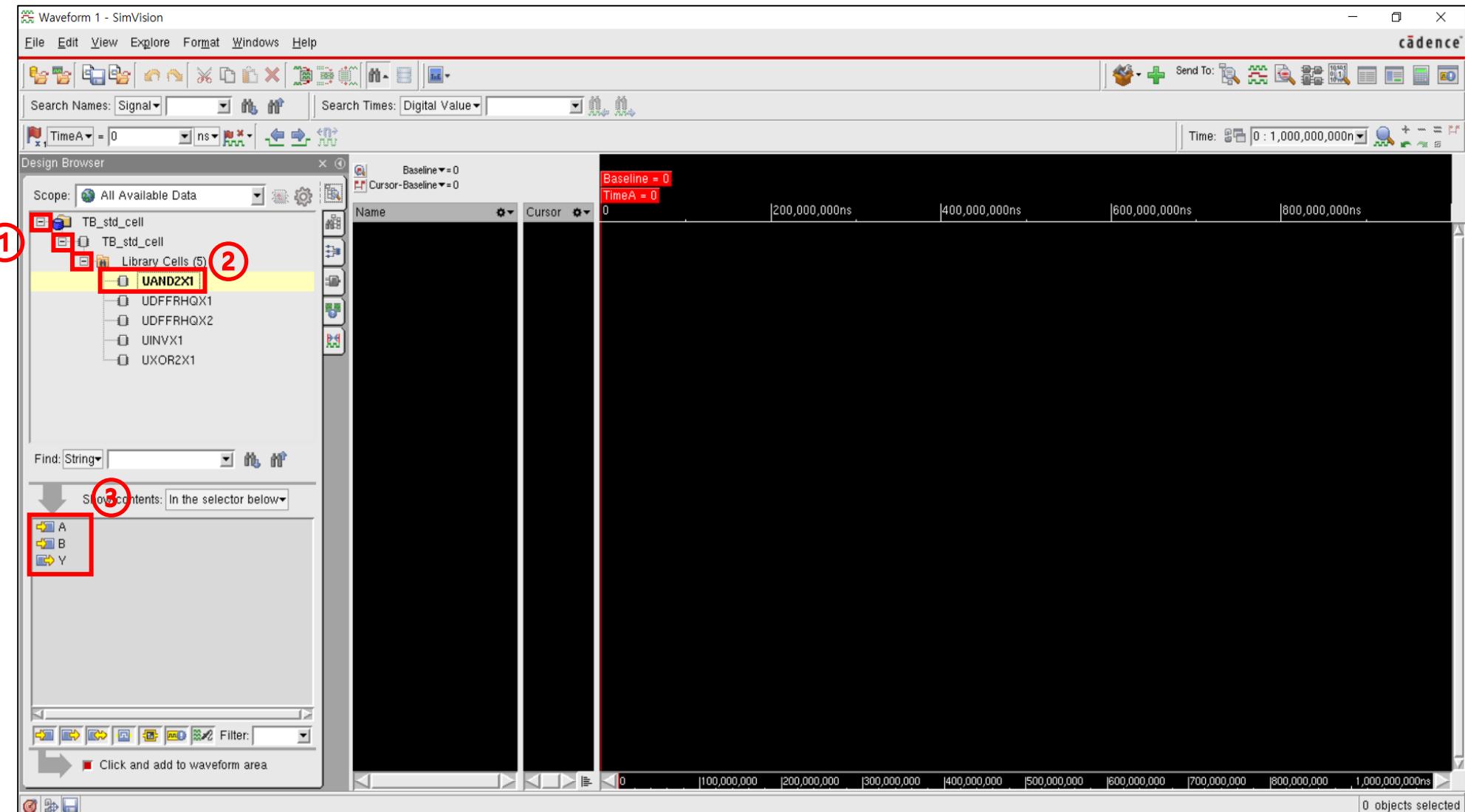


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. \$> simvision

- ① Library Cell 오픈
- ② UAND2X1 클릭
- ③ A, B, Y 순서로 클릭

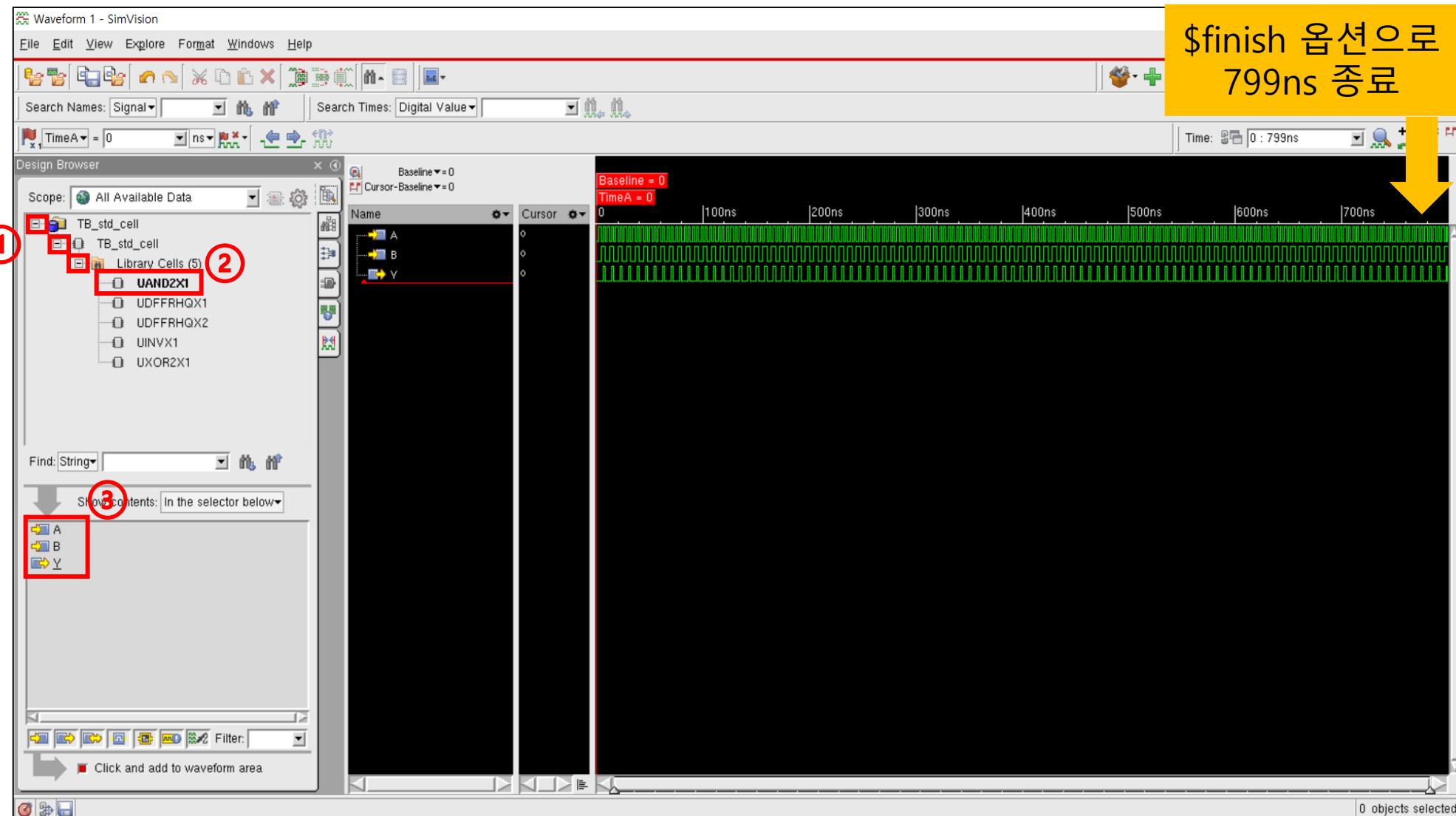


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. \$> simvision

- ① Library Cell 오픈
- ② UAND2X1 클릭
- ③ A, B, Y 순서로 클릭



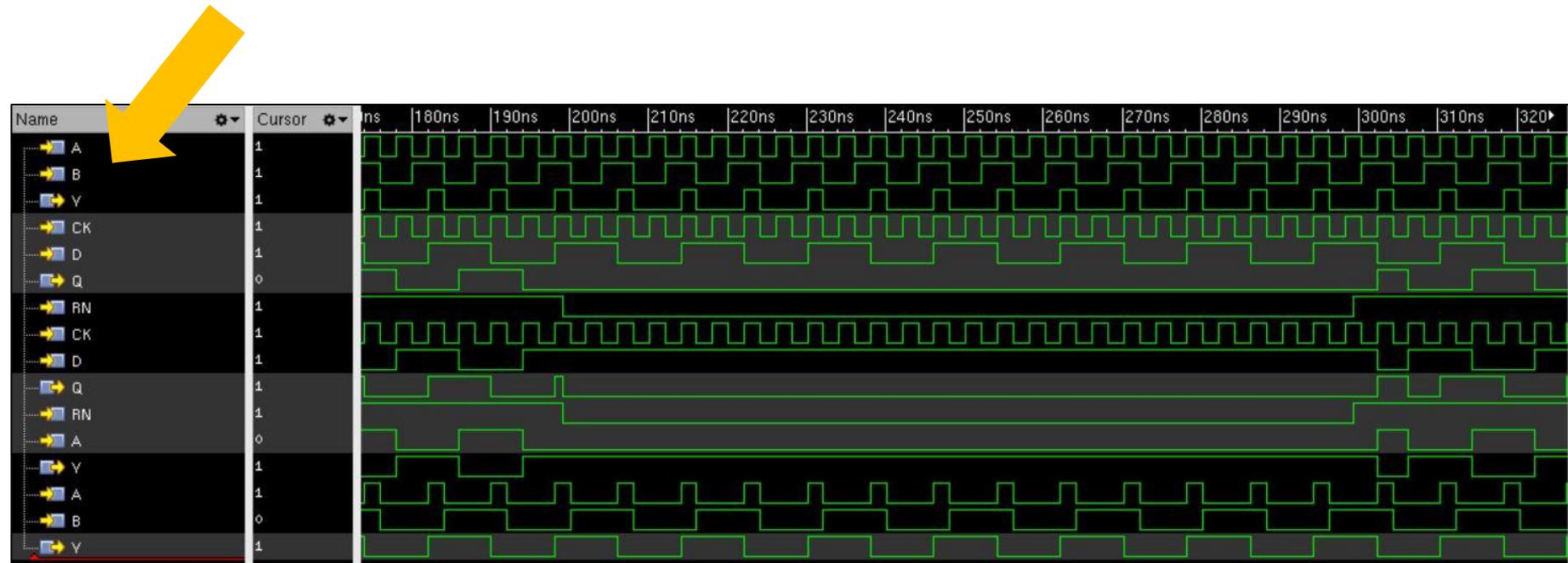
# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. \$> simvision

- LAB1과 결과가 같은 것을 알 수 있음

내가 원하는 파형만  
불러올 수 있음



# LAB3

# 디지털 라이브러리 테스트

## LAB3의 목적

1. -gui 옵션 사용 가능 simvision 확인
2. clock의 반전이 0.5ns 주기는 1ns
3. clock의 반전을 빠르게 했을 때 동작을 제대로 할 수 있나 확인
4. 최대 동작 주파수와 최대 동작 스피드 확인

# 디지털 라이브러리 테스트

## LAB3 파일 확인

1. \$>cd LAB3 ↵

2. \$> ll ↵

```
[ex_poly1@npit LAB3]$ ll
total 108
-rw-r--r-- 1 ex_poly1 rnd    178 Oct 20 22:31 README
-rw-r--r-- 1 ex_poly1 rnd    889 Oct 20 22:27 TB_std_cell.v
-rwxr-xr-x 1 ex_poly1 rnd     34 Oct 14 00:19 clean.tcl
drwxr-xr-x 3 ex_poly1 rnd    25 Oct 14 00:44 dump
-rw-r--r-- 1 ex_poly1 rnd   319 Nov  2 00:46 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd 42728 Nov  2 00:48 func_sim.log
drwxr-xr-x 2 ex_poly1 rnd    29 Oct 14 00:45 log
-rwxr-xr-x 1 ex_poly1 rnd   662 Oct 15 23:44 run_function.tcl
-rw-r--r-- 1 ex_poly1 rnd  7026 Oct 14 21:53 shm.prof
-rw-r--r-- 1 ex_poly1 rnd 23608 Nov  2 00:48 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd    448 Oct 21 23:31 xrun.history
-rw-r--r-- 1 ex_poly1 rnd    231 Nov  2 00:48 xrun.key
-rw-r--r-- 1 ex_poly1 rnd   331 Oct 21 23:31 xrun.log
```

# 디지털 라이브러리 테스트

## README 파일 확인

1. \$> vi README ↵

- 시뮬레이션이 이런 식으로 진행되는 것을 알 수 있음

```
with option -gui  
simvision auto started  
dump folder is auto opened  
press start button and stop at $stop function  
  
Lab3 is for high speed test  
1ns test, so toggle at each 0.5ns
```

# 디지털 라이브러리 테스트

## TB\_std\_cell.v 구조

1. \$> vi TB\_std\_cell.v



- **always begin**: CK신호 0.5ns마다 반전 주기는 1ns
- 나머진 LAB1과 동일

반복적으로 일어나는  
것에 대한 기술

```
62 // check the maximum speed
63 always begin
64 #0.5      CK      <= ~CK;
65 end
66
67 always begin
68 #2        IN      <= IN+1'b1;
69 end
```

# LAB3

## 시뮬레이션

# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- simvision 명령 후 시뮬레이션 결과 확인 함

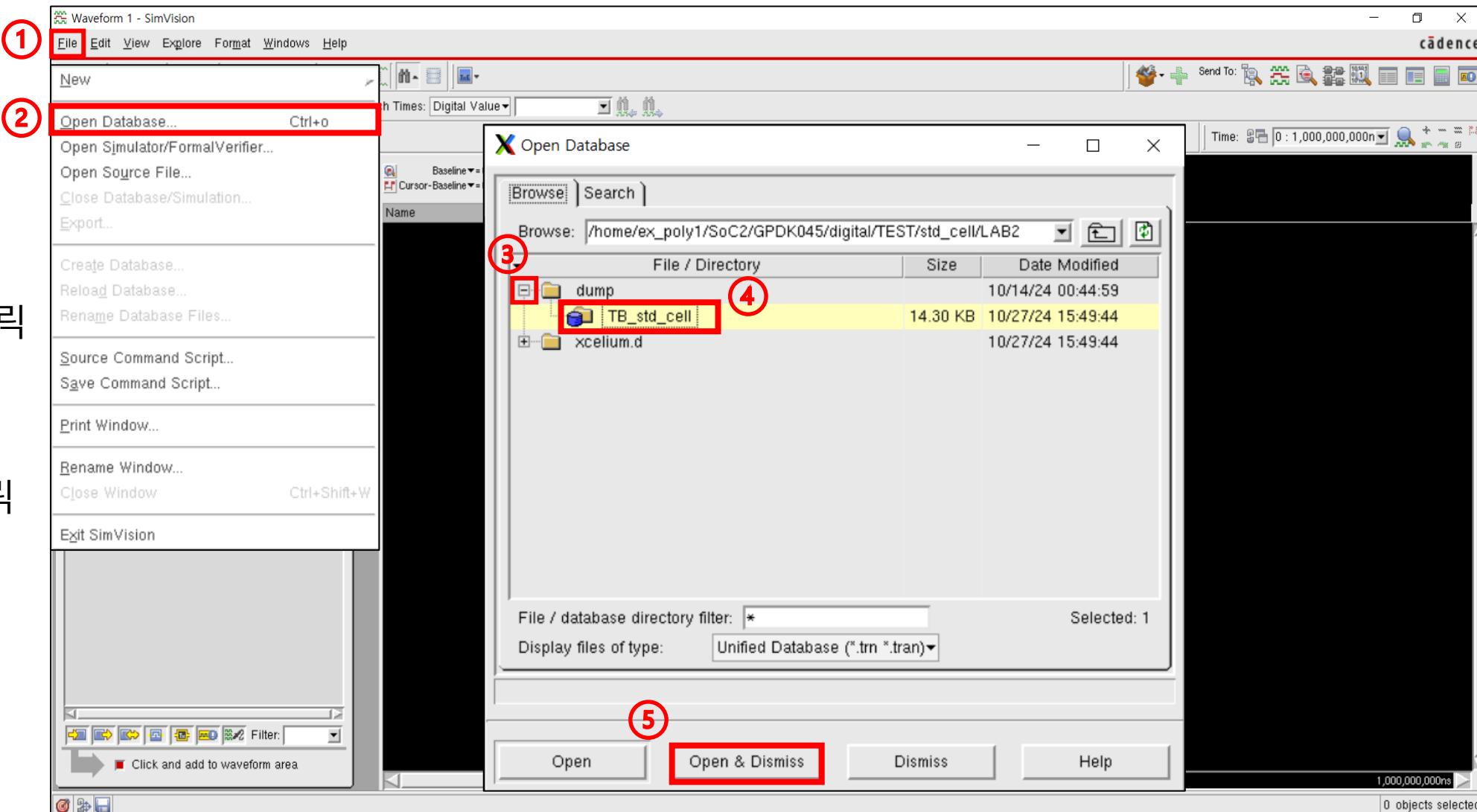
```
[ex_poly1@npit LAB2]$ simvision
simvision(64): 24.03-s005: (c) Copyright 1995-2024 Cadence Design Systems, Inc.
txe(64): 24.03-s005: (c) Copyright 1995-2024 Cadence Design Systems, Inc.
```

# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- ① File 클릭
- ② Open Database 클릭
- ③ dump 폴더 오픈
- ④ TB\_std\_cell 클릭
- ⑤ Open&Dismiss 클릭

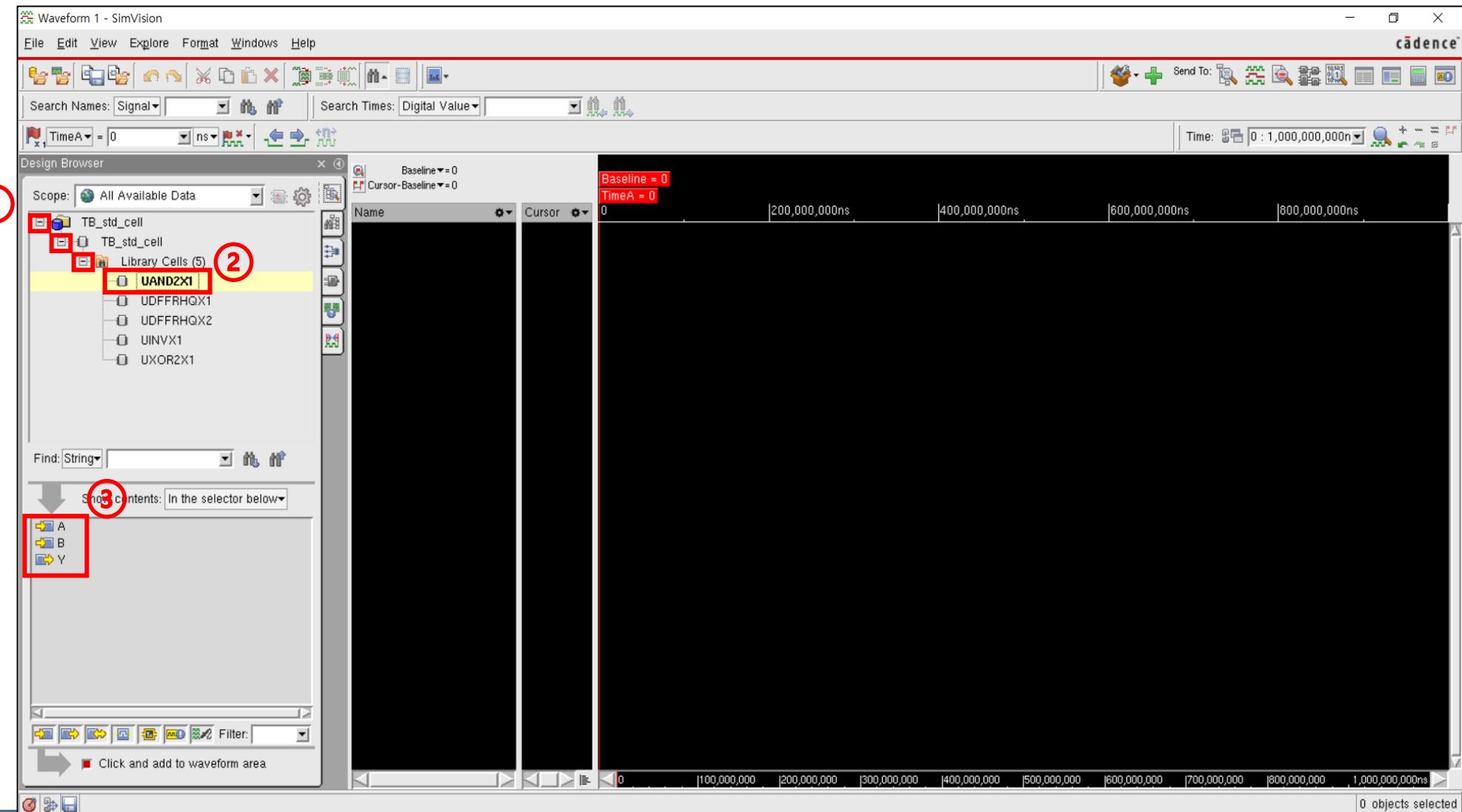


# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- ① Library Cell 오픈
- ② UAND2X1 클릭
- ③ A, B, Y 순서로 클릭

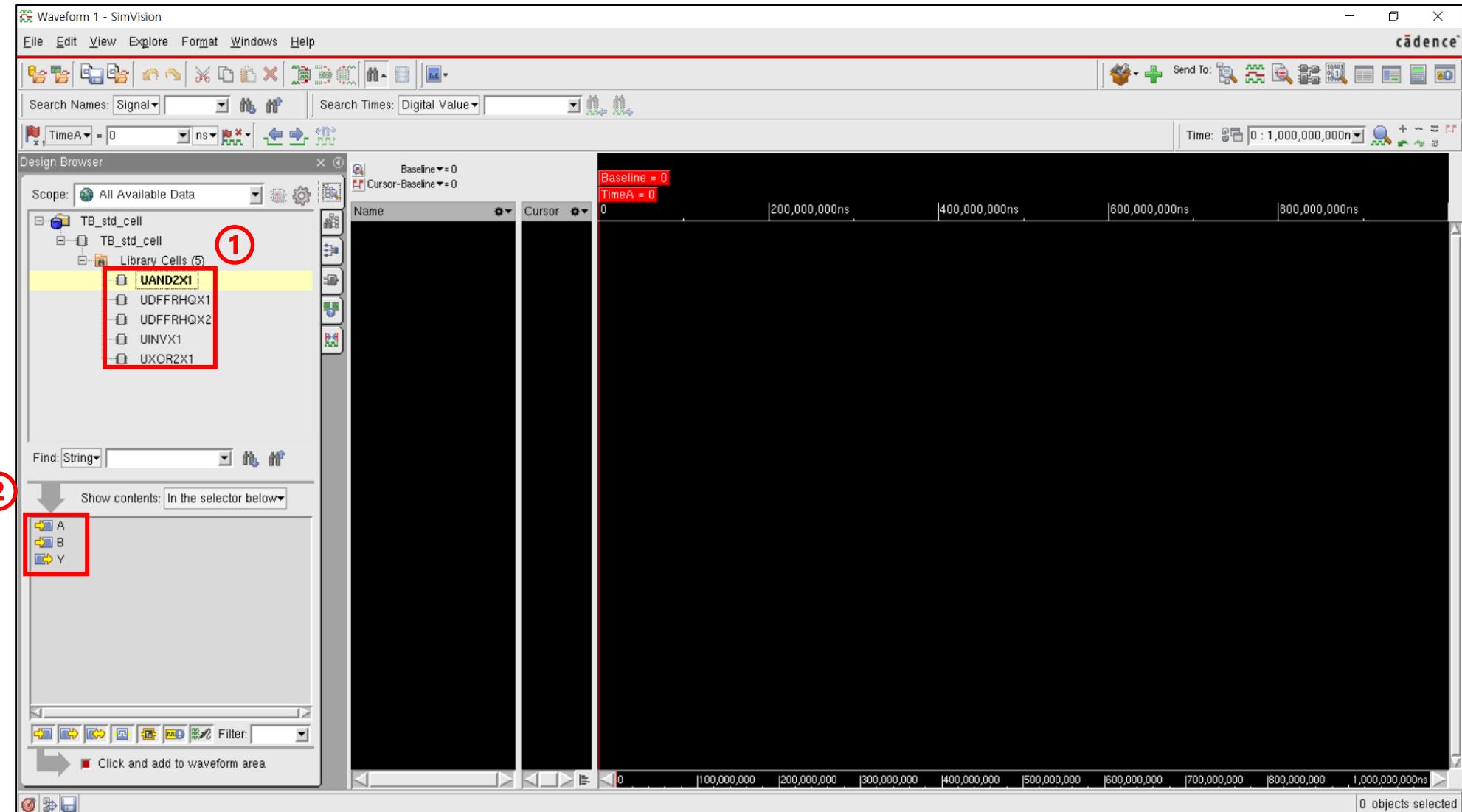


# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- 모든 파형 입력

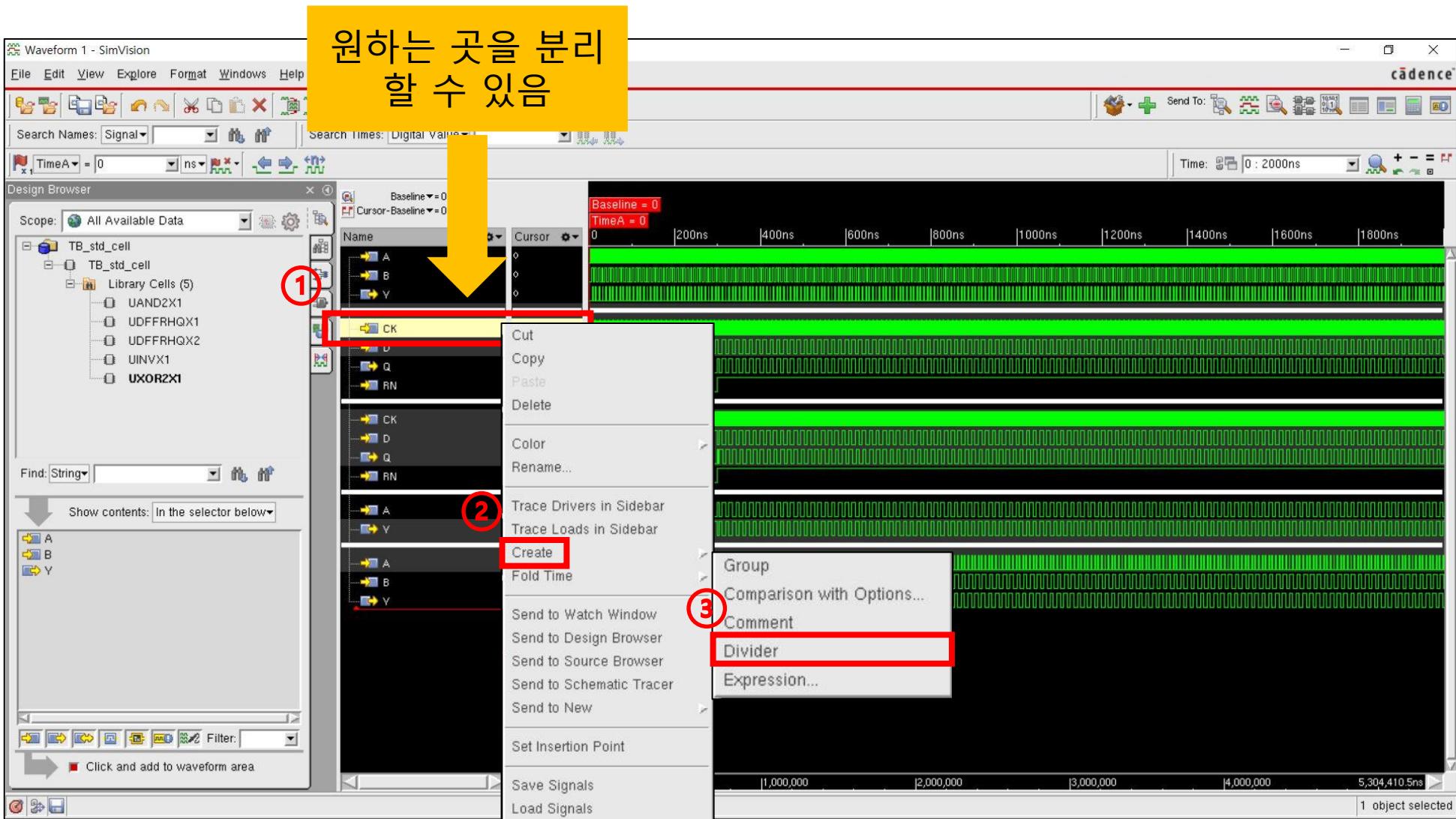


# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- ① CK 클릭 후 우클릭
- ② Create 클릭
- ③ Divider 클릭

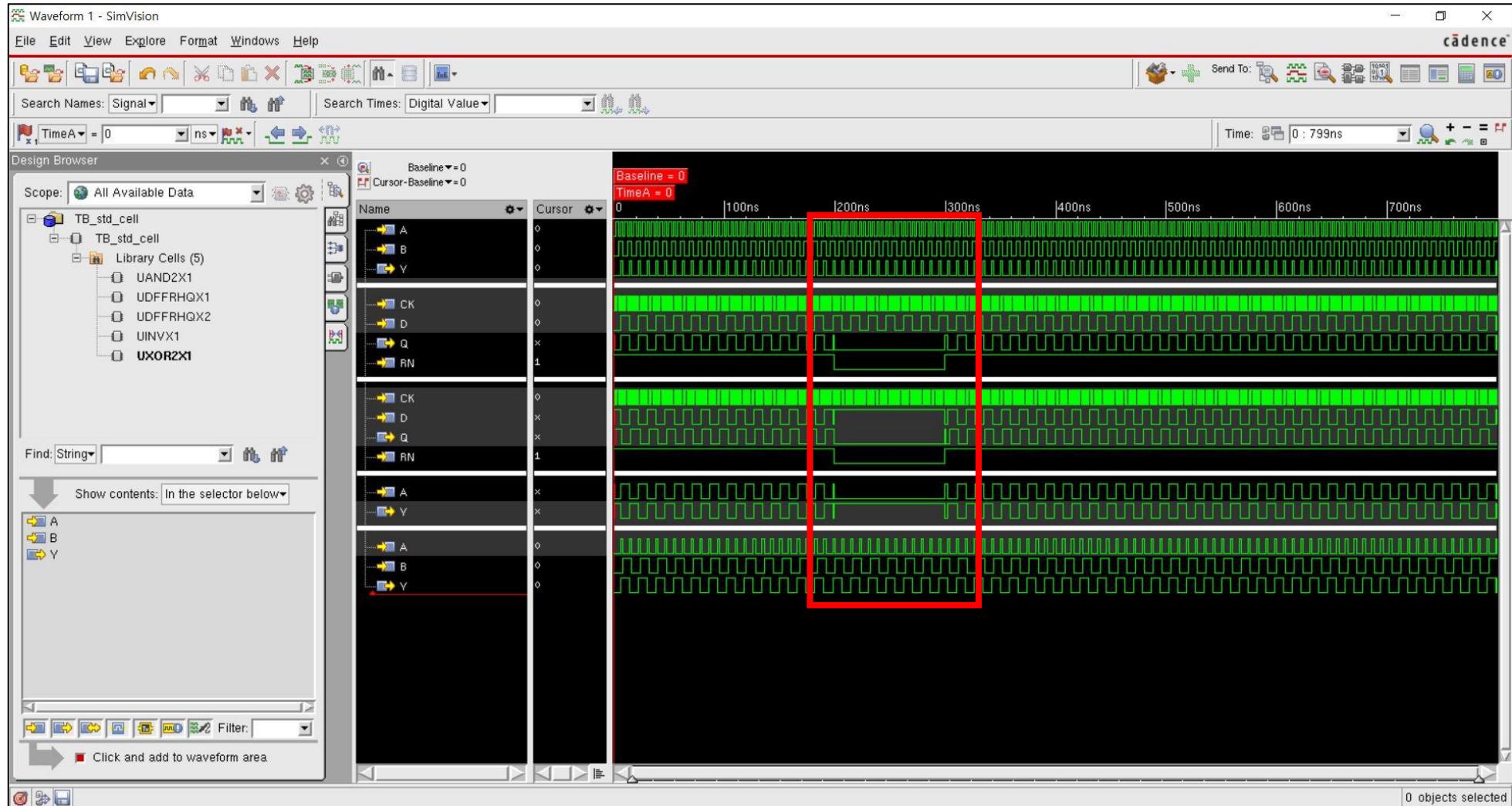


# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- 파형 드래그 하여 실행 결과 확인



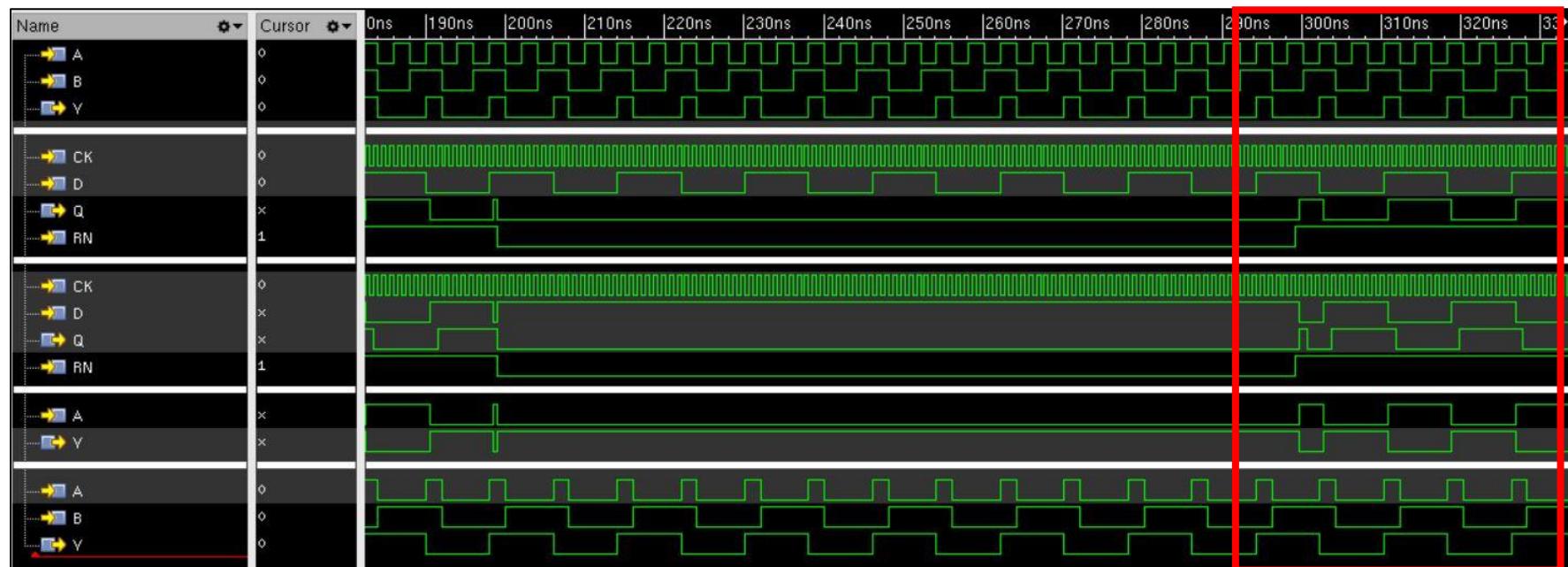
# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- 리셋 신호에 맞추어 출력 신호  
가 잘 반응 함
- 셀이  $f = 1/T$  이므로 최대  
1GHz에서도 잘 동작함

확대해서 확인



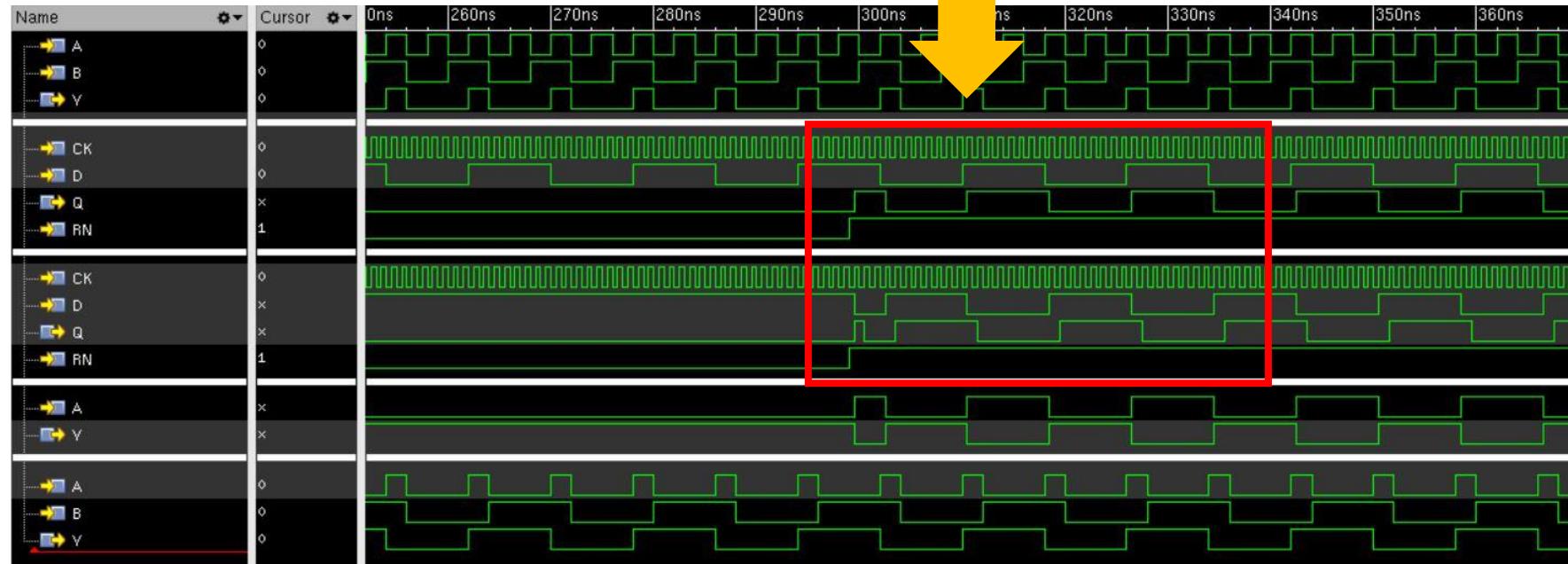
# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- D-FF의 리셋 직후의 동작을 살펴보아야 함
- 셀이 리셋 직후에도 CK신호와 D신호에 맞춰 출력이 잘 나옴
- 빨간 상자 확대해서 한번 더 확인

RN신호가 0 일 때 동작을 안하고  
RN에 1신호가 들어오면 다시 동작



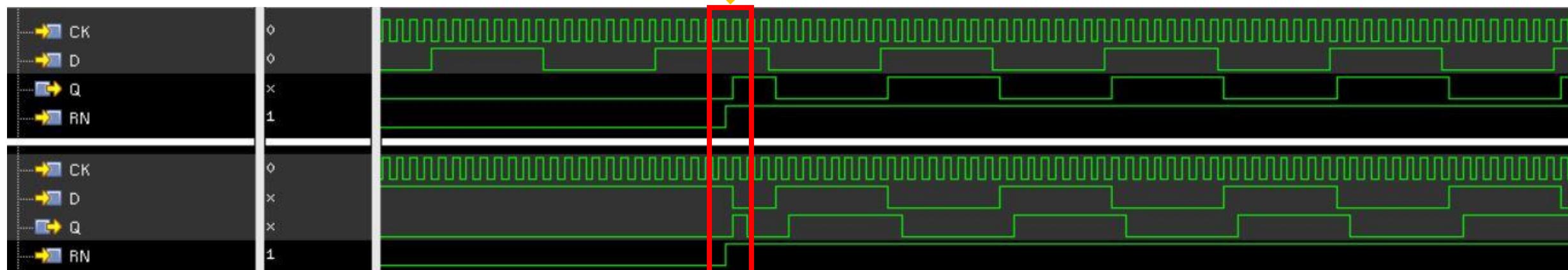
# 디지털 라이브러리 테스트

## LAB3 시뮬레이션

### 1. \$> simvision

- 동작에 문제는 없지만 사용 범위를 알아보기 위해 500MHz, 750MHz에서도 시뮬레이션 후 결과 확인 필요

리셋 직후 CK신호가 상승하는 타이밍에 맞추어 D값 출력



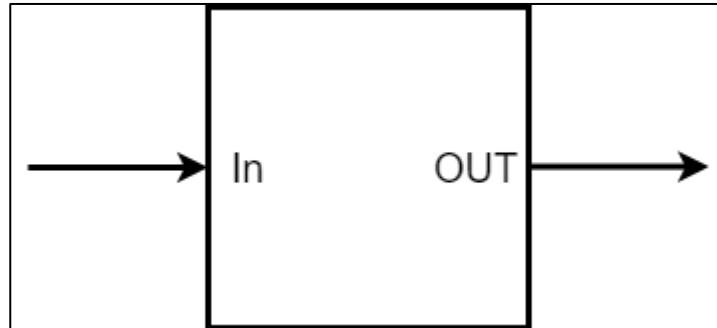
# IO셀 라이브러리 시뮬레이션

# 디지털 라이브러리 테스트

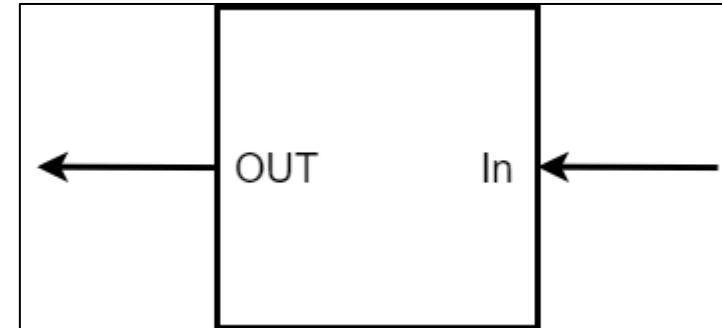
## IO셀 이란?

1. 신호가 들어가기도 나오기도 하는 양방향 셀(Input-Output)
2. ESD회로: 외부 정전기로 인한 회로 손상 방지용으로 사용
3. VDD, VSS를 연결해주는 셀(Core영역으로 Power 인가에 용이)

<하나의 셀에서 IN OUT 둘 다 가능>



<그림4>



<그림5>

# LAB1

# 디지털 라이브러리 테스트

## LAB1의 목적

1. IO셀의 기본적인 동작 확인
2. PADDB, PADDI, PADDO의 구조와 동작 이해

# 디지털 라이브러리 테스트

## LAB1 파일 확인

1. \$>cd LAB1 ↵

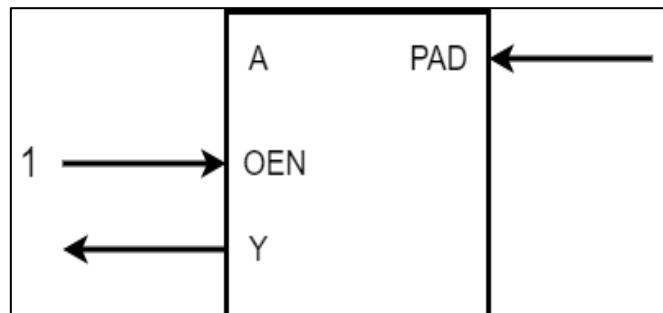
2. \$> ll ↵

```
[ex_poly1@npit LAB1]$ ll
total 32
-rw-r--r-- 1 ex_poly1 rnd 1008 Oct 19 02:53 TB_io_cell.v
-rwxr-xr-x 1 ex_poly1 rnd   34 Oct 19 02:05 clean.tcl
-rw-r--r-- 1 ex_poly1 rnd  272 Nov  2 17:08 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd 3421 Nov  2 17:32 func_sim.log
-rwxr-xr-x 1 ex_poly1 rnd  615 Oct 15 23:57 run_function.tcl
-rw-r--r-- 1 ex_poly1 rnd 4634 Nov  2 17:32 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd  388 Nov  2 17:32 xrun.key
```

# 디지털 라이브러리 테스트

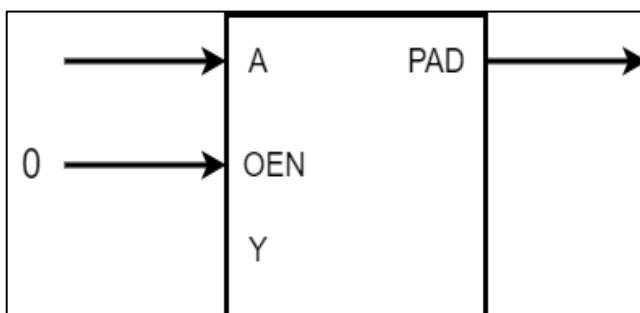
## TB\_io\_cell.v 구조

① PADDB\_IN



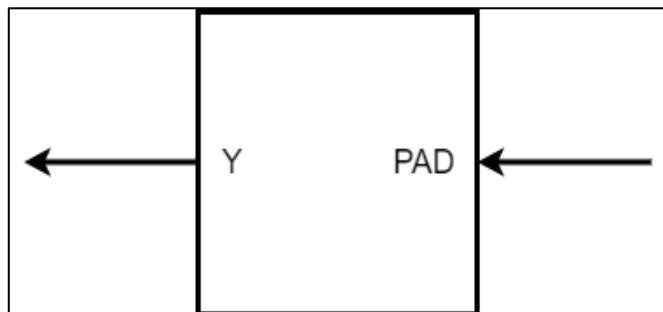
<그림3>

② PADDB\_OUT



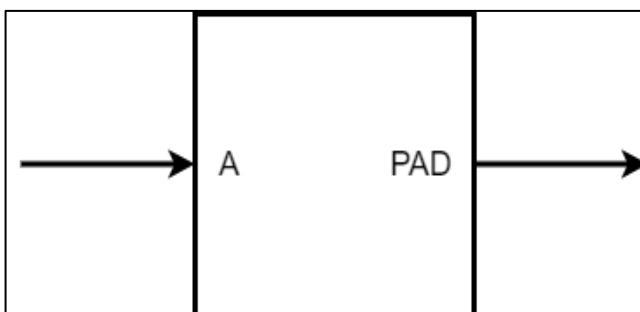
<그림6>

③ PADDI



<그림7>

④ PADDO



<그림8>

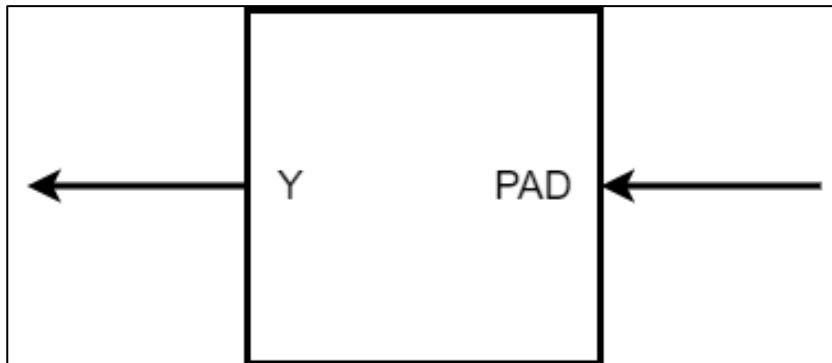
```
20 // Bi-directional Cell
21 // PAD to Y
22 PADDB UPADDB_IN(
23     .A(),
24     .OEN(OEN_IN),
25     .PAD(PAD_B_IN),
26     .Y(Y_B_IN)
27 );
28
29
30 // Bi-directional Cell
31 // A to PAD
32 PADDB UPADDB_OUT(
33     .A(A_B_OUT),
34     .OEN(OEN_OUT),
35     .PAD(PAD_B_OUT),
36     .Y()
37 );
38
39 // Input Buffer
40 PADDI UPADDI(
41     .PAD(PAD_IN),
42     .Y(Y_IN)
43 );
44
45 // Output Driver
46 PADDO UPADDO(
47     .A(A_OUT),
48     .PAD(PAD_OUT)
49 );
50
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

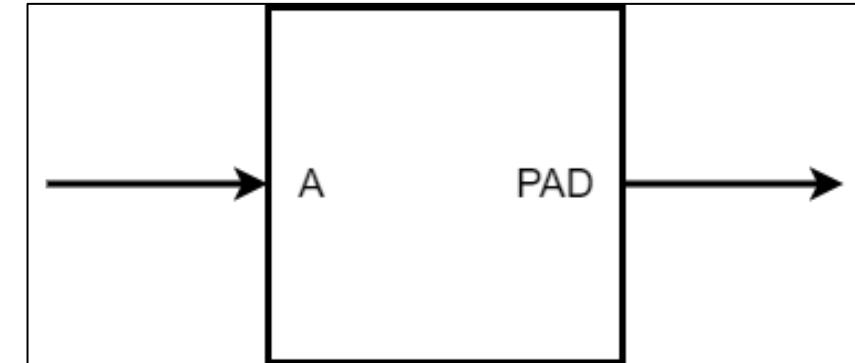
### 1. 한 방향 출력

1. PADDI



<그림7>

2. PADDO



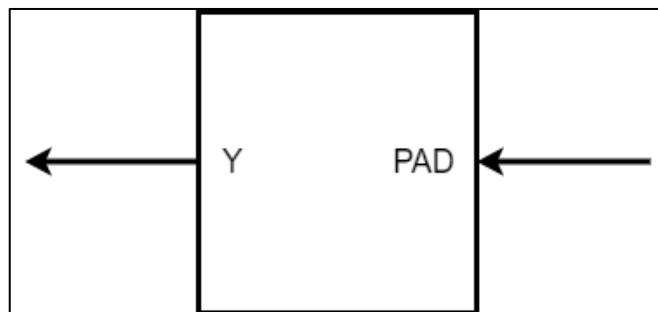
<그림8>

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

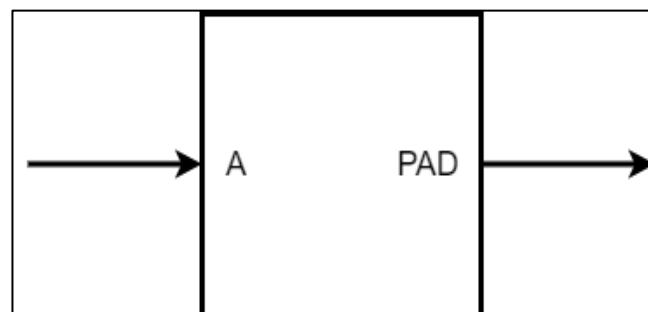
1. PADDI는 PAD가 입력하고 Y가 출력되는 방향으로만 사용 가능
2. PADDO는 A가 입력하고 PAD가 출력되는 방향으로만 사용 가능

1. PADDI



<그림7>

2. PADDO



<그림8>

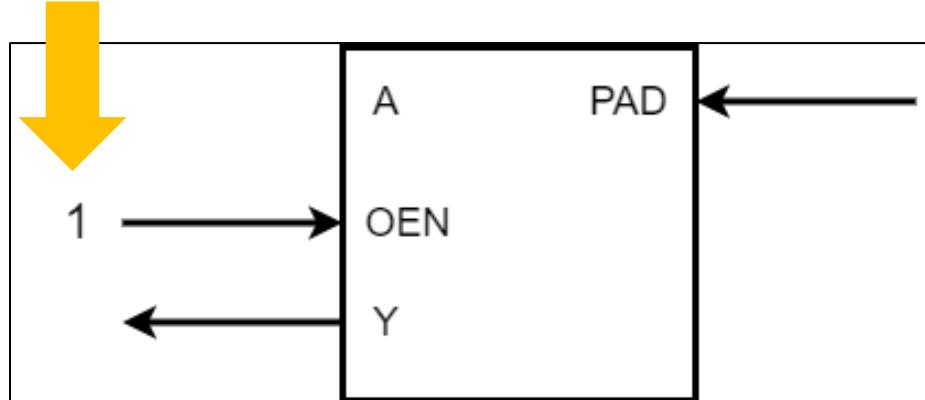
```
20 // Bi-directional Cell
21 // PAD to Y
22 PADDB UPADDB_IN(
23     .A(),
24     .OEN(OEN_IN),
25     .PAD(PAD_B_IN),
26     .Y(Y_B_IN)
27 );
28
29
30 // Bi-directional Cell
31 // A to PAD
32 PADDB UPADDB_OUT(
33     .A(A_B_OUT),
34     .OEN(OEN_OUT),
35     .PAD(PAD_B_OUT),
36     .Y()
37 );
38
39 // Input Buffer
40 PADDI UPADDI(
41     .PAD(PAD_IN),
42     .Y(Y_IN)
43 );
44
45 // Output Driver
46 PADDO UPADDO(
47     .A(A_OUT),
48     .PAD(PAD_OUT)
49 );
50
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

- 양방향 출력
- OEN 뜻은 Output Enable Negative이며 OEN값을 0 또는 1로 조절하면 하나의 셀로 입출력을 모두 처리할 수 있음

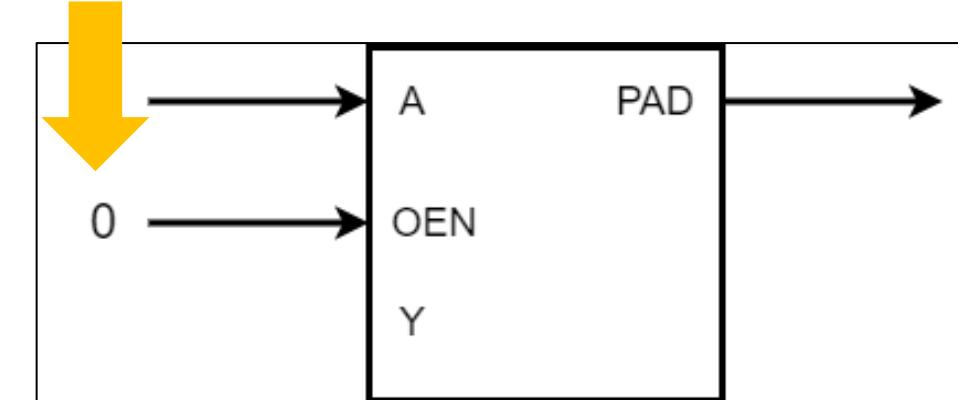
PAD에 입력을 하려면  
OEN 값이 1이어야 됨



1. PADDB\_IN

<그림3>

PAD가 출력을 하려면  
OEN 값이 0이어야 됨



2. PADDB\_OUT

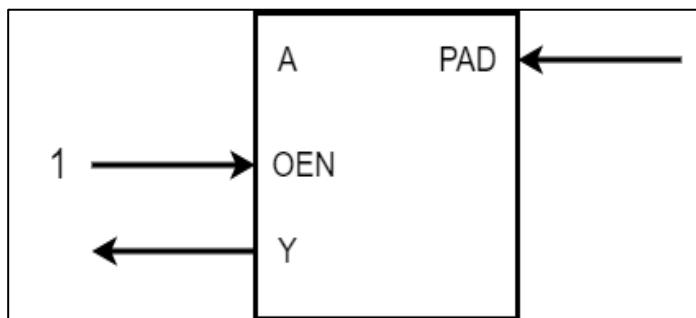
<그림6>

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

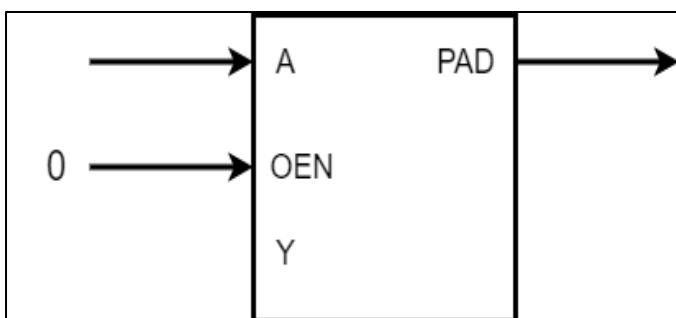
1. PADDB\_IN은 PAD로 입력 받고 Y로 출력함
2. PADDB\_OUT은 A로 입력 받고 PAD로 출력함

1. PADDB\_IN



<그림3>

2. PADDB\_OUT



<그림6>

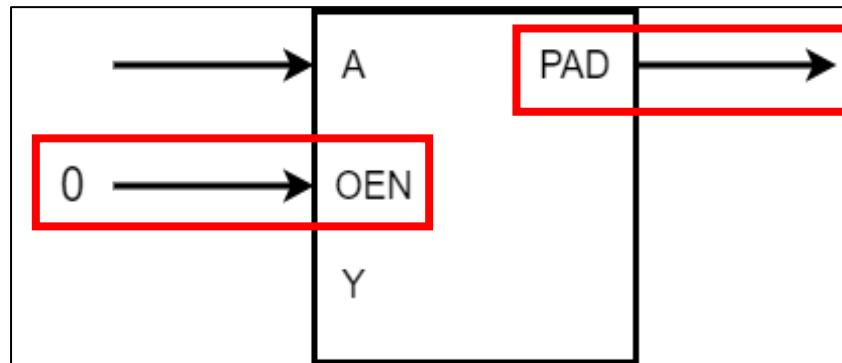
```
20 // Bi-directional Cell
21 // PAD to Y
22 PADDB_UPADDB_IN(
23     .A(),
24     .OEN(OEN_IN),
25     .PAD(PAD_B_IN),
26     .Y(Y_B_IN)
27 );
28
29
30 // Bi-directional Cell
31 // A to PAD
32 PADDB_UPADDB_OUT(
33     .A(A_B_OUT),
34     .OEN(OEN_OUT),
35     .PAD(PAD_B_OUT),
36     .Y()
37 );
38
39 // Input Buffer
40 PADDI_UPADDI(
41     .PAD(PAD_IN),
42     .Y(Y_IN)
43 );
44
45 // Output Driver
46 PADDO_UPADDO(
47     .A(A_OUT),
48     .PAD(PAD_OUT)
49 );
50
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

1. PAD가 출력을 하므로 OEN 값은 0이어야 함

### 1. PADDB\_OUT



<그림6>

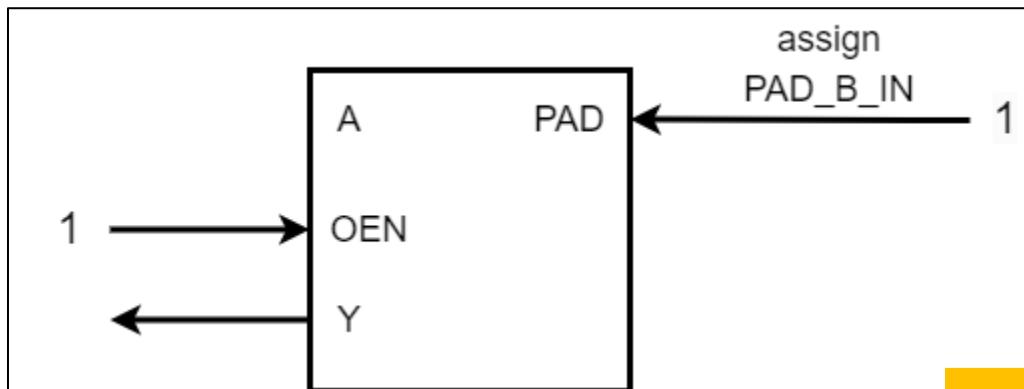
```
53 initial begin
54
55     OEN_IN <= 1'b1;
56
57     A_B_OUT<=1'b1;
58     OEN_OUT<=1'b1;
59     #20
60     OEN_OUT <=1'b0;
61     #20
62     A_B_OUT<=1'b0;
63
64     PAD_IN<= 1'b0;
65     #20
66     PAD_IN <=1'b1;
67
68     A_OUT<= 1'b0;
69     #20
70     A_OUT<= 1'b1;
71
72     #100
73     $stop;
74 end
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

1. PAD에 입력을 하므로 OEN 값이 1이어야 함

### 1. PADDB\_IN



<그림9>

assign으로 1 입력

PAD\_B\_IN을 reg로 했을 시  
오류 발생 wire로 수정

```
3
4
5 reg OEN_IN;
6 //reg PAD_B_IN;
7 wire PAD_B_IN;
8 wire Y_B_IN;
9
```

```
21 // PAD to Y
22 PADDB_UPADDB_IN(
23     .A(),
24     .OEN(OEN_IN),
25     .PAD(PAD_B_IN),
26     .Y(Y_B_IN)
27 );
```

```
51 assign PAD_B_IN = 1'b1;
```

# LAB1

# 시뮬레이션

# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

1. ./clean.tcl ↵

2. ./run\_function.tcl ↵

```
[ex_poly1@npit LAB1]$ ./clean.tcl  
[ex_poly1@npit LAB1]$ ./run_function.tcl
```

- xrun 실행

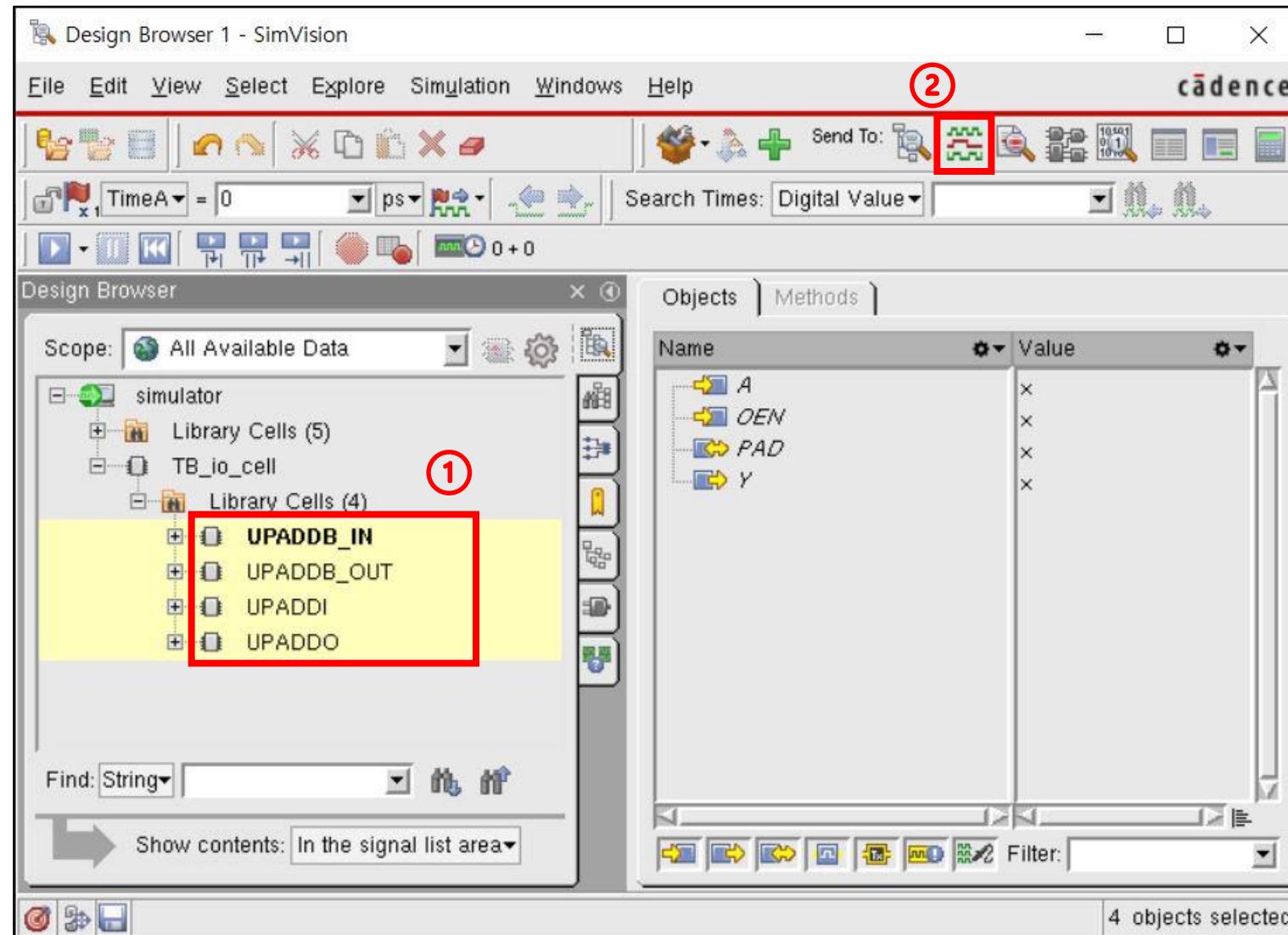
```
xrun -64bit \  
+max_err_count+50 \  
+define+function_sim \  
-access +rwc \  
-profile \  
-profthread \  
-gui \  
TB_io_cell.v \  
../../../../giolib045_v3.5/vlog/pads_FF_s1vg.v \  
-l ./func_sim.log
```

# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- ① 모두 클릭
- ② 웨이브 아이콘 클릭

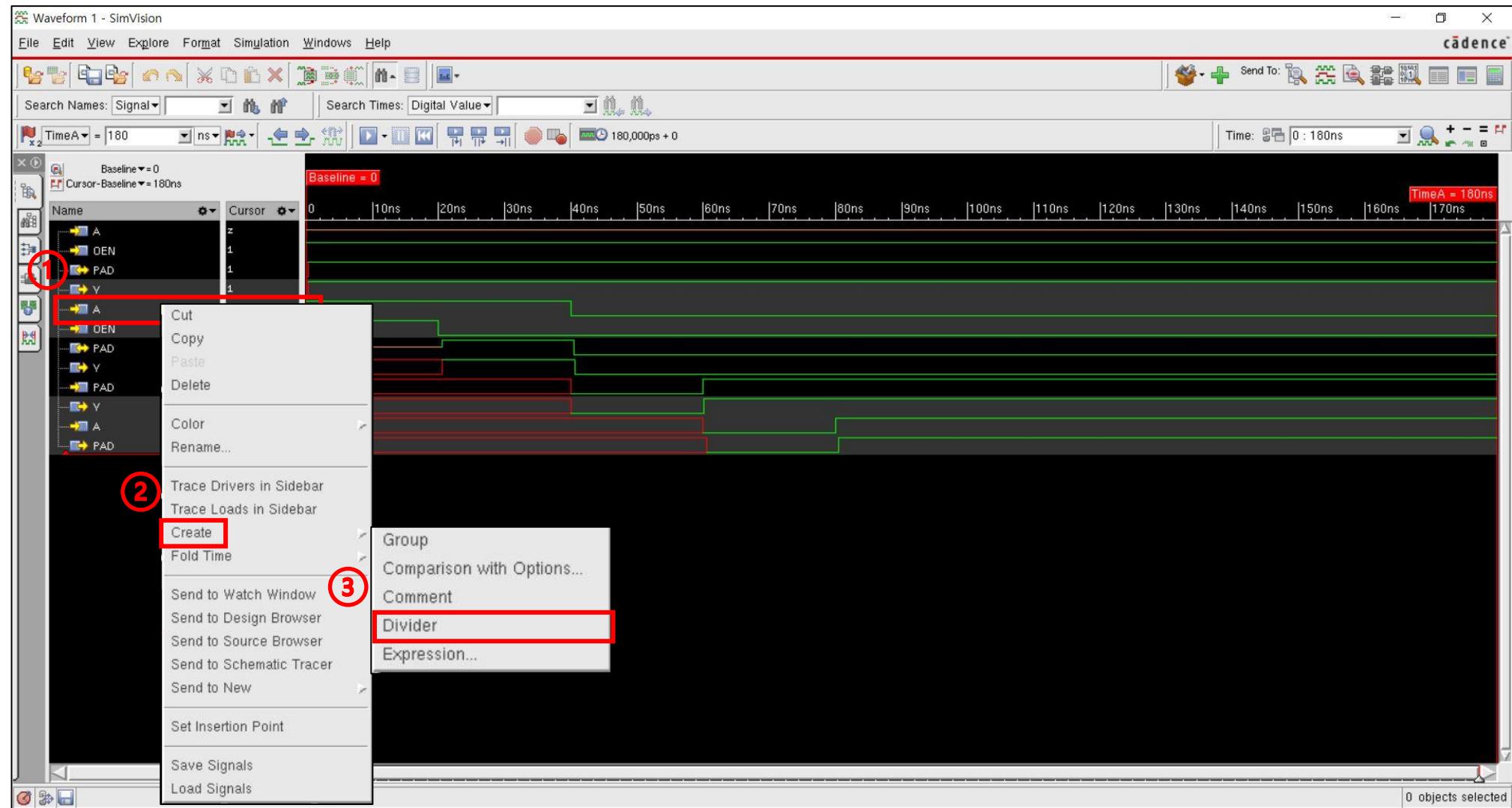


# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- ① A 클릭 후 우클릭
- ② Create 클릭
- ③ Divider 클릭

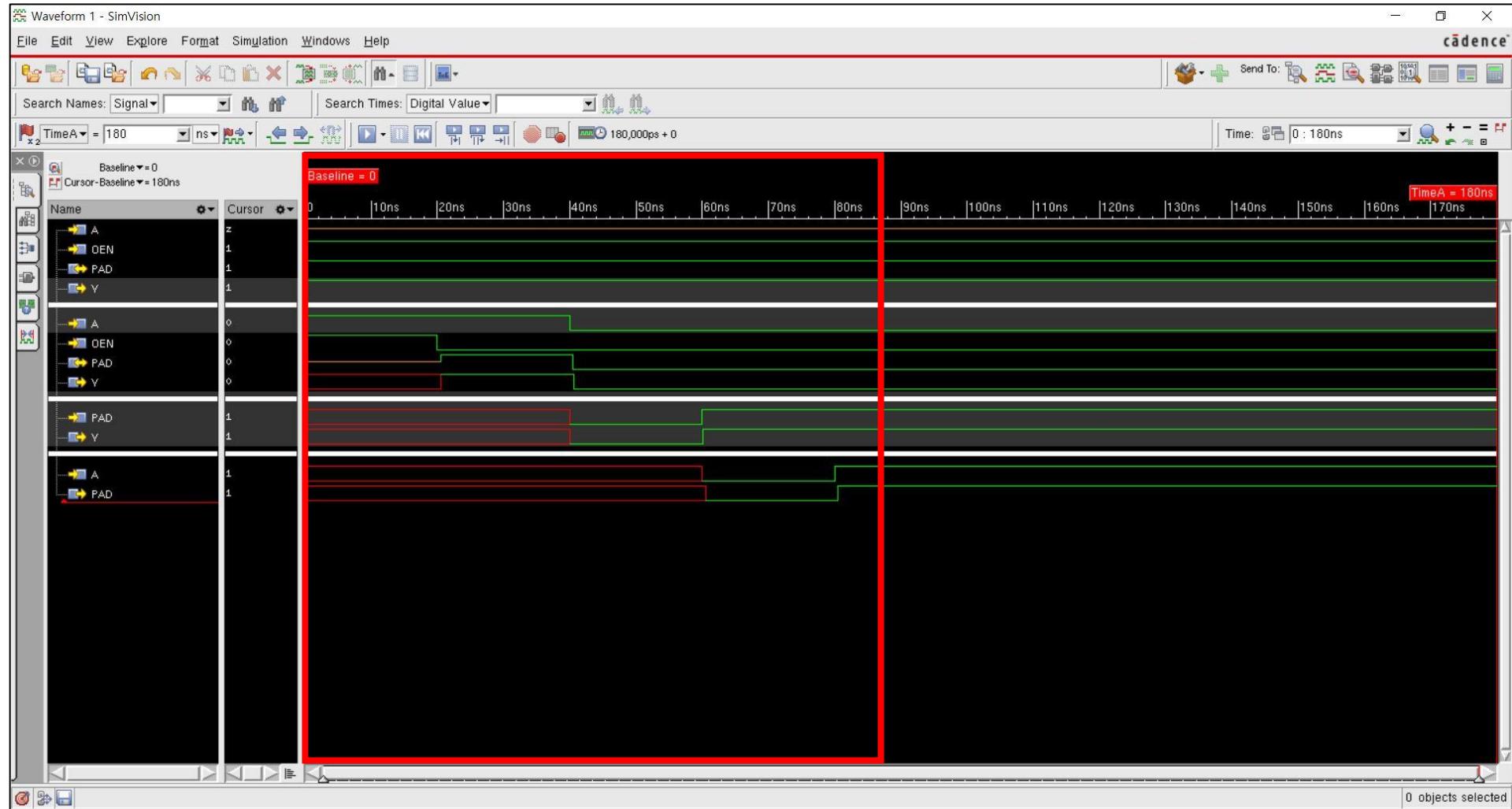


# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- 드래그 결과 확인



# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- 시뮬레이션 동작확인



# 디지털 라이브러리 테스트

## LAB1 시뮬레이션

### 1. xrun 실행 결과

- 시뮬레이션 동작확인
- 버퍼동작 문제없음

```
32 PADDB UPADDB_OUT(  
33     .A(A_B_OUT),  
34     .OEN(OEN_OUT),  
35     .PAD(PAD_B_OUT),  
36     .Y());  
37 );
```

```
53 initial begin  
54     OEN_IN <= 1'b1;  
55  
56     A_B_OUT<=1'b1;  
57     OEN_OUT<=1'b1;  
58     #20  
59     OEN_OUT <=1'b0;  
60     #20  
61     A_B_OUT<=1'b0;  
62  
63     PAD_IN<= 1'b0;  
64     #20  
65     PAD_IN <=1'b1;  
66  
67     A_OUT<= 1'b0;  
68     #20  
69     A_OUT<= 1'b1;  
70  
71     #100  
72     $stop;  
73  
74 end
```

OEN의 초기값을 1로 시작 후  
0으로 반전시켜 정상적으로  
동작하는지 확인



# LAB2

# 디지털 라이브러리 테스트

## LAB2의 목적

### 1. IO셀의 성능확인

- 최대 동작 스피드와 허용 가능한 신호의 가용범위 알아보기

# 디지털 라이브러리 테스트

## LAB2 파일 확인

1. \$>cd LAB2 ↵

2. \$> ll ↵

```
[ex_poly1@npit LAB2]$ ll
total 32
-rw-r--r-- 1 ex_poly1 rnd 1135 Oct 20 23:34 TB_io_cell.v
-rwxr-xr-x 1 ex_poly1 rnd    34 Oct 19 02:05 clean.tcl
-rw-r--r-- 1 ex_poly1 rnd   272 Nov  2 22:32 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd 3237 Nov  3 00:03 func_sim.log
-rwxr-xr-x 1 ex_poly1 rnd   615 Oct 15 23:57 run_function.tcl
-rw-r--r-- 1 ex_poly1 rnd 4812 Nov  3 00:03 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd   354 Nov  3 00:03 xrun.key
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

1. 8행에서 PAD\_B\_IN\_reg를 reg로 선언
2. 83행에서 0.5ns마다 반전을 해주기 위해 wire인 PAD\_B\_IN에 assign 명령으로 PAD\_B\_IN\_reg값을 입력해 줌

0.5ns 반전, 주기는 1ns

```
1 `timescale 1ns/1ps
2 module TB_io_cell(
3 );
4
5 reg OEN_IN;
6 //reg PAD_B_IN;
7 wire PAD_B_IN;
8 reg PAD_B_IN_reg; 초기값은 0으로 지정
9 wire Y_B_IN;
```

```
55 initial begin
56     PAD_B_IN_reg <= 1'b0;
57     OEN_IN <= 1'b1;
```

```
79 always begin
80     #0.5 PAD_B_IN_reg <= ~PAD_B_IN_reg;
81 end
82
83 assign PAD_B_IN = PAD_B_IN_reg;
```

# 디지털 라이브러리 테스트

## TB\_io\_cell.v 구조

1. \$> vi TB\_io\_cell.v -d ..../LAB1/TB\_io\_cell.v

- Tip: LAB1 TB파일과 LAB2 TB파일의 차이를 한눈에 보여줌

The screenshot shows a terminal window at the bottom with the command:

```
[ex_poly1@npit LAB2]$ vi TB_io_cell.v -d ..../LAB1/TB_io_cell.v
```

Two code editors are displayed above the terminal. The left editor is labeled "TB\_io\_cell.v" and the right is "B1/TB\_io\_cell.v". Both files contain Verilog code for a test bench. The right file has several lines highlighted in pink, while the left file has them in blue. A yellow callout box with the text "-d: Differential의 약자로 어디가 다른 지 보여줌" is pointing to the "-d" option in the terminal command.

```
File Edit View Search Terminal Help
50      .A(A_OUT),
51      .PAD(PAD_OUT)
52 );
53
54 initial begin
55     PAD_B_IN_reg <= 1'b0;
56     OEN_IN <= 1'b1;
57
58     A_B_OUT<=1'b1;
59     OEN_OUT<=1'b1;
60
61     #20
62     OEN_OUT <=1'b0;
+ 63 +-- 9 lines: #20-
63     A_OUT<= 1'b1;
64
65     #100
66     $stop;
67 end
68
69 always begin
70     #0.5 PAD_B_IN_reg <= ~PAD_B_IN_reg;
71 end
72
73 assign PAD_B_IN = PAD_B_IN_reg;
```

```
47      .A(A_OUT),
48      .PAD(PAD_OUT)
49 );
50
51 assign PAD_B_IN = 1'b1;
52 initial begin
53     OEN_IN <= 1'b1;
54
55     A_B_OUT<=1'b1;
56     OEN_OUT<=1'b1;
57
58     #20
59     OEN_OUT <=1'b0;
```

```
[ex_poly1@npit LAB2]$ vi TB_io_cell.v -d ..../LAB1/TB_io_cell.v
```

# LAB2

# 시뮬레이션

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

1. ./clean.tcl ↵

```
[ex_poly1@npit LAB1]$ ./clean.tcl  
[ex_poly1@npit LAB1]$ ./run_function.tcl
```

2. ./run\_function.tcl ↵

- xrun 실행

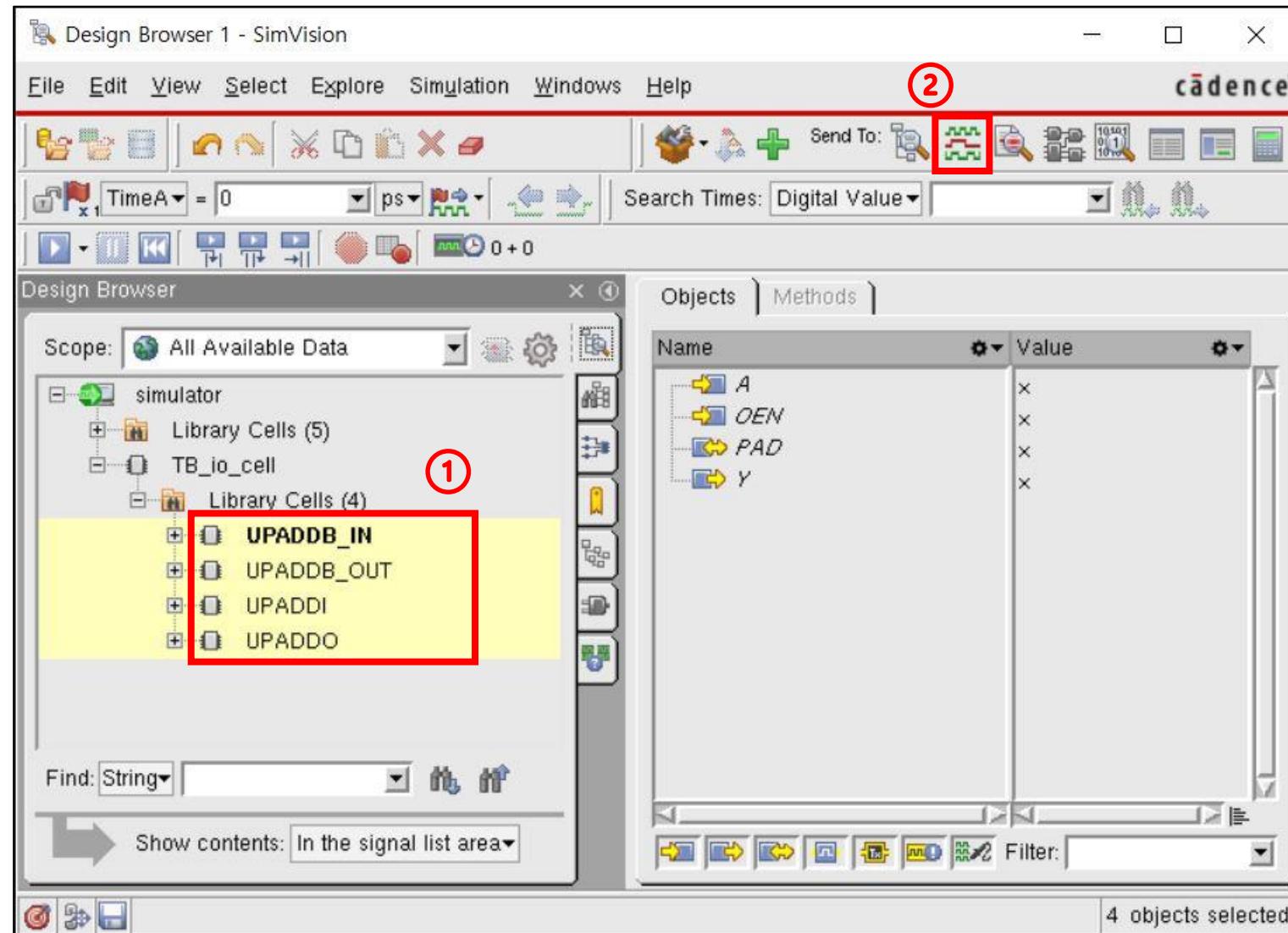
```
xrun -64bit \  
+max_err_count+50 \  
+define+function_sim \  
-access +rwc \  
-profile \  
-profthread \  
-gui \  
TB_io_cell.v \  
../../../../giolib045_v3.5/vlog/pads_FF_s1vg.v \  
-l ./func_sim.log
```

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. xrun 실행 결과

- ① 모두 클릭
- ② 웨이브 아이콘 클릭

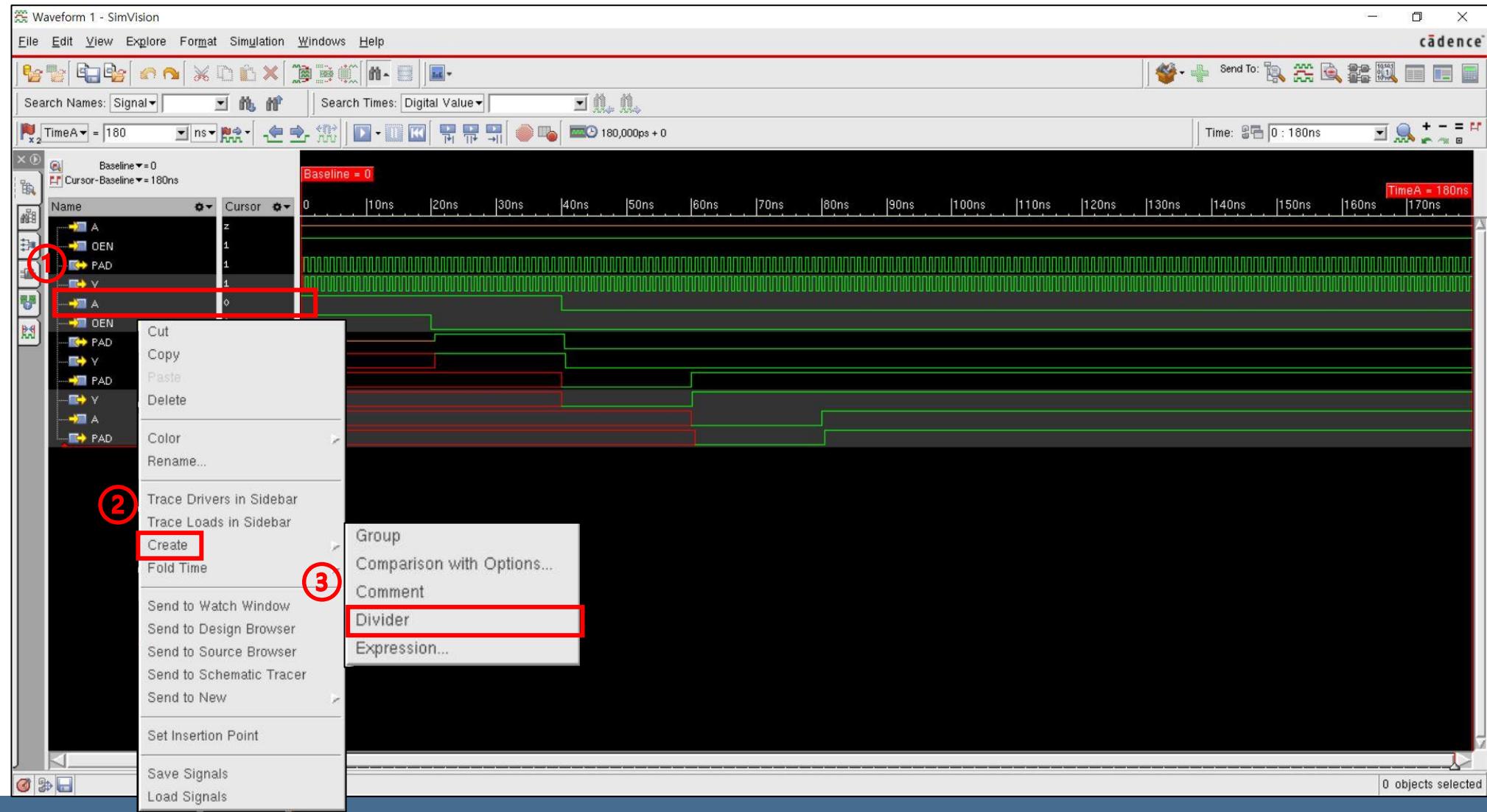


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. xrun 실행 결과

- ①: A 클릭 후 우클릭
- ②: Create 클릭
- ③: Divider 클릭

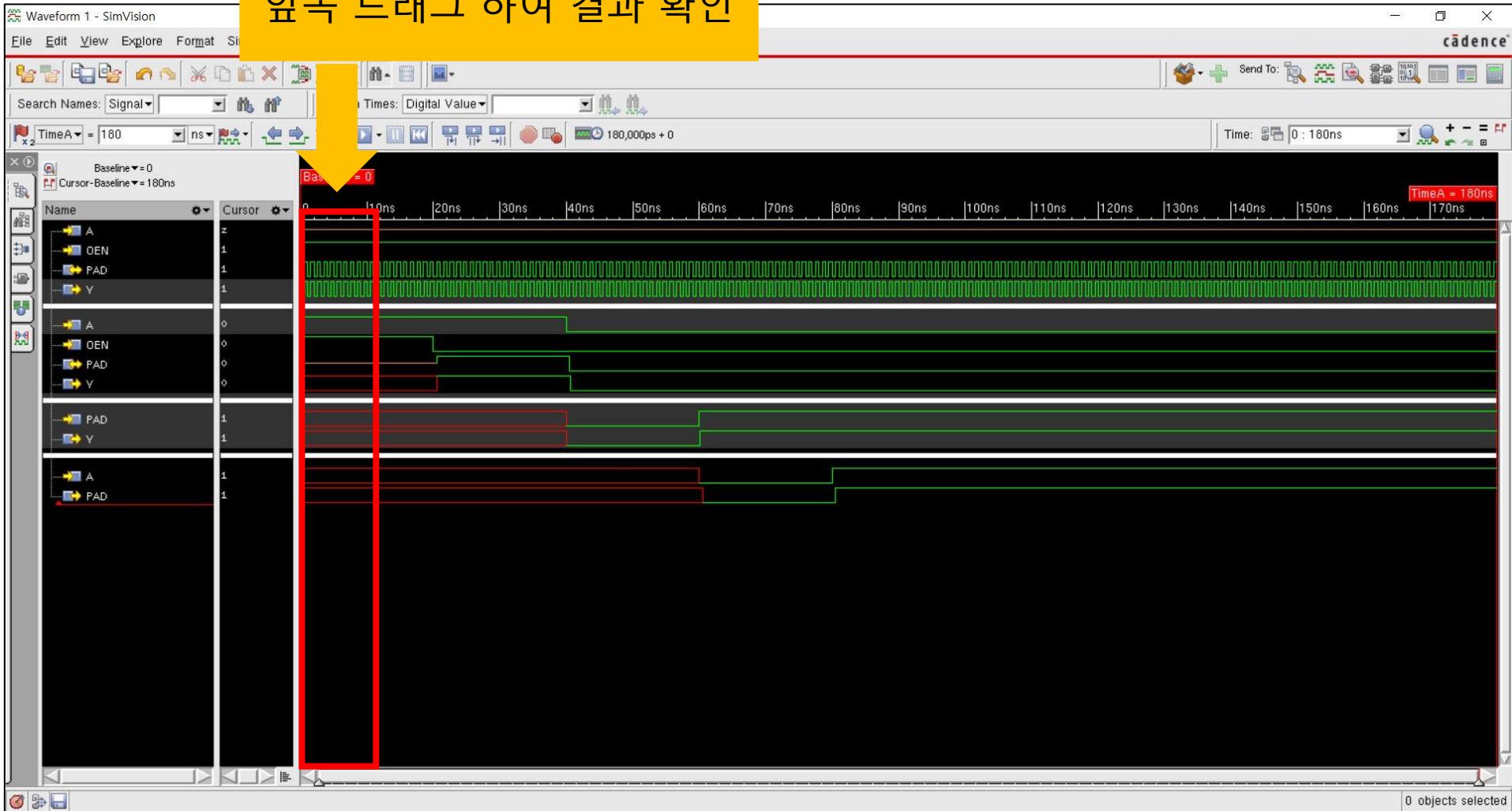


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

앞쪽 드래그 하여 결과 확인

### 1. xrun 실행 결과

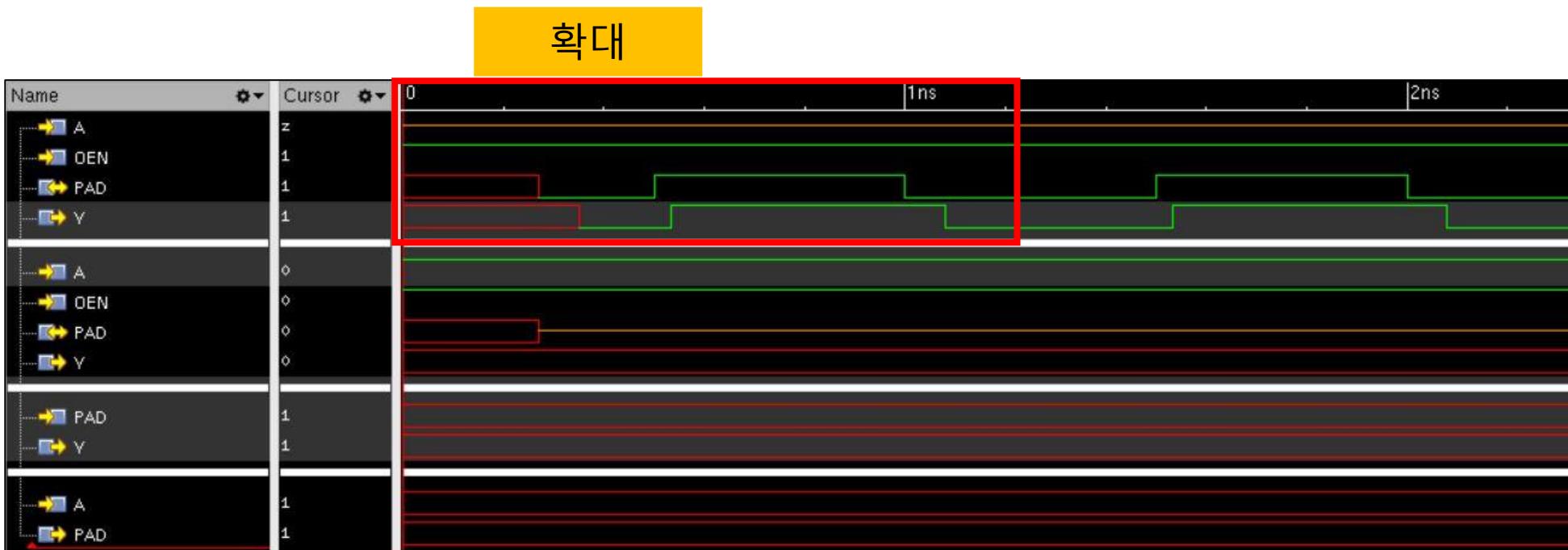


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. xrun 실행 결과

- PADDB\_IN의 동작 확인

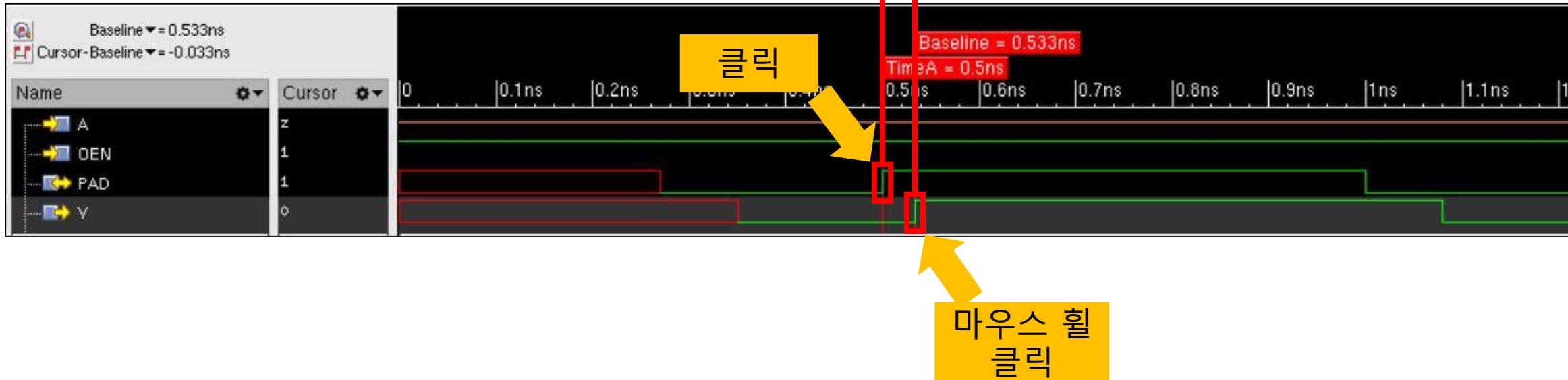


# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_IN 시뮬레이션

- Rising 신호



# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_IN 시뮬레이션

- Falling 신호



# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_IN 시뮬레이션 결과

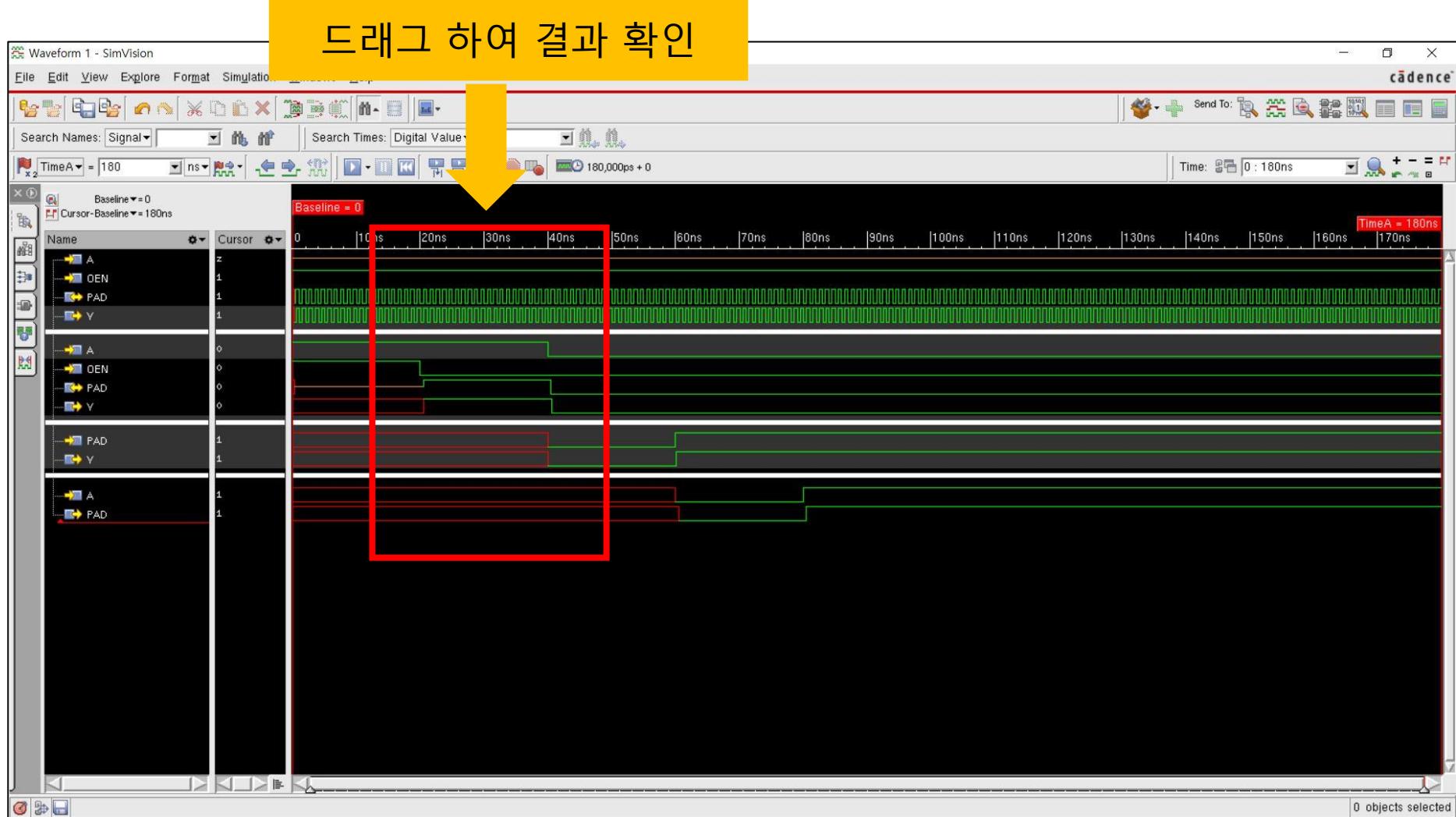
- IO셀의 Cell delay가 크지 않기 때문에 고속 신호(1GHz) 처리에 적합

# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_OUT 실행 결과

- OEN의 반응 시간



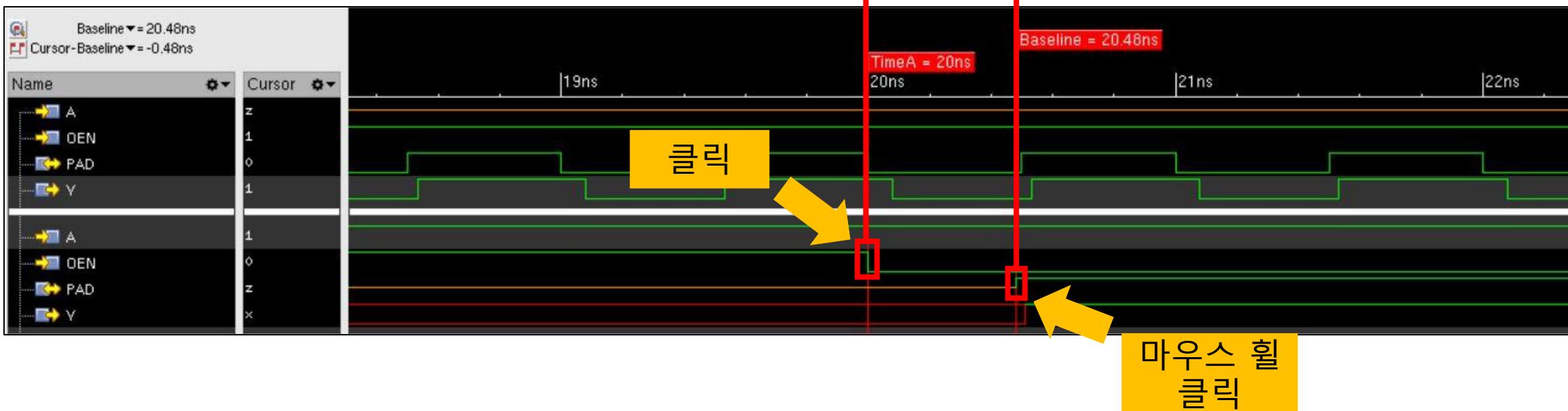
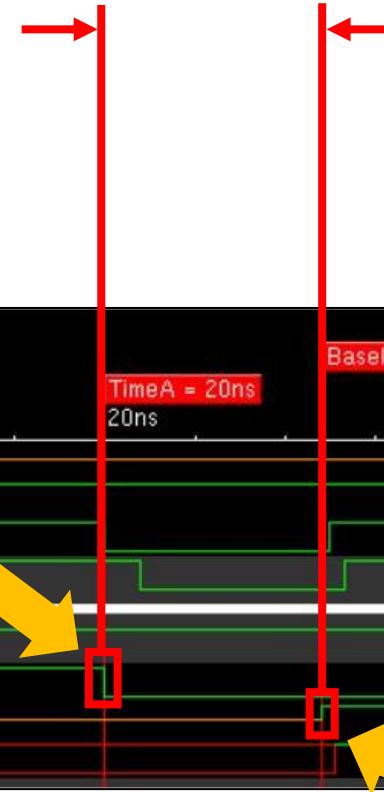
# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_OUT 시뮬레이션

- Rising 신호

Cell delay  
0.48ns



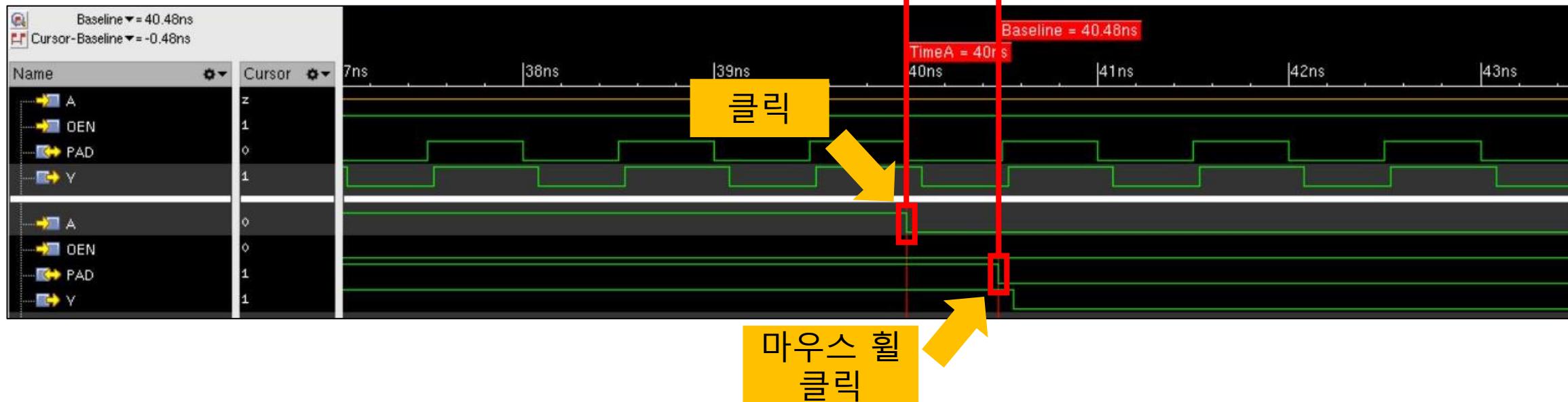
# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_OUT 시뮬레이션

- Falling 신호

Cell delay  
0.48ns



# 디지털 라이브러리 테스트

## LAB2 시뮬레이션

### 1. PADDB\_OUT 시뮬레이션

- IO셀의 Cell delay가 어느 정도 생기기 때문에 고속 신호(1GHz) 처리에 부적합
- 안정적으로 동작되는 가용 범위를 알아보기 위해 500MHz, 750MHz에서도 시뮬레이션 후 결과 확인 필요

# Function Simulation

# 실습 준비

# Function Simulation

## 실습준비

1. cd SoC1 ↴
2. cd SoC ↴
3. cd smkcowl\_make\_s28 ↴
4. cd TOP ↴
5. cd SIM ↴
6. cd FUNCTION ↴

/home/ex\_poly1/SoC2/SoC/smkcowl\_make\_S28/TOP/SIM/FUNCTION

# Function Simulation

## 파일 확인

1. cd SoC

```
[ex_poly1@npit FUNCTION]$ ll
total 180
drwxr-xr-x 2 ex_poly1 rnd 4096 Jan  3 2024 Verdi-SXLog
-rwxr-xr-x 1 ex_poly1 rnd  296 Oct  1 2022 clean.tcl
-rw-r--r-- 1 ex_poly1 rnd  225 Jul 21 2021 fsdb.tcl
-rwxr-xr-x 1 ex_poly1 rnd   49 Apr 15 2022 fsdb2saif.tcl
-rw-r--r-- 1 ex_poly1 rnd 83516 Nov  8 17:24 func_sim.history
-rw-r--r-- 1 ex_poly1 rnd  3616 Oct  2 2022 hello.hex
drwxr-xr-x 2 ex_poly1 rnd 4096 May  7 2020 logs
-rwxr-xr-x 1 ex_poly1 rnd 4422 Nov  4 20:24 run_function.tcl
-rwxr-xr-x 1 ex_poly1 rnd 4336 Oct 22 00:31 run_function.tcl_ori
-rw-r--r-- 1 ex_poly1 rnd 21340 Jul 25 2022 session.inter.vpd.tcl
-rwxr-xr-x 1 ex_poly1 rnd   450 Jan  2 2024 simv_function.tcl
drwxr-xr-x 2 ex_poly1 rnd    52 Nov  6 23:58 waves.shm
drwxr-xr-x 6 ex_poly1 rnd   214 Nov  8 17:24 xcelium.d
-rw-r--r-- 1 ex_poly1 rnd 23624 Nov  8 17:55 xmprof.out
-rw-r--r-- 1 ex_poly1 rnd      5 Nov  8 17:55 xrun.key
```

# FUNCTION SIMULATION

# Function Simulation

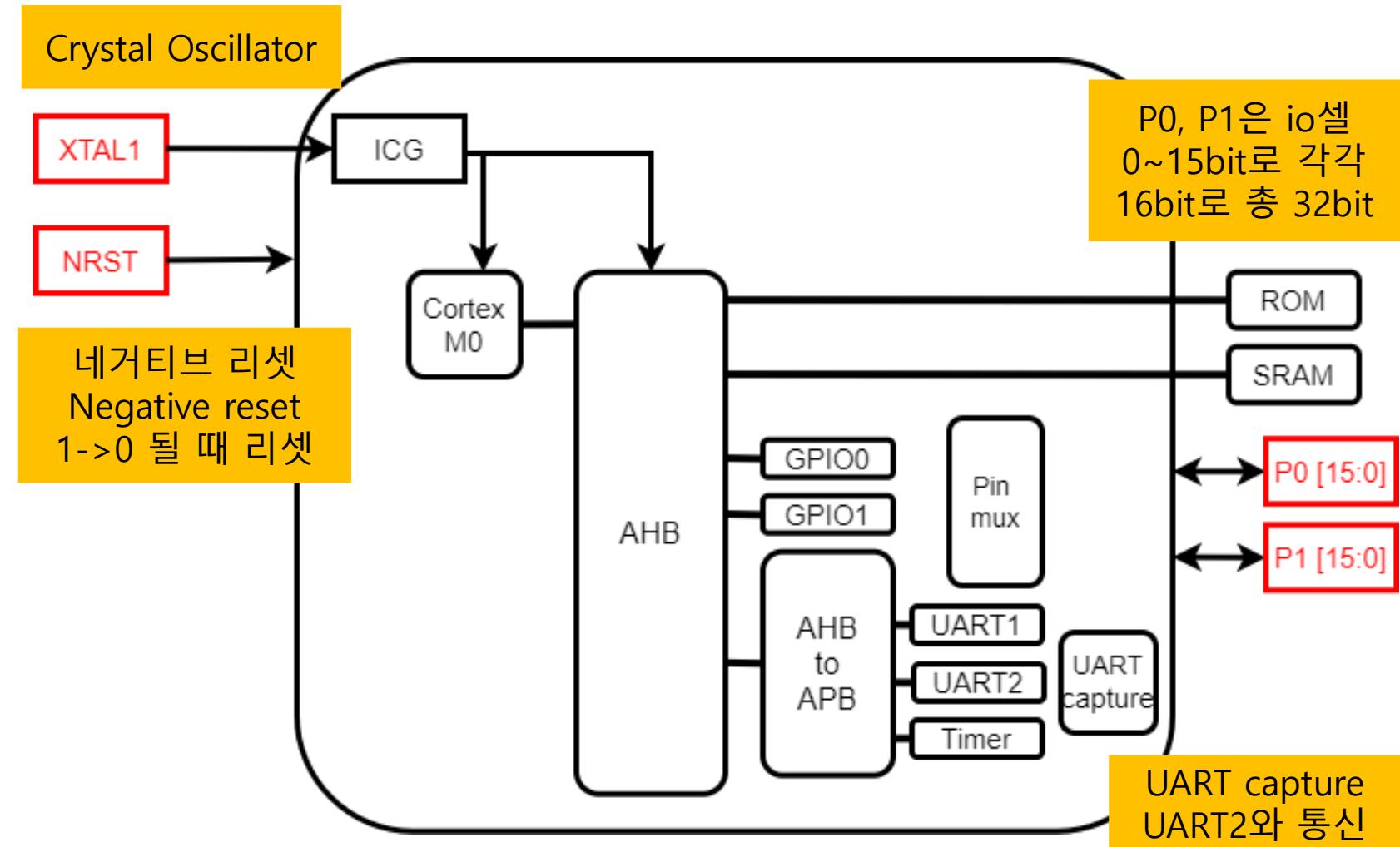
## Function 시뮬레이션의 목적

1. Verilog 소스를 시뮬레이션을 통해 기능 검증하는 단계
2. hex 코드가 rom으로 잘 다운로드 되는지 확인
3. 검증 완료 후 Hello World와 smkcow를 출력시킴

# Function Simulation

## 시스템 구조

1. 빨간색 표시된 것만 시뮬레이션에서 확인할 예정임
2. **ICG**: Integrated Clock Gating의 약자로 hex코드가 rom에 다운로드 되는 동안 클럭을 닫아서 모듈의 동작을 멈춤
3. **Behavioral Model**: 동작만 하는 메모리 모델임 즉 시뮬레이션만 가능하고 합성 불가능



# Function Simulation

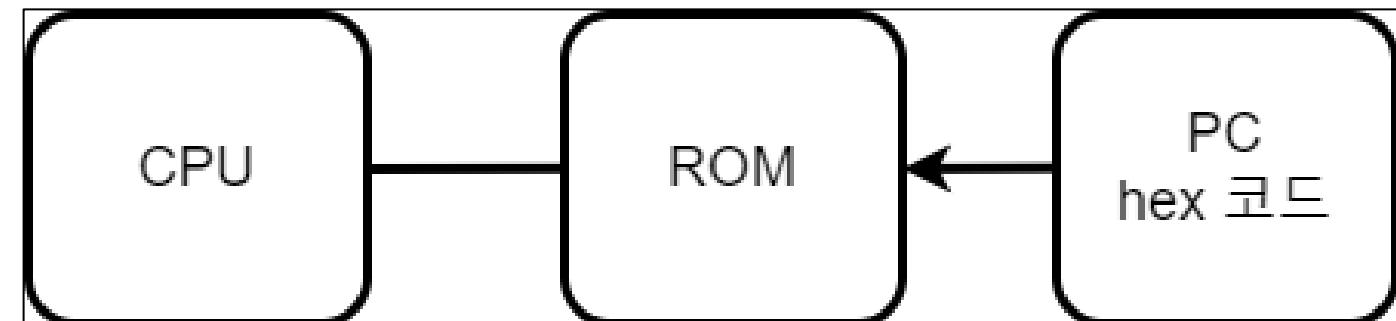
## Hex코드

### 1. C 코드가 CPU로 이동하는 과정

- I. C코드는 컴파일러를 통해 hex로 변환됨(Binary code)
- II. hex 코드를 ROM에 전달

### 2. CPU의 코드 처리 방식

- **Little Endian:** 모바일 시장 – ARM (hex 코드를 거꾸로 입력하는 방식)
- **Big Endian:** PC 시장(hex 코드를 있는 그대로 입력하는 방식)



```
86 //-----  
87 // AHB_ROM_BEH_MODEL : Simple behavioral model (default)  
88  
89 generate if (MEM_TYPE == `AHB_ROM_BEH_MODEL) begin  
90     // Behavioral memory model  
91     cmsdk_ahb_rom_beh  
92     #( .AW(AW),  
93         .filename(filename),  
94         .WS_N(WS_N),  
95         .WS_S(WS_S)  
96     )
```

# Function Simulation

## 시뮬레이션

1. ./clean.tcl ↵

2. ./run\_function.tcl ↵

- xrun 실행

```
[ex_poly1@npit FUNCTION]$ ./clean.tcl
[ex_poly1@npit FUNCTION]$ ./run_function.tcl
```

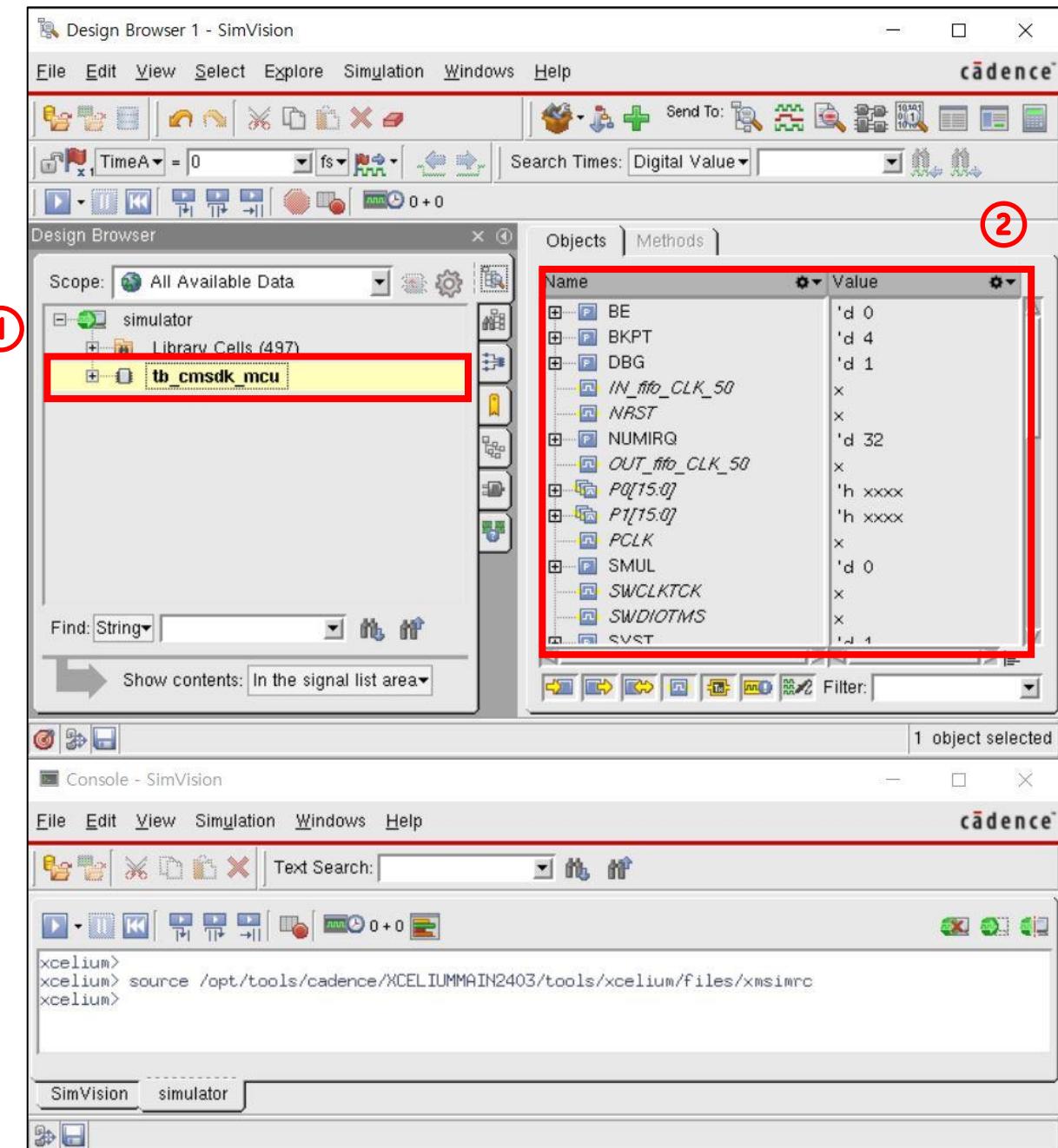
```
xrun -64bit \
  +max_err_count+50 \
  +define+function_sim \
  -access +rwc \
  -profile \
  -profthread \
  -gui \
+libext+.v \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_timer/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_dualtimers/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_uart/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_watchdog/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_slave_mux/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_apb_subsystem/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_ahb_slave_mux/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_ahb_default_slave/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_ahb_gpio/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_ahb_to_apb/verilog \
-incdir ../../../../../../cortexm0_designstart/logical/models/clkgate \
-incdir ../../../../../../cortexm0_designstart/logical/cmsdk_iop_gpio/verilog \
-incdir ../../../../../../cortexm0_designstart/cores/cortexm0_designstart_r1p0/logical/cortexm0_integration/verilog \
```

# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

- ① tb\_cmsdk\_mcu 클릭
- ② 포트 확인



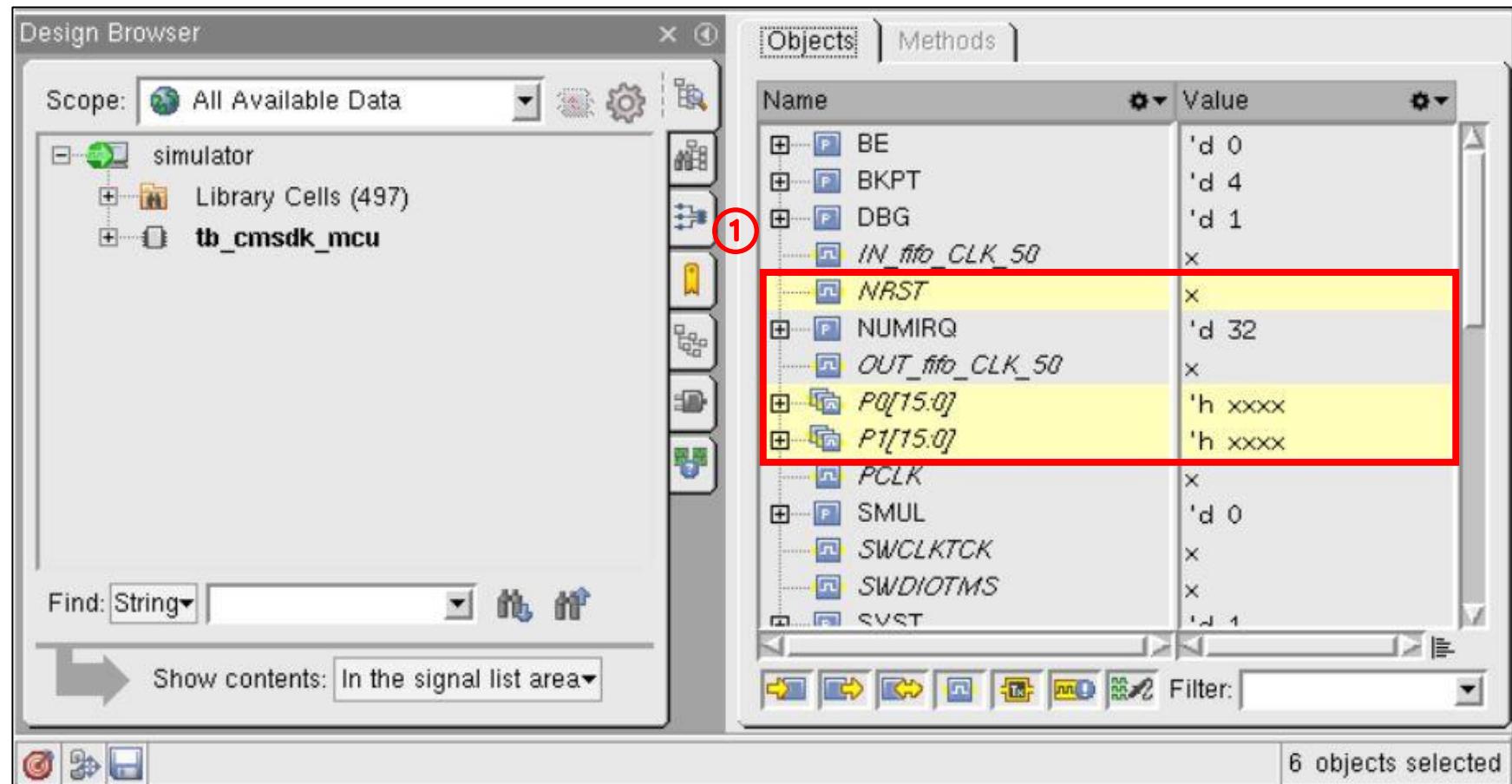
# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 시뮬레이션에서 확인할 포트 선택

(XTAL1, NRST, P0, P1, srx, stx)



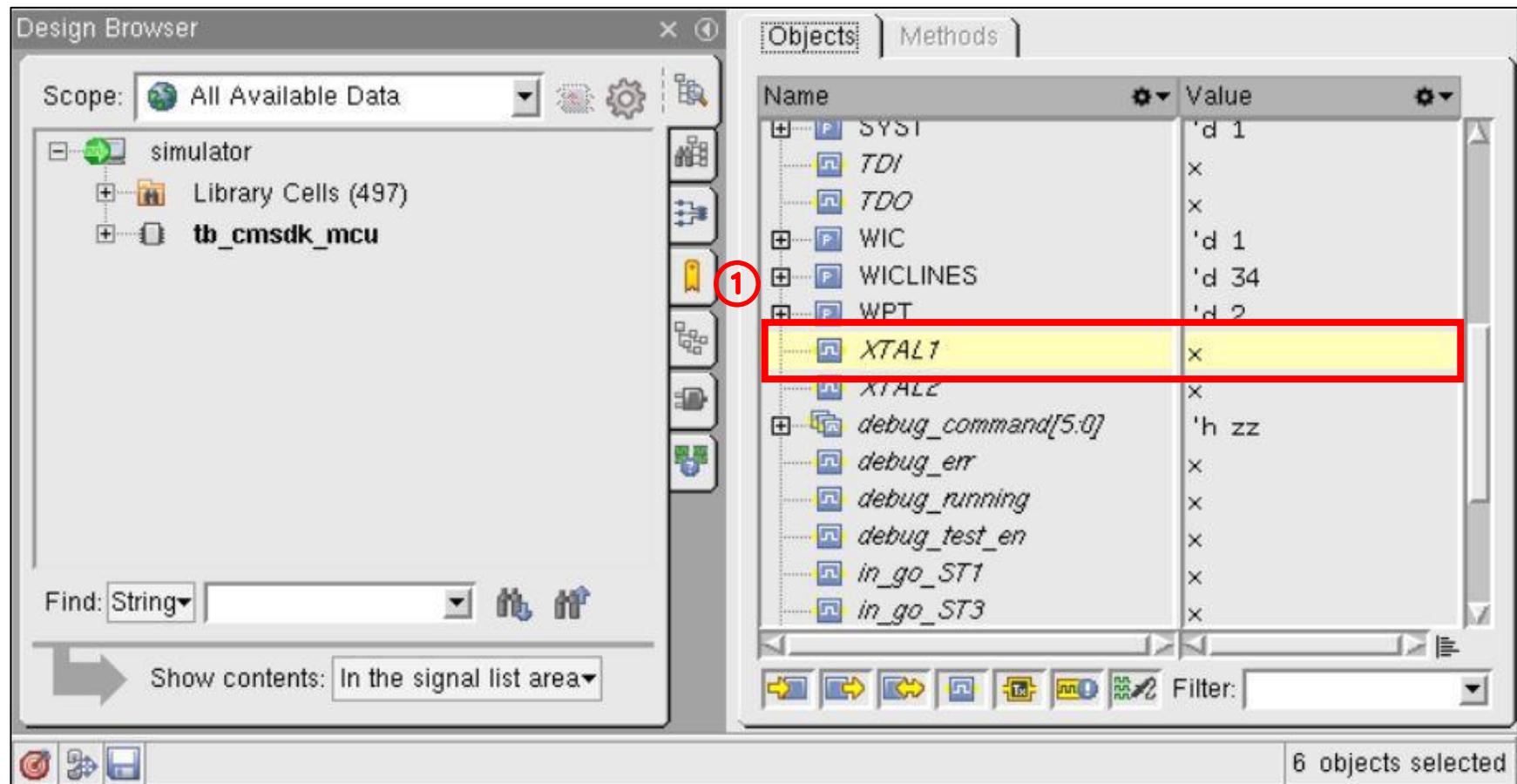
# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 시뮬레이션에서 확인할 포트 선택

(XTAL1, NRST, P0, P1, srx, stx)



# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 시뮬레이션에서 확인할 포트 선택

(XTAL1, NRST, P0, P1, srx, stx)

The screenshot shows the Xilinx Vivado Design Suite's Design Browser window during a function simulation. The left pane displays the project structure under 'Scope: All Available Data'. The 'tb\_cmsdk\_mcu' item is expanded, showing its internal components. The right pane shows a table of simulation objects. A red box highlights the last two rows, which correspond to the ports selected in the previous step: 'srx' and 'stx'. The row for 'srx' is also circled with a red number '1'.

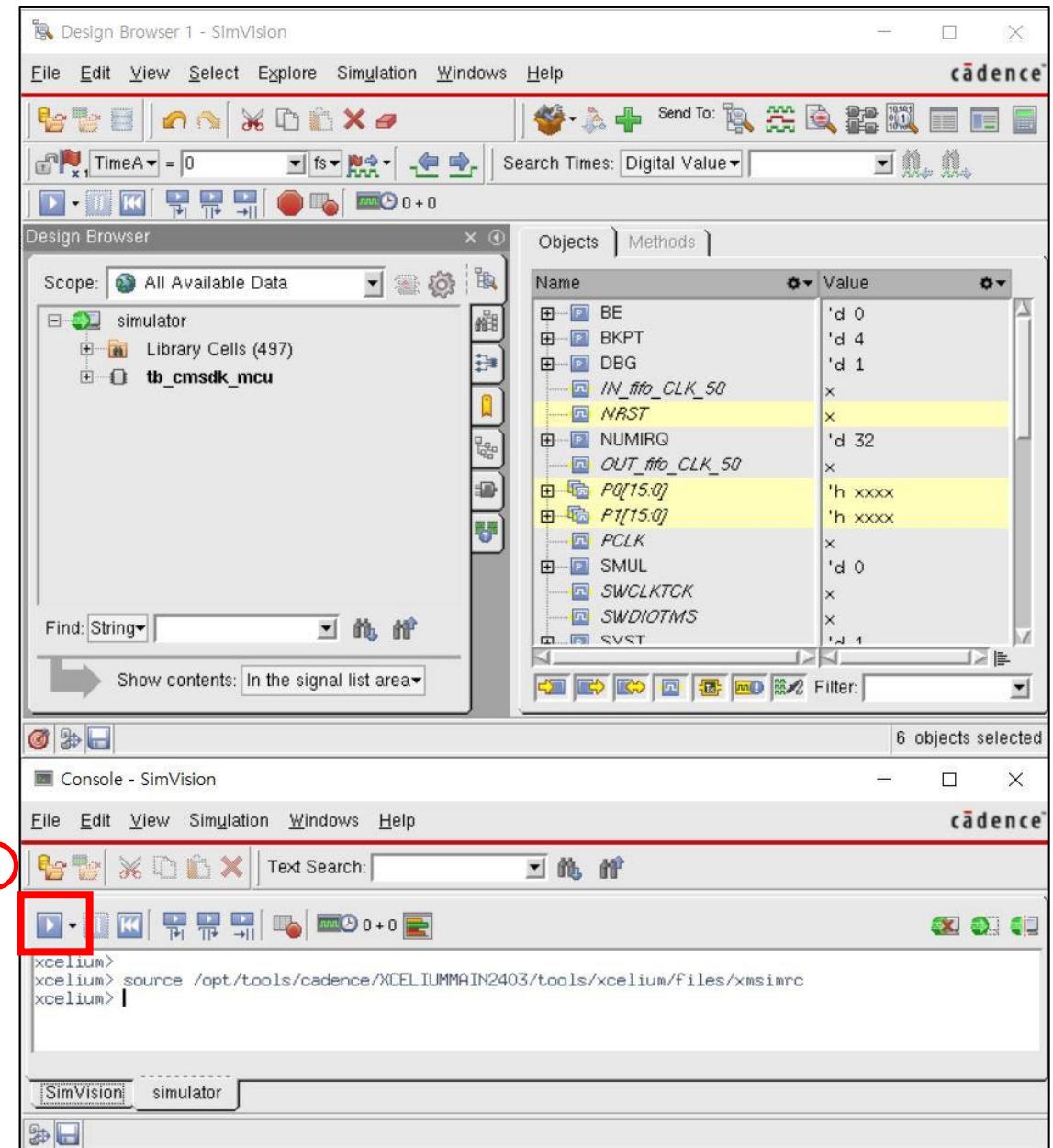
Name	Value
XTAL1	x
debug_command[5:0]	'h zz
debug_err	x
debug_running	x
debug_test_en	x
in_go_ST1	x
in_go_ST3	x
in_go_ST4	x
in_wait_ST1	x
in_wait_ST3	x
in_wait_ST4	x
nTRST	x
srx	x
stx	x

# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 모두 선택 후 실행 버튼 클릭

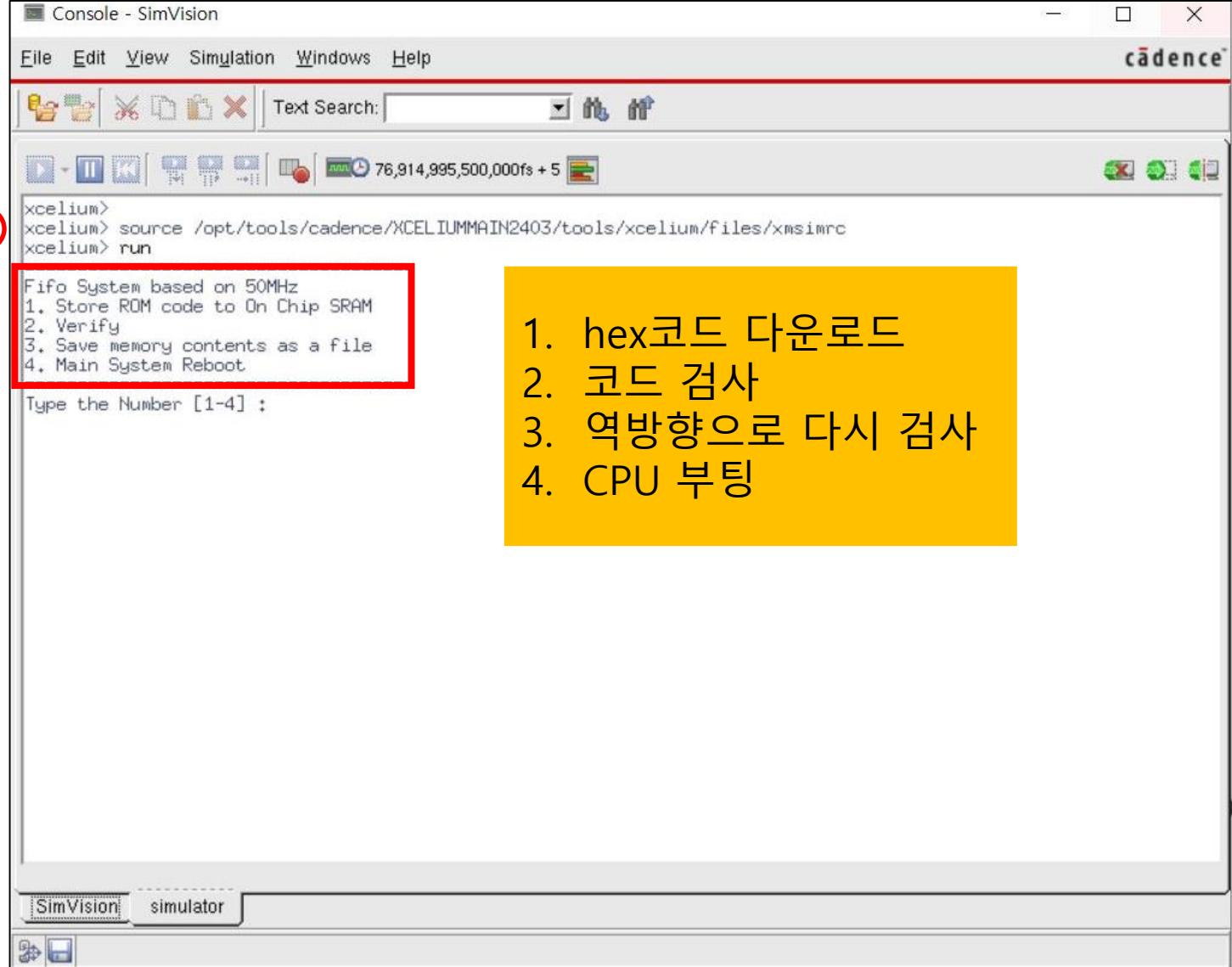


# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 1~4까지 숫자가 뜨는데 자동으로 1번이 실행 됨



The screenshot shows the Cadence SimVision console window. The title bar reads "Console - SimVision". The menu bar includes File, Edit, View, Simulation, Windows, Help, and the Cadence logo. The toolbar has various icons for simulation control. The main area shows a command-line interface:

```
xcelium>
xcelium> source /opt/tools/cadence/XCELIUMMAIN2403/tools/xcelium/files/xmsimrc
xcelium> run
```

Below the command line, a red box highlights the following text:

```
Fifo System based on 50MHz
1. Store ROM code to On Chip SRAM
2. Verify
3. Save memory contents as a file
4. Main System Reboot
```

Below this box, the text "Type the Number [1-4] :" is displayed. A red circle with the number "1" is drawn over the first option in the list.

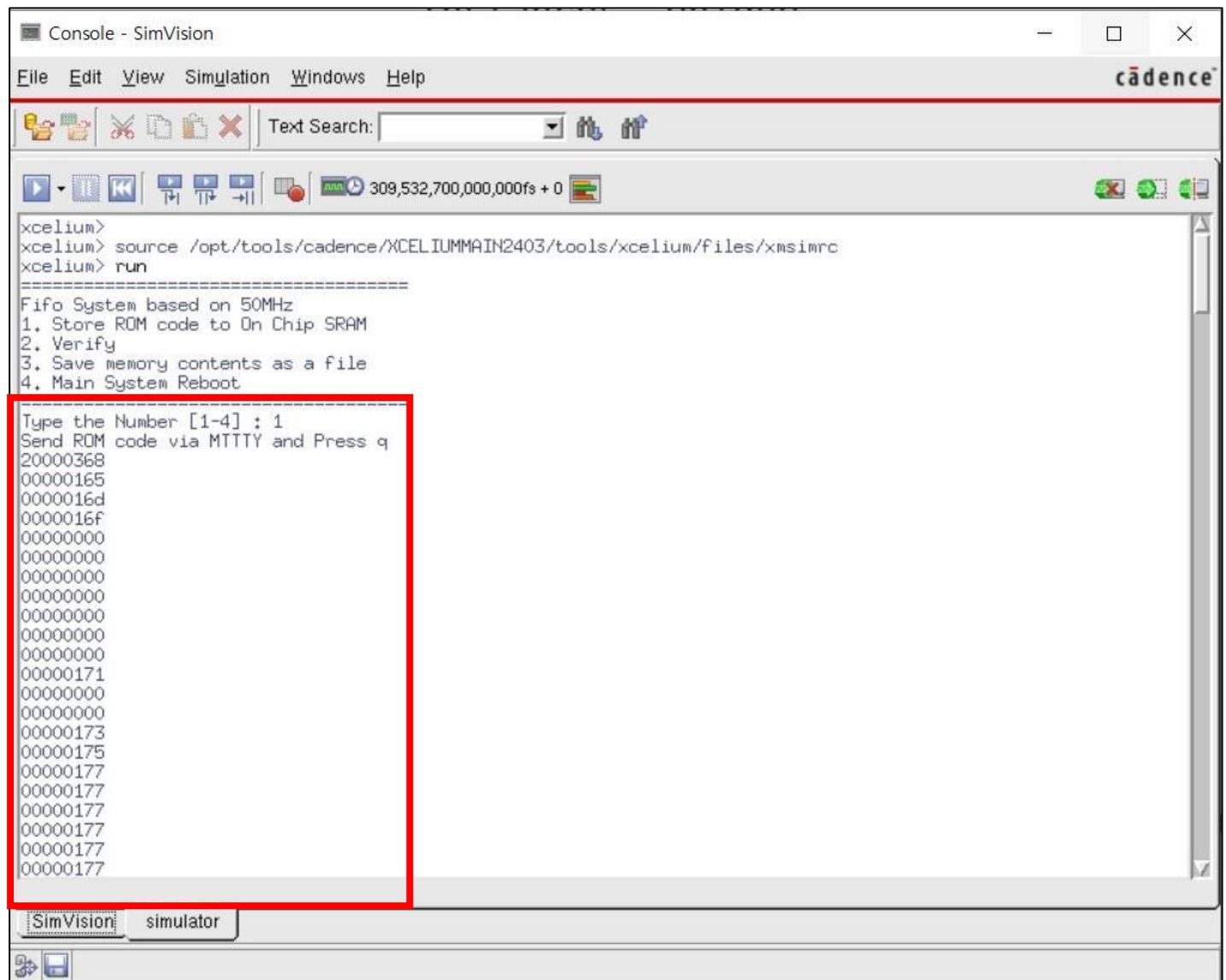
1. hex코드 다운로드
2. 코드 검사
3. 역방향으로 다시 검사
4. CPU 부팅

# Function Simulation

# 시뮬레이션

## 1. xrun 실행 결과

① 1번이 실행되면서 1번 ~ 4번의 과정이 자동 실행

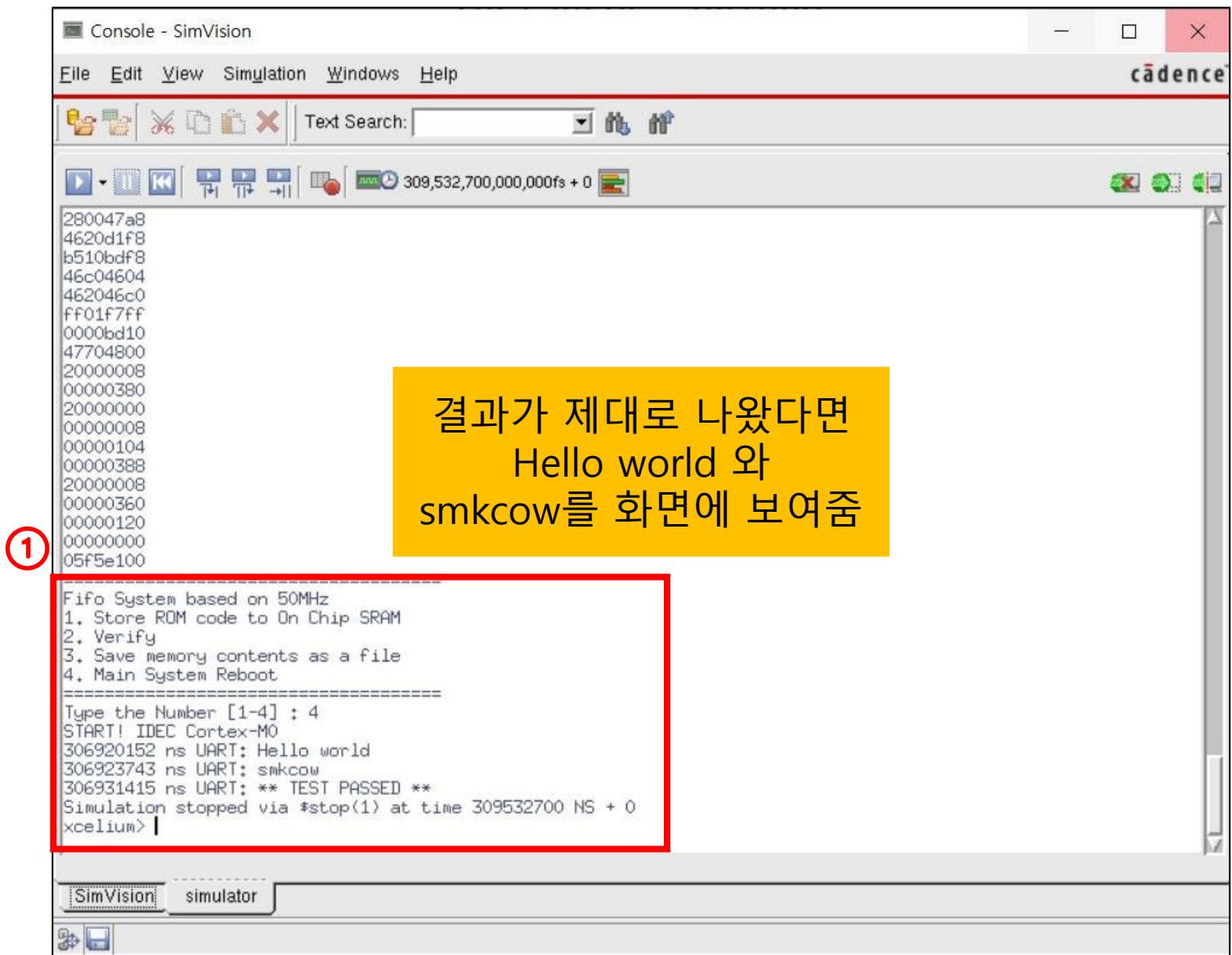


# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

① 검사과정은 나타내지 않고 부팅이 잘 되었는지 4번으로 결과 확인



Console - SimVision

File Edit View Simulation Windows Help

Text Search:

309,532,700,000,000fs + 0

280047a8  
4620d1f8  
b510bdf8  
46c04604  
462046c0  
ff01f7ff  
0000bd10  
47704800  
20000008  
00000380  
20000000  
00000008  
00000104  
00000388  
20000008  
00000360  
00000120  
00000000  
05F5e100

결과가 제대로 나왔다면  
Hello world 와  
smkcow를 화면에 보여줌

1

Fifo System based on 50MHz  
1. Store ROM code to On Chip SRAM  
2. Verify  
3. Save memory contents as a file  
4. Main System Reboot  
=====  
Type the Number [1-4] : 4  
START! IDEC Cortex-M0  
306920152 ns UART: Hello world  
306923743 ns UART: smkcow  
306931415 ns UART: \*\* TEST PASSED \*\*  
Simulation stopped via \$stop(1) at time 309532700 NS + 0  
xcelium> |

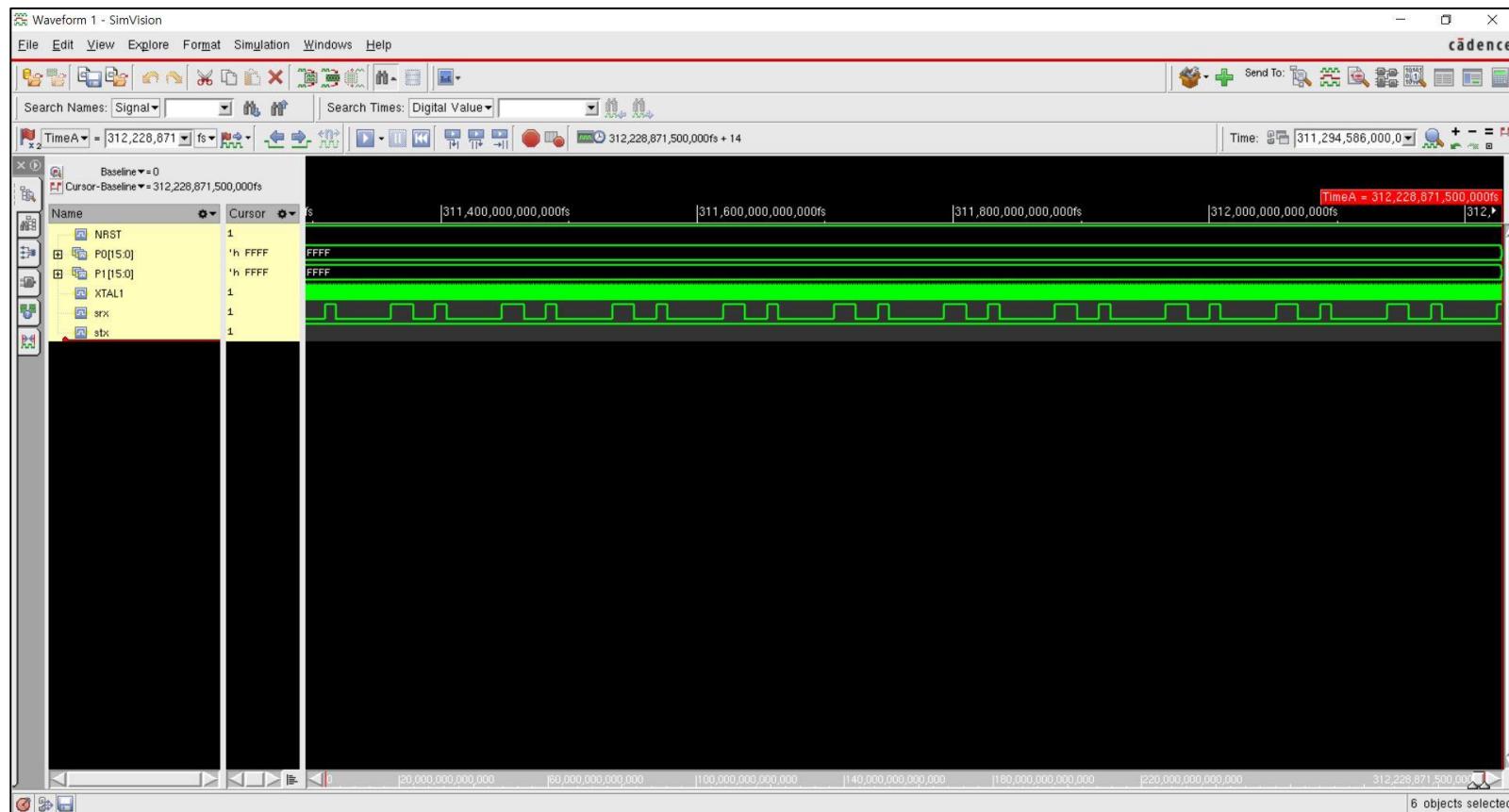
SimVision simulator

# Function Simulation

## 시뮬레이션

### 1. xrun 실행 결과

- 파형도 확인은 가능하나 보기 쉽게 텍스트로 확인
- 개인이 보고자 하는 포트나 시그널들을 자유롭게 추가 삭제 가능



# Function Simulation

## ■ 시뮬레이션 분석

1. vi로 hex코드 확인

```
$> vi hello.hex
```

```
[ex_poly1@npit FUNCTION]$ vi hello.hex
```

# Function Simulation

## 시뮬레이션 분석

## 1. hello.hex

- 시뮬레이션 결과와 비교

# Function Simulation

## 시뮬레이션 분석

## 1. hello.hex

## Little Endian

- 한 줄의 각 숫자를 2진수로 표현하면 숫자 하나당 4bit이므로 총 32bit임

- Cortex-M0은 32bit 프로세서임  
을 알 수 있음

Type the Number [1-4] : 1  
Send ROM code via MTTTY and Press enter  
**20000368**  
00000165  
0000016d  
0000016f  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000171  
00000000  
00000000  
00000173  
00000175  
00000177  
00000177  
00000177  
00000177  
00000177

```
File Edit View Search Terminal Help
1 68
2 03
3 00
4 20
5 65
6 01
7 00
8 00
9 6D
10 01
11 00
12 00
13 6F
14 01
15 00
16 00
17 00
18 00
19 00
20 00
21 00
22 00
23 00
24 00
```

# 합성(Synthesis)

# Introduction to Synthesis

```
always @ (posedge CLK or posedge in_Rom_select)
begin
    if (in_Rom_select == 1'b1)
        reset_connect <= 1'b0;
    else
        reset_connect <= 1'b1;
end
:
:
```

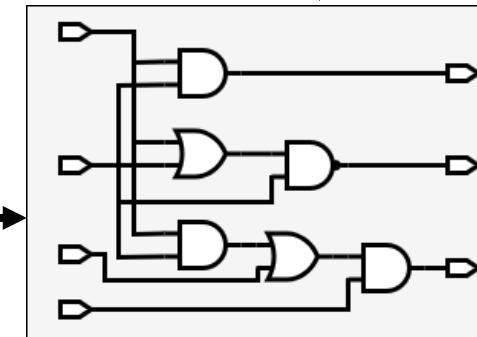
RTL source

① Translate (read\_Verilog)

Constraints

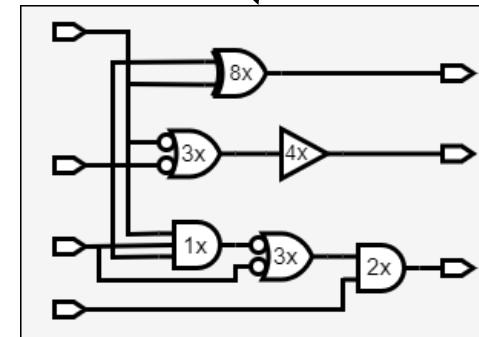
```
set_max_area ...
create_clock...
set_input_delay...
```

② Constraint (source)



Generic Boolean Gates  
(GTECH/UNMAPPED)

③ Map + Optimize



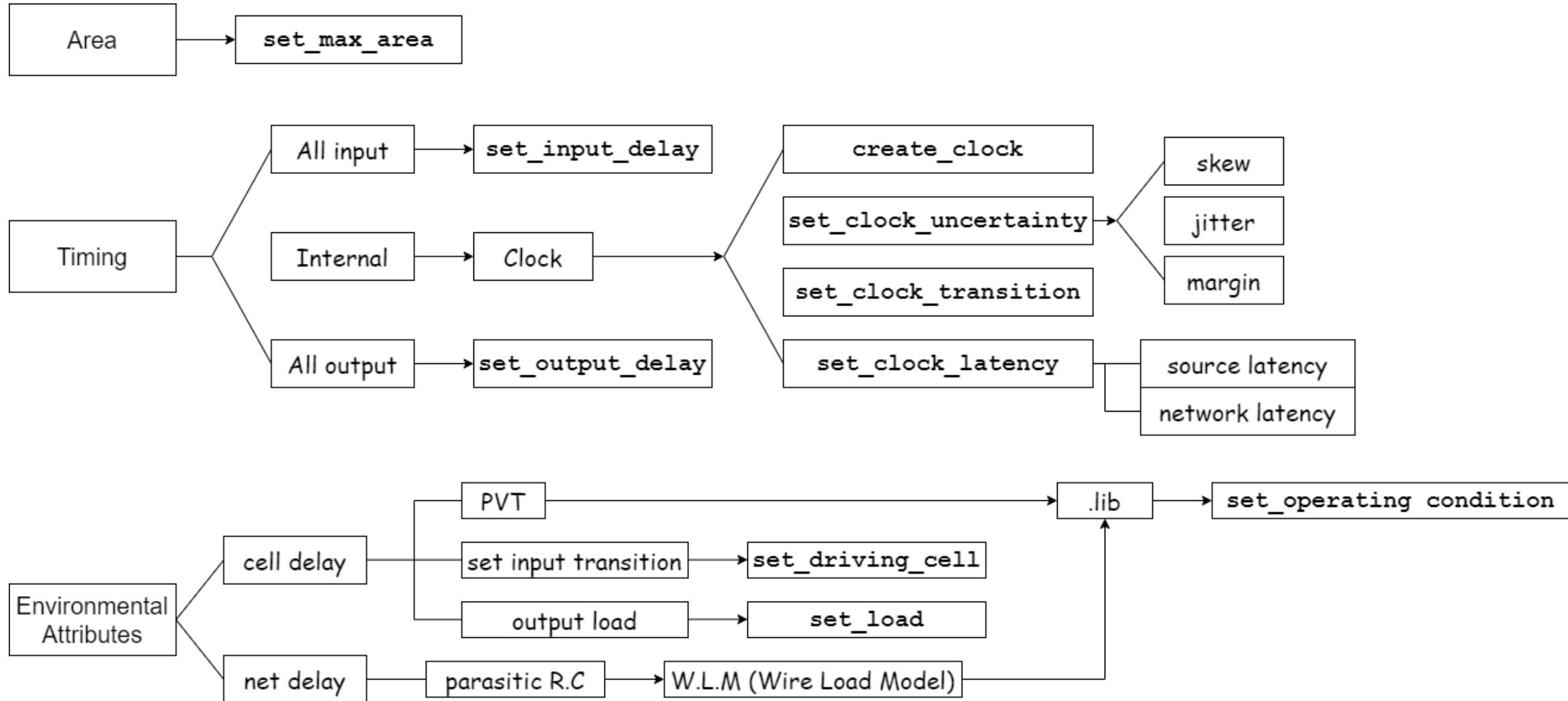
Technology specific Gates  
(MAPPED)

④ Save (Write out Gate Level Netlist)

# Constraints

# Synthesis

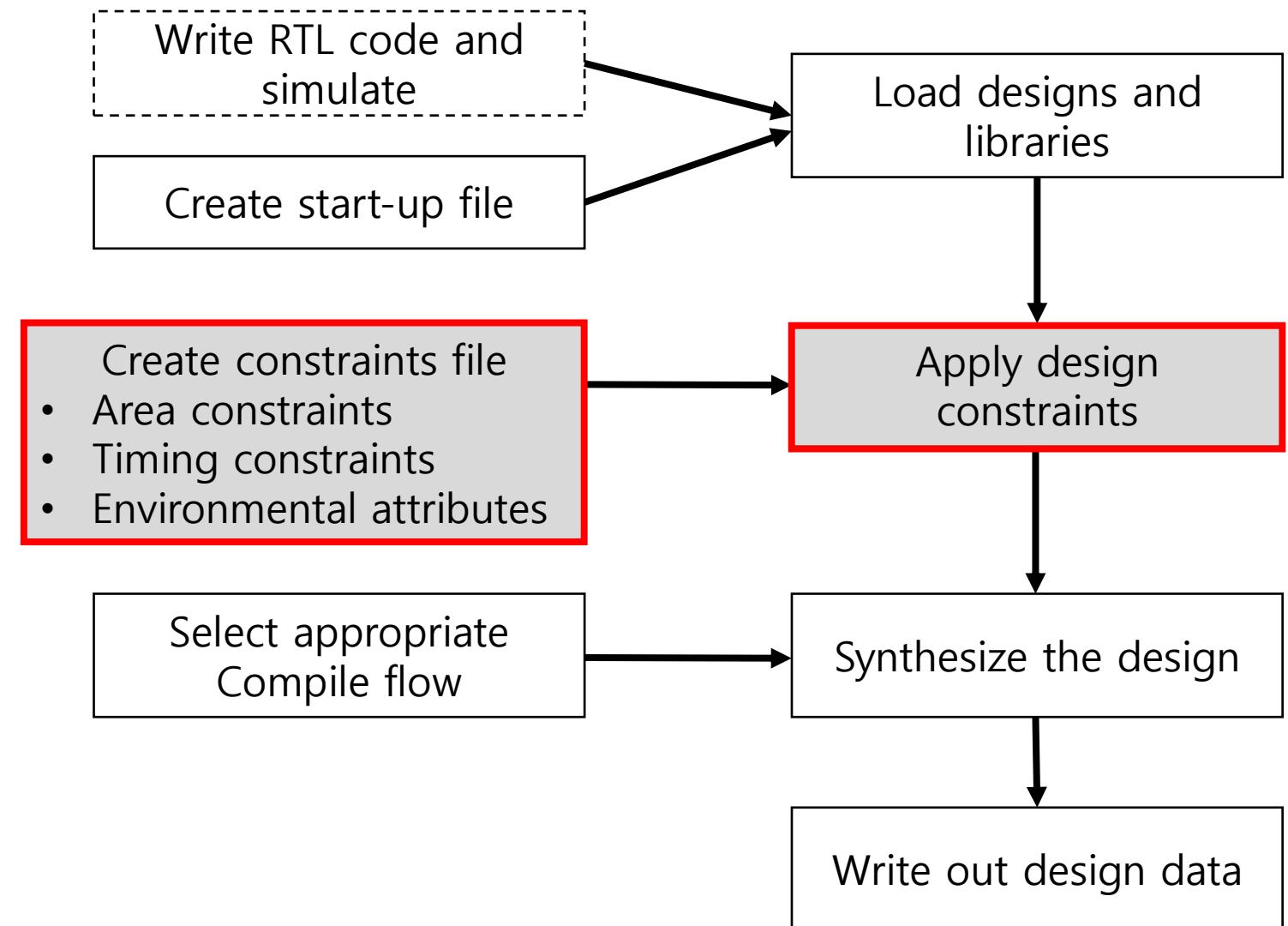
## Constraint



# Constraints

## Area and Timing Constraints

### RTL Synthesis Flow



# Constraints

## Area Constraints

### 면적 조건 설정

```
genus > read_Verilog MY_DESIGN.v  
genus > current_design TOP_CHIP_or_BLOCK  
genus > link  
genus > set_max_area 245000
```

- **Area unit is defined by the library supplier**  
-it's not in the library so ask!
  - 2-input NAND gates
  - Transistors
  - mils<sup>2</sup>, mm<sup>2</sup> or μm<sup>2</sup>
- **How do you determine what value to use?**
  - From the spec or project lead
  - If migrating to a newer technology use a smaller % of the old design size
  - Estimate based on experience

**set\_max\_area 0** 의 의미  
→ 사이즈를 최소한으로 줄여라



Is **set\_max\_area 0** acceptable?

# Constraints

## Timing Constraints

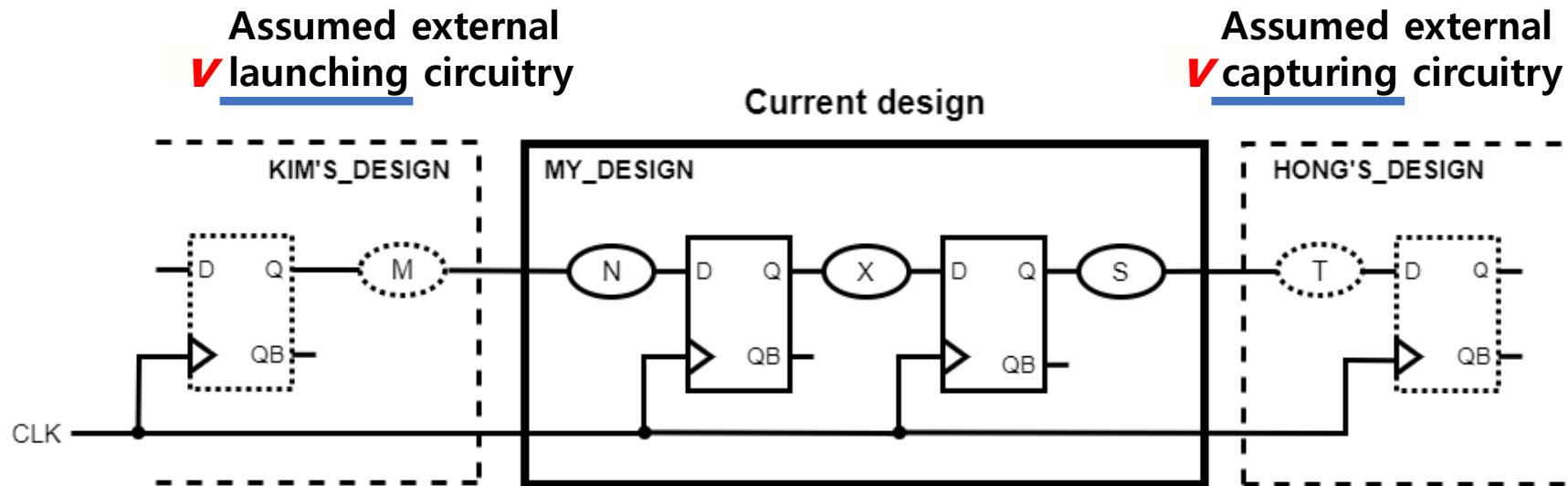
### 셋업 타임 조건 설정

- **Objective: Define setup timing constraints for all paths within a sequential design**
  - All input logic paths (starting at input ports)
  - The internal (register to register) paths
  - All output paths (ending at output ports)
- **How do you determine what value to use?**
  - You are given the design's specs
  - Block-or chip-level design
  - Single clock, single cycle or environment

# Constraints

## Timing Constraints

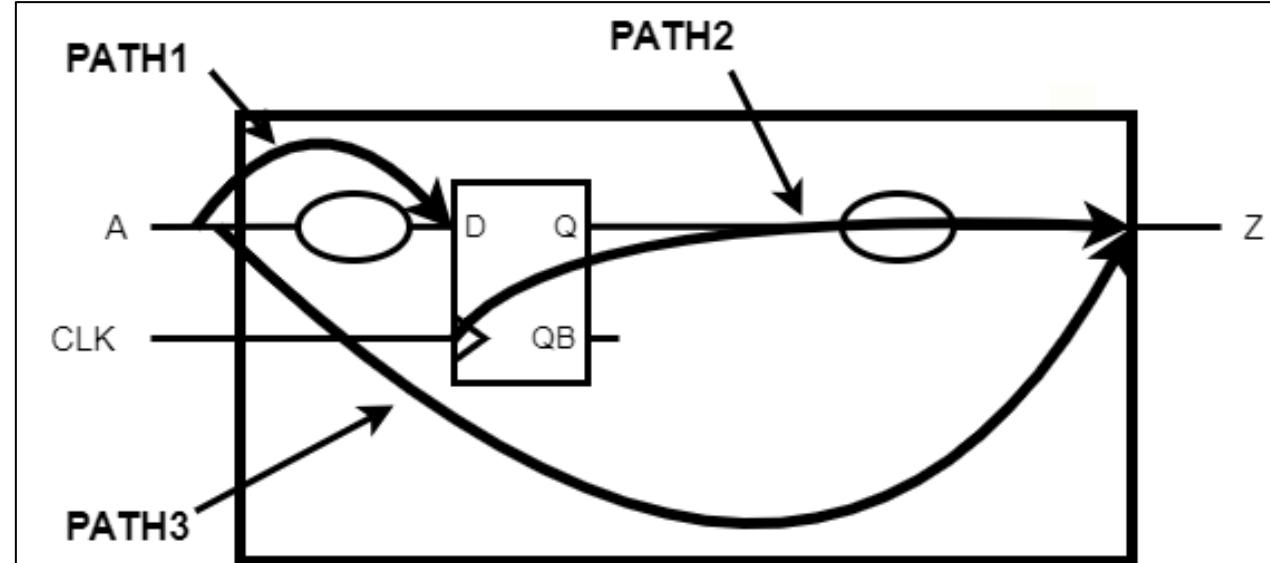
### 기본적인 디자인(예시)



# Constraints

## Timing Constraints

### 합성 진행 시 타이밍 분석

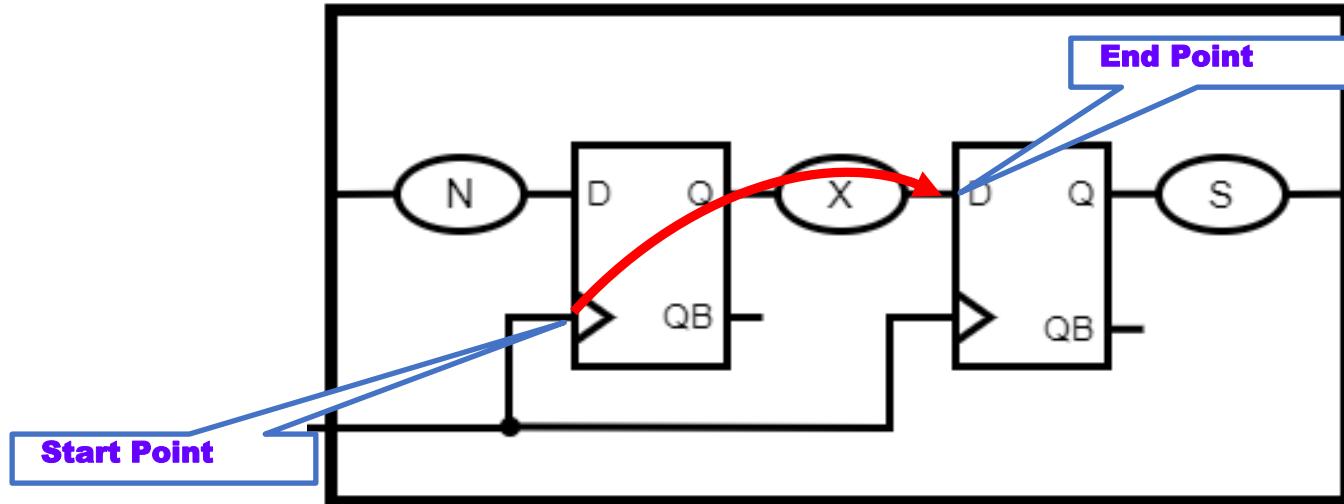


- Internal paths (Reg-to-Reg): 디자인 내부 레지스터 간의 신호 경로를 정의한다.
- All input logic paths: 외부 입력에서 내부 회로로 이어지는 신호 경로를 설정한다.
- All output paths: 내부 회로에서 외부 출력으로 이어지는 신호 경로를 설정한다.

# Constraints

## Timing Constraints

### 합성 진행 시 타이밍 분석



- Internal paths (Reg-to-Reg): 디자인 내부 레지스터 간의 신호 경로를 정의한다.

# Constraints

## Timing Constraints

- 목표: 하나의 순차적인 디자인 내부의 모든 path들에 대한 셋업 타임 조건을 정의한다.

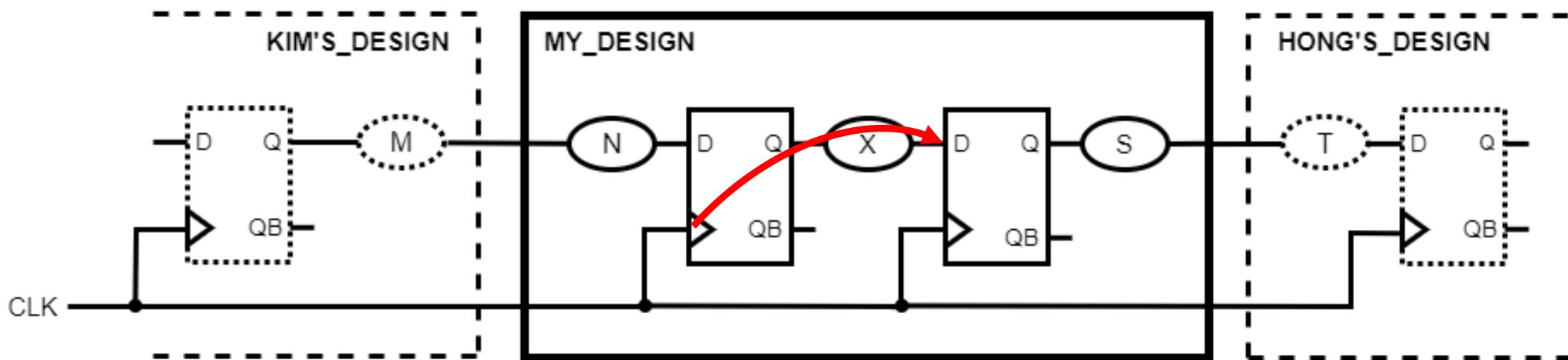
- All input logic paths (starting at input ports)
- The internal (register to register) paths
- All output paths (ending at output ports)

1. 입력 단
2. 디자인 내부
3. 출력 단

- CLK 및 아래 3가지 중요
- Uncertainty(skew)
  - Transition
  - Latency

# Constraints

## Timing Constraints



# Constraints

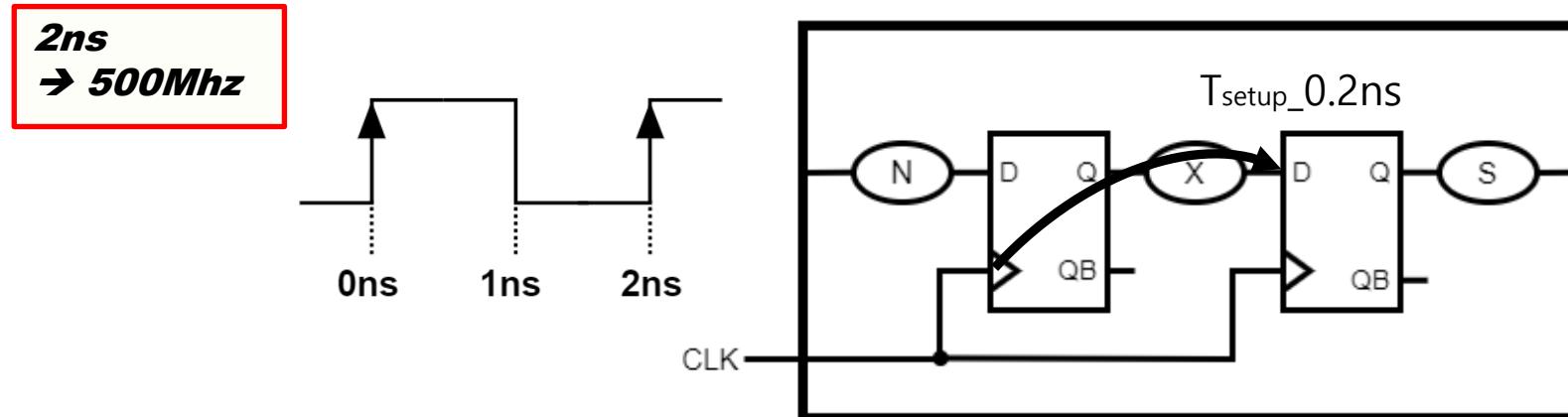
## Timing Constraints

### Constraining Reg-to-Reg Path Example

Spec:

Clock Period = 2ns

```
create_clock -period 2 [get_ports CLK]
```

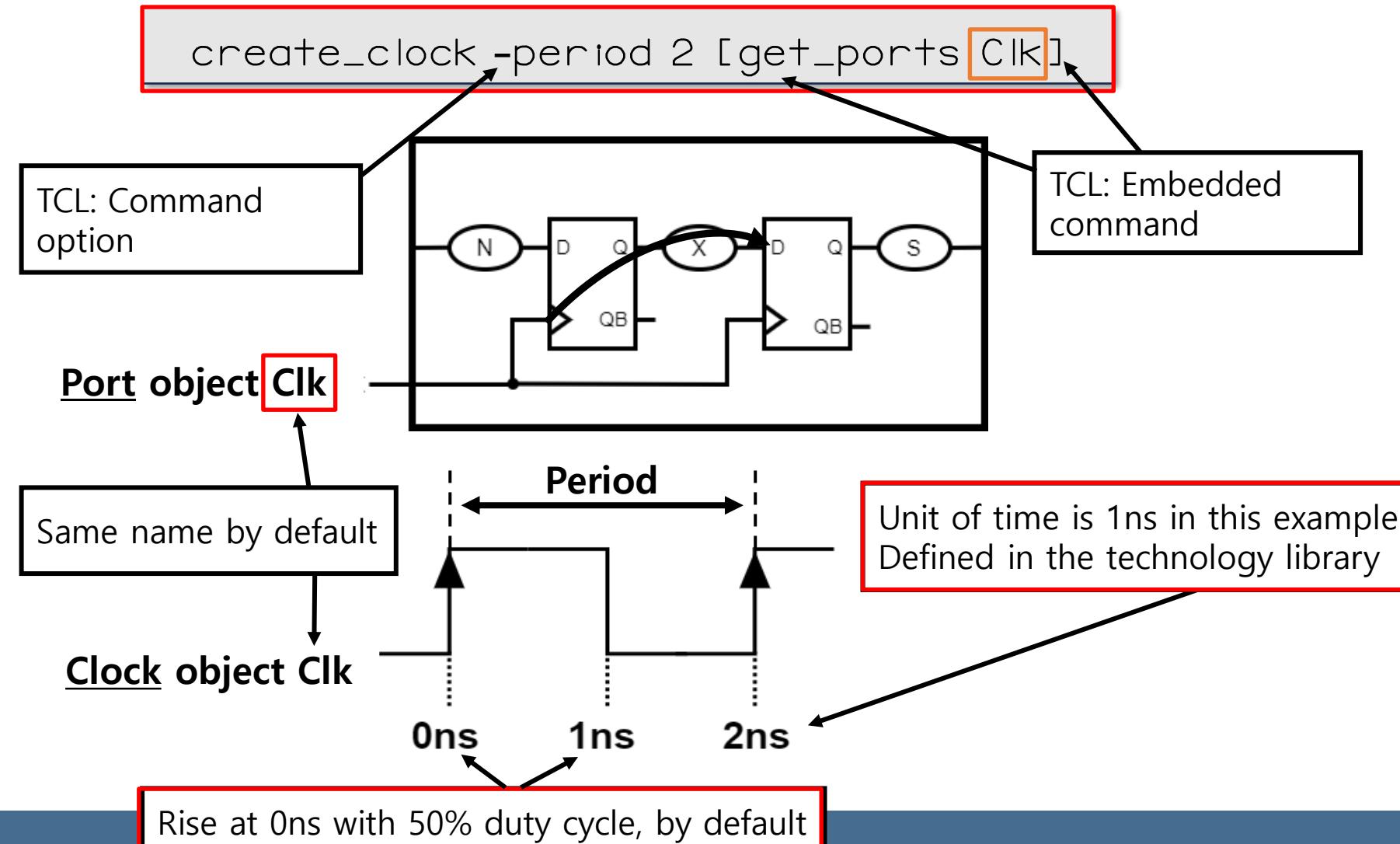


- 설계에서 셋업 타임(setup time)은 신호가 안정적으로 전달되도록 보장하는 핵심 요소다. 셋업 타임은 클럭의 상승 엣지 이전에 신호가 안정적으로 도착해야 하는 시간이다. 위 그림에서는 0.2ns로 표시되어 있다. 예를 들어, 2ns의 클럭 주기와 0.2ns의 셋업 타임이 주어질 경우 신호는 1.8ns 이전에 들어와야 한다.

# Constraints

## Timing Constraints

### create\_clock Required Arguments



# Constraints

## Timing Constraints

### 기본적인 클럭 동작

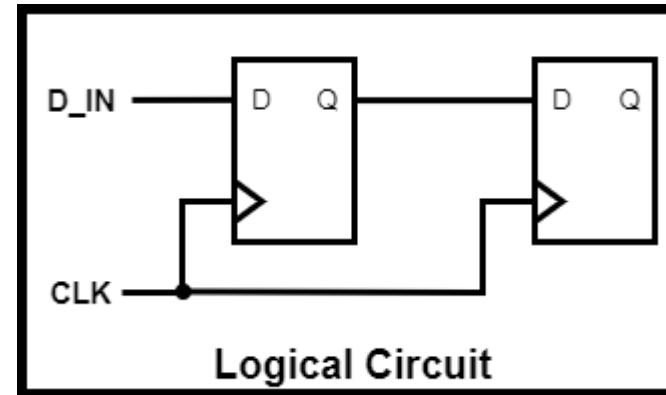
- 단일 클럭 디자인에서 클럭을 정의하면 레지스터 간 모든 타이밍 path(Reg to Reg)들이 단일 주기, 셋업 타임에 의해 결정된다.  
기본적으로 클럭은 0ns에서 상승, Duty cycle은 50%
- By default DC will not "buffer up" the clock network, even when connected to many clock/enable pins of flip-flop/latches
  - 클럭 신호는 "이상적(ideal)"이라고 가정한다. (infinite drive capability)
    - ◆ Zero rise/fall transition times
    - ◆ Zero skew -> **(uncertainty)**
    - ◆ Zero insertion delay or latency
  - Skew, Latency, Transition 시간은 클럭 동작을 더 정확하게 나타내기 위해 모델링할 수 있어야 하며 이는 필수적인 과정이다. (behavior)

# Constraints

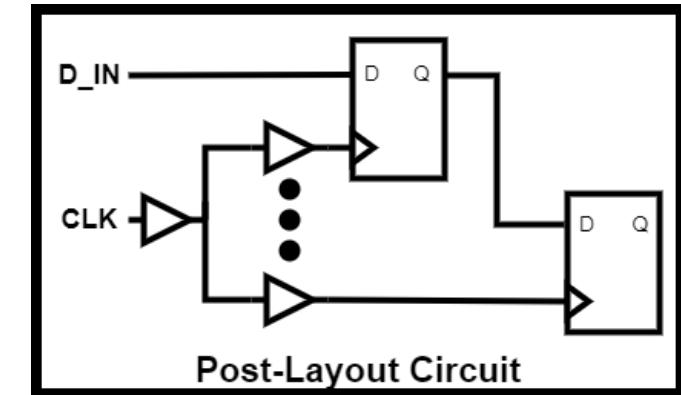
## Timing Constraints

### Clock Trees 모델링

- Front End 단계 Synthesis Constraints에서는 Internal Path에 영향을 미치는 시간적 변수들을 추정값으로 설정한다. 이는 설계 초기 단계에서 실제 클럭 트리가 없기 때문에 정확한 값을 알 수 없기 때문이다.
- 따라서 예상되는 최악의 시나리오를 기준으로 보수적인(pessimistic) 값을 설정한다.



**Design Compiler**



**Astro 또는 ICC**

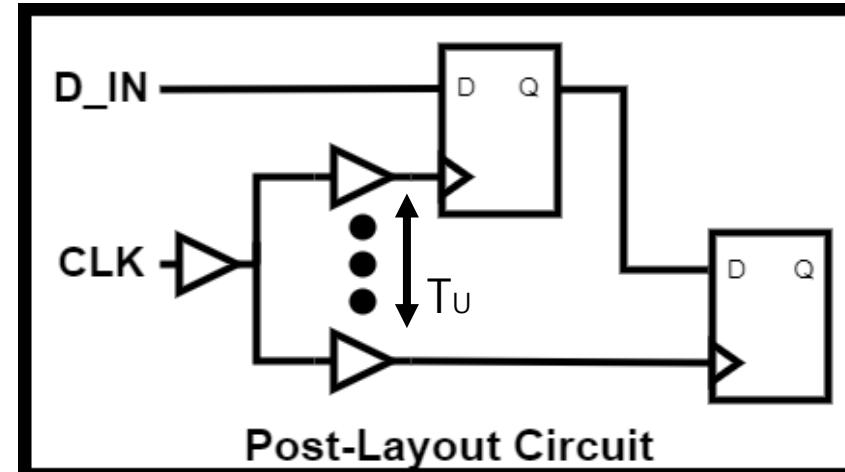
- Back End 단계 Post-CTS STA Constraints에서는 CTS(Clock Tree Synthesis)를 통해 실제 클럭 트리가 생성되고, Skew, Network Latency, Transition의 실제 값이 계산된다.
- 이 단계에서는 Front End에서 설정했던 추정값을 제거하고, 실제 값을 기반으로 타이밍 제약 조건을 재설정한다.

# Constraints

## Timing Constraints

### 클럭 Skew 모델링

- Skew는 클럭 신호가 동일한 소스에서 발생했음에도 불구하고 설계 내의 각 지점에 도달하는 시간이 달라지는 현상을 의미한다.
- Jitter는 클럭 신호의 위상(phase)에 미세한 변동이 생기는 현상으로 위상차에 따른 지글거림으로 이해할 수 있다.
- Margin은 설계와 제조 과정에서 발생할 수 있는 미세한 오차를 대비해 추가적으로 설정하는 여유 시간이다.



Set\_clock\_uncertainty -setup\_Tu [get\_clocks CLK]

Pre-Layout: clock skew + jitter + margin

Uncertainty는 클럭 신호가 실제 전달 과정에서 겪는 다양한 불확실성을 보정하기 위하여 설정하는 여유 시간이다. 이는 세부적으로 Skew, Jitter, Margin으로 구성된다

# Constraints

## Timing Constraints

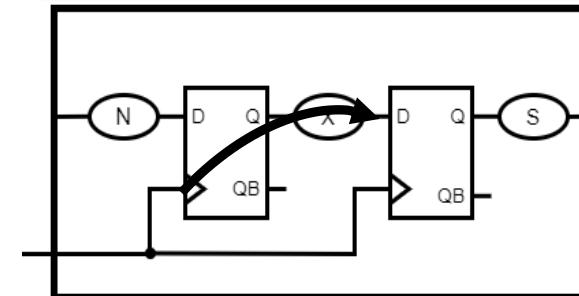
- Uncertainty를 구성하는 세 가지 변수는 실제 그 특정 값을 정확하게 파악하기 쉽지 않다.
- 칩의 안정적인 동작을 위해 시 간적 여유를 세 변수를 모두 합산하여 보수적(pessimistic)으로 설정한다.

### Set\_clock\_uncertainty and Setup Timing

#### Example:

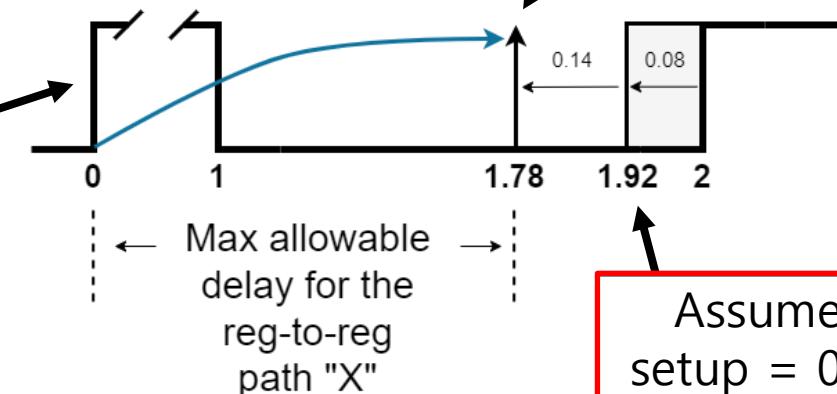
```
create_clock -period 2 [get_ports CLK]
```

```
Set_clock_uncertainty -setup 0.14 [get_clocks CLK]
```



FF1 Data Launch Edge  
(No uncertainty)

FF2 setup check at:  
 $2 - 0.14 - 0.08 = 1.78$



# Constraints

## Timing Constraints

### Latency 모델링 (지연 시간 부여)

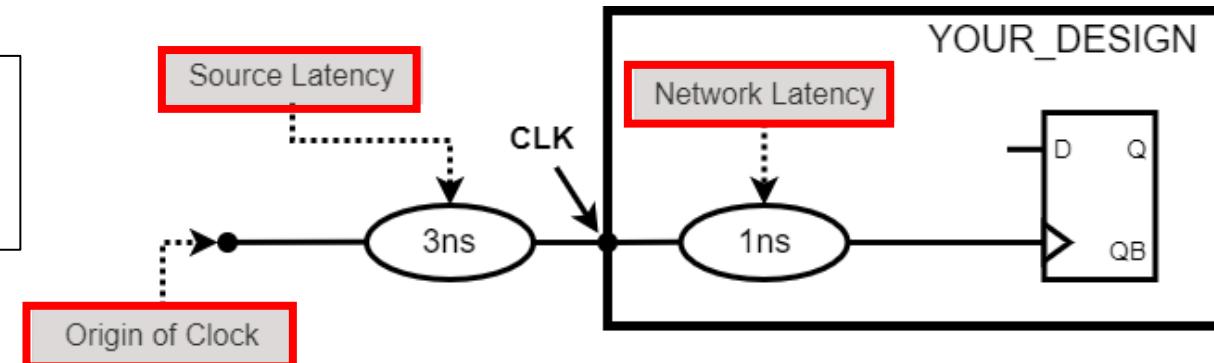
```
Genus> set_clock_latency -source -max 3 [get_clocks CLK]
```

: Source latency 최대 3ns 설정

```
Genus> set_clock_latency -max 1 [get_clocks CLK]
```

: Network latency 최대 1ns 설정

- Latency는 Source Latency(소스 레이턴시)와 Network Latency(네트워크 레이턴시)로 구분된다.

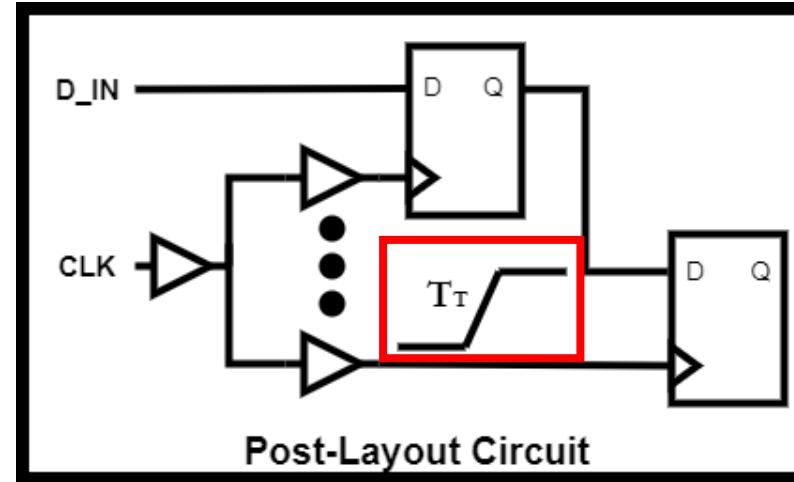


- Latency는 클럭 신호가 전달되는 과정에서 발생하는 시간적 지연(delay)을 의미한다.
- 오실레이터(Oscillator)와 같은 클럭 소스에서 시작하여 내부의 레지스터까지 신호가 도달하는 동안 발생하는 모든 지연 시간을 포함한다.

# Constraints

## Timing Constraints

### Transition 시간 모델링



```
Genus> set_clock_transition 0.08 [get_clock CLK]
```

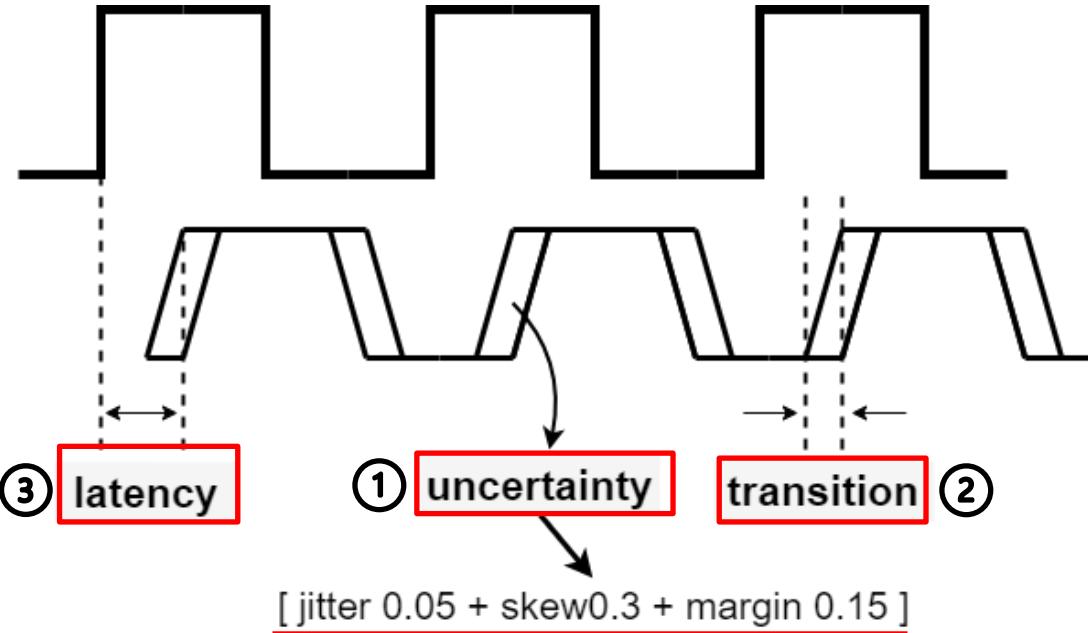
: 클럭에 0.08ns의 transition 설정

- Transition은 신호가 0에서 1로 상승하거나 1에서 0으로 하강할 때 걸리는 시간을 의미한다. 신호가 바뀌는 임계 전압(Threshold)까지 걸리는 시간으로 transition이 너무 길면 신호가 왜곡되거나 타이밍에 오류가 발생할 수 있다.

# Constraints

## Timing Constraints

### Pre/Post Layout Clock

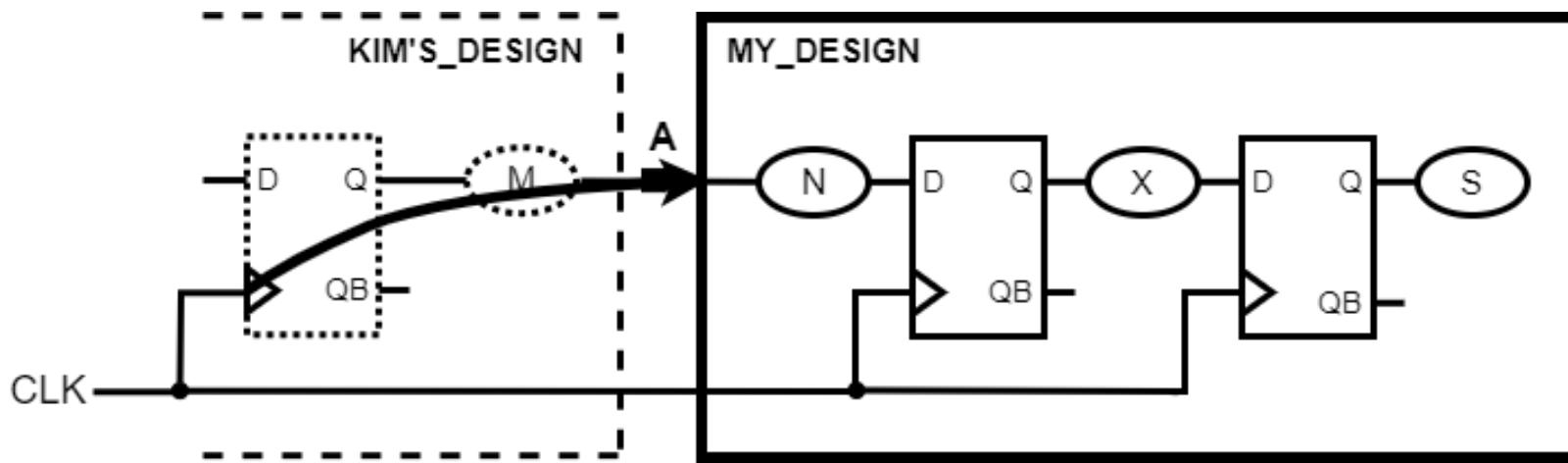


- timing constraint의 3가지 변수를 종합하여 선언하는 예시이다.
- 이러한 선언은 Front End와 Back End 단계에서 사용하는 값과 방식에 차이가 있으며, 각각의 단계에서 적절한 설정이 필요하다.

# Constraints

## Timing Constraints

### Constraining Input Paths



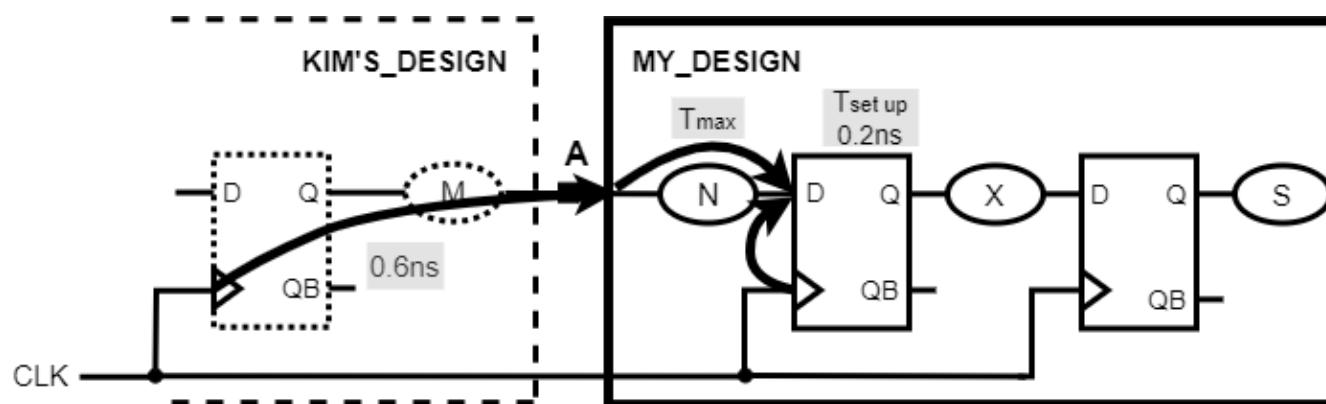
- Input path는 외부 신호가 칩 내부로 전달되는 경로를 의미한다.
- 외부 신호는 입력 핀을 통해 내부로 들어오며, 최종적으로 레지스터의 데이터 입력 핀(D 핀)까지 도달하게 된다

# Constraints

## Timing Constraints

### Constraining Input Paths: Example1

```
create_clock -period 2 [get_ports CLK]  
set_clock_uncertainty -setup 0.3 [get_clocks CLK]  
set_input_delay -max 0.6 -clock CLK [get_ports A]
```

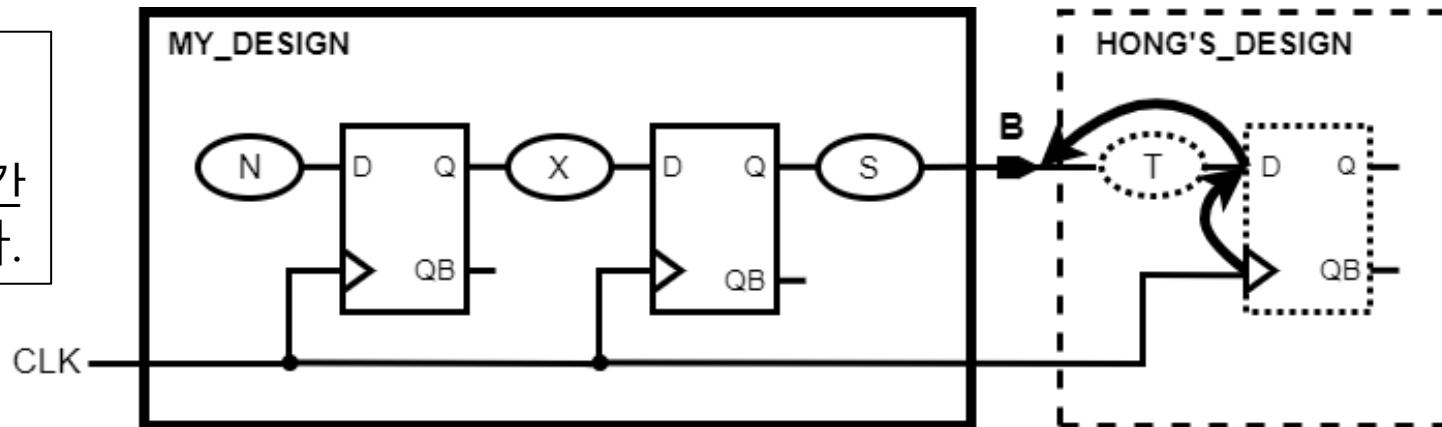


# Constraints

## Timing Constraints

### Constraining output Paths

- 외부 설계에서 소요되는 정확한 시간을 설계자가 직접 알 수 없으므로 시간 적 여유를 설정해야 한다.

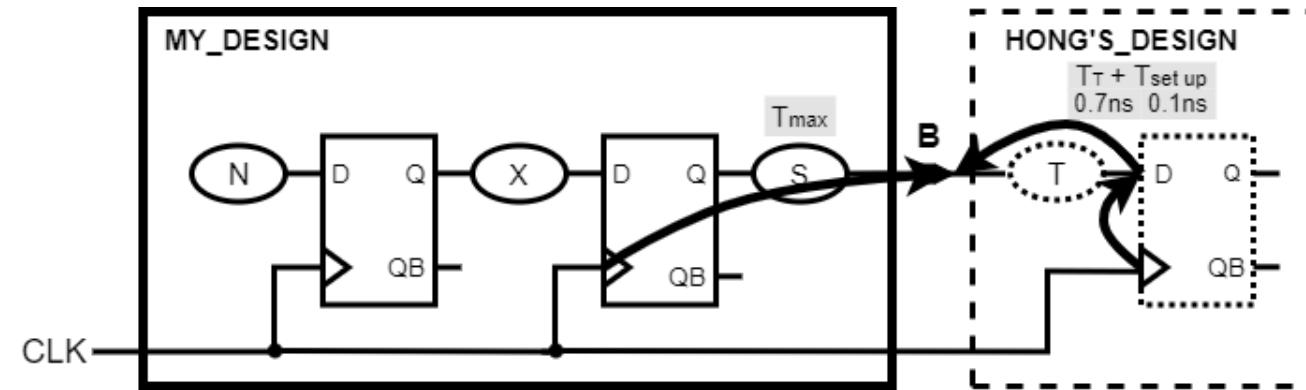


- Output path는 input path와 반대로 내부 신호가 외부 신호로 전달되는 경로이다.
- Input path와 마찬가지로 각 디자인들은 같은 클럭 신호를 공유하고 있기에 현재 디자인의 내부 레지스터의 출력부터 다음 디자인의 레지스터 입력까지 하나의 path를 형성하게 된다.

# Constraints

## Timing Constraints

### Constraining output Paths: Example 1



```
Genus> set_output_delay -max T -clock Clk [get_ports_B]  
: T만큼의 output delay를 준다.
```

# Constraints

## Virtual clk 실제 사용 예

- 현업에서는 Virtual clk을 자주 사용하고, 모든 input과 output의 타이밍의 기준으로 활용
  - 일반적인 디자인은 클럭의 종류가 많고, 입, 출력이 매우 많기 때문
  - 일일히 딜레이를 기술하기 어렵기 때문에 Vclk을 사용하게 됨

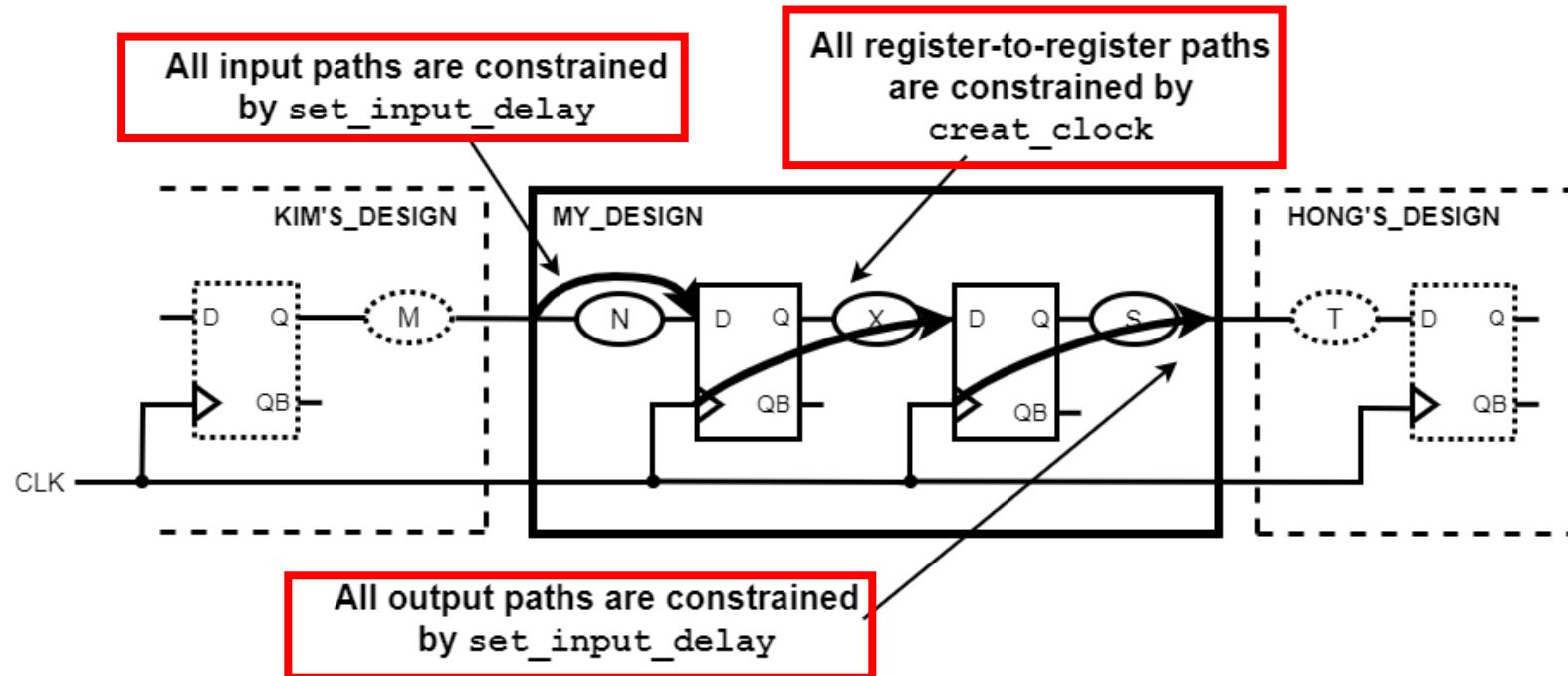
```
# Apply timing constraint
create_clock -period 10 -name CLK [get_port clk]

create_clock -period 10 -name VCLK
set_input_delay -max 6.0 -clock VCLK [get_port A]
set_output_delay -max 6.0 -clock VCLK [get_port Z]
```

# Constraints

## Timing Constraints

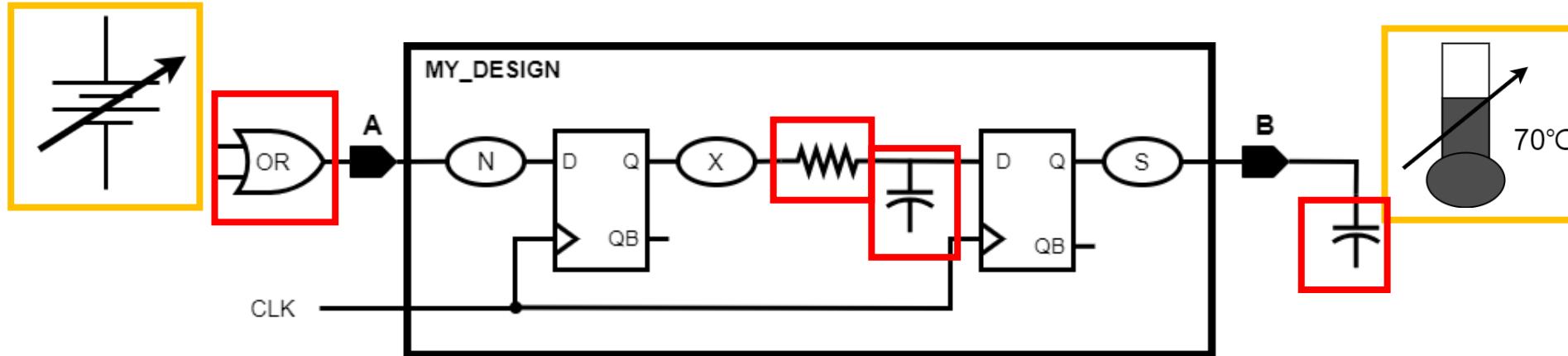
### Timing Constraint Summary



# Environmental Attributes

# Environmental Attributes

## Timing에 영향을 주는 요인들



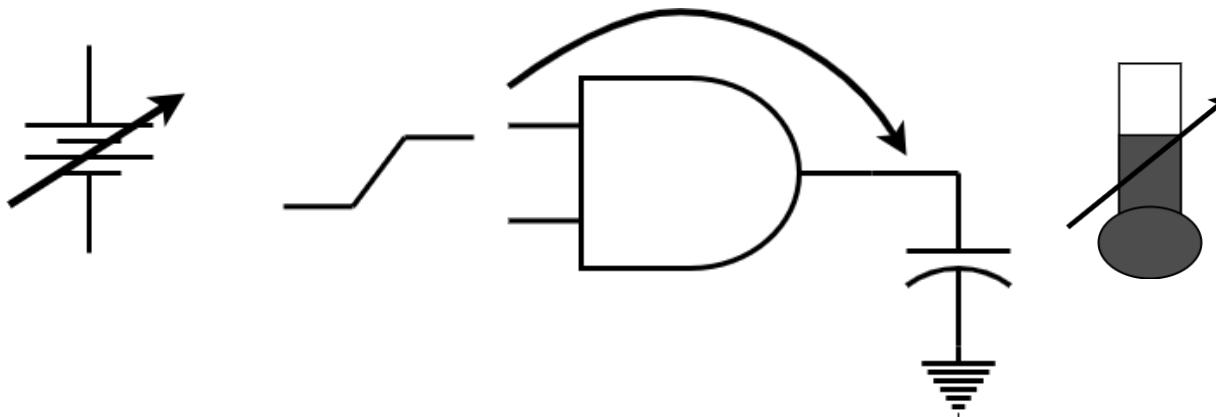
- 환경적 요인(Environmental Attributes)은 설계 및 검증 과정에서 타이밍과 성능에 영향을 미치는 물리적 요인들을 의미한다.
- 이러한 요인들은 설계 및 동작 환경에 항상 존재하기 때문에 신뢰성과 안정성을 위해 반드시 고려되어야 한다.
- 환경적 요인은 Cell Delay와 Net Delay가 있다. Cell Delay에는 PVT corner와 Output Load(출력 부하) 및 Input Transition(입력 기울기)이 있으며 Net Delay에는 Parasitic RC(기생 RC)가 있다.

# Environmental Attributes

## Cell Delay

### Cell Delay에 영향을 미치는 요인

높은 전압에서는 속도가 빨라지고 전력 소모가 증가하며, 낮은 전압에서는 속도가 느려지고 전력 소모가 감소한다



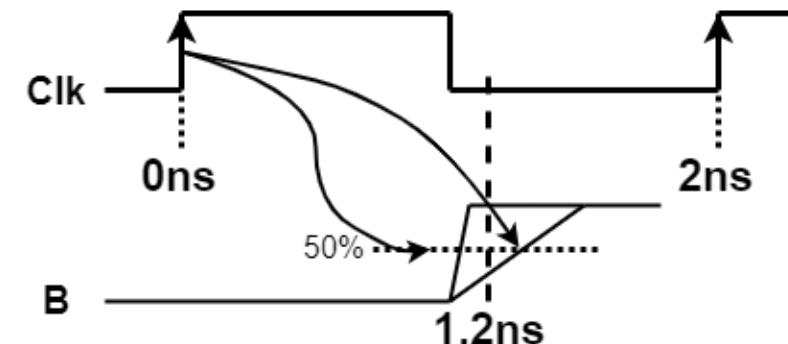
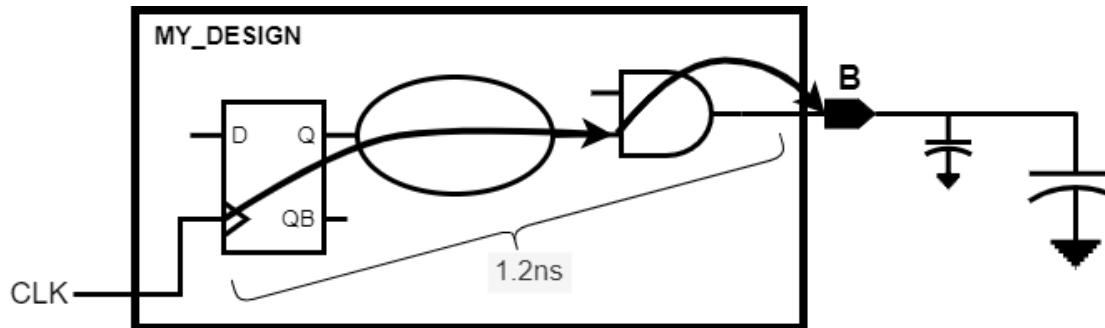
온도는 칩의 동작 성능에 영향을 준다. 일반적으로 반도체는 낮은 온도(-40°C)에서는 빠른 성능을 보이며, 높은 온도(125°C)에서는 느린 성능을 보인다.

- 셀 딜레이는 특정 셀이 신호를 처리하는 데 걸리는 시간으로 Input Transition 와 Output Load에 따라 달라진다.
- PVT(Process, Voltage, Temperature) 조건은 셀 딜레이에 큰 영향을 미친다.
- 칩 설계에 적용할 공정이 결정되면 해당 공정의 PVT특성을 파악하고 설계 환경에 적합한 PVT corner를 선택해야 한다.
- 결정된 PVT corner에 따라 공정에서 사용하는 소자들이 달라지게 된다.
- 일반적으로 PVT Corner는 온도와 전압에 따라 FF(Fast-Fast), SS(Slow-Slow), TT(Typical-Typical) 등으로 표현된다.

# Environmental Attributes

## Cell Delay

### Output 부하 용량(Capacitive Load)의 영향

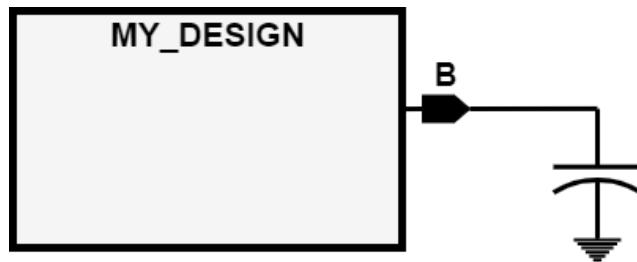


- Output load는 디자인의 출력 포트에 연결된 부하(Load) 용량으로 신호의 전달 속도와 타이밍에 영향을 준다.
- 출력 부하가 클수록 신호가 전달되는 속도가 느려지고 타이밍에 오류가 발생할 가능성이 높아진다.

# Environmental Attributes

## Cell Delay

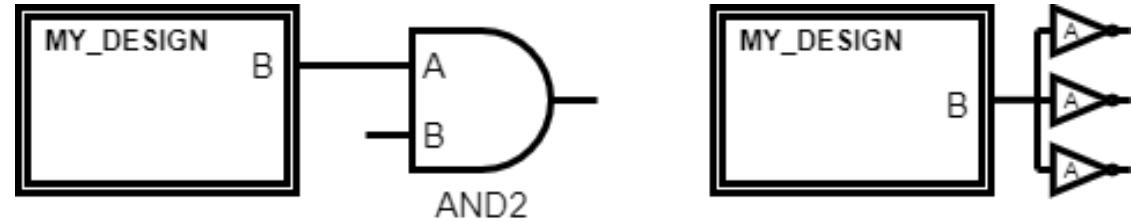
### Modeling Output Capacitive Load: Example1



# Environmental Attributes

## Cell Delay

### Modeling Output Capacitive Load: Example 2



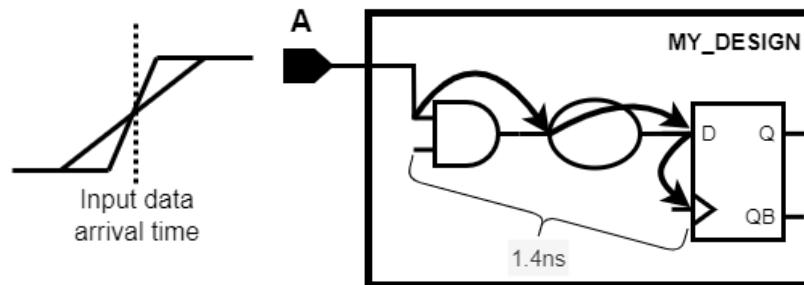
- set\_load 명령어를 사용하여 부하를 불러올 라이브러리 이름과 셀의 이름을 선언하여 포트와 연결한다.
- 최대한 pessimistic한 조건을 부여해야 하므로 큰 셀을 사용하는 것이 일반적이다.

```
Genus> set_load [load_of basciCells/BUFX20/A] [get_ports_B]
```

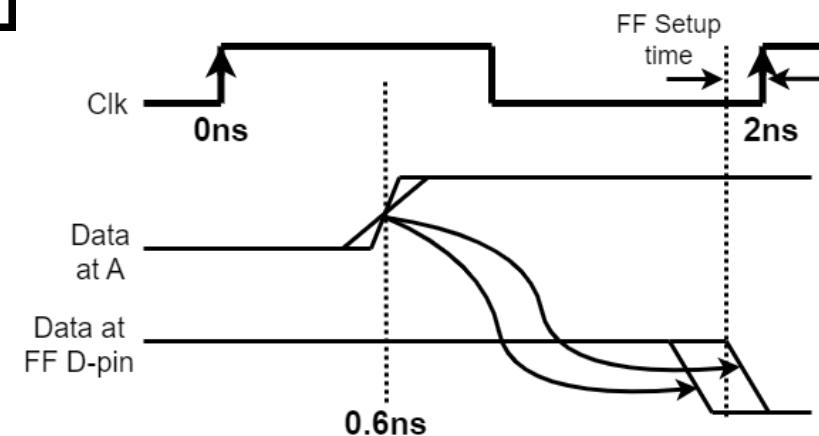
# Environmental Attributes

## Cell Delay

### Input Transition Time 영향



- Input transition은 설계의 입력 포트에 들어오는 신호의 기울기(Slew Rate)를 의미한다.
- Slew Rate는 입력 신호가 0에서 1로 상승하거나 1에서 0으로 하강할 때 소요되는 시간이다.

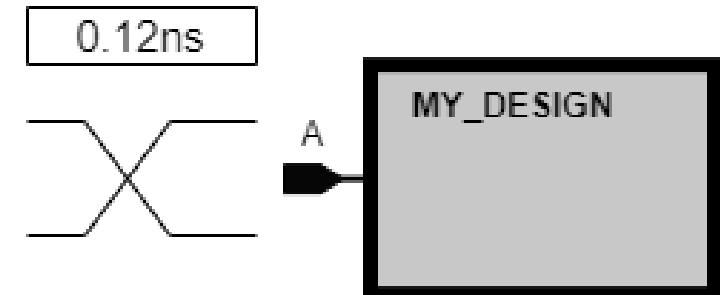


# Environmental Attributes

## Cell Delay

### Modeling Input Transition: Example 1

```
Genus> set_driving_cell PADDI -pin Y -library [get_libs $STD_LIB]
```

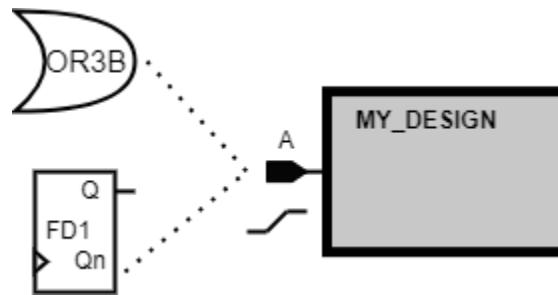


- Transition으로 발생하는 지연시간은 Output load와 마찬가지로 입력 포트에 걸리는 transition 수치를 직접 설정하는 방법과 라이브러리 내의 특정 셀이 신호를 보내주고 있다고 가정하는 방법이 있다.
- 일반적으로 입력포트에 특정 셀이 있다고 가정하는 경우는 다음과 같은 명령어를 사용한다. 이런 방법을 Driving Cell이라고 한다.

# Environmental Attributes

## Cell Delay

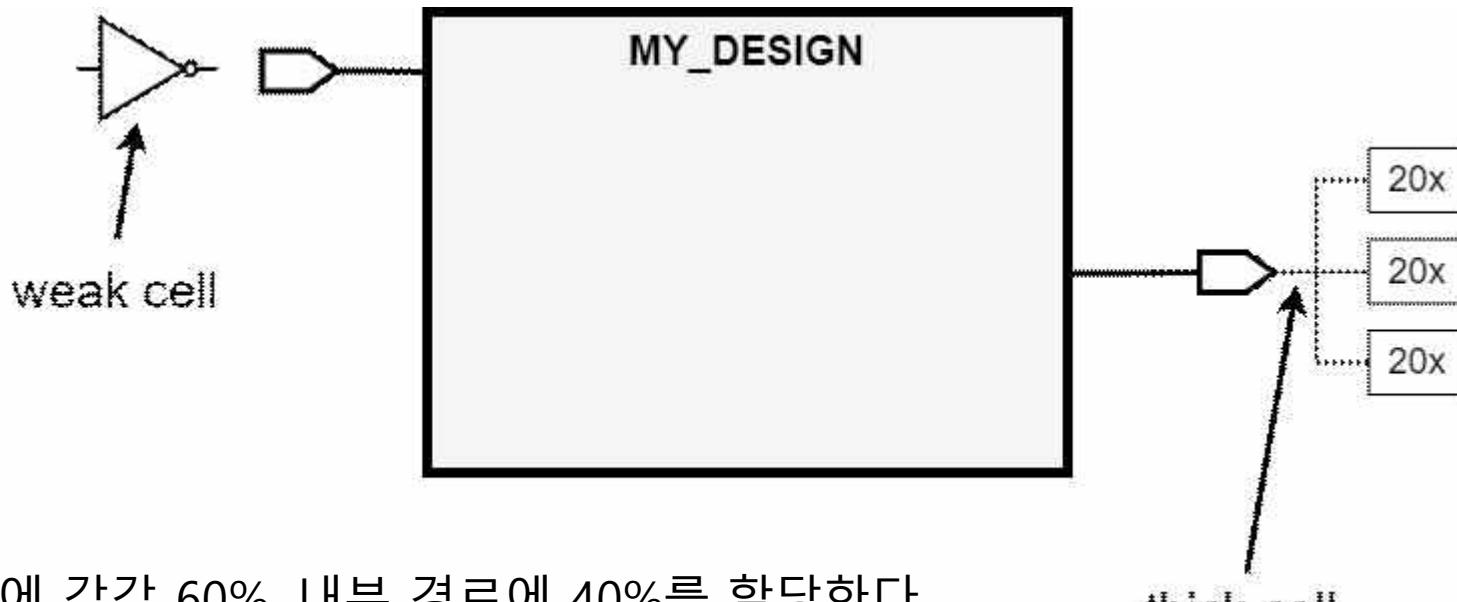
### Modeling Input Transition: Example 2



- Load Budgeting은 Constraint 과정에서 입력과 출력 경로에 타이밍적인 여유를 뒀 신호의 안정성 확보하는 기법이다.
- Load Budgeting는 설계 목표를 만족하기 위해 경로마다 확 보해야 하는 시간적 마진으로, 이를 효율적으로 분배하지 않으면 타이밍 위반이 발생할 수 있다.

# Environmental Attributes

## Load Budgeting



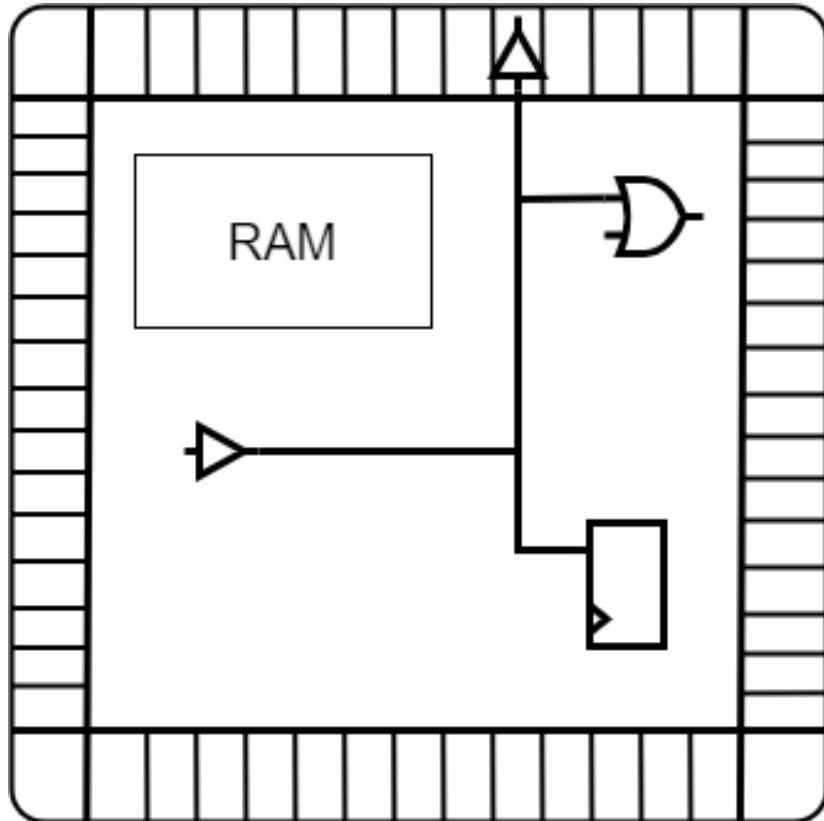
- 보통 입력과 출력에 각각 60%, 내부 경로에 40%를 할당한다.
- Load Budgeting 적용 시 입력 경로에서는 약한 셀(Weak Cell)이 신호를 드라이빙한다고 가정하고 출력 경로에서는 큰 부하가 걸린다고 가정한다.

# Environmental Attributes

## Net Delay

Path Delay 는  
Cell Delay 와 Net Delay 의 합  
본 페이지는 Net Delay 만 설명

### Modeling Interconnect or Net Parasitics

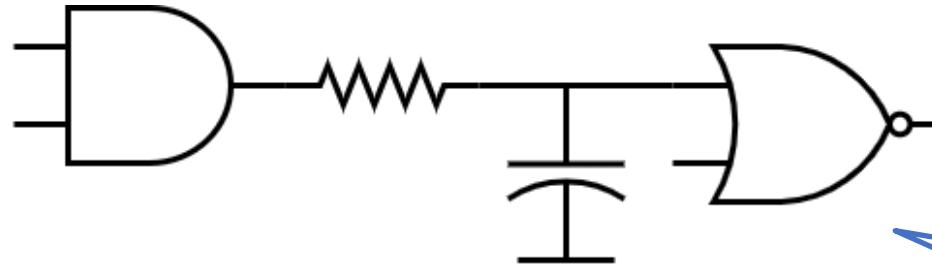


- Net Delay는 배선에 존재하는 저항(Resistance)과 커패시턴스(Capacitance)에 의해 발생하며, 이를 Parasitic RC라고 한다.

# Environmental Attributes

## Net Delay

### Modeling Net RCs with Wire Load Models



Net 딜레이는 **RC** 값으로 구현됨  
공정사에서 **estimation** 한 값에서 기반함  
→ 딜레이 정보는 **WLM**에 담겨 있음

- RC 지역 효과를 효율적으로 분석하고 모델링하기 위해 Wire Load Model(WLM)을 사용한다.
- WLM은 RC 값을 기반으로 Net Delay를 예측하며 공정사에서 제공하는 .lib 파일에 저장되어 있다.
- 설계 툴은 설계하는 면적에 따른 WLM 정보를 자동으로 선택한다.

# Environmental Attributes

Net De

## Wire Load Model Examples

아래 정보는  
list\_libs 후에 report\_lib 명령으로 확인 가능

What does this mean???

Name	:	140000
Location	:	90nm_com
Resistance	:	0.000331
Capacitance	:	8.6e-05
Area	:	0.1
Slope	:	93.7215

Fanout	Length
1	14.15
2	32.31
3	52.48
4	74.91
.	.
.	.
20	952.16

kΩ/unit length  
pF/unit length  
area/unit length  
Extrapolation slope

Name	:	8000000
Location	:	90nm_com
Resistance	:	0.000331
Capacitance	:	8.6e-05
Area	:	0.1
Slope	:	334.957

Fanout	Length
1	24.58
2	58.28
3	98.54
4	146.54
.	.
.	.
20	2946.37

Ask your library provider how to  
select the appropriate model!!

# Environmental Attributes

Net

## Specifying Wire Loads in Design Compiler

- Manual selection: `set_wire_load_model -name 8000000`
- *Automatic model selection* selects an appropriate wireload model during compile:

(enabled by default if selection table available in library)

Selection		Wire load name
min area	max area	
<hr/>		
0.00	43478.00	140000
43478.00	86956.00	280000
86956.00	173913.00	560000
173913.00	347826.00	1000000
<hr/>		
...		

- To override automatic selection and select manually:

```
set auto_wire_load_selection false  
set_wire_load_model -name 8000000
```

# Environmental Attributes

## Net Delay

### Wire Load Model vs 'Topographical Mode'

- Wire load models (WLMs) are based on statistical averages and are not specific to your design

WLM 은 통계적이지만  
유저의 디자인에는 **specific** 하지 않음
- In Ultra Deep Sub-Micron (UDSM) designs the interconnect parasitics have a major impact on path delays → need accurate RC estimates

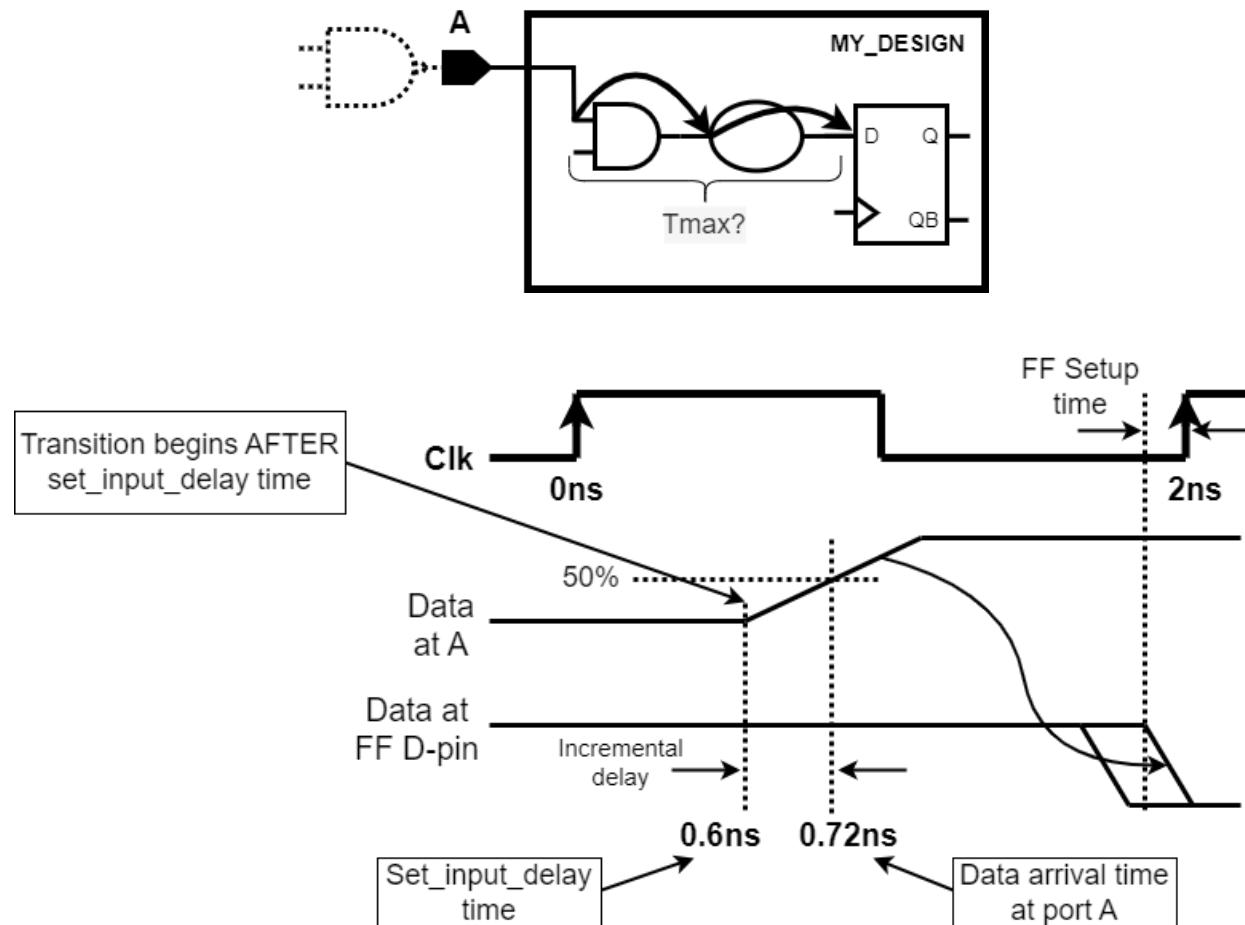
UDSM 디자인에서는  
**interconnect parasitic**이 매우 중요  
아직 공정사에서 지원하지 못함
- For UDSM designs WLMs are not adequate: It is recommended to use DC's 'Topographical Mode' (discussed in a later Unit)
  - Topographical mode eliminates the use of WLMs
  - Uses placement algorithms under-the-hood to estimate wire lengths and parasitics
  - Provides much better timing correlation to that of the actual physical layout

**Topographical Mode**는  
**Place** 알고리즘을 통해서  
좀 더 실제적인 **net** 정보를 갖고 합성하도록 함

# More Constraint Considerations

# More Constraint Considerations

## Effect of Driving Cell on Input Delay



# More Constraint Considerations

## Set\_driving\_cell Recommendation

If you are confident that your design constraint specs are accurate, and you want to model your input data arrival time with precision, make sure that:

1. The `set_input_delay` number you apply is based on zero output load on Jane's block (the intrinsic delay to Jane's output port)
2. The `set_driving_cell` gate matches Jane's output driver

If not, your input constraints will include some built-in pessimism.

Spec:

1. **Zero output load**를 가정하고 지정
2. **Set\_driving\_cell**로 실제를 지정

Latest Data Arrival Time at Port A, after Jane's launching clock:

Use this number!

0.60ns, with 50fF load  
0.48ns, with 0.0fF load

Driving cell on input port A:

Qn pin of FD1 flip-flop

```
create_clock -period 2 [get_ports Clk]  
set_input_delay -max 0.48 -clock Clk [get_ports A]  
set_driving_cell -lib_cell FD1 -pin Qn [get_ports A]
```

Constraint  
cons 폴더 확인

# Synthesis

## genus

- 합성에 사용할 Constraint가 담긴 cons 폴더 확인

\$> cd cons

```
/home/ex_poly1/SoC2/SoC/smkcowl_make_S28/TOP/SYN
```

```
clean.tcl cons fv log mapped report run_synthesis script unmapped
```

# Synthesis

## genus

- cmsdk\_mcu.sdc 파일 내용 확인

```
$> vi cmsdk_mcu.sdc
```

```
[ex_poly1@npit cons]$ ls  
cmsdk_mcu.sdc dont_use_45nm.tcl
```

# Synthesis

## genus

- cons 파일 내용 확인
- 13행: STD\_LIB 중에 더 중점적으로 사용할 .lib 파일
- 15, 16행: 기본적인 단위 설정

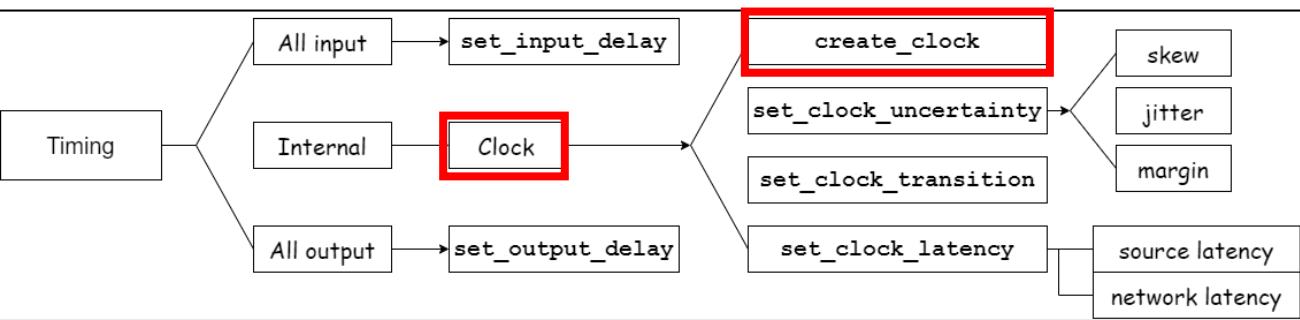
```
1 #####  
2  
3 # Created by write_sdc on Sat Mar 15 14:01:37 2014  
4  
5 #####  
6 # STD_LIB slow_vdd1v0_basicCells.lib:slow_vdd1v0  
7 # operating_conditions (PVT_0P9V_125C) {  
8 #   process : 1;  
9 #   temperature : 125;  
10 #   voltage : 0.9;  
11 # }  
12  
13 set STD_LIB slow_vdd1v0_basicCells.lib:slow_vdd1v0  
14  
15 set_units -capacitance 1000.0fF  
16 set_units -time 1000.0ps
```

Operating Condition 중 worst 상황인 slow 파일 사용

# Synthesis

## genus

- timing 내용 확인



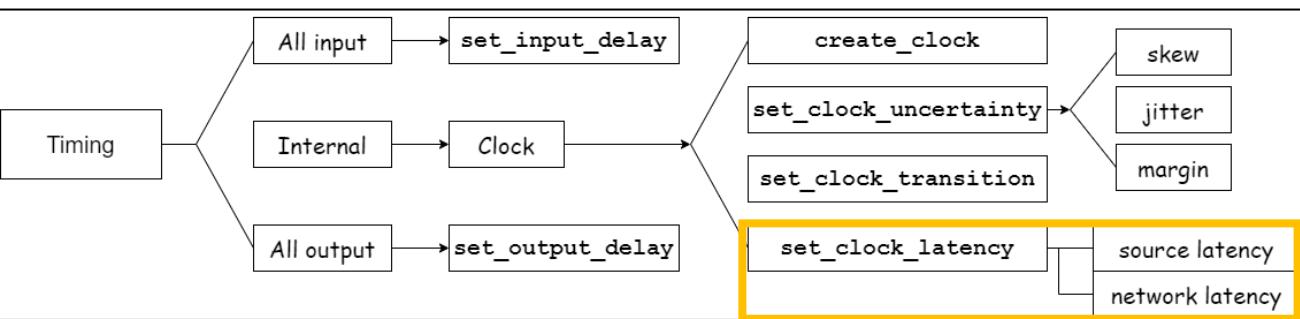
10ns를 사용하지만 목표는  
7ns로 더 어렵게 설정함

```
18 #####; #####; #####
19 reset_design
20 #####
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]
22
23 # External clock source and network latency is 0.2ns
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]
26
27 # 60ps skew (+300ps and -300ps)
28 # 20ps jitter
29 # 20ps setup margin;
30 # This equals 0.1 ns of total uncertainty.
31 set_clock_uncertainty -setup 0.1 [get_clocks MAIN_CLOCK]
32
33 # The maximum clock transition is 0.1ns
34 set_clock_transition 0.1 [get_clocks MAIN_CLOCK]
35
36 #####
37 # REFERENCE
38 #set_clock_latency -source -max 0.3 [all_clocks]
39 #set_clock_latency -max 0.2 [all_clocks]
40 #set_clock_uncertainty -setup 0.3 [all_clocks]
41 #set_clock_uncertainty -hold 0.2 [all_clocks]
42 #set_clock_transition -max -rise 0.3 [all_clocks]
43 #set_clock_transition -max -fall 0.3 [all_clocks]
44 #set_clock_transition -min -rise 0.2 [all_clocks]
45 #set_clock_transition -min -fall 0.2 [all_clocks]
```

# Synthesis

genus

- timing 내용 확인

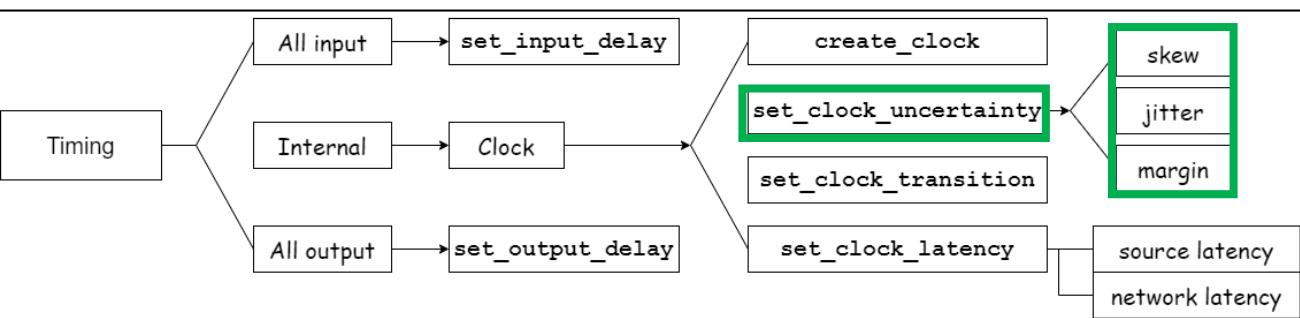


```
18 #####  
19 reset_design  
20 #####  
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]  
22  
23 # External clock source and network latency is 0.2ns  
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]  
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]  
26  
27 # 60ps skew (+300ps and -300ps)  
28 # 20ps jitter  
29 # 20ps setup margin;  
30 # This equals 0.1 ns of total uncertainty.  
31 set_clock_uncertainty -setup 0.1 [get_clocks MAIN_CLOCK]  
32  
33 # The maximum clock transition is 0.1ns  
34 set_clock_transition 0.1 [get_clocks MAIN_CLOCK]  
35  
36 #####  
37 # REFERENCE  
38 #set_clock_latency -source -max 0.3 [all_clocks]  
39 #set_clock_latency -max 0.2 [all_clocks]  
40 #set_clock_uncertainty -setup 0.3 [all_clocks]  
41 #set_clock_uncertainty -hold 0.2 [all_clocks]  
42 #set_clock_transition -max -rise 0.3 [all_clocks]  
43 #set_clock_transition -max -fall 0.3 [all_clocks]  
44 #set_clock_transition -min -rise 0.2 [all_clocks]  
45 #set_clock_transition -min -fall 0.2 [all_clocks]
```

# Synthesis

## genus

- timing 내용 확인

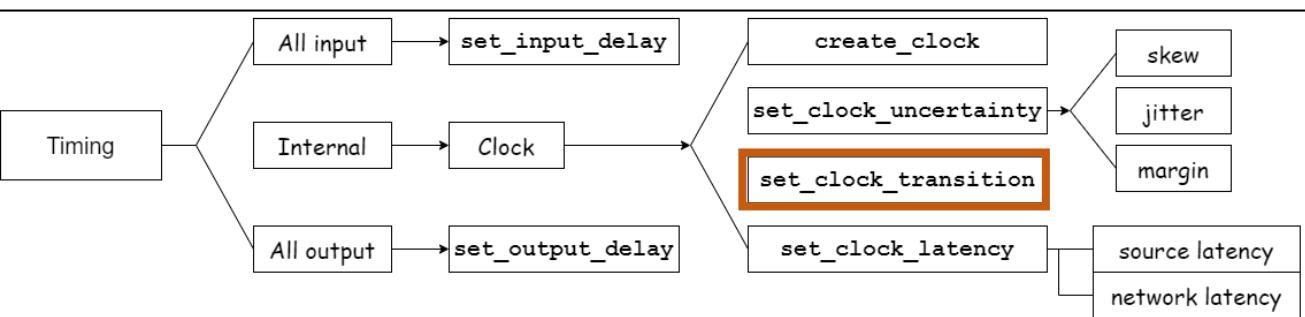


```
18 #####  
19 reset_design  
20 #####  
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]  
22  
23 # External clock source and network latency is 0.2ns  
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]  
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]  
26  
27 # 60ps skew (+300ps and -300ps)  
28 # 20ps jitter  
29 # 20ps setup margin;  
30 # This equals 0.1 ns of total uncertainty.  
31 set_clock_uncertainty -setup 0.1 [get_clocks MAIN_CLOCK]  
32  
33 # The maximum clock transition is 0.1ns  
34 set_clock_transition 0.1 [get_clocks MAIN_CLOCK]  
35  
36 #####  
37 # REFERENCE  
38 #set_clock_latency -source -max 0.3 [all_clocks]  
39 #set_clock_latency -max 0.2 [all_clocks]  
40 #set_clock_uncertainty -setup 0.3 [all_clocks]  
41 #set_clock_uncertainty -hold 0.2 [all_clocks]  
42 #set_clock_transition -max -rise 0.3 [all_clocks]  
43 #set_clock_transition -max -fall 0.3 [all_clocks]  
44 #set_clock_transition -min -rise 0.2 [all_clocks]  
45 #set_clock_transition -min -fall 0.2 [all_clocks]
```

# Synthesis

genus

- timing 내용 확인

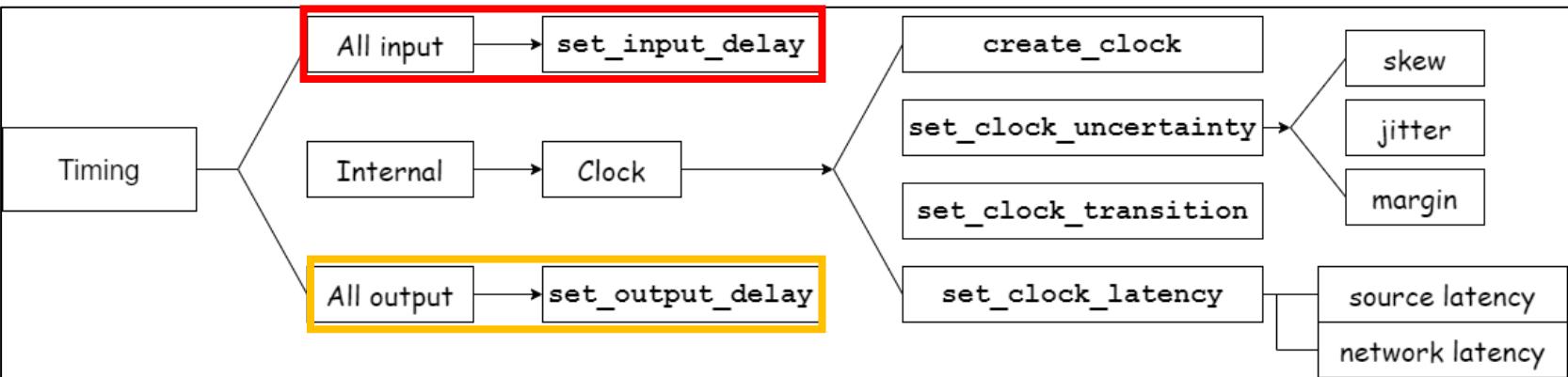


```
18 #####  
19 reset_design  
20 #####  
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]  
22  
23 # External clock source and network latency is 0.2ns  
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]  
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]  
26  
27 # 60ps skew (+300ps and -300ps)  
28 # 20ps jitter  
29 # 20ps setup margin;  
30 # This equals 0.1 ns of total uncertainty.  
31 set_clock_uncertainty -setup 0.1 [get_clocks MAIN_CLOCK]  
32  
33 # The maximum clock transition is 0.1ns  
34 set_clock_transition 0.1 [get_clocks MAIN_CLOCK]  
35  
36 #####  
37 # REFERENCE  
38 #set_clock_latency -source -max 0.3 [all_clocks]  
39 #set_clock_latency -max 0.2 [all_clocks]  
40 #set_clock_uncertainty -setup 0.3 [all_clocks]  
41 #set_clock_uncertainty -hold 0.2 [all_clocks]  
42 #set_clock_transition -max -rise 0.3 [all_clocks]  
43 #set_clock_transition -max -fall 0.3 [all_clocks]  
44 #set_clock_transition -min -rise 0.2 [all_clocks]  
45 #set_clock_transition -min -fall 0.2 [all_clocks]
```

# Synthesis

genus

- timing 내용 확인
- 62행: set\_max\_area 명령은 cadence에서 사용 X



```
47 #####  
48 ##### input, output delay #####  
49 #####  
50 set_input_delay -max 0.1 -clock MAIN_CLOCK [all_inputs]  
51 set_input_delay -min 0.01 -clock MAIN_CLOCK [all_inputs]  
52  
53 remove_input_delay [get_ports "XTAL1"]  
54  
55 set_output_delay -max 0.1 -clock MAIN_CLOCK [all_outputs]  
56 set_output_delay -min -0.01 -clock MAIN_CLOCK [all_outputs]  
57  
58 #####  
59 # DESIGN AREA #  
60 #####  
61 # Below is no use in Genus tool  
62 #set_max_area 0
```

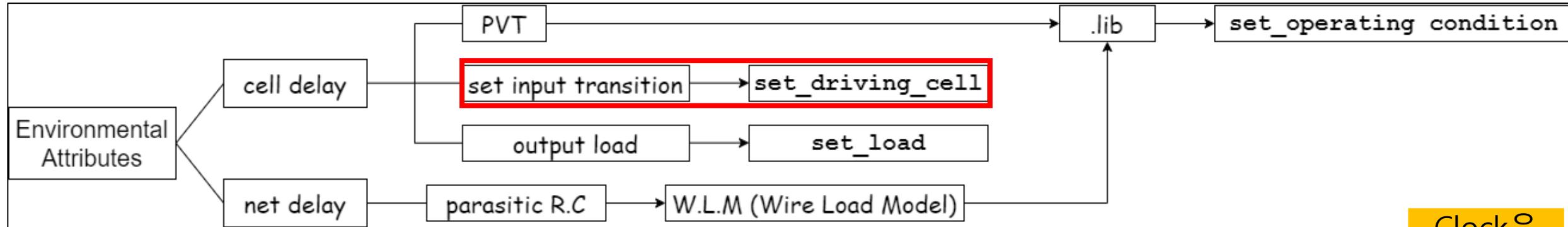
-max: setup time  
-min: hold time

Clock은 제외시킴

# Synthesis

## genus

- Environmental Attributes 내용 확인



```
64 #####  
65 ##### Select this point! #####  
66 #####  
67 #set_driving_cell -lib_cell DFFHQX1 -pin Q [remove_from_collection [all_inputs] [get_ports clk]]  
68  
69 set_driving_cell -lib_cell PADDI -pin Y -library [get_libs $STD_LIB] [remove_from_collection [all_inputs] [get_ports XTAL1]]  
70  
71 #set_load [load_of $STD_LIB/PADD0/A] [get_ports {TDO XTAL2}]  
72 set_load [load_of PADD0/A] [all_outputs]  
73  
74 #####  
75 ##### Operating Condition #####  
76 #####  
77  
78 set_operating_conditions PVT_0P9V_125C -library [get_libs $STD_LIB] -analysis_type single
```

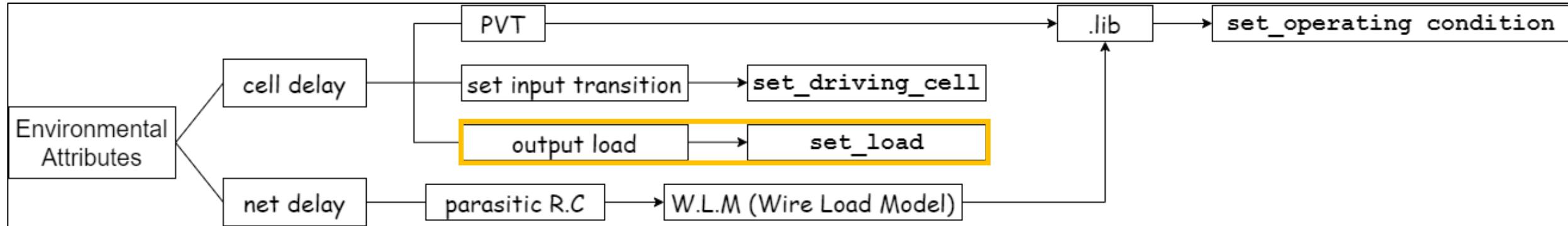
Clock은 제외시킴

서브 모듈에서는 set\_input\_transition에 weak한 셀을 사용하고 output\_load에는 무거운 셀을 사용하지만 io셀이 포함된 SoC에는 실제 input과 output에 사용하는 io셀을 사용한다.

# Synthesis

## genus

- Environmental Attributes 내용 확인



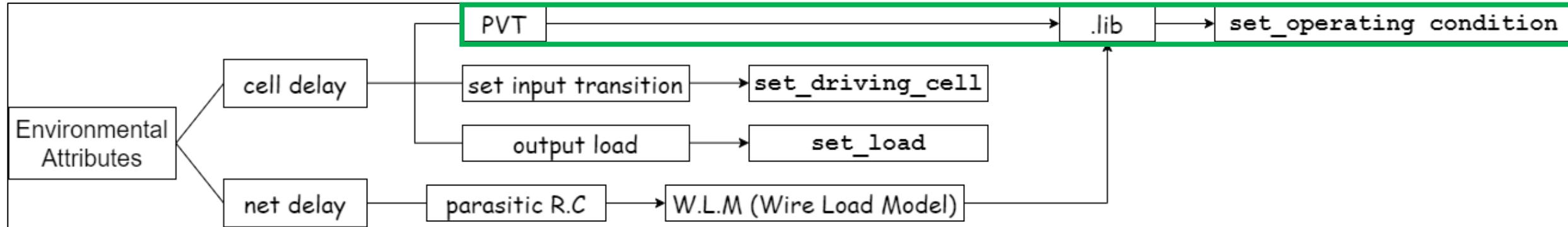
```
64 #####
65 ##### Select this point! #####
66 #####
67 #set_driving_cell -lib_cell DFFHQX1 -pin Q [remove_from_collection [all_inputs] [get_ports clk]]
68
69 set_driving_cell -lib_cell PADDI -pin Y -library [get_libs $STD_LIB] [remove_from_collection [all_inputs] [get_ports XTAL1]]
70
71 #set_load [load_of $STD_LIB/PADD0/A1] [get_ports {TDO XTAL2}]
72 set_load [load_of PADD0/A] [all_outputs]
73
74 #####
75 ##### Operating Condition #####
76 #####
77
78 set_operating_conditions PVT_0P9V_125C -library [get_libs $STD_LIB] -analysis_type single
```

서브 모듈에서는 set\_input\_transition에 weak한 셀을 사용하고 output\_load  
에는 무거운 셀을 사용하지만  
io셀이 포함된 SoC에는 실제 input과 output에 사용하는 io셀을 사용한다.

# Synthesis

## genus

- Environmental Attributes 내용 확인



```
64 #####
65 ##### Select this point! #####
66 #####
67 #set_driving_cell -lib_cell DFFHQX1 -pin Q [remove_from_collection [all_inputs] [get_ports clk]]
68
69 set_driving_cell -lib_cell PADDI -pin Y -library [get_libs $STD_LIB] [remove_from_collection [all_inputs] [get_ports XTAL1]]
70
71 #set_load [load_of $STD_LIB/PADD0/A] [get_ports {TDO XTAL2}]
72 set_load [load_of PADD0/A] [all_outputs]
73
74 #####
75 ##### Operating Condition #####
76 #####
77
78 set_operating_conditions PVT_0P9V_125C -library [get_libs $STD_LIB] -analysis_type single
```

# Synthesis

## genus

- cons 파일 내용 확인

```
81 #####  
82 #  
83 # ETC ATTRIBUTES  
84 #  
85 #####  
86  
87 set_dont_touch u_cmsdk_mcu_system  
88  
89 # Below is no use in Genus tool  
90 #set_isolate_ports [all_outputs]  
91  
92 set_ideal_network [get_ports "XTAL1 NRST"]  
93  
94 # no through pass assign syntax => insert buffers  
95 # Below is no use in Genus tool  
96 #set_fix_multiple_port_nets -all -buffer_constants -feedthroughs -constants
```

미리 합성한 뒤 불러온  
모듈은 건드리지 않도록  
하는 명령임

Clock과 reset에 대해서는  
ideal한 network로 처리함

Ideal한 network에는  
버퍼를 삽입하지 않음

# Synthesis

## genus

- don't\_use\_45nm.tcl 파일 내용 확인

```
$> vi don't_use_45nm.tcl
```

```
[ex_poly1@npit cons]$ ls  
cmsdk_mcu.sdc dont_use_45nm.tcl
```

# Synthesis

## genus

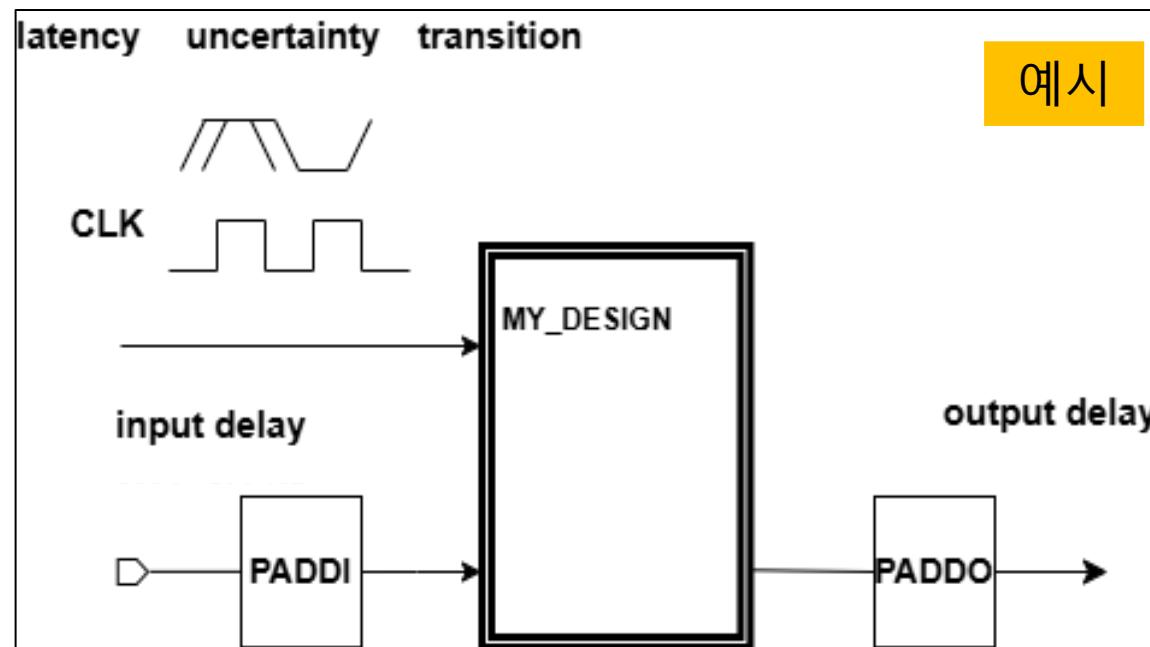
- don't\_use\_45nm.tcl 파일은 주로 Back-End에서 사용할 셀들을 기재함

```
1 set_dont_use [get_lib_cells */ANTENNA*]
2 set_dont_use [get_lib_cells */DLY*]
3 #set_dont_use [get_lib_cells */FILL*]
4 set_dont_use [get_lib_cells */HOLD*]
5 set_dont_use [get_lib_cells */TIE*]
6 set_dont_use [get_lib_cells */TLA*]
7 set_dont_use [get_lib_cells */DFFS*]
8 set_dont_use [get_lib_cells */SDFF*]
```

# Synthesis

## genus

- all\_input, all\_output, internal, cell\_delay의 정보들을 constraints 파일에서 찾아내 이를 토대로 책에 그려져 있는 my design을 참고하여 그려볼 것
- 각각의 조건들이 어떻게 적용되어 있는지 파악해 볼 것

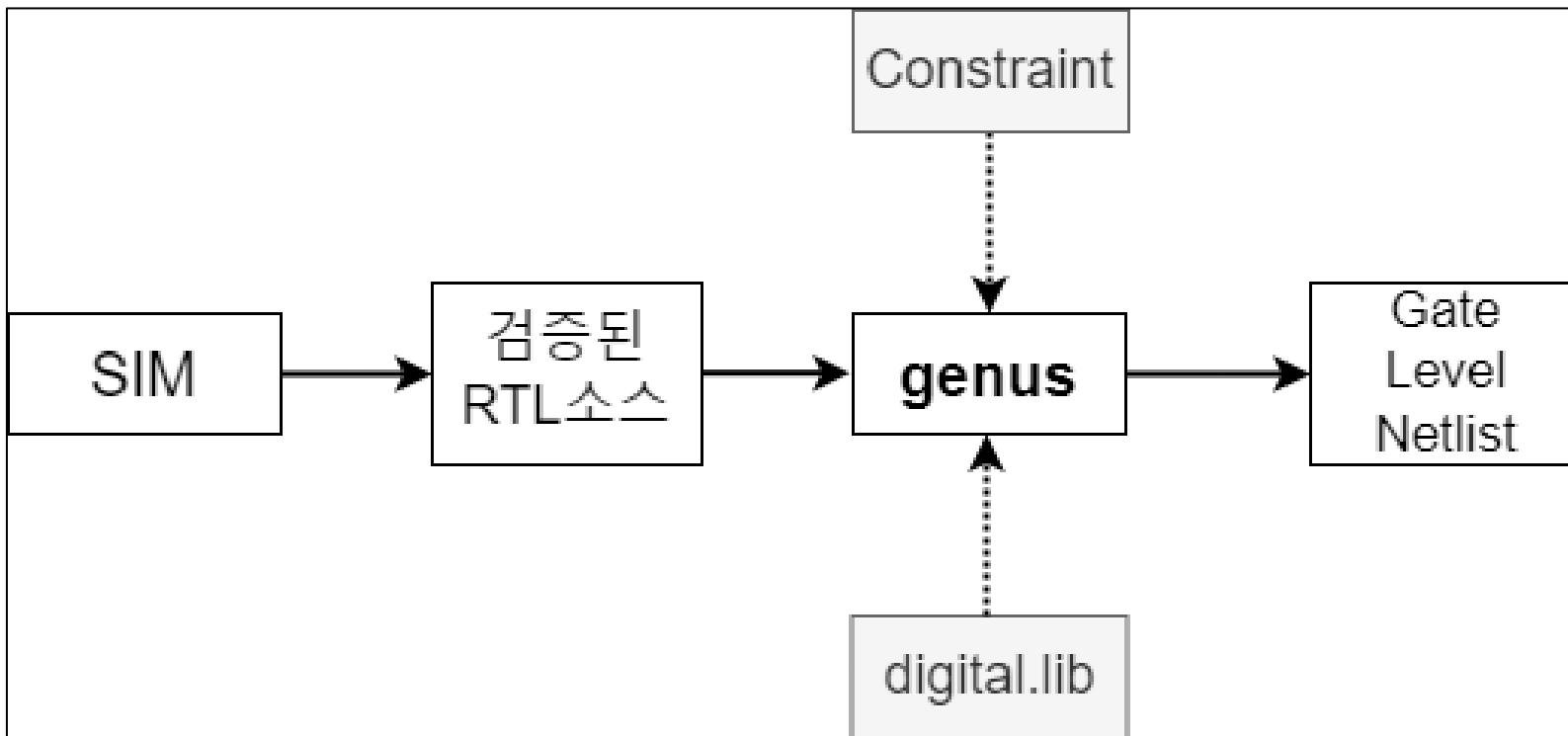


genus

# Synthesis

## genus란?

- 시뮬레이션을 통해 검증된 RTL소스를 합성하여 G.L.N(Gate Level Netlist)를 만들어내는 툴
- 합성 과정에서 Constraint 목표를 만족해야 하며 공정사에서 제공한 라이브러리(.lib)를 사용함



# Synthesis

## genus

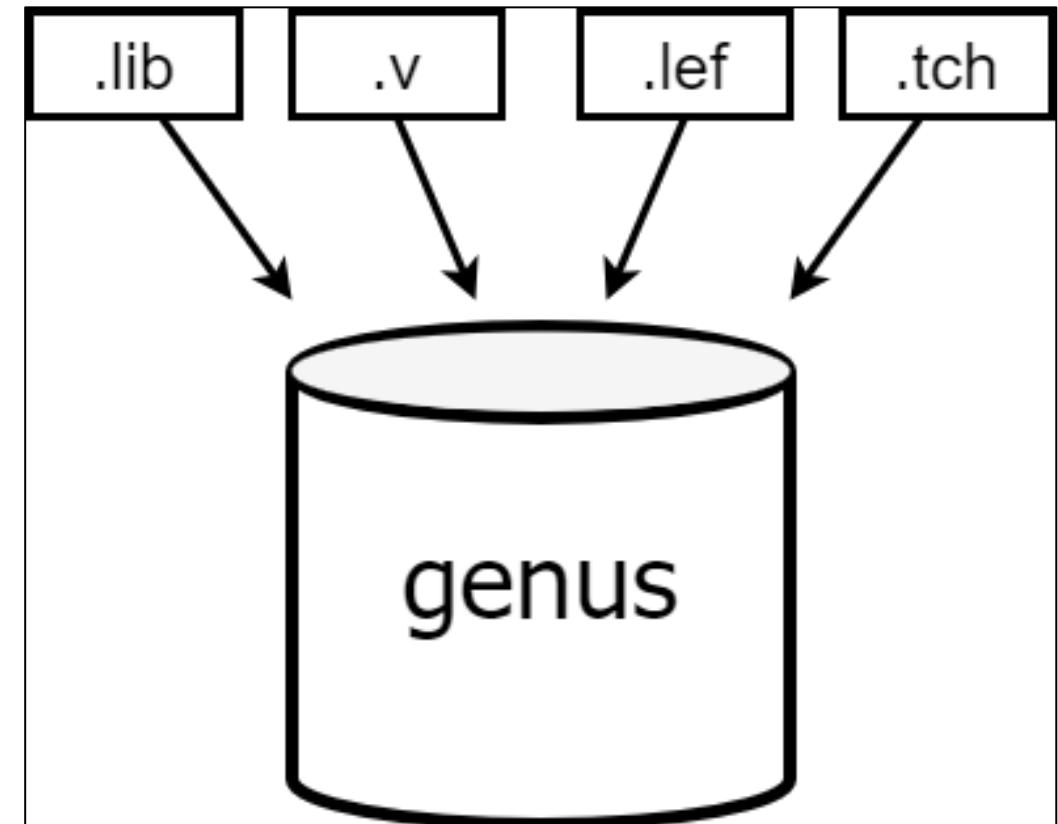
- 합성에 필요한 확장자 파일

**.lib:** STD셀과 IO셀 등이 있는 공정사의 라이브러리 파일

**.v:** 디자인의 Verilog 소스코드가 적혀 있는 파일

**.lef:** STD셀의 프레임 셀

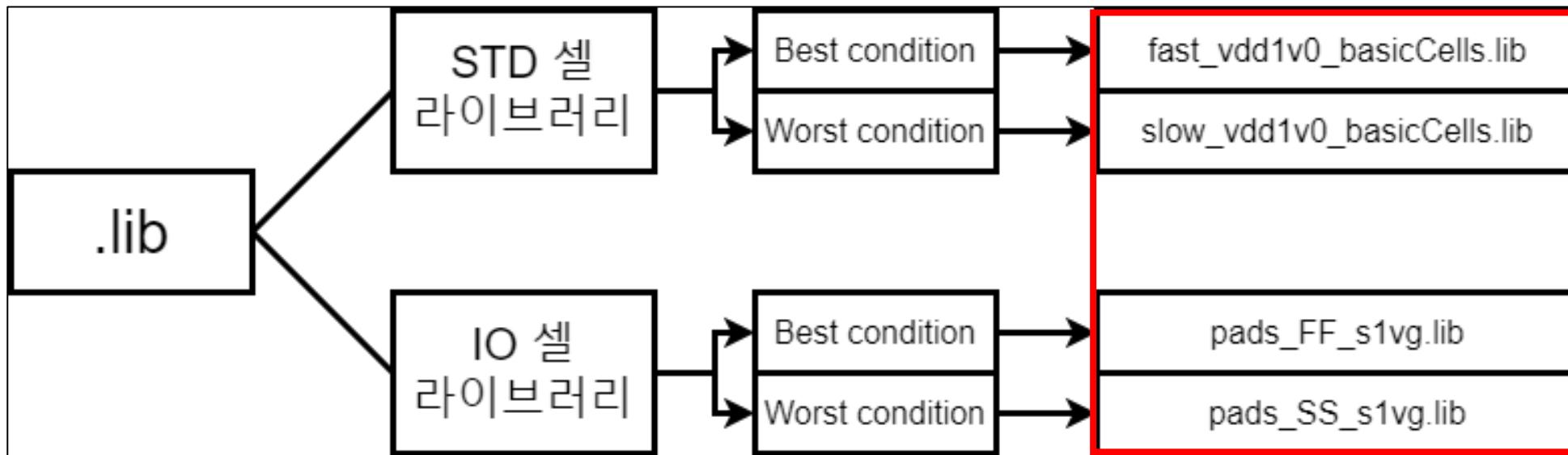
**.tch:** 기생 성분을 담고 있는 파일



# Synthesis

genus

Genus 툴에 사용할 4개의 .lib  
파일을 결정

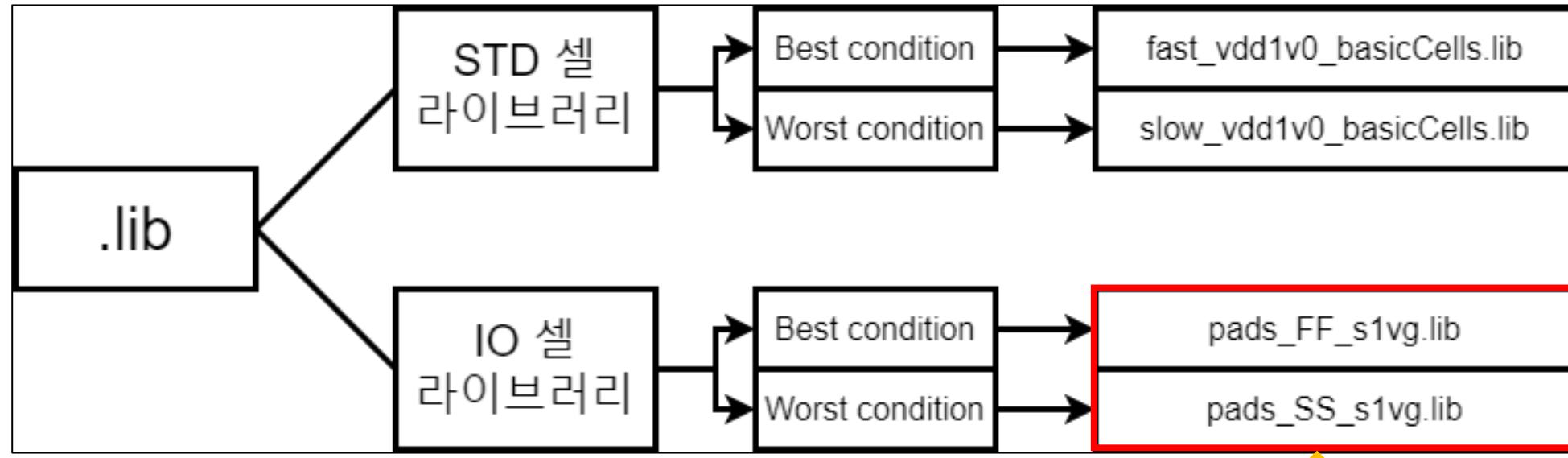


set\_db 명령어를 통해 라이브러리의 검색 경로를 정의함

`set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]`

# Synthesis

genus



**충돌 이슈 발생:** IO셀 .lib 파일의 라이브러리 내용을 보면 이름이 겹쳐 충돌하는 문제가 있음

**해결 제안 :** IO셀 라이브러리 파일은 하나만 기입

# Synthesis

genus

- IO셀 충돌 이슈 확인 방법

```
/home/ex_poly1/SoC2/GPDK045/digital/giolib045_v3.5/timing
```

pads\_FF\_s1vg.lib    pads\_SS\_s1vg.lib    pads\_TT\_s1vg.lib

vi 진행 방법

```
$> vi pads_FF_s1vg.lib
```

```
: vs
```

```
: e .
```

```
pads_SS_s1vg.lib -> Enter
```

```
2 delay model : typ
3 check model : typ
4 power model : typ
5 capacitance model : typ
6 other model : typ
7 */
8 library(giolib045) {
9
```

라이브러리 이름이 같으므로 충돌 발생

해결 제안 : IO셀 라이브러리 파일은 하나만 기입

```
29 set LIB_LIST {fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib
pads_SS_s1vg.lib}
```

# Synthesis

genus

\$> genus

genus> gui\_show

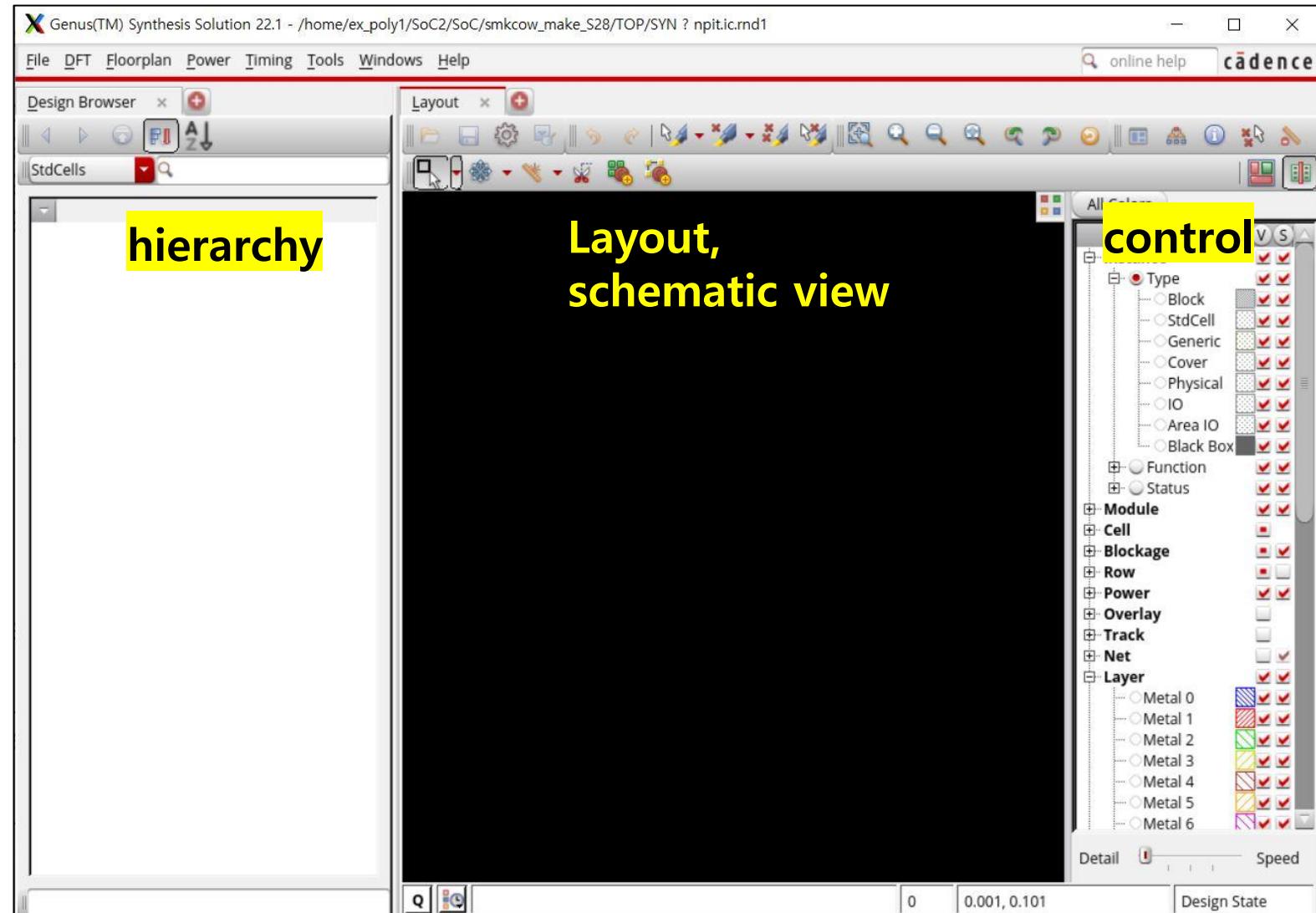
[ex\_poly1@npit SYN]\$ genus

@genus:root: 1> gui\_show

# Synthesis

genus

- Gui 확인



# Synthesis

genus

\$> vi run\_synthesis

/home/ex\_poly1/SoC2/SoC/smkcw\_make\_S28/TOP/SYN

clean.tcl cons fv log mapped report **run\_synthesis** script unmapped

자동 배치모드로 합성을  
진행하기 위한  
실행 파일

# Synthesis

## genus

- 2행 cortexm0\_45nm.tcl 실행할 것임

-f : 자동으로  
script 실행

-log : 툴 실행부터  
종료까지의 전체  
log를 저장

```
1 #!/bin/csh -T
2 genus -f ./script/cortexm0_45nm.tcl -log ./log/all.log
3
```

LAB

# Synthesis

## genus를 사용 준비

- 두개의 터미널 사용 (아래의 경로에서 진행)
- 하나는 genus를 실행, 나머지는 vi run\_synthesis

/home/ex\_poly1/SoC2/SoC/smkcowl\_make\_S28/TOP/SYN

The screenshot shows two terminal windows side-by-side. The left terminal window displays the output of the 'genus' command, which includes the Cadence Genus Synthesis Solution copyright notice, system configuration details (version 22.16-s078\_1, built on Sun Jun 09 22:32:57 PDT 2024), and license information. It also shows the startup of the tool and loading of tool scripts. The right terminal window shows the content of the 'run\_synthesis' script, which is a template for RTL-to-Gate-Level flow, generated from GENUS 17.10-p007\_1. The script sets various environment variables and paths related to the GPKD045 library and design sources.

```
[ex_poly1@npit:SYN]$ genus
TMPDIR is being set to /tmp/genus_temp_22450_npit.ic.rnd1_ex_poly1_Fn4CGL
Cadence Genus(TM) Synthesis Solution.
Copyright 2024 Cadence Design Systems, Inc. All rights reserved worldwide.
Cadence and the Cadence logo are registered trademarks and Genus is a trademark
of Cadence Design Systems, Inc. in the United States and other countries.

[13:11:50.440153] Configured Lic search path (22.01-s003): 35266@npit-service1.localhost.org

Version: 22.16-s078_1, built Sun Jun 09 22:32:57 PDT 2024
Options:
Date: Thu Feb 06 13:11:50 2025
Host: npit.ic.rnd1 (x86_64 w/Linux 3.10.0-1160.119.1.el7.x86_64) (20cores*80cpus*
2physical cpus*Intel(R) Xeon(R) Silver 4316 CPU @ 2.30GHz 30720KB) (395262912KB)
PID: 22450
OS: Red Hat Enterprise Linux Server release 7.9 (Maipo)

Checking out license: Genus_Synthesis
[13:11:50.411130] Periodic Lic check successful
[13:11:50.411145] Feature usage summary:
[13:11:50.411145] Genus_Synthesis

*****
Finished executable startup (1 second elapsed).

Loading tool scripts...
Finished loading tool scripts (16 seconds elapsed).

WARNING: This version of the tool is 240 days old.
@genus:root: 1>
```

```
## Template Script for RTL->Gate-Level Flow (generated from GENUS 17.10-p007_1)
if {[file exists /proc/cpuinfo]} {
    sh grep "model name" /proc/cpuinfo
    sh grep "cpu MHz"      /proc/cpuinfo
}

puts "Hostname : [info hostname]"

#####
# Preset global variables and attributes
#####

set TOP_DESIGN cmsdk_mcu

set LOG_DIR      "./log"
set RPT_DIR      "./report"
set UNMAPPED_DIR "./unmapped"
set MAPPED_DIR   "./mapped"

#GPKD 045 SVT
set STD_LIB_PATH "../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/timing"
set IO_LIB_PATH  "../../../../GPKD045/digital/giolib045_v3.5/timing"
set LEF_LIB_PATH "../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/lef ../../../../../../giolib045_v3.5/lef"
set RTL_PATH     "../RTL/"
set QRC_FILE_PATH "../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045_tech/qrc/qx"

#GPKD 045 SVT
set LIB_LIST {fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib pads_SS_s1vg.lib}
set LEF_LIST {gsclib045_tech.lef gsclib045_macro.lef giolib045.lef}
set QRC_LIST {gpkd045.tch}

set RTL_SOURCES "../RTL/cmsdk_mcu.v \
                ../RTL/verilog_smkcowl/cmsdk_mcu_clkctrl.v \
                ../RTL/verilog_smkcowl/cmsdk_mcu_pin_mux.v \
                ../../../../../../cortexm0_designstart/implementation/cortex_m0_mcu_system_cadence_k_mcu_system_syn_final.v \
                "
```

# Synthesis

## genus

◆ 터미널 1

\$> genus

**genus> gui\_show**

```
[ex_poly1@npit SYN]$ genus  
@genus:root: 1> gui_show
```

```
[ex_poly1@npit SYN]$ genus
TMPDIR is being set to /tmp/genus_temp_78333_npit.ic.rnd1_ex_poly1_iGZGh9
Cadence Genus(TM) Synthesis Solution.
Copyright 2024 Cadence Design Systems, Inc. All rights reserved worldwide.
Cadence and the Cadence logo are registered trademarks and Genus is a trademark
of Cadence Design Systems, Inc. in the United States and other countries.

[16:27:16.440249] Configured Lic search path (22.01-s003): 35266@npit-service1.iptime.org

Version: 22.16-s078_1, built Sun Jun 09 22:32:57 PDT 2024
Options:
Date:    Thu Jan 16 16:27:16 2025
Host:    npit.ic.rnd1 (x86_64 w/Linux 3.10.0-1160.119.1.el7.x86_64) (20cores*80cpus*2physi
PID:    78333
OS:     Red Hat Enterprise Linux Server release 7.9 (Maipo)

Checking out license: Genus_Synthesis
[16:27:16.310939] Periodic Lic check successful
[16:27:16.310947] Feature usage summary:
[16:27:16.310947] Genus_Synthesis

*****
*****
```

Finished executable startup (1 second elapsed).

Loading tool scripts...

Finished loading tool scripts (16 seconds elapsed).

WARNING: This version of the tool is 220 days old.

@genus:root: 1> █

# Synthesis

genus

## ◆ 터미널 2

```
$> vi run_synthesis
```

```
clean.tcl  cons  fv  log  mapped  report  run_synthesis  script  unmapped
```

# Synthesis

genus

## ◆ 터미널 2

- 2행의 스크립트 파일 진행예정

커서를 위치시킨 뒤 g+f

```
1#!/bin/csh -f
2genus -f ./script/cortexm0_45nm.tcl -log ./log/all.log
3
```

# Synthesis

genus

## ◆ script/cortexm0\_45nm.tcl

```
### Template Script for RTL->Gate-Level Flow (generated from GENUS 17.10-p007_1)

if {[file exists /proc/cpuinfo]} {
    sh grep "model name" /proc/cpuinfo
    sh grep "cpu MHz"      /proc/cpuinfo
}

puts "Hostname : [info hostname]"

#####
# Preset global variables and attributes
#####

set TOP DESIGN cmsdk_mcu

set LOG_DIR      "./log"
set RPT_DIR      "./report"
set UNMAPPED_DIR "./unmapped"
set MAPPED_DIR   "./mapped"
```



# Synthesis

## Genus 툴 실행 결과

### genus

- 합성 과정에서 생성되는 폴더의 경로를 설정하는 명령임

```
10 #####
11 # Preset global variables and attributes
12 #####
13
14 set TOP_DESIGN cmsdk_mcu
15
16 set LOG_DIR      "./log"
17 set RPT_DIR      "./report"
18 set UNMAPPED_DIR "./unmapped"
19 set MAPPED_DIR   "./mapped"
20
21 #GPKD 045 SVT
22 set STD_LIB_PATH "../../../../../GPKD045/digital/gsclib045_all_v4.4/gsc
lib045/timing"
23 set IO_LIB_PATH  "../../../../../GPKD045/digital/giolib045_v3.5/timing"
24 set LEF_LIB_PATH "../../../../../GPKD045/digital/gsclib045_all_v4.4/gsc
lib045/lef ../../../../../GPKD045/digital/giolib045_v3.5/lef"
25 set RTL_PATH     "../RTL/"
26 set QRC_FILE_PATH "../../../../../GPKD045/digital/gsclib045_all_v4.4/gsc
lib045_tech/qrc/qx"
```

합성 전의 결과는  
unmapped에 입력되고  
합성 후의 결과는  
mapped에 입력됨

```
@genus:root: 8> set TOP_DESIGN cmsdk_mcu
cmsdk_mcu
@genus:root: 9>
@genus:root: 9> set LOG_DIR      "./log"
./log
@genus:root: 10> set RPT_DIR      "./report"
./report
@genus:root: 11> set UNMAPPED_DIR "./unmapped"
./unmapped
@genus:root: 12> set MAPPED_DIR   "./mapped"
./mapped
```

주의: 행 번호가 보이지 않는 상태에서  
진행하도록 함

# Synthesis

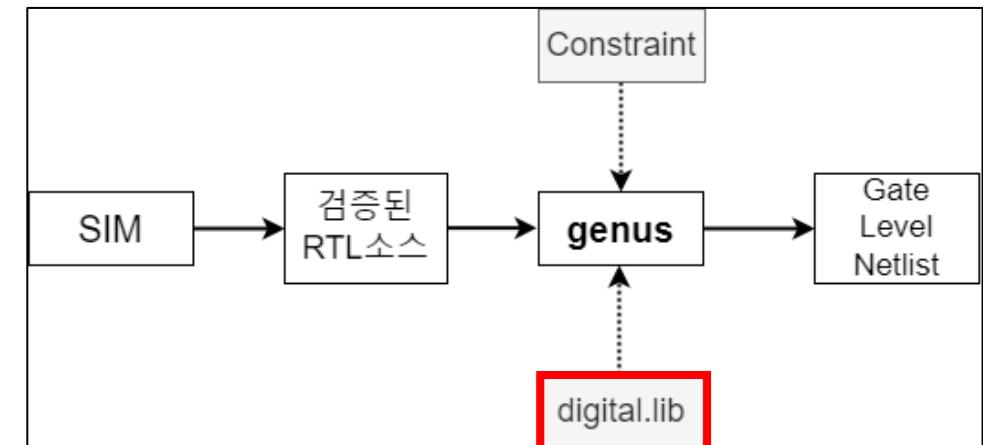
## genus

- genus를 사용하기 위한 환경 setup을 함
- digital.lib을 불러오는 과정임

## Genus 툴 실행 결과

```
@genus:root: 14> set STD_LIB_PATH "../../../..../GPDK045/digital/gsclib045_all_v4.4/gs  
clib045/timing"  
../../../..../GPDK045/digital/gsclib045_all_v4.4/gsclib045/timing  
@genus:root: 15> set IO_LIB_PATH "../../../..../GPDK045/digital/giolib045_v3.5/timing"  
../../../..../GPDK045/digital/giolib045_v3.5/timing  
@genus:root: 16> set LEF_LIB_PATH "../../../..../GPDK045/digital/gsclib045_all_v4.4/gs  
clib045/lef ../../../..../GPDK045/digital/giolib045_v3.5/lef"  
../../../..../GPDK045/digital/gsclib045_all_v4.4/gsclib045/lef ../../../..../GPDK045/dig  
ital/giolib045_v3.5/lef  
@genus:root: 17> set RTL_PATH      ".../RTL/"  
.../RTL/  
@genus:root: 18> set QRC_FILE_PATH ".../..../GPDK045/digital/gsclib045_all_v4.4/g  
sclib045_tech/qrc/qx"  
.../..../..../GPDK045/digital/gsclib045_all_v4.4/gsclib045_tech/qrc/qx
```

```
10 #####  
11 # Preset global variables and attributes  
12 #####  
13  
14 set TOP_DESIGN cmsdk_mcu  
15  
16 set LOG_DIR      "./log"  
17 set RPT_DIR      "./report"  
18 set UNMAPPED_DIR "./unmapped"  
19 set MAPPED_DIR   "./mapped"  
20  
21 #GPDK 045 SVT  
22 set STD_LIB_PATH "../../../..../GPDK045/digital/gsclib045_all_v4.4/gsc  
lib045/timing"  
23 set IO_LIB_PATH "../../../..../GPDK045/digital/giolib045_v3.5/timing"  
24 set LEF_LIB_PATH "../../../..../GPDK045/digital/gsclib045_all_v4.4/gs  
clib045/lef ../../../..../GPDK045/digital/giolib045_v3.5/lef"  
25 set RTL_PATH      ".../RTL/"  
26 set QRC_FILE_PATH ".../..../GPDK045/digital/gsclib045_all_v4.4/gs  
clib045_tech/qrc/qx"
```



주의: 행 번호가 보이지 않는 상태에서  
진행하도록 함

# Synthesis

genus

경로확인\_0

22행 STD\_LIB\_PATH

- 스탠다드 셀에 대한 .lib 파일 확인

22행 빨간색 상대경로에  
커서를 위치시킨 뒤 g+f

```
22 set STD_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/timing"  
23 set IO_LIB_PATH "../../../../../GPK045/digital/giolib045_v3.5/timing"  
24 set LEF_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/lef ../../../../../GPK045/digital/giolib045_v3.5/lef"  
25 set RTL_PATH "../RTL/"  
26 set QRC_FILE_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045 tech/qrc/qx"
```

이 현황이 확인된다면 경로에  
오타, 오류가 없는 것임  
돌아오기 위해서는 Ctrl+o

10 README.txt  
11 fast\_vdd1v0\_basicCells.lib  
12 fast\_vdd1v0\_extvdd1v0.lib  
13 fast\_vdd1v0\_extvdd1v2.lib  
14 fast\_vdd1v0\_multibitsDFF.lib  
15 fast\_vdd1v2\_basicCells.lib  
16 fast\_vdd1v2\_extvdd1v0.lib  
17 fast\_vdd1v2\_extvdd1v2.lib  
18 fast\_vdd1v2\_multibitsDFF.lib  
19 report  
20 slow\_vdd1v0\_basicCells.lib  
21 slow\_vdd1v0\_extvdd1v0.lib  
22 slow\_vdd1v0\_extvdd1v2.lib  
23 slow\_vdd1v0\_multibitsDFF.lib  
24 slow\_vdd1v2\_basicCells.lib  
25 slow\_vdd1v2\_extvdd1v0.lib  
26 slow\_vdd1v2\_extvdd1v2.lib  
27 slow\_vdd1v2\_multibitsDFF.lib

# Synthesis

genus

파란 상자와 같이 현황이 확인된다  
면 경로에 오타, 오류가 없는 것임  
돌아오기 위해서는 Ctrl+o

경로확인\_1

23행 IO\_LIB\_PATH

- Io셀에 대한 .lib 파일 확인

10 pads\_FF\_s1vg.lib  
11 pads\_SS\_s1vg.lib  
12 pads\_TT\_s1vg.lib

23행 빨간색 상대경로에  
커서를 위치시킨 뒤 g+f

경로확인\_2

24행 LEF\_LIB\_PATH

- lef셀에 대한 .lef 파일 확인

.lef 파일이란?  
STD셀의 프레임 셀

```
22 set STD_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/timing"  
23 set IO_LIB_PATH "../../../../../GPK045/digital/giolib045_v3.5/timing"  
24 set LEF_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/lef ../../../../../GPK045/digital/giolib045_v3.5/lef"  
25 set RTL_PATH "../RTL/"  
26 set QRC_FILE_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gs  
clib045 tech/qrc/qx"
```

24행 빨간색 상대경로에  
커서를 위치시킨 뒤 g+f

10 gsclib045\_macro.lef  
11 gsclib045\_multibitsDFF.lef  
12 gsclib045\_tech.lef

# Synthesis

genus

경로확인\_3

25행 RTL\_PATH

- 작성한 RTL소스 코드에 대한 경로

25행 빨간색 상대경로에  
커서를 위치시킨 뒤 g+f

```
22 set STD_LIB_PATH "../../../../GPDK045/digital/gsclib045_all_v4.4/gsc  
lib045/timing"  
23 set IO_LIB_PATH "../../../../GPDK045/digital/giolib045_v3.5/timing"  
24 set LEF_LIB_PATH "../../../../GPDK045/digital/gsclib045_all_v4.4/gsc  
lib045/lef ../../../../../../GPDK045/digital/giolib045_v3.5/lef"  
25 set RTL_PATH      "../RTL/"  
26 set QRC_FILE_PATH "../../../../GPDK045/digital/gsclib045_all_v4.4/gs  
clib045_tech/qrc/qx"
```

이 현황이 확인된다면 경로에  
오타, 오류가 없는 것임  
돌아오기 위해서는 Ctrl+o

```
10 memories_smkcw/  
11 memories_smkcw_1/  
12 verilog_smkcw/  
13 add_pad*  
14 cmsdk_ahb_memory_models_defs.v  
15 cmsdk_mcu.v  
16 cmsdk_mcu_defs.v
```

# Synthesis

genus

경로확인\_4

26행 QRC\_FILE\_PATH

- QRC 파일에 대한 경로

26행 빨간색 상대경로에  
커서를 위치시킨 뒤 g+f

```
22 set STD_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/timing"  
23 set IO_LIB_PATH "../../../../../GPK045/digital/giolib045_v3.5/timing"  
24 set LEF_LIB_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gsc  
lib045/lef ../../../../../GPK045/digital/giolib045_v3.5/lef"  
25 set RTL_PATH      "../RTL/"  
26 set QRC_FILE_PATH "../../../../../GPK045/digital/gsclib045_all_v4.4/gs  
clib045_tech/qrc/qx"
```

이 현황이 확인된다면 경로에  
오타, 오류가 없는 것임  
돌아오기 위해서는 Ctrl+o

8 ../  
9 ./  
10 README  
11 gpk045.tch  
12 hosts

# Synthesis

## genus

- 29행 ~ 31행은 사용할 셀과 파일을 지정
- 33행은 사용할 verilog 소스 코드를 지정

## Genus 툴 실행 결과

```
@genus:root: 24> set LIB_LIST {fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib  
pads_SS_s1vg.lib}  
fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib pads_SS_s1vg.lib  
@genus:root: 25> set LEF_LIST {gsclib045_tech.lef gsclib045_macro.lef giolib045.lef}  
gsclib045_tech.lef gsclib045_macro.lef giolib045.lef  
@genus:root: 26> set QRC_LIST {gpdk045.tch}  
gpdk045.tch  
@genus:root: 27>  
@genus:root: 27> set RTL_SOURCES "../RTL/cmsdk_mcu.v \  
==>           ../RTL/verilog_smkcow/cmsdk_mcu_clkctrl.v \  
==>           ../RTL/verilog_smkcow/cmsdk_mcu_pin_mux.v \  
==>           ../../../../../../cortexm0_designstart/implementation/cortex_m0_mcu_syste  
m_cadence_7ns/mapped/cmsdk_mcu_system_syn_final.v \  
==> "  
..../RTL/cmsdk_mcu.v ..../RTL/verilog_smkcow/cmsdk_mcu_clkctrl.v ..../RTL/verilog_smkcow/  
/cmsdk_mcu_pin_mux.v ..../../../../cortexm0_designstart/implementation/cortex_m0_mcu_sys  
tem_cadence_7ns/mapped/cmsdk_mcu_system_syn_final.v
```

```
28 #GPDK 045 SVT  
29 set LIB_LIST {fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib  
pads_SS_s1vg.lib}  
30 set LEF_LIST {gsclib045_tech.lef gsclib045_macro.lef giolib045.lef}  
31 set QRC_LIST {gpdk045.tch}  
32  
33 set RTL_SOURCES "../RTL/cmsdk_mcu.v \  
34           ../RTL/verilog_smkcow/cmsdk_mcu_clkctrl.v \  
35           ../RTL/verilog_smkcow/cmsdk_mcu_pin_mux.v \  
36           ../../../../../../cortexm0_designstart/implementation/cortex_  
m0_mcu_system_cadence_7ns/mapped/cmsdk_mcu_system_syn_final.v \  
37 "  
38 set_db auto_ungroup none  
39 set_db max_cpus_per_server 2
```

# Synthesis

genus

```
/home/ex_poly1/SoC2/GPDK045/digital/gsclib045_all_v4.4/gsclib045/timing
```

```
$> vi README.txt
```

- 29행은 사용할 STD셀과 io셀 라이브러리를 지정
- 29행 LIB\_LIST와 해당 경로의 README.txt 파일에 동일한 파일 이름이 있는지 확인

```
7 Operating Conditions
8 =====
9 PROJECT SELECTION
10 ##fast_vdd1v0_basicCells.lib : PVT_1P1V_0C
11 ##slow_vdd1v0_basicCells.lib : PVT_0P9V_125C
12 =====
13 SUMMARY OF basicCells
14 ##fast_vdd1v0_basicCells.lib : PVT_1P1V_0C
15 ##fast_vdd1v2_basicCells.lib : PVT_1P32V_0C
16 ##slow_vdd1v0_basicCells.lib : PVT_0P9V_125C
17 ##slow_vdd1v2_basicCells.lib : PVT_1P08V_125C
18 =====
```

```
29 set LIB_LIST [fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib
pads SS_s1vg.lib}
```

# Synthesis

genus

```
/home/ex_poly1/SoC2/GPDK045/digital/giolib045_v3.5/timing
```

pads\_FF\_s1vg.lib    pads\_SS\_s1vg.lib    pads\_TT\_s1vg.lib

vi 진행 방법

```
$> vi pads_FF_s1vg.lib
```

```
: vs
```

```
: e .
```

pads\_SS\_s1vg.lib -> Enter

```
2 delay model : typ  
3 check model :  
4 power model :  
5 capacitance model :  
6 other model : typ  
7 */  
8 library(giolib045) {  
9
```

```
2 delay model : typ  
3 check model : typ  
4 power model : typ  
5 capacitance model : typ  
6 other model : typ  
7 */  
8 library(giolib045) {  
9
```

라이브러리 이름이  
같으므로 충돌 발생

Io셀 라이브러리 파일은  
하나만 기입

```
2 vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib  
pads_SS_s1vg.lib}
```

# Synthesis

## genus

### 30행 LEF\_LIST

- 사용할 LEF파일을 지정

```
30 set LEF_LIST {gsclib045_tech.lef gsclib045_macro.lef giolib045.lef}
```

### 31행 QRC\_LIST

- Parasitic을 뽑거나 Back-End 작업을 할 때 주로 사용되는 파일임

```
31 set QRC_LIST {gpdk045.tch}
```

# Synthesis

genus

## 33행 RTL\_LIST

- 사용할 verilog 파일을 지정
- 33행: IO셀을 포함한 TOP모듈
- 34행: clkctrl에 대한 Verilog 코드
- 35행: pin\_mux에 대한 Verilog 코드
- 36행: clkctrl과 pin\_mux를 제외한 나머지 모듈들에 대한 verilog 코드 (미리 완성한 Gate Level Netlist )

```
33 set RTL_SOURCES "../RTL/cmsdk_mcu.v \
34                         ../RTL/verilog_smkcowl/cmsdk_mcu_clkctrl.v \
35                         ../RTL/verilog_smkcowl/cmsdk_mcu_pin_mux.v \
36                         ../../../../cortexm0_designstart/implementation/cortex_ \
                           m0_mcu_system_cadence_7ns/mapped/cmsdk_mcu_system_syn_final.v \
```

# Synthesis

Genus 툴 실행 결과

genus

```
@genus:root: 28> set_db auto_ungroup none
    Setting attribute of root '/': 'auto_ungroup' = none
1 none
@genus:root: 29> set_db max_cpus_per_server 2
    Setting attribute of root '/': 'max_cpus_per_server' = 2
1 2
```

- set\_db는 genus에서 환경을 설정하는 명령임

```
28 #GDK 045 SVT
29 set LIB_LIST {fast_vdd1v0_basicCells.lib slow_vdd1v0_basicCells.lib
  pads_SS_s1vg.lib}
30 set LEF_LIST {gsclib045_tech.lef gsclib045_macro.lef giolib045.lef}
31 set QRC_LIST {gdk045.tch}
32
33 set RTL_SOURCES "../RTL/cmsdk_mcu.v \
34           ../RTL/verilog_smkcw/cmsdk_mcu_clkctrl.v \
35           ../RTL/verilog_smkcw/cmsdk_mcu_pin_mux.v \
36           ../../cortexm0_designstart/implementation/cortex_
  m0_mcu_system_cadence_7ns/mapped/cmsdk_mcu_system_syn_final.v \
37 "
38 set_db auto_ungroup none
39 set db max cpus per server 2
```

# Synthesis

genus

Effort는 high에 가까울수록 좋은 결과를 기대할 수 있음  
(Default 값은 medium)

Genus 툴 실행 결과

```
@genus:root: 30> set_db syn_generic_effort high; # Default medium
Setting attribute of root '/': 'syn_generic_effort' = high
1 high
@genus:root: 31> set_db syn_map_effort high;      # Default medium
Setting attribute of root '/': 'syn_map_effort' = high
1 high
@genus:root: 32> set_db syn_opt_effort high;      # Default medium
Setting attribute of root '/': 'syn_opt_effort' = high
1 high
```

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;      # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LIB_LIST}
50 read_physical -lef ${LEF_LIST}
51 read_qrc ${QRC_LIST}
52
53 ##generates <signal>_reg[<bit_width>] format
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
```

- set\_db는 genus에서 환경을 설정하는 명령임

# Synthesis

## Genus 툴 실행 결과

### genus

- 최초 라이브러리의 검색 경로를 연결시켜 줌
- hdl소스를 어디서 읽을 것인지 경로를 연결시켜 줌

```
@genus:root: 33> set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
  Setting attribute of root '/': 'init_lib_search_path' = ../../../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/timing ../../../../../../GPKD045/digital/giolib045_v3.5/timing ../../../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/lef ../../../../../../GPKD045/digital/giolib045_v3.5/lef ../../../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045_tech/qrc/qx
1 {../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/timing ../../../../../../GPKD045/digital/giolib045_v3.5/timing ../../../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/lef ../../../../../../GPKD045/digital/giolib045_v3.5/lef ../../../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045_tech/qrc/qx}
@genus:root: 34> set_db init_hdl_search_path [concat $RTL_PATH]
  Setting attribute of root '/': 'init_hdl_search_path' = ./RTL/
1 ./RTL/
```

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;     # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LIB_LIST}
50 read_physical -lef ${LEF_LIST} 
51 read_qrc ${QRC_LIST}
52
53 ##generates <signal>_re mat
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
```

연결한다는 뜻을  
가지고 있음

# Synthesis

## Genus 툴 실행 결과

genus

- 29행의 LIB\_LIST 파일을 읽음

```
@genus:root: 35> read_libs ${LIB_LIST}

Message Summary for Library all 3 libraries:
*****
Missing a function attribute in the output pin definition. [LBR-518]: 2
Missing library level attribute. [LBR-516]: 1
An unsupported construct was detected in this library. [LBR-40]: 16
*****


Warning : Libraries have inconsistent nominal operating conditions. In the Liberty l
ibrary, there are attributes called nom_voltage, nom_process and nom_temperature. Ge
nus reports the message, if the respective values of the 2 given .libs differ. [LBR-
38]
    : The libraries are 'fast_vdd1v0' and 'slow_vdd1v0'.
    : This is a common source of delay calculation confusion and should be avoid
ed.
Warning : Libraries have inconsistent nominal operating conditions. In the Liberty l
```

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;     # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LIB_LIST}
50 read_physical -lef ${LEF_LIST}
51 read_qrc ${QRC_LIST}
52
53 ##generates <signal>_reg[<bit_width>] format
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
```

# Synthesis

## Genus 툴 실행 결과

genus

- 30행의 LEF\_LIST 파일을 읽음

```
@genus:root: 36> read_physical -lef ${LEF_LIST}
Info      : Via with no resistance will have a value of '0.0' assigned for resistance
value. [PHYS-129]
          : Via 'M2_M1_HV' has no resistance value.
          : If this is the expected behavior, this message can be ignored.
Info      : Via with no resistance will have a value of '0.0' assigned for resistance
value. [PHYS-129]
          : Via 'M2_M1_VV' has no resistance value.
Info      : Via with no resistance will have a value of '0.0' assigned for resistance
value. [PHYS-129]
          : Via 'M2_M1_VH' has no resistance value.
Info      : Via with no resistance will have a value of '0.0' assigned for resistance
value. [PHYS-129]
```

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;     # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LTB_LTST}
50 read_physical -lef ${LEF_LIST} // This line is highlighted with a red box
51 read_qrc ${QRC_LIST}
52
53 ##generates <signal>_reg[<bit_width>] format
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
```

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 37> read_qrc ${QRC_LIST}
```

```
According to qrc_tech_file, there are total 11 routing layers [ V(5) / H(6) ]
```

```
Done reading qrc_tech_file
```

- 31행의 QRC\_LIST 파일을 읽음

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;     # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LIB_LIST}
50 read_physical -lef ${LEF_LIST}
51 read_qrc ${QRC_LIST} // Line 51 is highlighted with a red box
52
53 ##generates <signal>_reg[<bit_width>] format
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP_DESIGN}_%s_%d
```

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 38> ##generates <signal>_reg[<bit_width>] format
@genus:root: 39> set_db hdl_array_naming_style %s_%d
    Setting attribute of root '/': 'hdl_array_naming_style' = %s_%d
1 %s_%d
@genus:root: 40> set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
    Setting attribute of root '/': 'uniquify_naming_style' = cmsdk_mcu_%s_%d
1 cmsdk_mcu_%s_%d
```

- 디자인 내에서 instance 이름 충돌을 방지하기 위해 이름 생성 과정에서 규칙을 설정함

```
41 ##Set synthesizing effort for each synthesis stage
42 set_db syn_generic_effort high; # Default medium
43 set_db syn_map_effort high;      # Default medium
44 set_db syn_opt_effort high;     # Default medium
45
46 set_db init_lib_search_path [concat $STD_LIB_PATH $IO_LIB_PATH $LEF_LIB_PATH $QRC_FILE_PATH]
47 set_db init_hdl_search_path [concat $RTL_PATH]
48
49 read_libs ${LIB_LIST}
50 read_physical -lef ${LEF_LIST}
51 read_qrc ${QRC_LIST}
52
53 ##generates <signal>_req[<bit width>] format
54 set_db hdl_array_naming_style %s_%d
55 set_db / .uniquify_naming_style ${TOP DESIGN}_%s_%d
```

# Synthesis

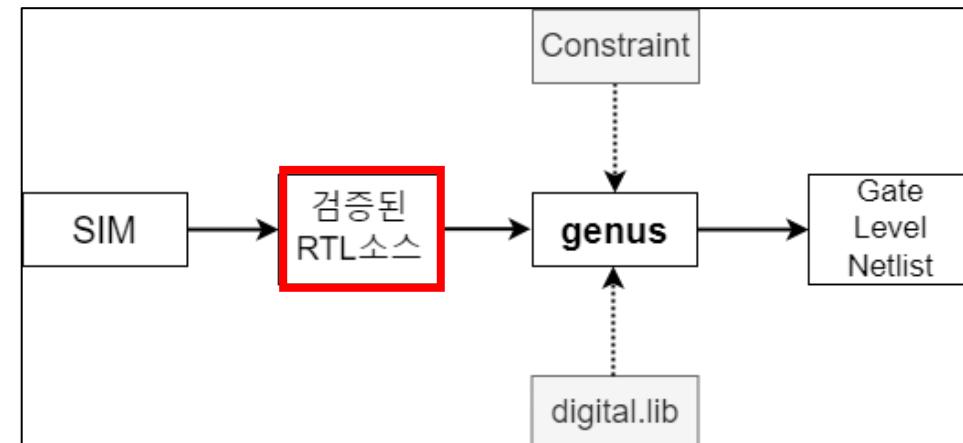
## Genus 툴 실행 결과

genus

```
@genus:root: 41> read_hdl $RTL_SOURCES  
@genus:root: 42>
```

```
57 #####  
58 ## Load Design  
59 #####  
60  
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm  
62 # -f option is for the case that all files are defined in a specific  
file  
63 read_hdl $RTL_SOURCES  
64  
65 elaborate ${TOP DESIGN}  
66 puts "Runtime & Memory after 'read_hdl'"  
67 time_info Elaboration  
68  
69 set_db hinst:u_cmsdk_mcu_system .preserve true  
70 set_db optimize_merge_flops false  
71 set_db optimize_merge_latches false  
72  
73 uniquify -verbose ${TOP DESIGN}  
74  
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

- 검증된 RTL소스를 불러옴



# Synthesis

## Genus 툴 실행 결과

genus

- 읽어 들인 RTL소스 파일을 elaborate

```
57 #####
58 ## Load Design
59 #####
60
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadenc
62 # -f option is for the case that all files are defined in
   file
63 read_hdl $RTL_SOURCES
64
65 elaborate ${TOP_DESIGN}
66 puts "Runtime & Memory after 'read_hdl'"
67 time_info Elaboration
68
69 set_db hinst:u_cmsdk_mcu_system .preserve true
70 set_db optimize_merge_flops false
71 set_db optimize_merge_latches false
72
73 uniquify -verbose ${TOP_DESIGN}
74
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

```
@genus:root: 42> elaborate ${TOP_DESIGN}
Libraries have 324 usable logic and 128 usable sequential lib-cells.
Info      : Elaborating Design. [ELAB-1]
            : Elaborating top-level block 'cmsdk_mcu' from file '../RTL/../RTL/cmsdk_mcu.v'.
Warning   : Using default parameter value for module elaboration. [CDFG-818]
            : Elaborating block 'cmsdk_mcu' with default parameters value.
Warning   : Removing unused register. [CDFG-508]
            : Removing unused flip-flop register 'dbgrst_reg' in module 'cmsdk_mcu_clkctrl_CLKGATE_PRESENT0' in file '../RTL/../RTL/verilog_smkcw/cmsdk_mcu_clkctrl.v' on line 115.
            : Genus removes the flip-flop or latch inferred for an unused signal or variable. To preserve the flip-flop or latch, set the hdl_preserve_unused_registers attribute to true or use a pragma in the RTL.
```

Stage: post\_elab

Trick	Accepts	Rejects	Runtime (s)
ume_constant_bmux	0	0	0.00
ume_merge	0	0	0.00
ume_ssm	0	0	0.00
ume_cse	0	0	0.01
ume_shrink	0	0	0.00
ume_sweep	0	0	0.00

Stage: post\_elab

Transform	Accepts	Rejects	Runtime (s)
hlo_optimize_datapath_shifters	0	0	0.00
hlo_clip_mux_input	0	0	0.00
hlo_clip	0	0	0.00
hlo_cleanup	0	0	0.00

```
UM:    flow.cputime    flow.realtime    timing.setup.tns    timing.setup.wns    snapshot
UM:*    elaborate
```

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 43> puts "Runtime & Memory after 'read_hdl'"  
Runtime & Memory after 'read_hdl'
```

```
57 #####  
58 ## Load Design  
59 #####  
60  
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm  
62 # -f option is for the case that all files are defined in a specific  
file  
63 read_hdl $RTL_SOURCES  
64  
65 elaborate ${TOP DESIGN}  
66 puts "Runtime & Memory after 'read_hdl'"  
67 time_info Elaboration  
68  
69 set_db hinst:u_cmsdk_mcu_system .preserve true  
70 set_db optimize_merge_flops false  
71 set_db optimize_merge_latches false  
72  
73 uniquify -verbose ${TOP DESIGN}  
74  
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

- puts는 메시지를 출력하는 명령임

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 44> time_info Elaboration
stamp 'Elaboration' being created for table 'default'

Total Time (Wall) | Stage Time (Wall) | % (Wall) | Date - Time | Memory | Stage
-----+-----+-----+-----+-----+-----+
00:00:08(00:00:00) | 00:00:00(00:00:00) | 0.0( 0.0) | 20:44:02 (Jan26) | 241.0 MB | init
-----+-----+-----+-----+-----+
00:00:46(01:00:25) | 00:00:38(01:00:25) | 100.0(100.0) | 21:44:27 (Jan26) | 531.7 MB | Elaboration
-----+-----+-----+-----+-----+
Number of threads: 2 * 1 (id: default, time_info v1.57)
Info: (*N*) indicates data that was populated from previously saved time_info database
Info: CPU time includes time of parent + longest thread
```

```
57 #####
58 ## Load Design
59 #####
60
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm
62 # -f option is for the case that all files are defined in a specific
   file
63 read_hdl $RTL_SOURCES
64
65 elaborate ${TOP DESIGN}
66 puts "Runtime & Memory after 'read_hdl'"
67 time_info Elaboration
68
69 set_db hinst:u_cmsdk_mcu_system .preserve true
70 set_db optimize_merge_flops false
71 set_db optimize_merge_latches false
72
73 uniquify -verbose ${TOP DESIGN}
74
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

- Elaborate에서 소요된 시간과 메모리를 확인하는 명령임

# Synthesis

genus

## Genus 툴 실행 결과

```
@genus:root: 45> set_db hinst:u_cmsdk_mcu_system .preserve true
    Setting attribute of hinst 'u_cmsdk_mcu_system': 'preserve' = true
1 true
@genus:root: 46> set_db optimize_merge_flops false
    Setting attribute of root '/': 'optimize_merge_flops' = false
1 false
@genus:root: 47> set_db optimize_merge_latches false
    Setting attribute of root '/': 'optimize_merge_latches' = false
1 false
```

```
57 #####
58 ## Load Design
59 #####
60
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm
62 # -f option is for the case that all files are defined in a specific
   file
63 read_hdl $RTL_SOURCES
64
65 elaborate ${TOP DESIGN}
66 puts "Runtime & Memory after 'read_hdl'"
67 time_info Elaboration
68
69 set_db hinst:u_cmsdk_mcu_system .preserve true
70 set_db optimize_merge_flops false
71 set_db optimize_merge_latches false
72
73 uniquify -verbose ${TOP DESIGN}
74
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

- 기존의 u\_cmsdk\_mcu\_system 모듈은 건드리지 않고 보존할 것임
- Set\_don't\_touch와 비슷한 개념임

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 48> uniquify -verbose ${TOP DESIGN}
Uniquified instance: u_pin_mux/mux_119_30, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_10
Uniquified instance: u_pin_mux/mux_117_30, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_9
Uniquified instance: u_pin_mux/mux_115_30, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_8
Uniquified instance: u_pin_mux/mux_130_33, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_7
Uniquified instance: u_pin_mux/mux_128_33, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_6
Uniquified instance: u_pin_mux/mux_126_33, and its subdesign name is: cmsdk_mcu_cmsdk_mcu_bmux_1_5
Uniquified instance: u_cmsdk_mcu_clkctrl/mux_165_22, and its subdesign name is: cmsdk_mcu_cmsdk_mcu
bmux_1_4
Uniquified instance: u_cmsdk_mcu_clkctrl/mux_166_22, and its subdesign name is: cmsdk_mcu_cmsdk_mcu
bmux_1_3
```

```
57 #####
58 ## Load Design
59 #####
60
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm
62 # -f option is for the case that all files are defined in a specific
   file
63 read_hdl $RTL_SOURCES
64
65 elaborate ${TOP DESIGN}
66 puts "Runtime & Memory after 'read_hdl'"
67 time_info Elaboration
68
69 set_db hinst:u_cmsdk_mcu_system .preserve true
70 set_db optimize_merge_flops false
71 set_db optimize_merge_latches false
72
73 uniquify -verbose ${TOP DESIGN}
74
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

- 디자인 내에서 instance 이름을 겹치지 않도록 정리해주는 명령임

# Synthesis

genus

- RTL소스의 무결성을 확인하는 명령임

```
57 #####
58 ## Load Design
59 #####
60
61 #read_hdl -f $RTL_SOURCES -define Func_mode -define cadence_045nm
62 # -f option is for the case that all files are defined in a specific
   file
63 read_hdl $RTL_SOURCES
64
65 elaborate ${TOP_DESIGN}
66 puts "Runtime & Memory after 'read_hdl'"
67 time_info Elaboration
68
69 set_db hinst:u_cmsdk_mcu_system .preserve true
70 set_db optimize_merge_flops false
71 set_db optimize_merge_latches false
72
73 uniquify -verbose ${TOP_DESIGN}
74
75 check_design > ${RPT_DIR}/00_check_design.rpt
```

## Genus 툴 실행 결과

큰 문제가 없음을  
확인함

Check Design Report (c)

Summary

Name	Total
Unresolved References	0
Empty Modules	0
Unloaded Port(s)	0
Unloaded Sequential Pin(s)	0
Unloaded Combinational Pin(s)	11
Assigns	137
Undriven Port(s)	0
Undriven Leaf Pin(s)	0
Undriven hierarchical pin(s)	1241
Multidriven Port(s)	0
Multidriven Leaf Pin(s)	0
Multidriven hierarchical Pin(s)	0
Multidriven unloaded net(s)	0
Constant Port(s)	0
Constant Leaf Pin(s)	880
Constant hierarchical Pin(s)	56
Preserved leaf instance(s)	13566
Preserved hierarchical instance(s)	30
Feedthrough Modules(s)	0
Libcells with no LEF cell	4
Physical (LEF) cells with no libcell	107
Subdesigns with long module name	0
Physical only instance(s)	0
Logical only instance(s)	0

Done Checking the design.

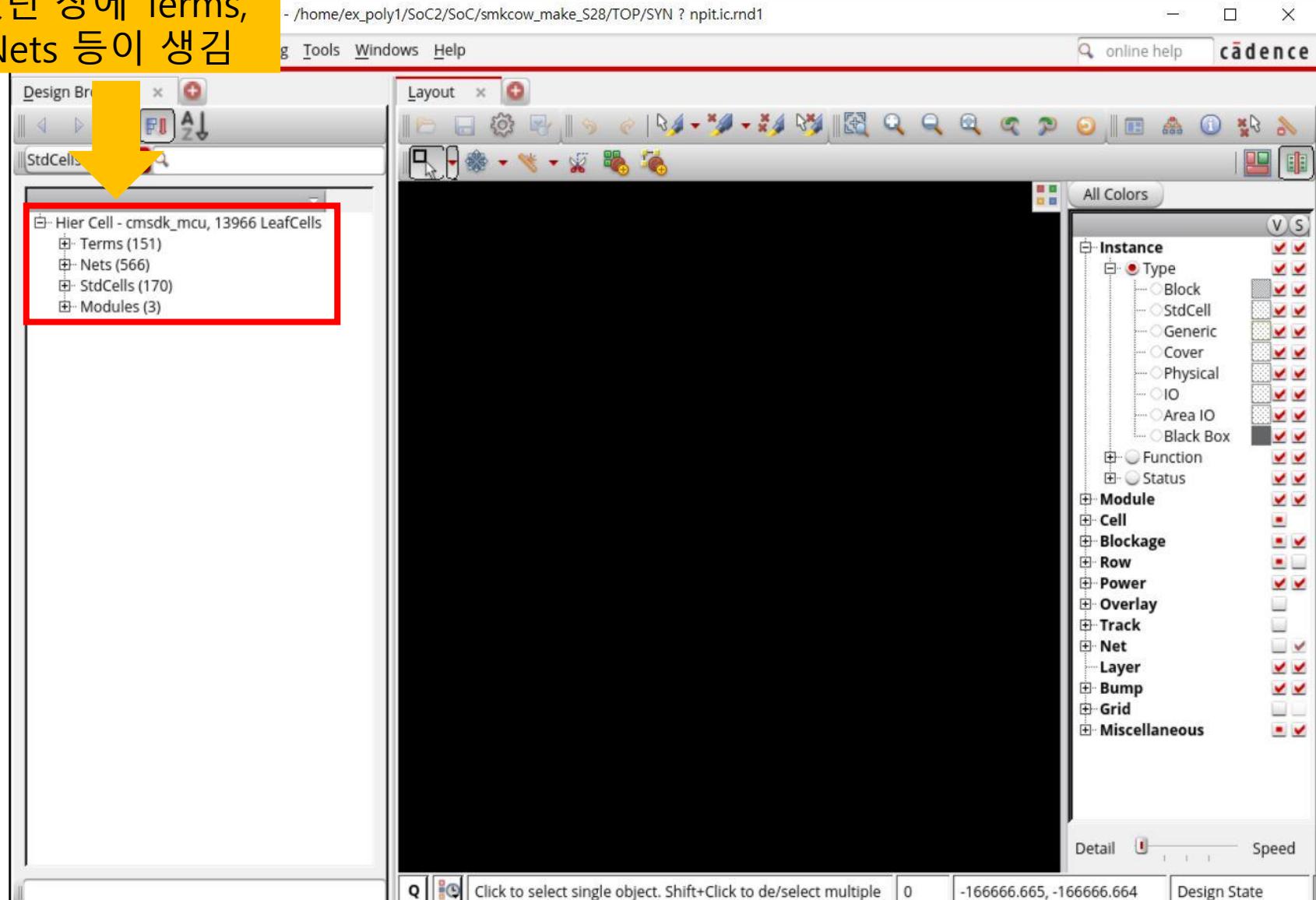
# Synthesis

genus

\$> **gui\_show**

- Check Design 후 gui 확인

처음엔 아무것도  
없던 창에 Terms,  
Nets 등이 생김

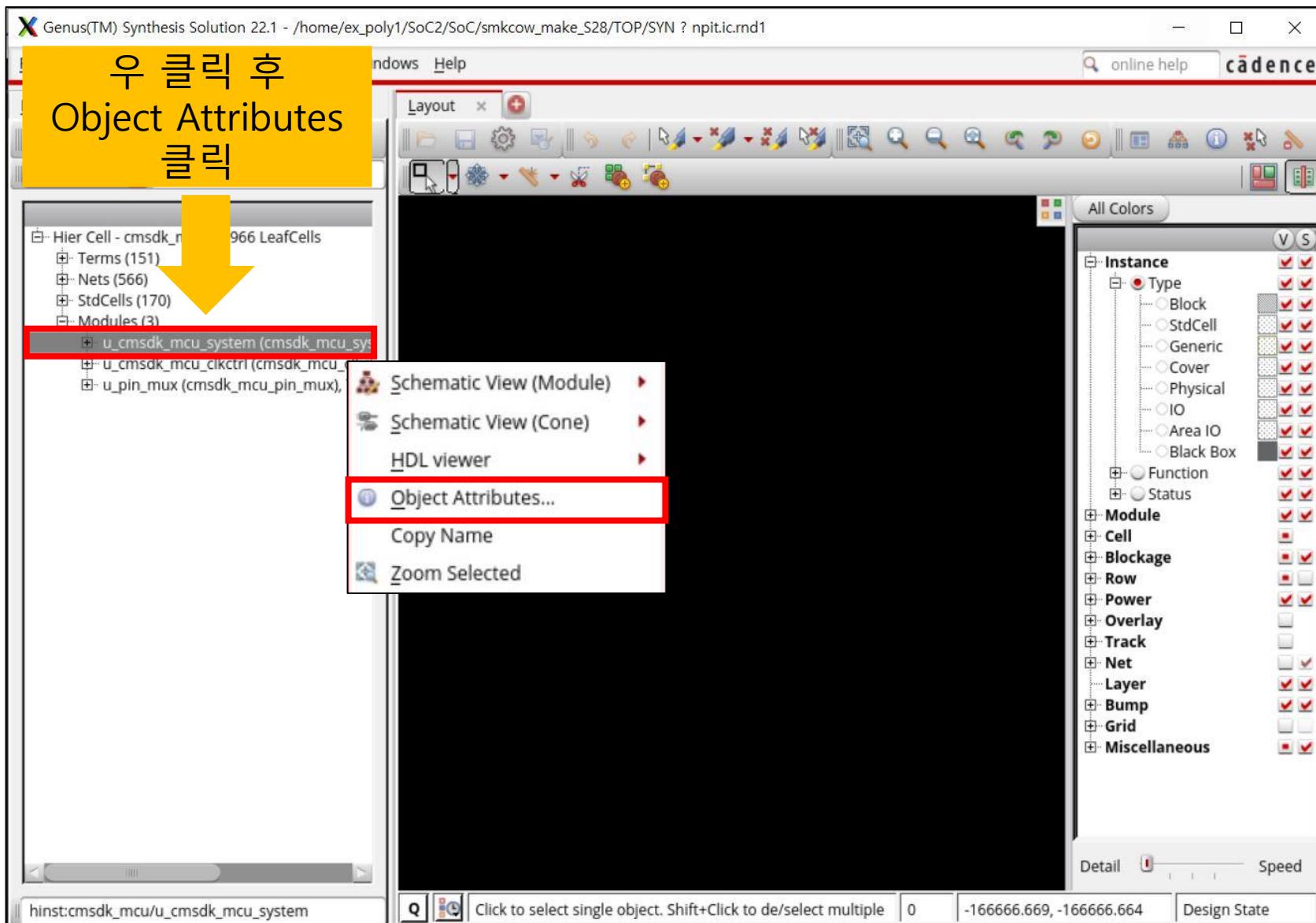


# Synthesis

genus

\$> **gui\_show**

- **gui 확인**

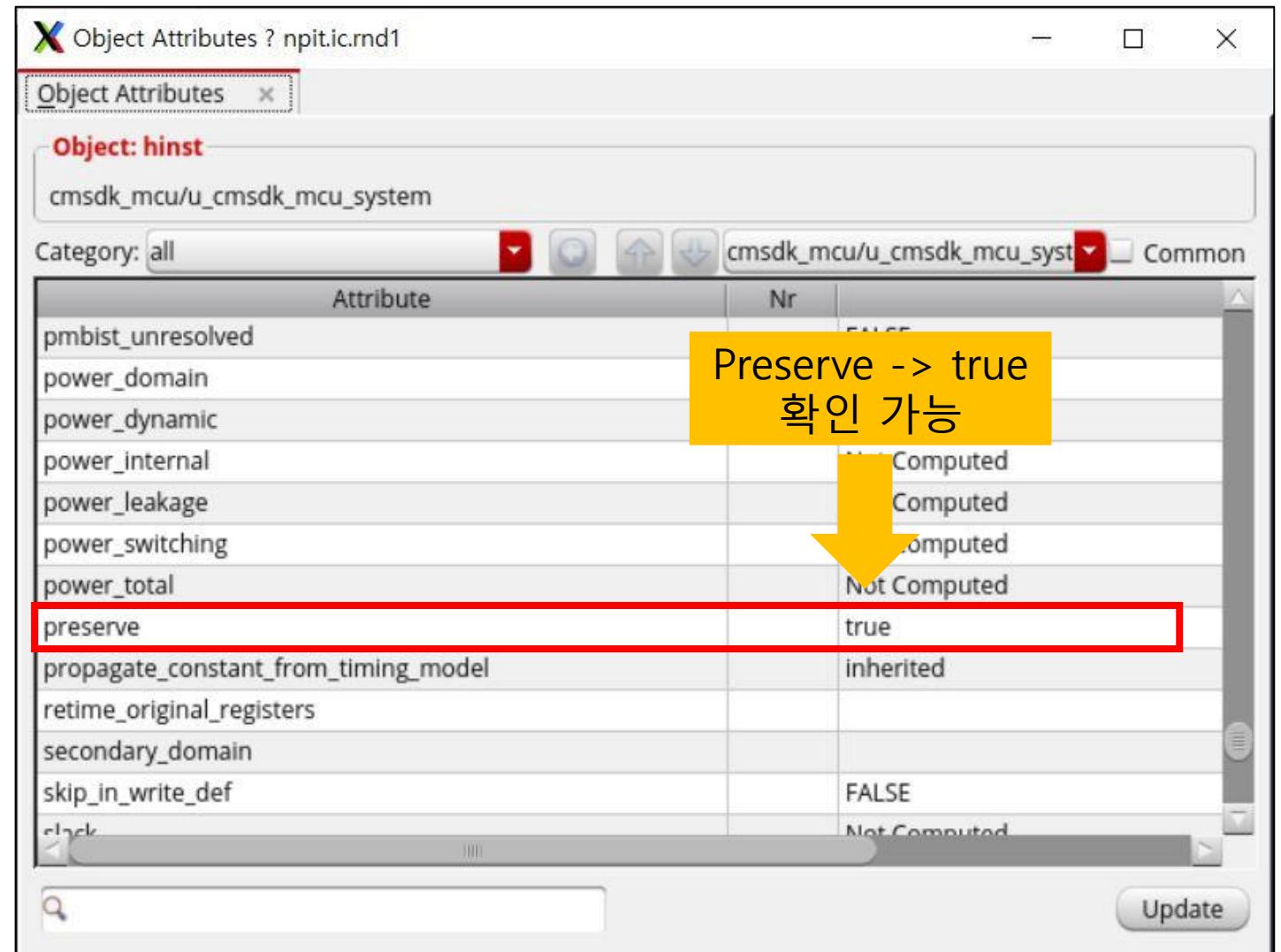


# Synthesis

genus

\$> gui\_show

- gui 확인

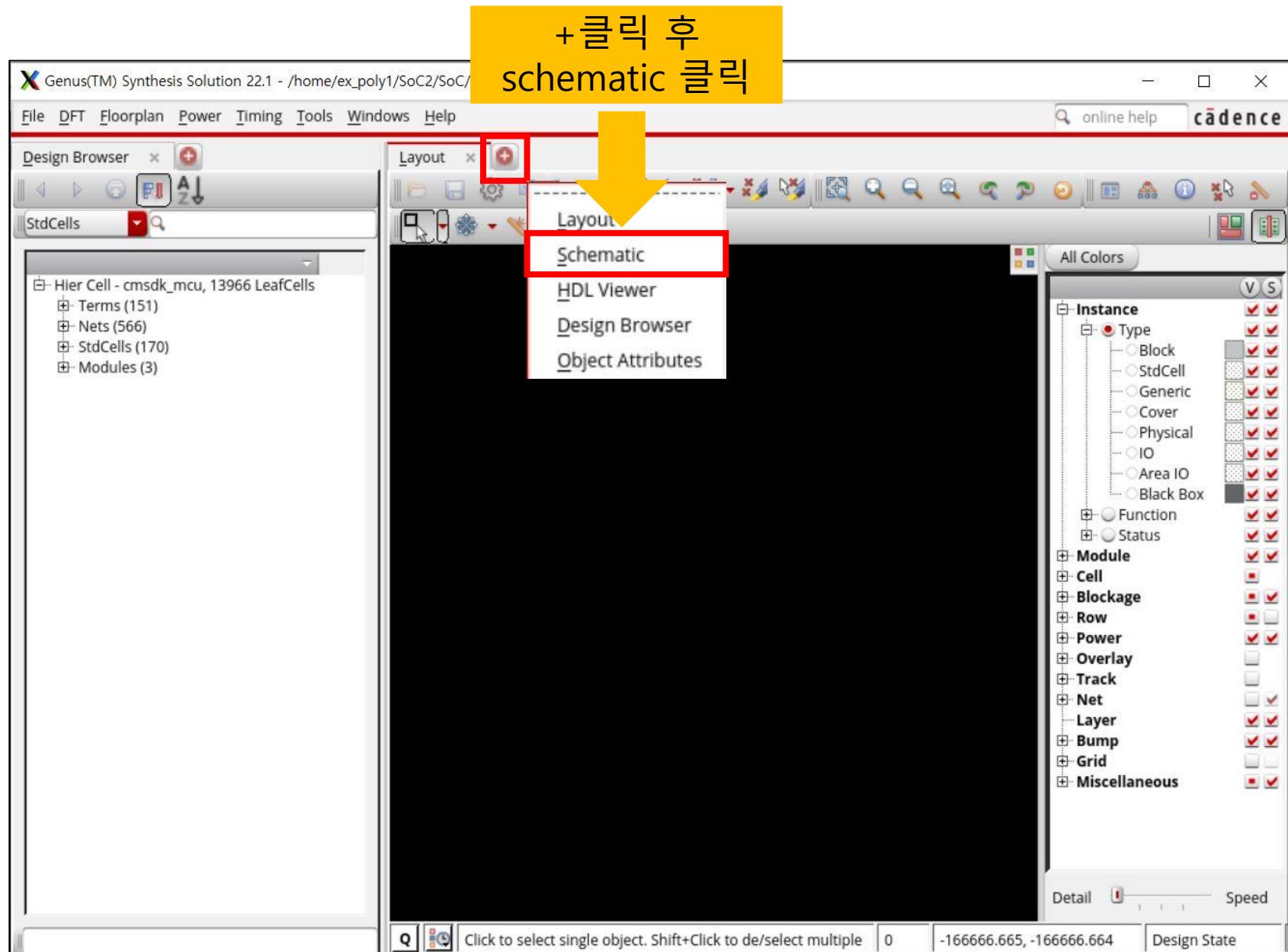


# Synthesis

genus

\$> **gui\_show**

- **gui 확인**

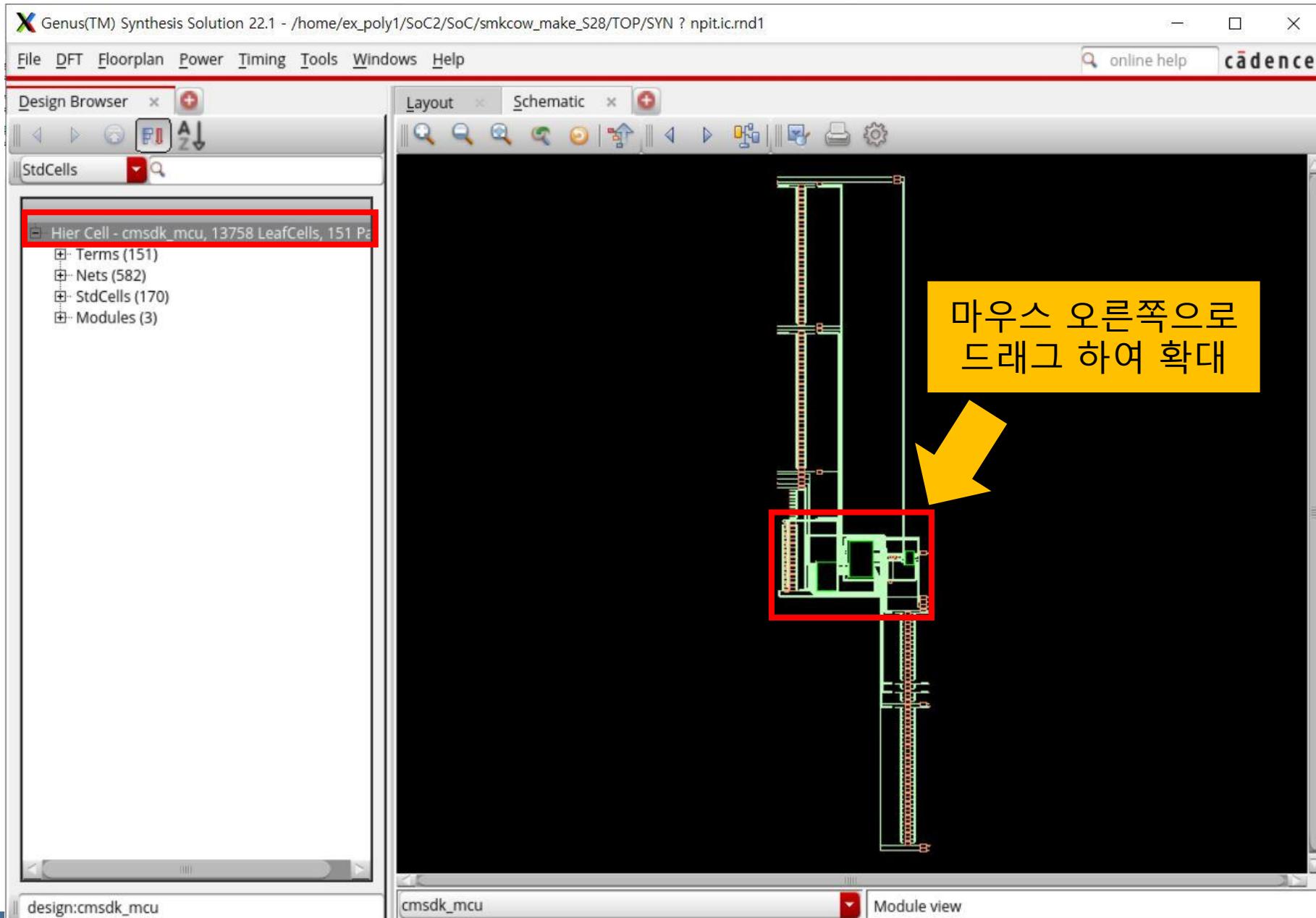


# Synthesis

genus

\$> **gui\_show**

- Gui로 TOP모듈 확인



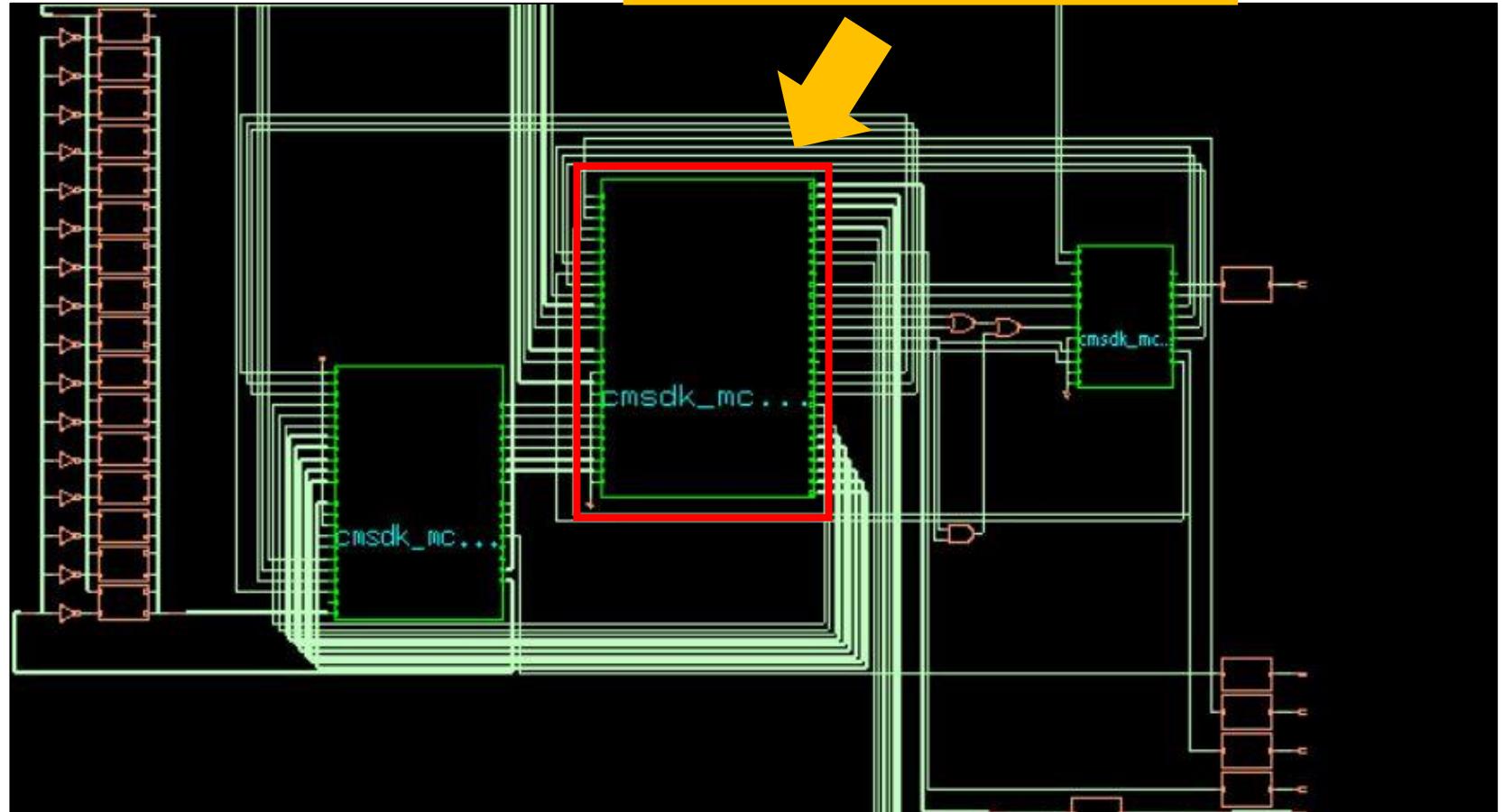
# Synthesis

genus

\$> **gui\_show**

- gui 확인

미리 합성했던 모듈도 잘  
load 되어있음을 확인  
해당 모듈은 Gate Mapping이  
끝나 있음

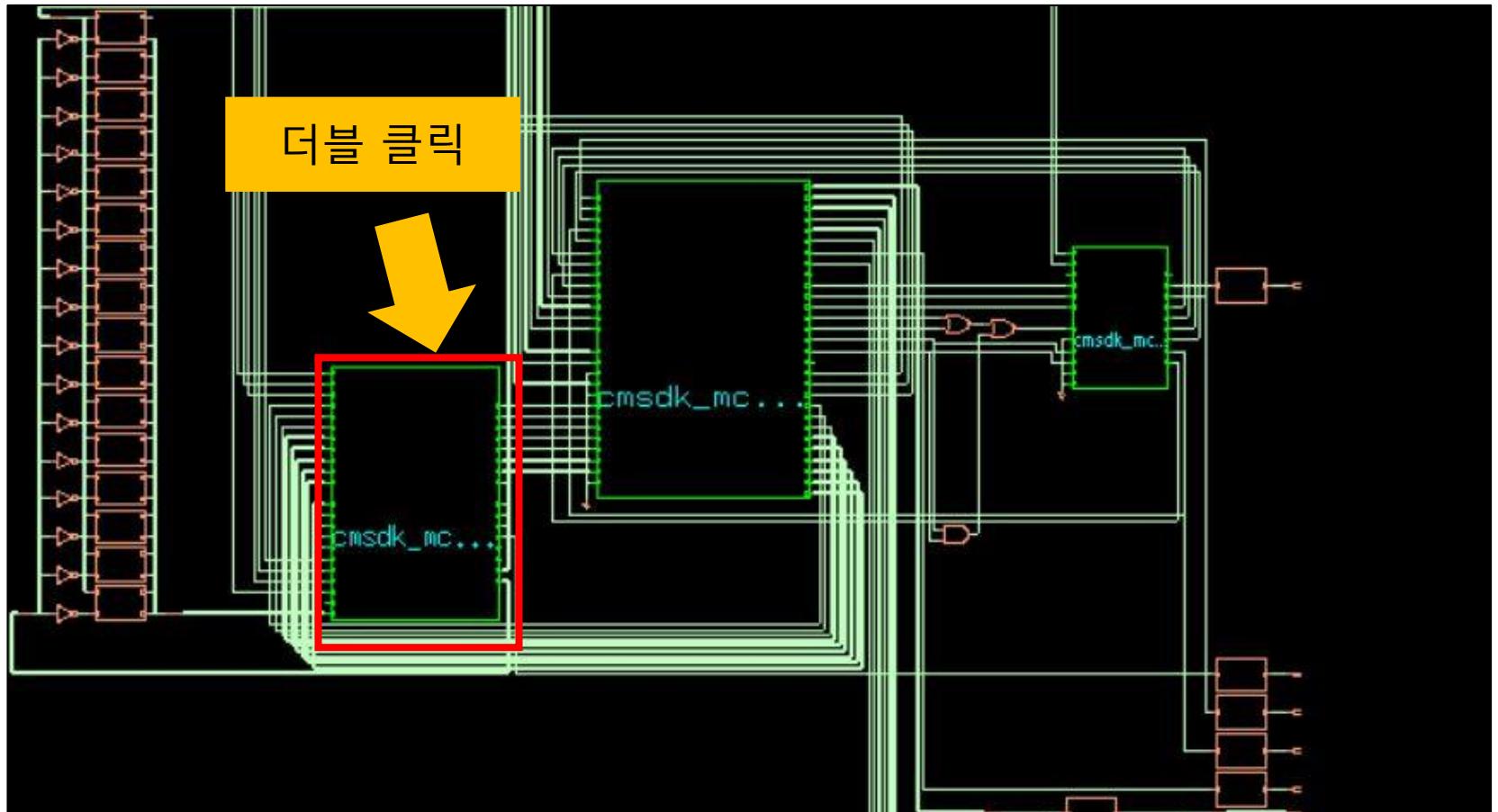


# Synthesis

genus

\$> **gui\_show**

- 합성 전 STD셀 이름 확인



## Synthesis

## genus

\$> **gui\_show**

- 합성 전 STD셀 이름 확인

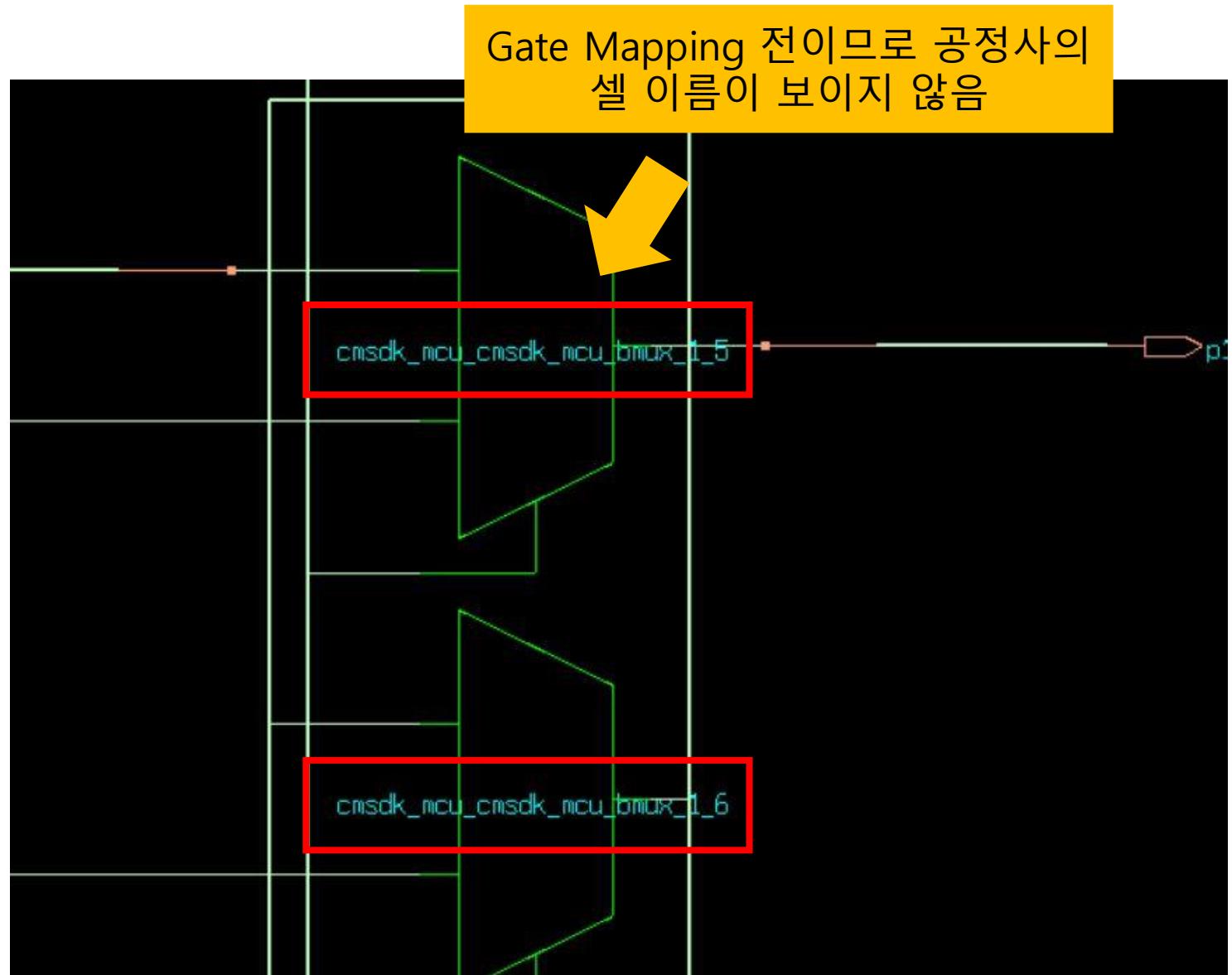


# Synthesis

genus

\$> **gui\_show**

- 합성 전 STD셀 이름 확인



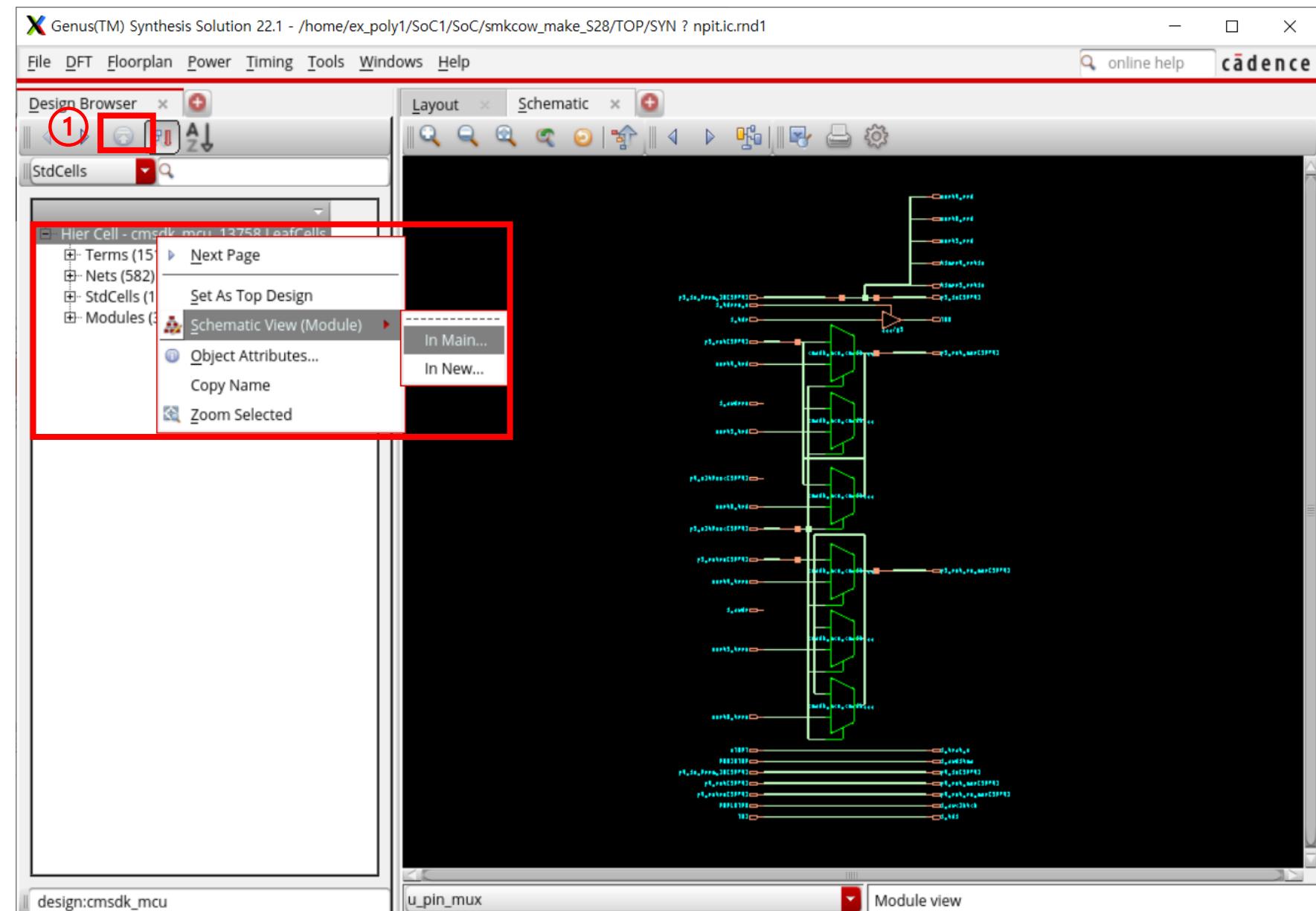
# Synthesis

genus

- TOP 디자인으로 이동

1번 클릭

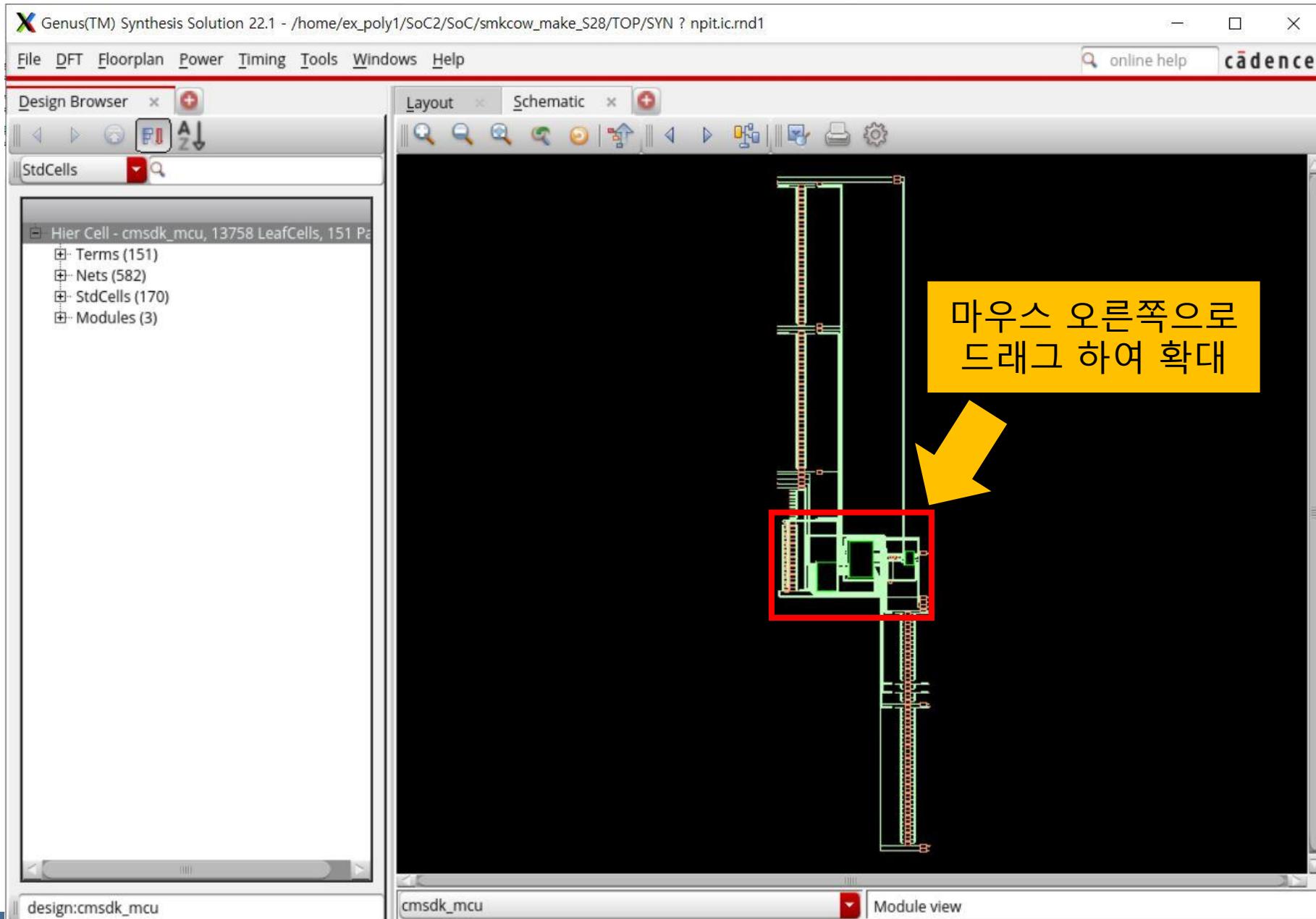
- Hier Cell 우 클릭
- schematic view 클릭
- In Main 클릭



# Synthesis

genus

\$> **gui\_show**

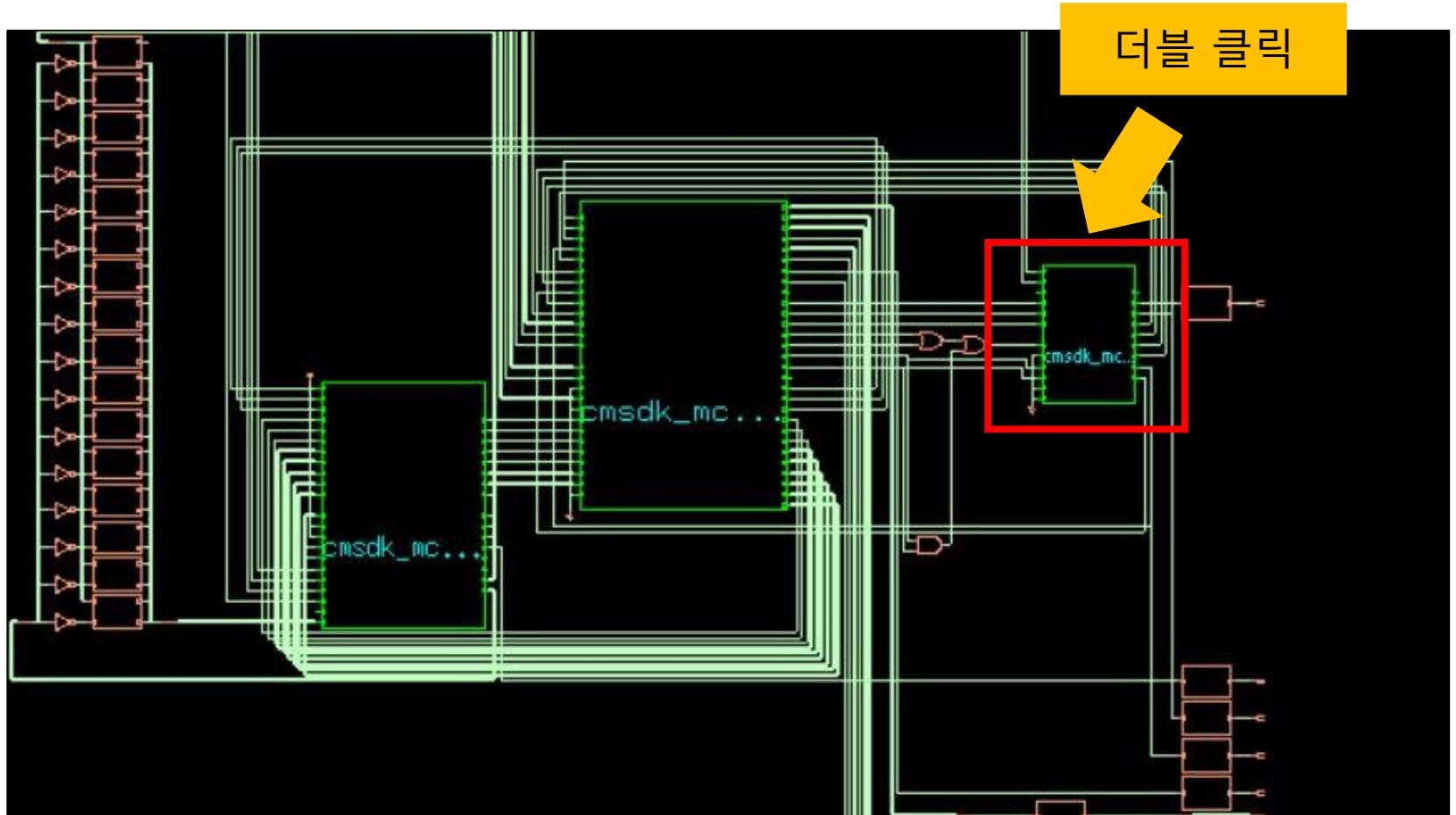


# Synthesis

genus

\$> gui\_show

- 합성 전 schematic 확인

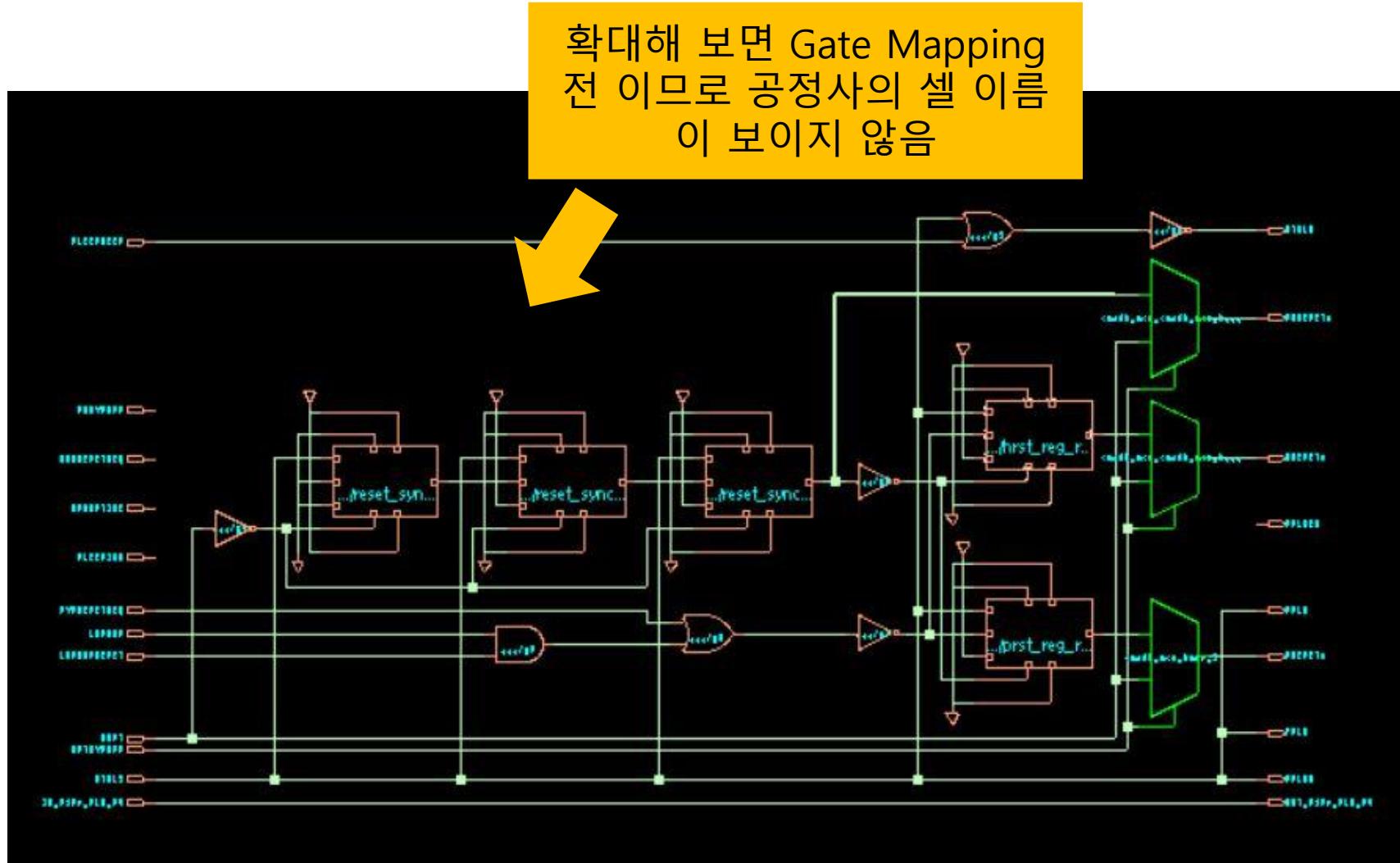


# Synthesis

genus

\$> **gui\_show**

- 합성 전 schematic 확인



# Synthesis

genus

## ◆ script/cortexm0\_45nm.tcl

Setup이 완료되었으므로 83행 부터  
차례차례 진행할 예정

```
78 #####
79 ## Constraints Setup
80 #####
81
82 #read_sdc ./cons/cmsdk_mcu.sdc
83 source -verbose ./cons/cmsdk_mcu.sdc      ; #This command can't write report by using the mark '>'
84 source -verbose ./cons/dont_use_45nm.tcl ; #This command can't write report by using the mark '>'
85 puts "The number of exceptions is [llength [vfind "design:$TOP_DESIGN" -exception *]]"
86 #set_db "design:$DESIGN" .force_wireload <wireload name>
87
88 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt
89
90 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.rpt
91 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.rpt
92
93 echo -n "\n TIMEX > compile is ready : ";
94 sh date
```

# Synthesis

## Genus 툴 실행 결과

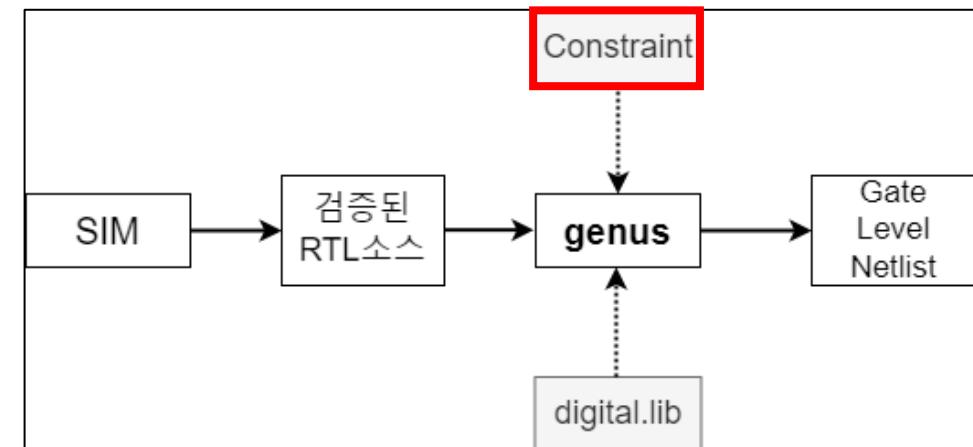
### genus

- cons의 .sdc 파일을 읽어 들임
- 검사 결과 Warning 발생

```
78 #####
79 ## Constraints Setup
80 #####
81
82 #read sdc ./cons/cmsdk_mcu.sdc
83 source -verbose ./cons/cmsdk_mcu.sdc ; #This command can
   e report by using the mark '>'
84 source -verbose ./cons/dont_use_45nm.tcl ; #This command can
   e report by using the mark '>'

85 puts "The number of exceptions is [llength [vfind "design:$DESIGN"
      N" -exception *]]"
86 #set_db "design:$DESIGN" .force_wireload <wireload name>
87
88 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt
89
90 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.
rpt
91 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.
rpt
92
93 echo -n "\n TIMEX > compile is ready : "
94 sh date
```

```
@genus:root: 52> source -verbose ./cons/cmsdk_mcu.sdc ; #This command can't write report by using the mark '>'  
Sourcing './cons/cmsdk_mcu.sdc' (Sun Jan 26 21:55:43 KST 2025)...  
#@ Begin verbose source ./cons/cmsdk_mcu.sdc  
@file/cmsdk_mcu.sdc 13: set STD_LIB slow_vdd1v0_basicCells.lib:slow_vdd1v0  
@file/cmsdk_mcu.sdc 15: set_units -capacitance 1000.0fF  
@file/cmsdk_mcu.sdc 16: set_units -time 1000.0ps  
@file/cmsdk_mcu.sdc 19: reset_design  
@file/cmsdk_mcu.sdc 21: create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]  
@file/cmsdk_mcu.sdc 24: set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]  
@file/cmsdk_mcu.sdc 25: set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]  
@file/cmsdk_mcu.sdc 31: set_clock_uncertainty -setup 0.1 [get_clocks MAIN_CLOCK]  
@file/cmsdk_mcu.sdc 34: set_clock_transition 0.1 [get_clocks MAIN_CLOCK]  
@file/cmsdk_mcu.sdc 50: set_input_delay -max 0.1 -clock MAIN_CLOCK [all_inputs]  
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]  
          : The set_input_delay command is not supported on ports which have a clock already defined 'port:cmsdk_mcu/XTAL1'.  
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]  
          : The set_input_delay command is not supported on ports which have a clock already defined 'port:cmsdk_mcu/XTAL1'.  
          : The current version does not support this SDC command option. However, future versions may be enhanced to support this option.  
@file/cmsdk_mcu.sdc 51: set_input_delay -min 0.01 -clock MAIN_CLOCK [all_inputs]  
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]  
          : The set_input_delay command is not supported on ports which have a clock already defined 'port:cmsdk_mcu/XTAL1'.
```



# Synthesis

genus

Clock을  
제외시킴



- set\_input\_delay는 Clock에 적용할 수 없다는 Warning이 발생 (50, 51행)
- 이후에는 set\_input\_delay에 Clock을 제외시키는 명령이 들어가므로 이 Warning 무시

```
47 #####  
48 ##### input, output delay ####  
49 #####  
50 set_input_delay -max 0.1 -clock MAIN_CLOCK [all_inputs]  
51 set_input_delay -min 0.01 -clock MAIN_CLOCK [all_inputs]  
52  
53 remove_input_delay [get_ports "XTAL1"]  
54  
55 set_output_delay -max 0.1 -clock MAIN_CLOCK [all_outputs]  
56 set_output_delay -min -0.01 -clock MAIN_CLOCK [all_outputs]  
57  
58 #####  
59 # DESIGN AREA #  
60 #####  
61 # Below is no use in Genus tool  
62 #set_max_area 0
```

```
@file(cmsdk_mcu.sdc) 50: set_input_delay -max 0.1 -clock MAIN_CLOCK [all_inputs]  
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]  
          : The set_input_delay command is not supported on ports which have a clock already defined 'p  
ort:cmsdk_mcu/XTAL1'.  
          : The current version does not support this SDC command option. However, future versions may  
be enhanced to support this option.  
@file(cmsdk_mcu.sdc) 51: set_input_delay -min 0.01 -clock MAIN_CLOCK [all_inputs]  
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]  
          : The set_input_delay command is not supported on ports which have a clock already defined 'p  
ort:cmsdk_mcu/XTAL1'.
```

50, 51행에 대한 Warning 내용

# Synthesis

genus

```
78 #####
79 ## Constraints Setup
80 #####
81
82 #read_sdc ./cons/cmsdk_mcu.sdc
83 source -verbose ./cons/cmsdk_mcu.sdc ; #This command can't write report by using the mark '>'
84 source -verbose ./cons/dont_use_45nm.tcl ; #This command can't write report by using the mark '>'
85 puts "The number of exceptions is [llength [vfind "design:$TOPDESIGN -exception *]]"
86 #set_db "design:$DESIGN" .force_wireload <wireload name>
87
88 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt
89
90 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.rpt
91 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.rpt
92
93 echo -n "\n TIMEX > compile is ready : ";
94 sh date
```

## Genus 툴 실행 결과

```
@genus:root: 54> source -verbose ./cons/dont_use_45nm.tcl ; #This command can't write report by using the mark '>'  
Sourcing './cons/dont_use_45nm.tcl' (Sun Jan 26 22:03:58 KST 2025)...  
#@ Begin verbose source ./cons/dont_use_45nm.tcl  
@file(dont_use_45nm.tcl) 1: set_dont_use [get_lib_cells */ANTENNA*]  
@file(dont_use_45nm.tcl) 2: set_dont_use [get_lib_cells */DLY*]  
@file(dont_use_45nm.tcl) 4: set_dont_use [get_lib_cells */HOLD*]  
@file(dont_use_45nm.tcl) 5: set_dont_use [get_lib_cells */TIE*]  
@file(dont_use_45nm.tcl) 6: set_dont_use [get_lib_cells */TLA*]  
@file(dont_use_45nm.tcl) 7: set_dont_use [get_lib_cells */DFFS*]  
@file(dont_use_45nm.tcl) 8: set_dont_use [get_lib_cells */SDFF*]  
#@ End verbose source ./cons/dont_use_45nm.tcl
```

- cons의 .tcl 파일을 읽어 들임
- 사용하지 않는 셀들을 설정함

# Synthesis

## Genus 툴 실행 결과

genus

```
@genus:root: 55> puts "The number of exceptions is [llength [vfind "design:$TOP DESIGN" -exception *]]"  
The number of exceptions is 1
```

```
78 #####  
79 ## Constraints Setup  
80 #####  
81 #read_sdc ./cons/cmsdk_mcu.sdc  
82 source -verbose ./cons/cmsdk_mcu.sdc ; #This command can't write report by using the mark '>'  
83 source -verbose ./cons/dont_use_45nm.tcl ; #This command can't write report by using the mark '>'  
84 puts "The number of exceptions is [llength [vfind "design:$TOP DESIGN" -exception *]]"  
85 #set_db "design:$DESIGN" .force_wireload <wireload name>  
86 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt  
87 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.rpt  
88 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.rpt  
89  
90 echo -n "\n TIMEX > compile is ready : ";  
91 sh date
```

- puts는 메시지를 출력하는 명령임

# Synthesis

genus

## Genus 툴 실행 결과

Lint summary

Unconnected/logic driven clocks	0
Sequential data pins driven by a clock signal	0
Sequential clock pins without clock waveform	0
Sequential clock pins with multiple clock waveforms	0
Generated clocks without clock waveform	0
Generated clocks with incompatible options	0
Generated clocks with multi-master clock	0
Generated clocks with master clock not reaching the generated clock target	0
Paths constrained with different clocks	0
Loop-breaking cells for combinational feedback	0
Nets with multiple drivers	0
Timing exceptions with no effect	0
Suspicious multi_cycle exceptions	0
Pins/ports with conflicting case constants	0
Inputs without clocked external delays	0
Outputs without clocked external delays	0
Inputs without external driver/transition	0
Outputs without external load	0
Exceptions with invalid timing start-/endpoints	0

Total: 0

```
78 #####
79 ## Constraints Setup
80 #####
81
82 #read_sdc ./cons/cmsdk_mcu.sdc
83 source -verbose ./cons/cmsdk_mcu.sdc      ;
   e report by using the mark '>'
84 source -verbose ./cons/dont_use_45nm.tcl ;
   e report by using the mark '>'
85 puts "The number of exceptions is [llength [vfind "design:$TOP_DESIGN
   N" -exception *]]"
86 #set_db "design:$DESIGN" .force_wireload <wireload name>
87
88 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt
89
90 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.
   rpt
91 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.
   rpt
92
93 echo -n "\n TIMEX > compile is ready :           ";
94 sh date
```

- 내가 입력한 constraint에 문제가 없음을 확인

# Synthesis

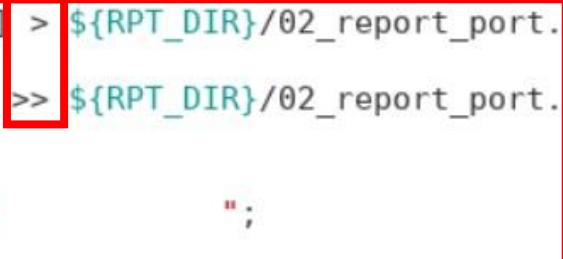
genus

## Genus 툴 실행 결과

```
@genus:root: 58> report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.rpt
@genus:root: 59> report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.rpt
@genus:root: 60> echo -n "\n TIMEX > compile is ready : ";
TIMEX > compile is ready :
Sun Jan 26 22:08:49 KST 2025
@genus:root: 61> sh date
```

```
78 #####
79 ## Constraints Setup
80 #####
81
82 #read_sdc ./cons/cmsdk_mcu.sdc
83 source -verbose ./cons/cmsdk_mcu.sdc      ; #This command can't write report by using the mark '>'
84 source -verbose ./cons/dont_use_45nm.tcl ; #This command can't write report by using the
85 puts "The number of exception is $TOP_DESIGN_N" -exception *]]"
86 #set_db "design:$DESIGN_NAME"
87
88 check_timing_intent -verbose > ${RPT_DIR}/01_check_timing.rpt
89
90 report_port -delay -driver [all_inputs] > ${RPT_DIR}/02_report_port.rpt
91 report_port -delay -load [all_outputs] >> ${RPT_DIR}/02_report_port.rpt
92
93 echo -n "\n TIMEX > compile is ready : ";
94 sh date
```

>: 새로운 레포트를 생성함  
>>: 기존 레포트에 이어서 씀



- 입력 포트와 출력 포트의 지연 값을 02\_report\_port에 입력함

# Synthesis

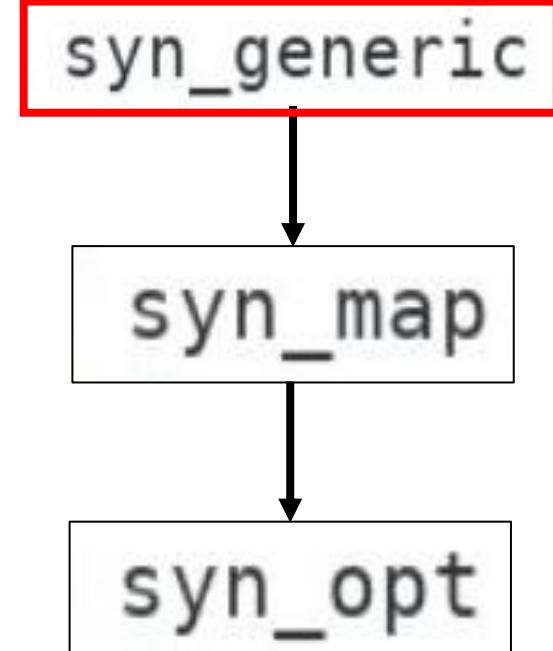
## Genus 툴 실행 결과

genus

```
@genus:root: 62> write_db -to_file ${UNMAPPED_DIR}/syn_generic.db  
Finished exporting design database to file './unmapped/syn_generic.db' for 'cmsdk_mc  
u' (command execution time mm:ss cpu = 00:01, real = 00:02).
```

- 합성의 세가지 단계 중 첫번째 단계임
- unmapped에 report 생성

```
96 #####  
97 ## Synthesizing to generic  
98 #####  
99  
100 syn_generic ; #This command can't write report by using the mark '>'  
101 #syn_generic -physical  
102 puts "Runtime & Memory after 'syn_generic'"  
103 time_info GENERIC  
104 report_datapath > ${RPT_DIR}/03_report_datapath_generic.rpt  
105  
106 write_snapshot -outdir ${RPT_DIR} -tag syn_generic  
107 # Below 2 lines are the same with the above command :: write_snapshot  
108 #report_summary -directory ${REPORTS_PATH}  
109 #summary_table -directory ${RPT_DIR}  
110  
111 write_db -to_file ${UNMAPPED_DIR}/syn_generic.db
```

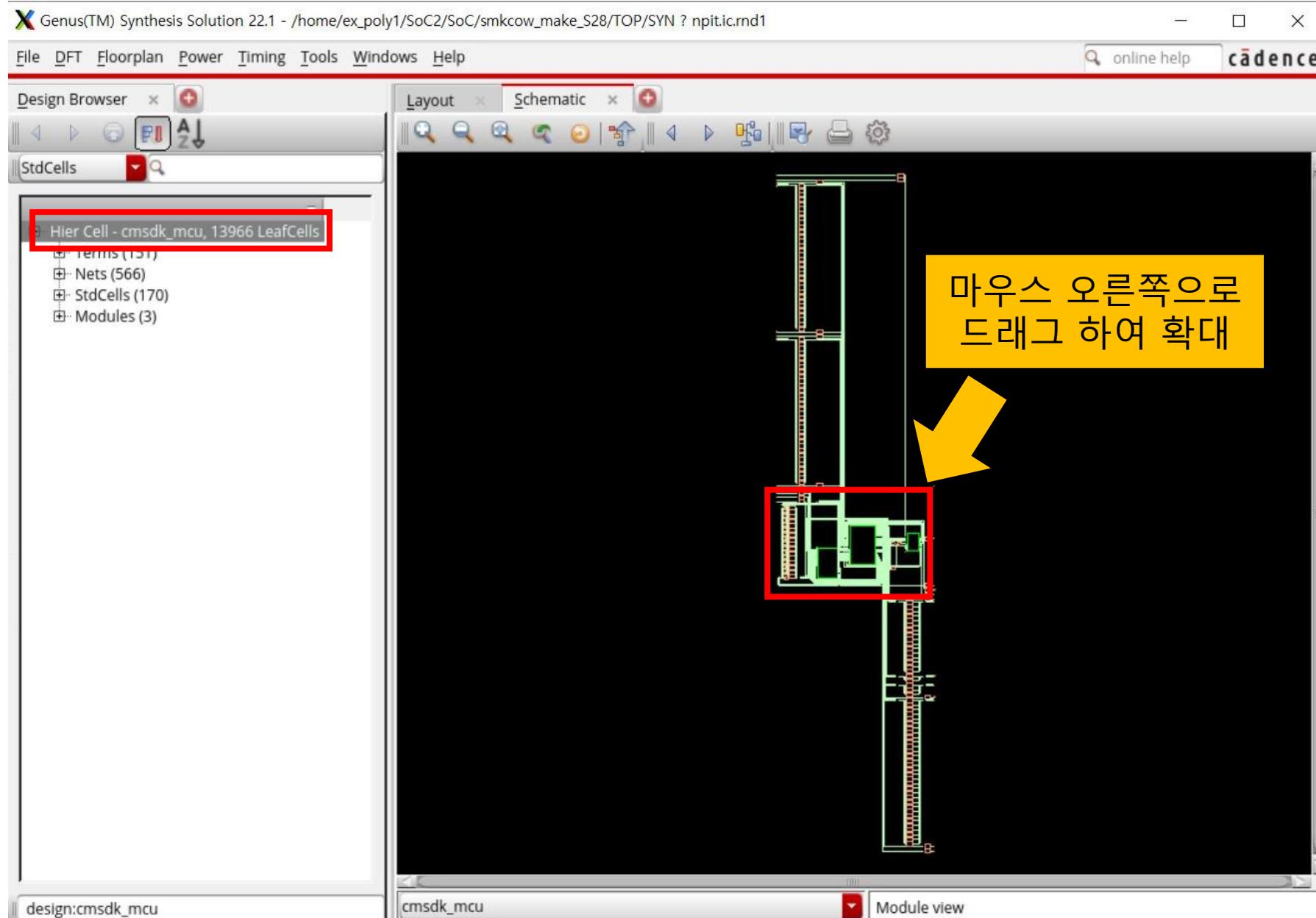


# Synthesis

genus

\$> gui\_show

- syn\_generic 실행 후  
TOP모듈 확인

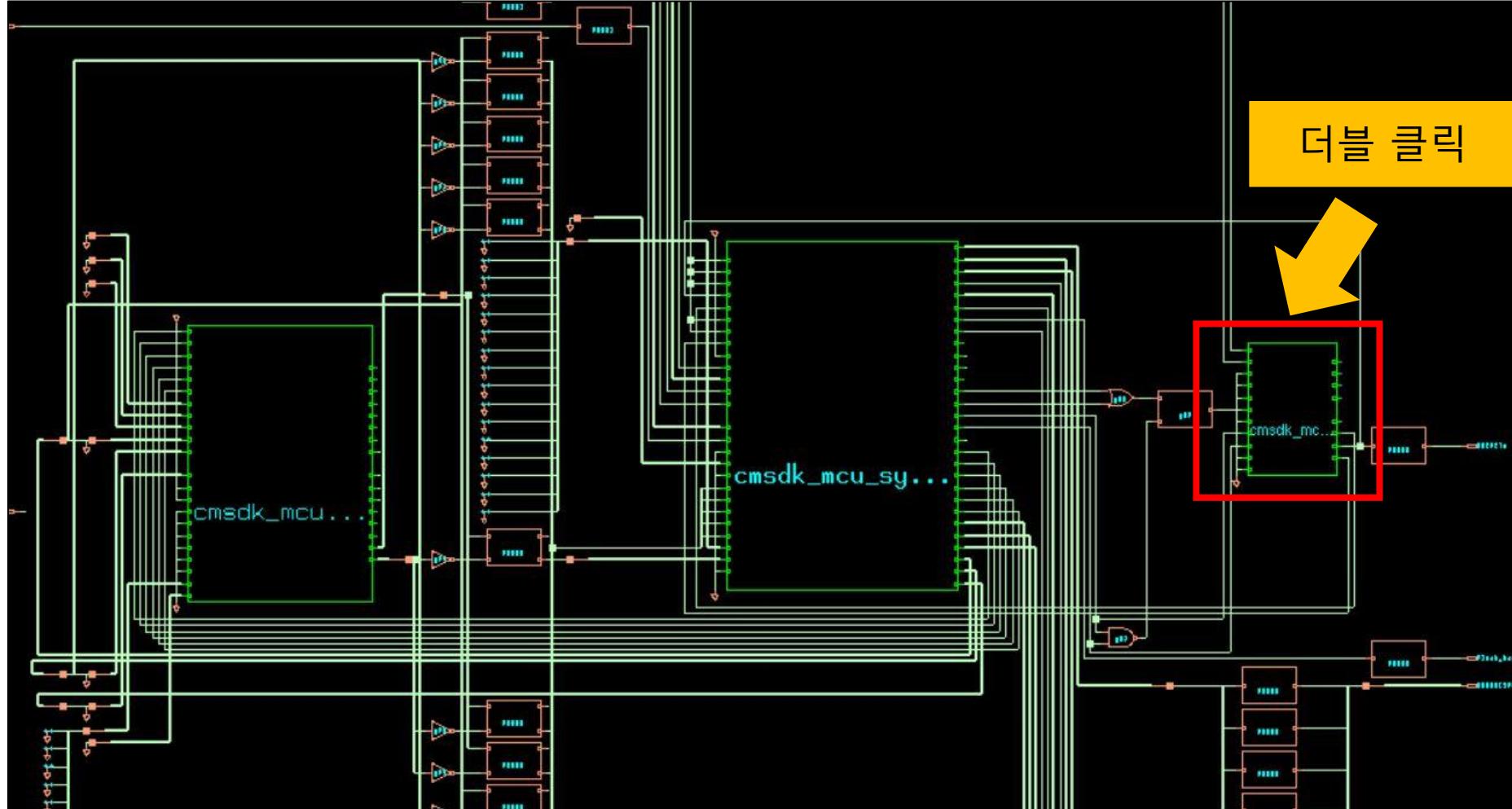


# Synthesis

genus

\$> gui\_show

- syn\_generic 실행 후 schematic 확인

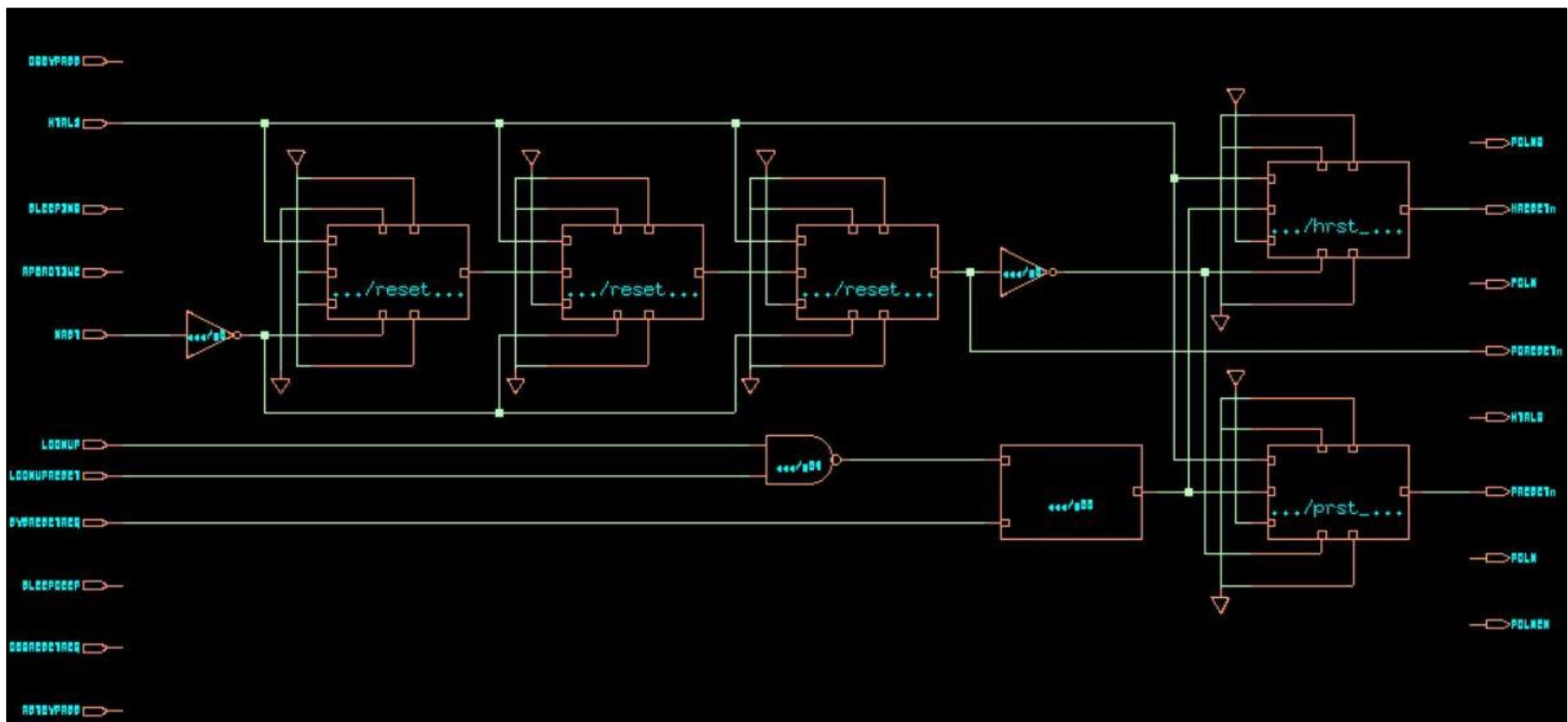


# Synthesis

genus

\$> gui\_show

- syn\_generic 실행 후 schematic 확인



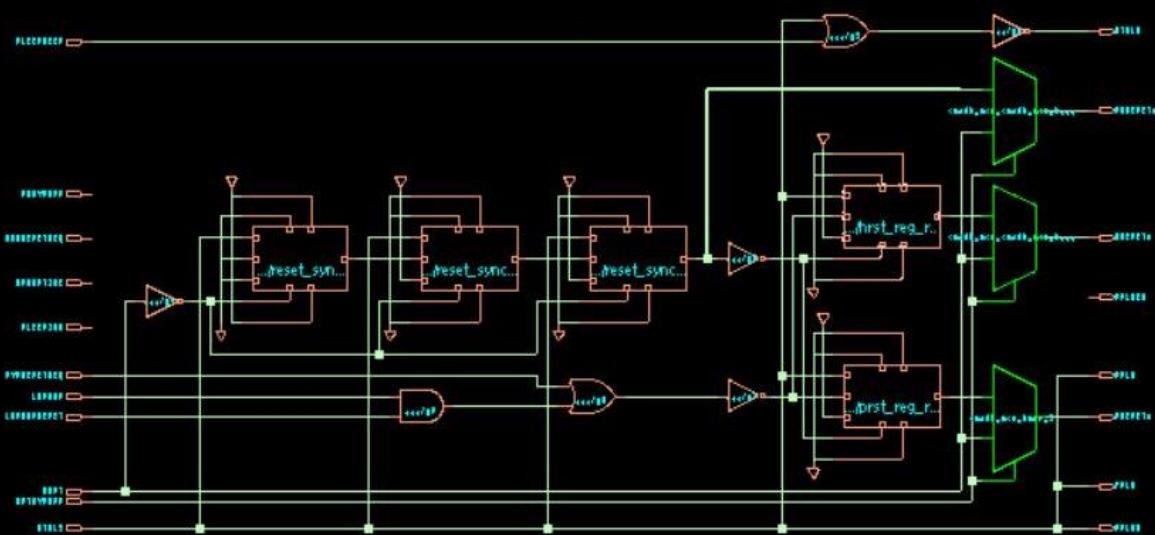
# Synthesis

genus

- 합성 전 후 비교

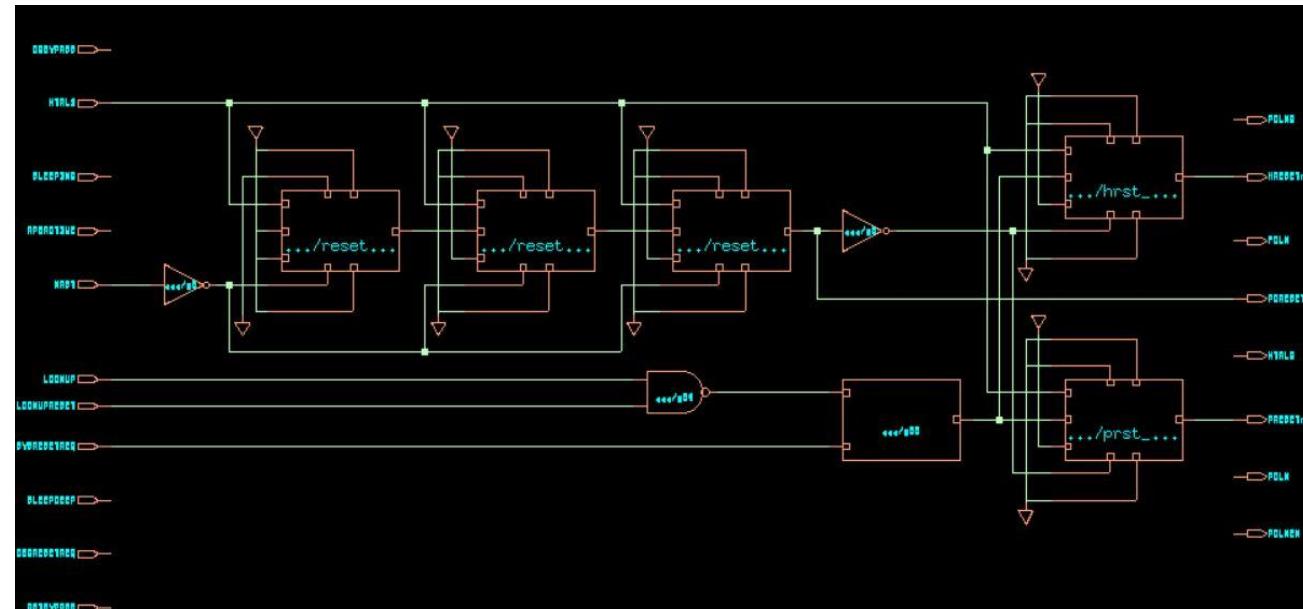
합성 전

(syn\_generic 실행 전)



합성 후

(syn\_generic 실행 후)

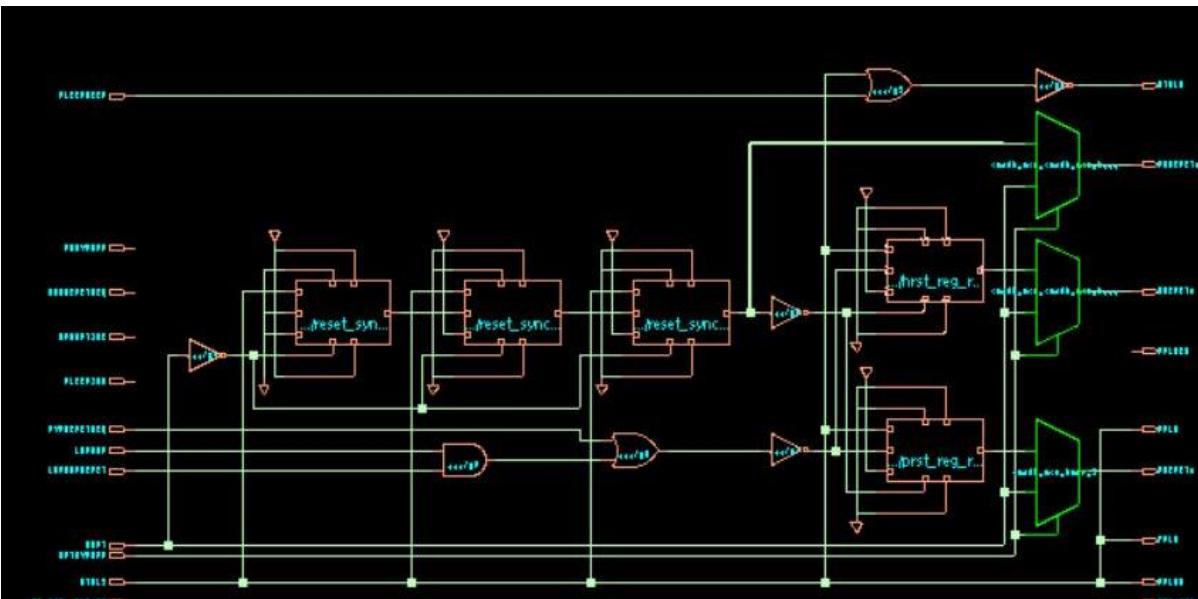


# Synthesis

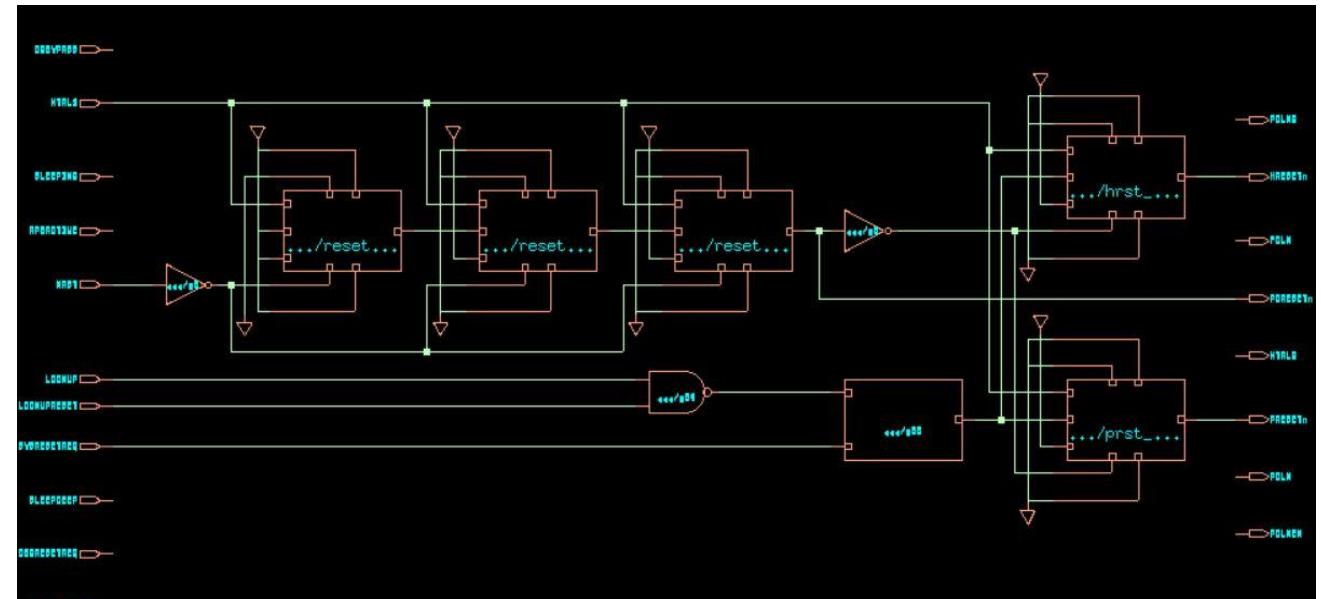
## genus

- 합성 전 후 비교
- 셀 이름은 동일함

합성 전  
(syn\_generic 실행 전)



합성 후  
(syn\_generic 실행 후)



# Synthesis

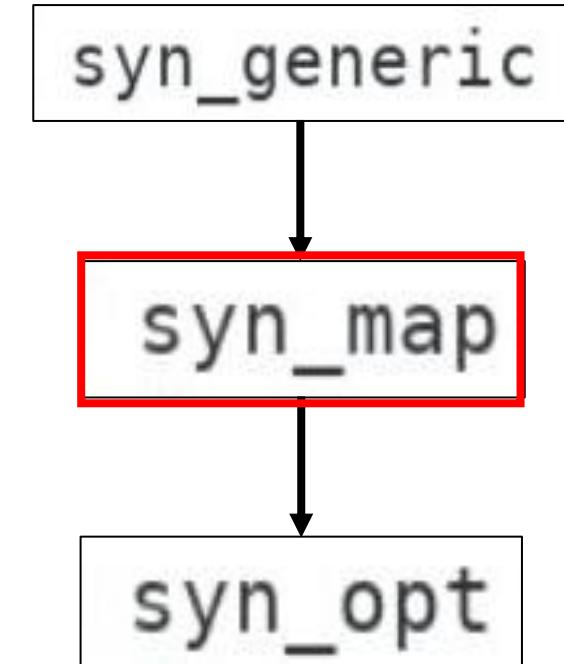
genus

Genus 툴 실행 결과

```
@genus:root: 73> write_db -to_file ${MAPPED_DIR}/syn_map.db
Finished exporting design database to file './mapped/syn_map.db' for 'cmsdk_mcu' (co
mmand execution time mm:ss cpu = 00:01, real = 00:01).
```

- 합성의 세가지 단계 중 두번째 단계임
- 공정사의 STD셀과 연결하는 Mapping 단계임

```
113 #####
114 ## Synthesizing to gates
115 #####
116
117 syn_map ; #This command can't write report by using the mark '>'
118 #syn_map -physical
119 puts "Runtime & Memory after 'syn_map'"
120 time_info MAPPED
121
122 report_datapath > ${RPT_DIR}/04_report_datapath_map.rpt
123
124 foreach cg [vfind / -cost_group *] {
125   report timing -cost_group [list $cg] > ${RPT_DIR}/05_[basename $cg
126 ]_post_syn_map.rpt
127 }
128
129 write_snapshot -outdir ${RPT_DIR} -tag syn_map
130 # Below line is the same with the above command :: write_snapshot
131 #summary_table -directory ${RPT_DIR}
132 write_db -to_file ${MAPPED_DIR}/syn_map.db
```

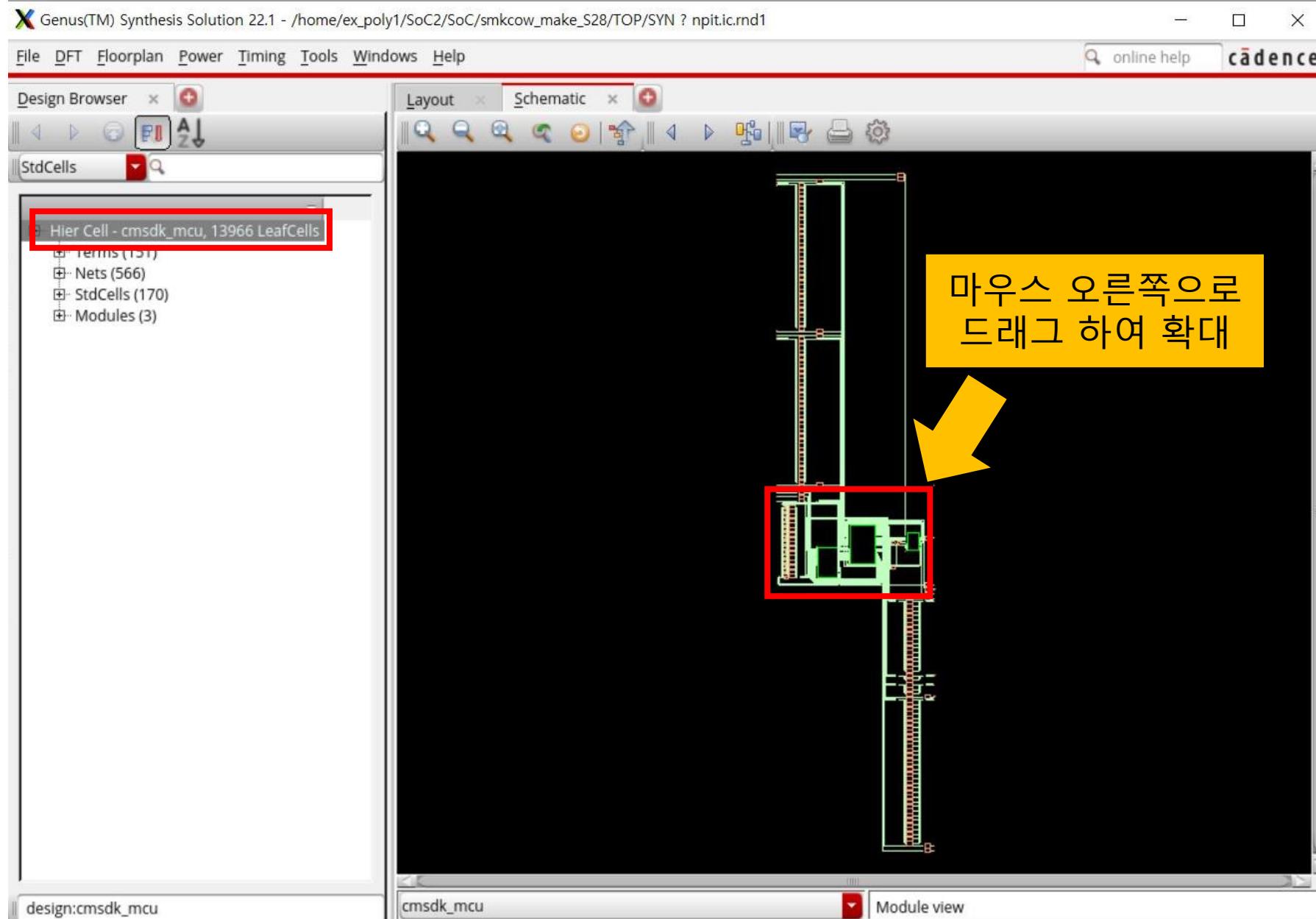


# Synthesis

genus

\$> gui\_show

- syn\_map 실행 후 TOP 모듈 확인

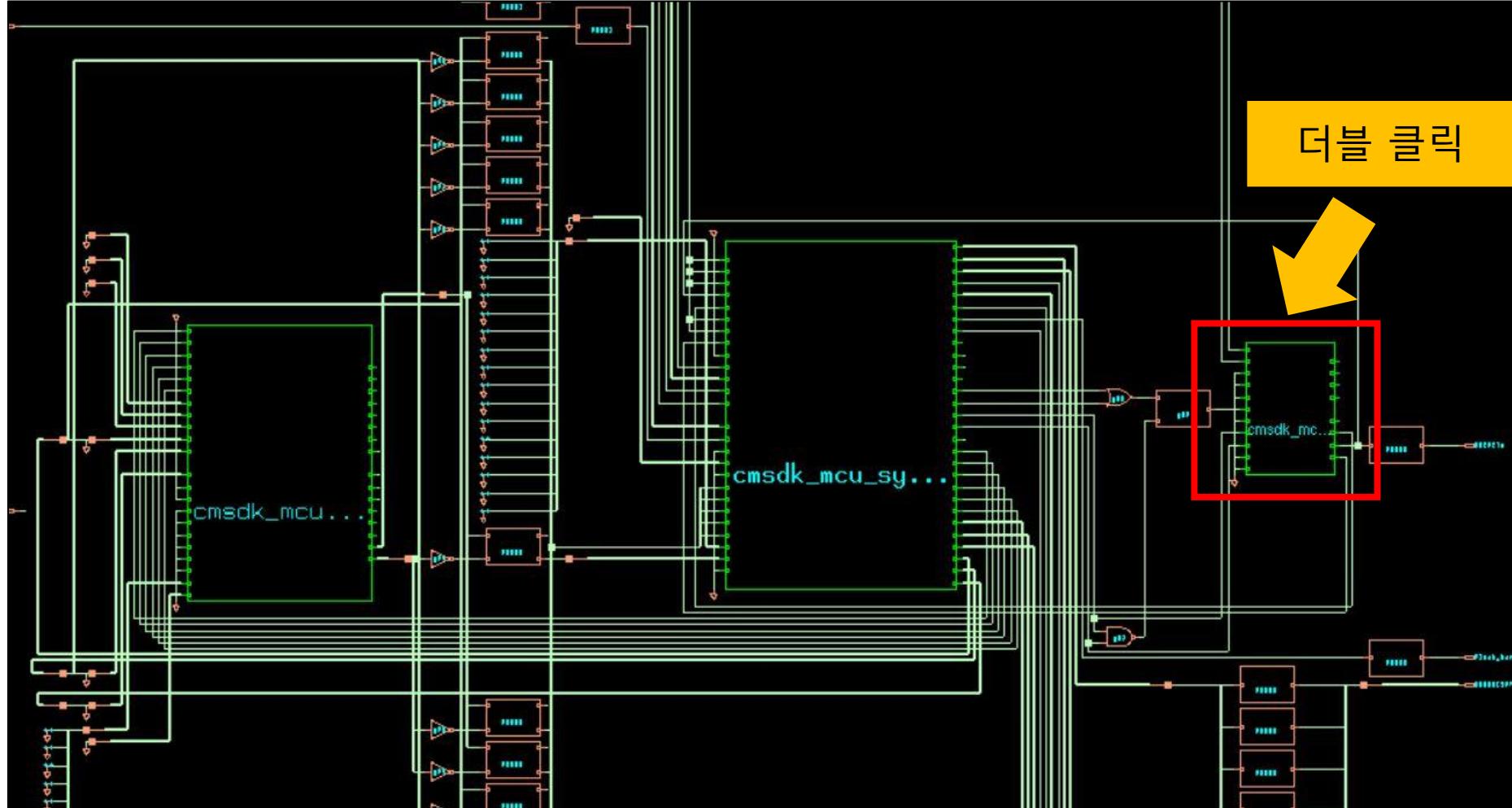


# Synthesis

genus

\$> gui\_show

- syn\_map 실행 후 schematic 확인

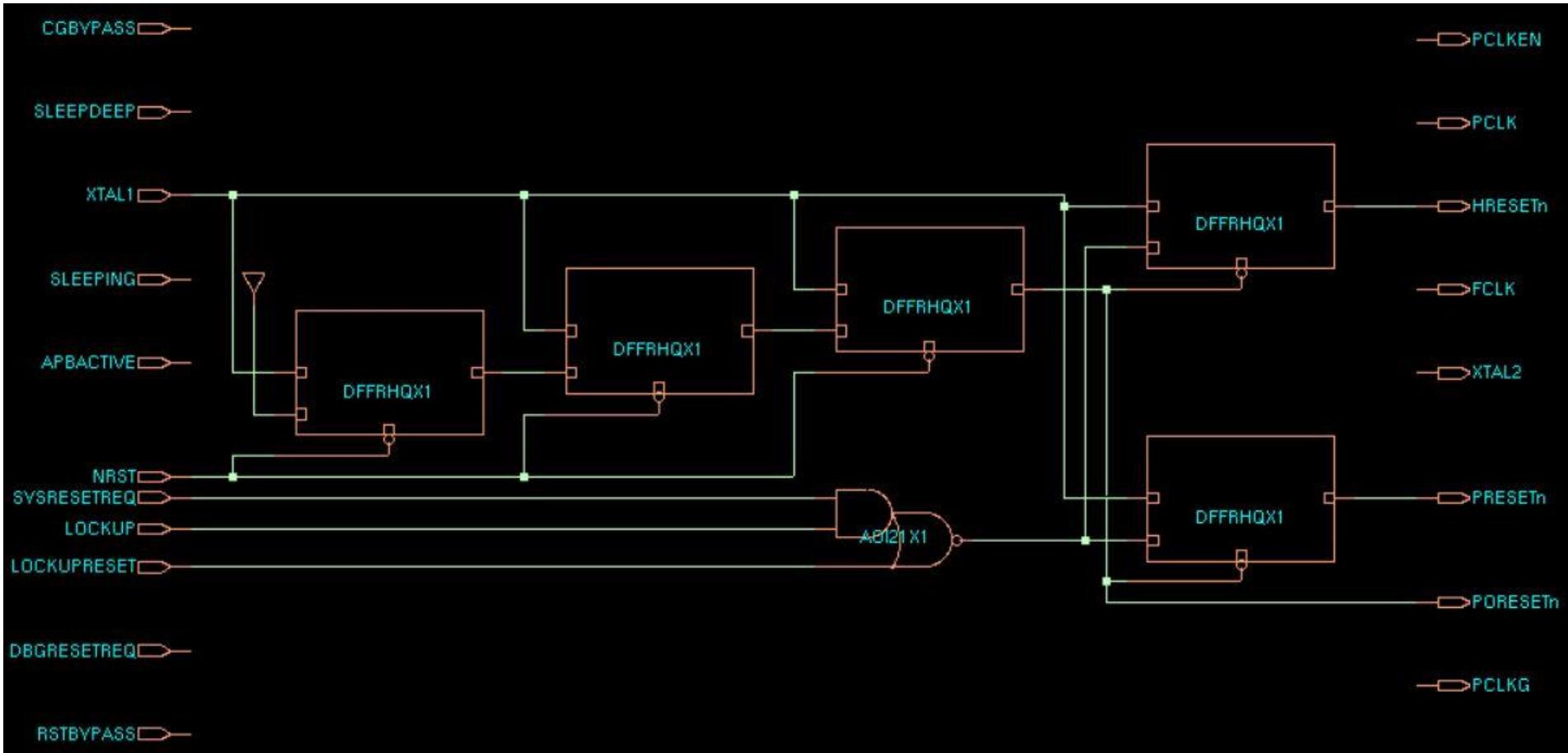


# Synthesis

genus

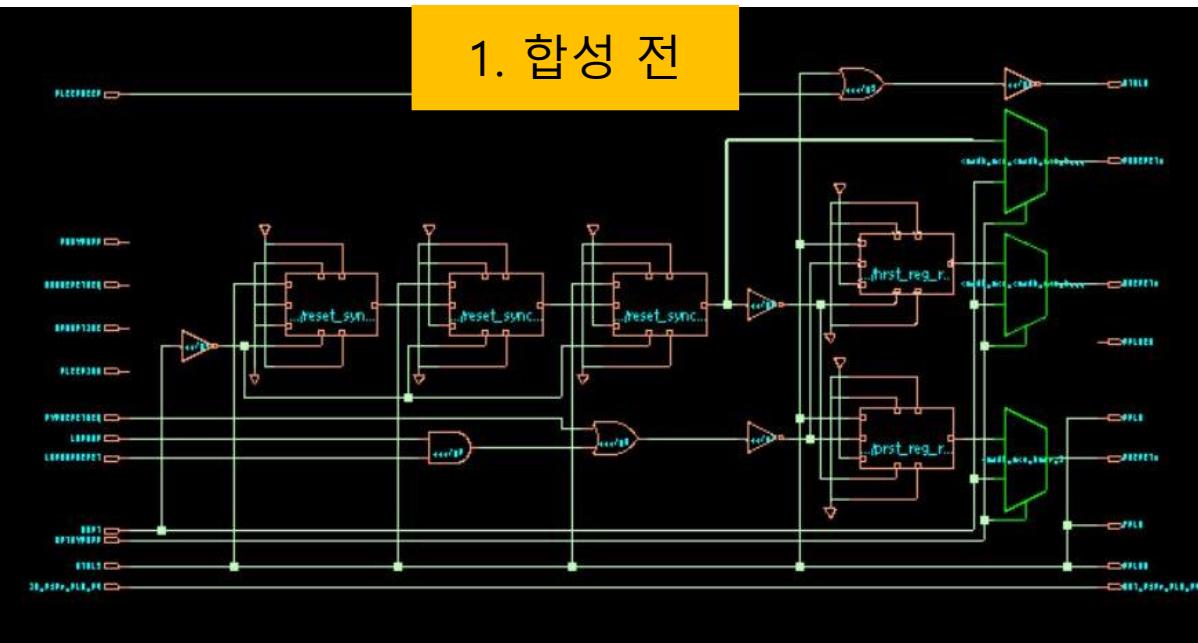
\$> gui\_show

- syn\_map 실행 후 schematic 확인
- 셀의 이름이 바뀌었고 잘 정리되어 있는 것을 볼 수 있음

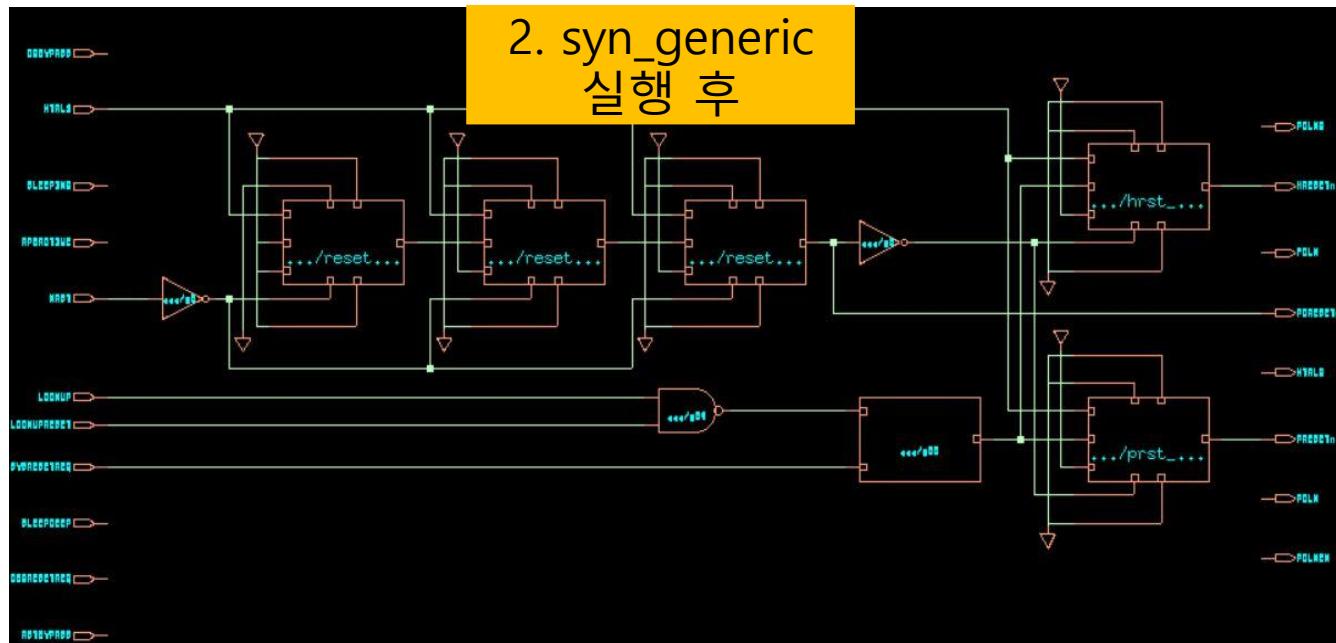


# Synthesis

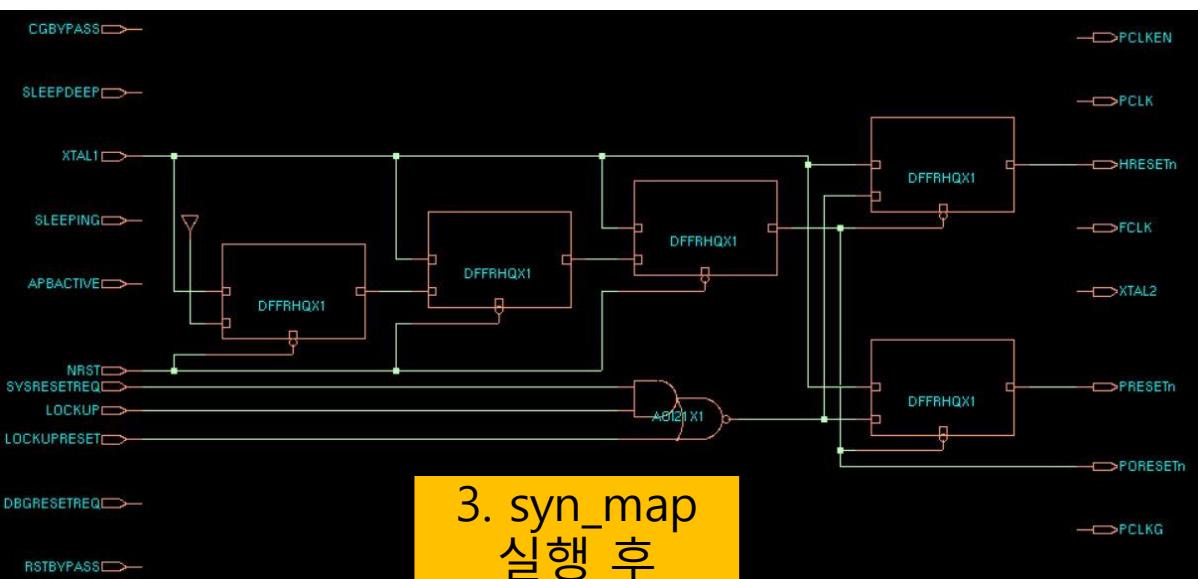
1. 합성 전



2. syn\_generic  
실행 후



3. syn\_map  
실행 후



# Synthesis

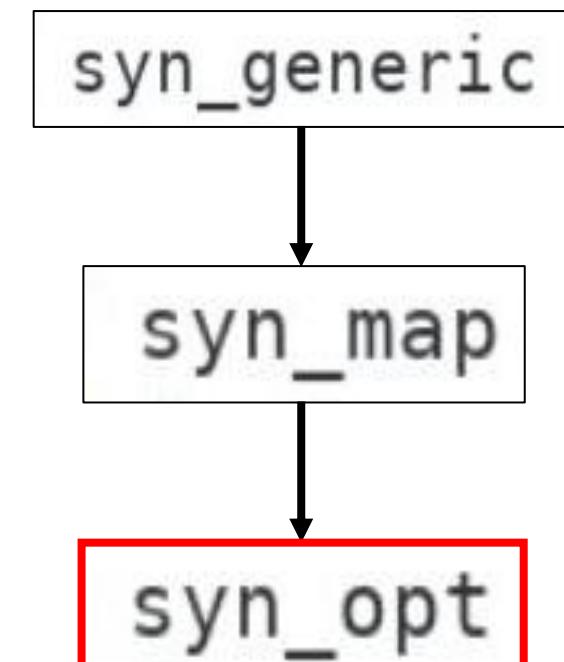
## Genus 툴 실행 결과

genus

```
@genus:root: 85> echo -n "\n TIMEX > Compile is completed : ";
TIMEX > Compile is completed :
Sun Jan 26 23:06:15 KST 2025
@genus:root: 86> sh date
```

- 합성의 세가지 단계 중 마지막 단계임
- Design을 최적화 하는 optimize 단계임

```
134 #####
135 ## Optimize Netlist + Incremental
136 #####
137
138 syn_opt ; #This command can't write report by using the mark '>'
139 #syn_opt -incremental
140 #syn_opt -physical
141
142 report_constraint -all_violators > ${RPT_DIR}/06_report_constraint.rpt
143
144 write_snapshot -outdir ${RPT_DIR} -tag syn_opt
145 # Below line is the same with the above command :: write_snapshot
146 #summary_table -directory ${RPT_DIR}
147
148 puts "Runtime & Memory after syn_opt"
149 time_info OPT
150
151 foreach cg [vfind / -cost_group *] {
152   report timing -cost_group [list $cg] > ${RPT_DIR}/07_[basename $cg]_post_syn_opt.rpt
153 }
154
155 write_db -to_file ${MAPPED_DIR}/syn_opt.db
156
157
158 echo -n "\n TIMEX > Compile is completed : ";
159 sh date
```



# Synthesis

## genus

- Constraint에 만족하는 디자인이 나왔는지 검증

### Genus 툴 실행 결과

```
134 ##### Incremental #####
13 report_constraint      Incremental
13                                     #####
13                                     실행
138 syn_opt ; "This command can't write report by using the mark
139 #syn_opt incremental
140 #syn_opt physical
141
142 report_constraint -all_violators > ${RPT_DIR}/06_report_cons
143
144 write_snapshot -outdir ${RPT_DIR} -tag syn_opt
145 # Below line is the same with the above command :: write_sna
146 #summary_table -directory ${RPT_DIR}
147
148 puts "Runtime & Memory after syn_opt"
149 time_info OPT
150
151 foreach cg [vfind / -cost_group *] {
152   report timing -cost_group [list $cg] > ${RPT_DIR}/07_[base
153 }
154
155 write_db -to_file ${MAPPED_DIR}/syn_opt.db
156
157
158 echo -n "\n TIMEX > Compile is completed :           ";
159 sh date
```

```
@genus:root: 8> report_constraint -all_violators
=====
Generated by:          Genus(TM) Synthesis Solution 22.16-s078_1
Generated on:          Feb 06 2025 11:18:16 pm
Module:                cmsdk_mcu
Operating conditions: PVT_0P9V_125C
Interconnect mode:    global
Area mode:             Setup Timing, clock_period,
                        pulse_width에 문제 없음
=====
Checking for violation type : Setup Timing Slack
-----
No timing slack violation found.
-----
Checking for violation type : clock_period
-----
No paths found.
-----
Checking for violation type : pulse_width
-----
No paths found.
```

# Synthesis

genus

## Genus 툴 실행 결과

Checking for violation type : max\_capacitance

Pin	Required Load (ff)	Actual Load (ff)	Slack Load (ff)	Mode
port:cmsdk_mcu/P1[0]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[1]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[2]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[3]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[4]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[5]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[6]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[7]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[8]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[9]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[10]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[11]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[12]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[13]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[14]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[15]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/flash_hrd[14]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[13]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[12]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[11]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[10]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[9]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[8]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrd[7]	888.30	2385.40	-1497.10	default

- max\_capacitance에 Required Load보다 Actual Load가 크게 걸림
- Io셀을 이미 넣은 상황이기 때문에 **수정 불가**
- BE과정에서 개선의 여지가 있기 때문에 일단 유지

# Synthesis

## genus

### Genus 툴 실행 결과

Checking for violation type : max\_transition

Pin	Required Slew (ps)	Actual Slew (ps)	Slack Slew (ps)	Mode
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4850/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4851/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4852/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4853/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4854/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4855/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4856/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4857/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4858/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4859/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4860/B1	280	915	-635	default
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4861/B1	280	915	-635	default

No max\_fanout rule violations.

fanout 문제 없음

# Synthesis

## genus

- report\_timing 명령으로 합성 결과 중 WNS(제일 좋지 않은 Path) 결과 확인 가능
- Slack에 음수가 아닌 양수 값이 나옴으로 문제가 없음을 확인할 수 있음

Genus 툴 실행 결과

```
134 #####  
135 ## Optimize Netlist + Incremental  
136 #####  
137  
138 syn_opt ; #This command can't write repo  
139 #syn_opt -incremental  
140 #syn_opt -physical  
141  
142 report_constraint -all_violators > ${RPT_DIR}/constraint.rpt  
143  
144 write_snapshot -outdir ${RPT_DIR} -tag snapshot  
145 # same with the above  
146 report_timing  
147 실행  
148 puts "Report time & Memory after syn_opt"  
149 time_in OPT  
150  
151 foreach cg [vfind / -cost_group *] {  
152   report timing -cost_group [list $cg]  
153 }  
154  
155 write_db -to_file ${MAPPED_DIR}/syn_opt.db  
156  
157  
158 echo -n "\n TIMEX > Compile is completed  
159 sh date
```

Path 1: MET (1645 ps) Setup Check with Pin u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/0o3l85\_reg/CKN->D  
Group: MAIN\_CLOCK  
Startpoint: (R) u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/Xc8l85\_reg/CK  
Clock: (R) MAIN\_CLOCK  
Endpoint: (F) u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/0o3l85\_reg/D  
Clock: (R) MAIN\_CLOCK

Capture	Launch
Clock Edge:+ 7000	0
Src Latency:+ 200	200
Net Latency:+ 200 (I)	200 (I)
Arrival:= 7400	400

Setup:- 31  
Uncertainty:- 100  
Required Time:= 7269  
Launch Clock:- 400  
Data Path:- 5225  
Slack:= 1645

Timing Point

Arrival Instance	Flags	Arc	Edge	Cell	Fanout	Load	Trans	Delay
(ps) Location	(ff)	(ps)	(ps)					

u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/Xc8l85\_reg/CK - - R (arrival) 1445 - 100 0  
400 (-,-)  
u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/Xc8l85\_reg/Q (P) CK->Q R DFFRX1 37 45.0 306 242  
642 (-,-)  
u\_cmsdk\_mcu\_system/u\_cortexm0integration/u\_cortexm0/u\_logic/g215877/Y (P) A->Y R OR2X1 19 22.5 155 152  
794 (-,-)

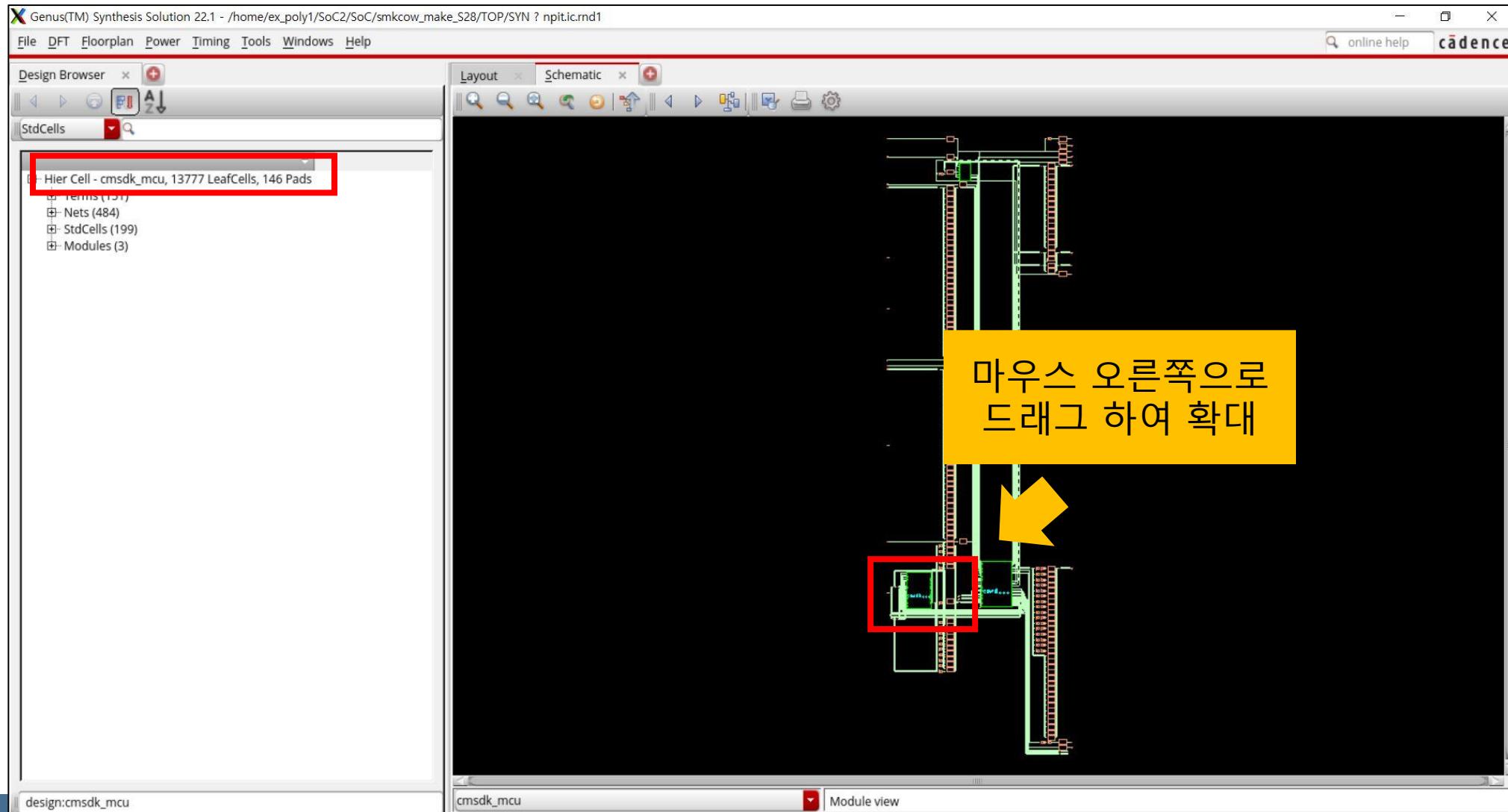
문제 없음을 확인

# Synthesis

\$> gui\_show

genus

- syn\_opt 실행 후 TOP모듈 확인

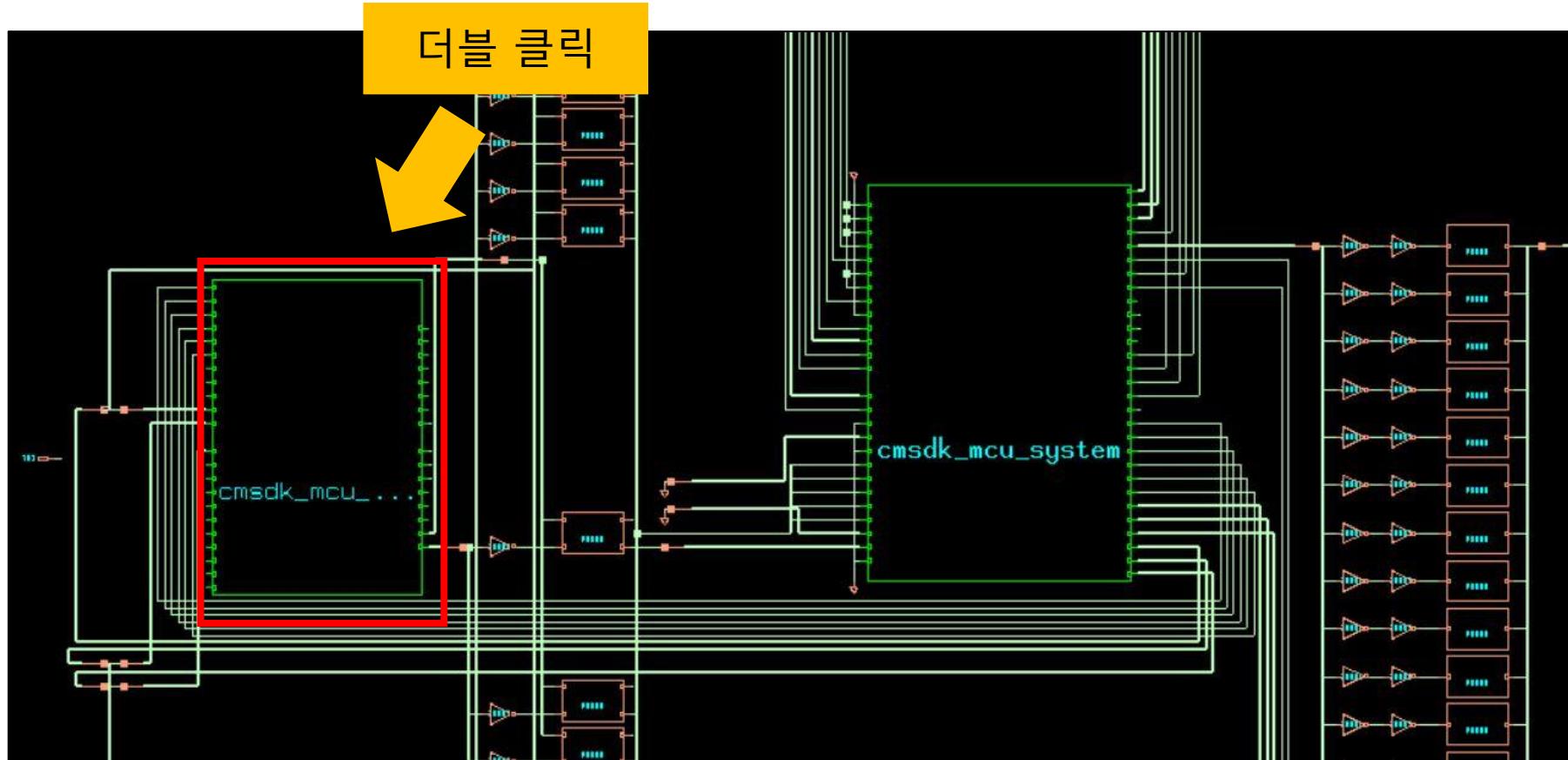


# Synthesis

genus

\$> gui\_show

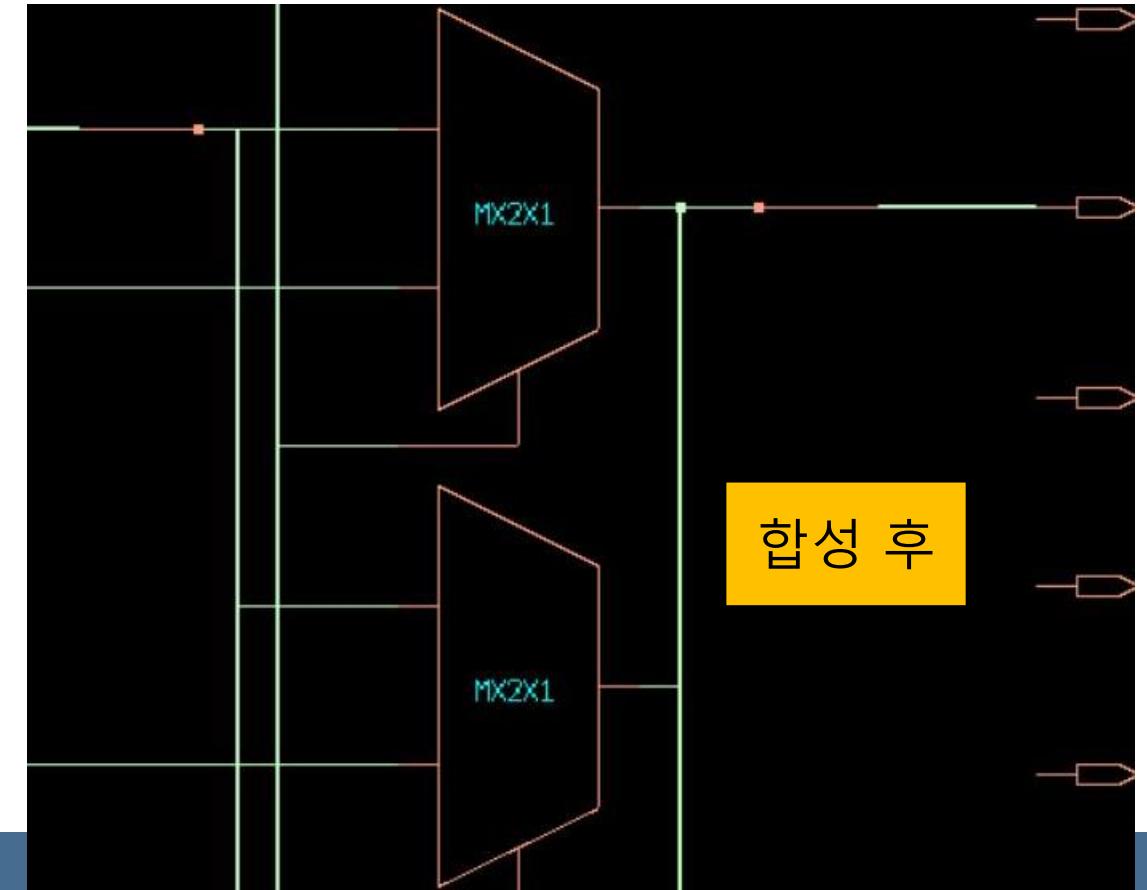
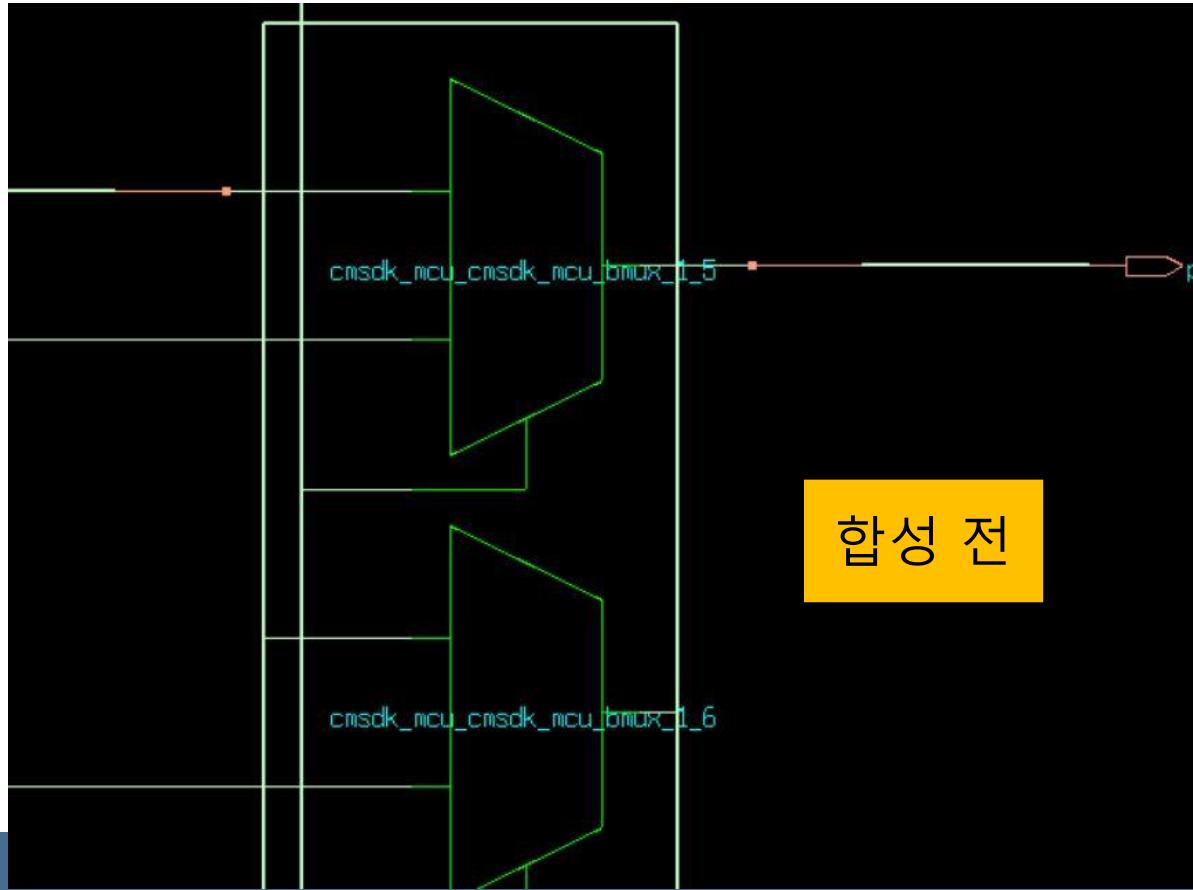
- syn\_opt 실행 후 schematic 확인



# Synthesis

## genus

- 합성 전 후 비교
- 공정사의 STD셀로 바뀐 것을 확인할 수 있음



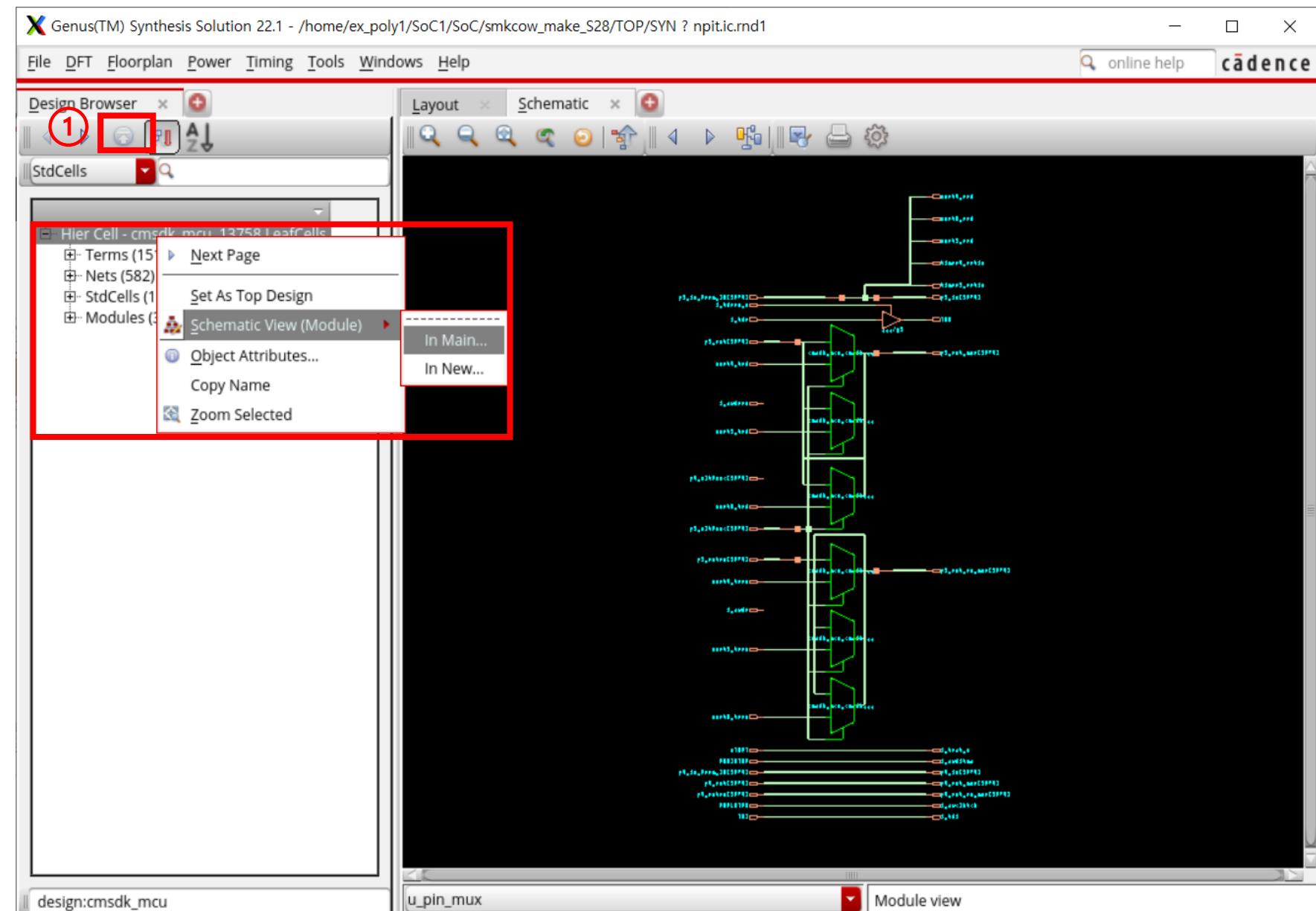
# Synthesis

genus

- TOP 디자인으로 이동

1번 클릭

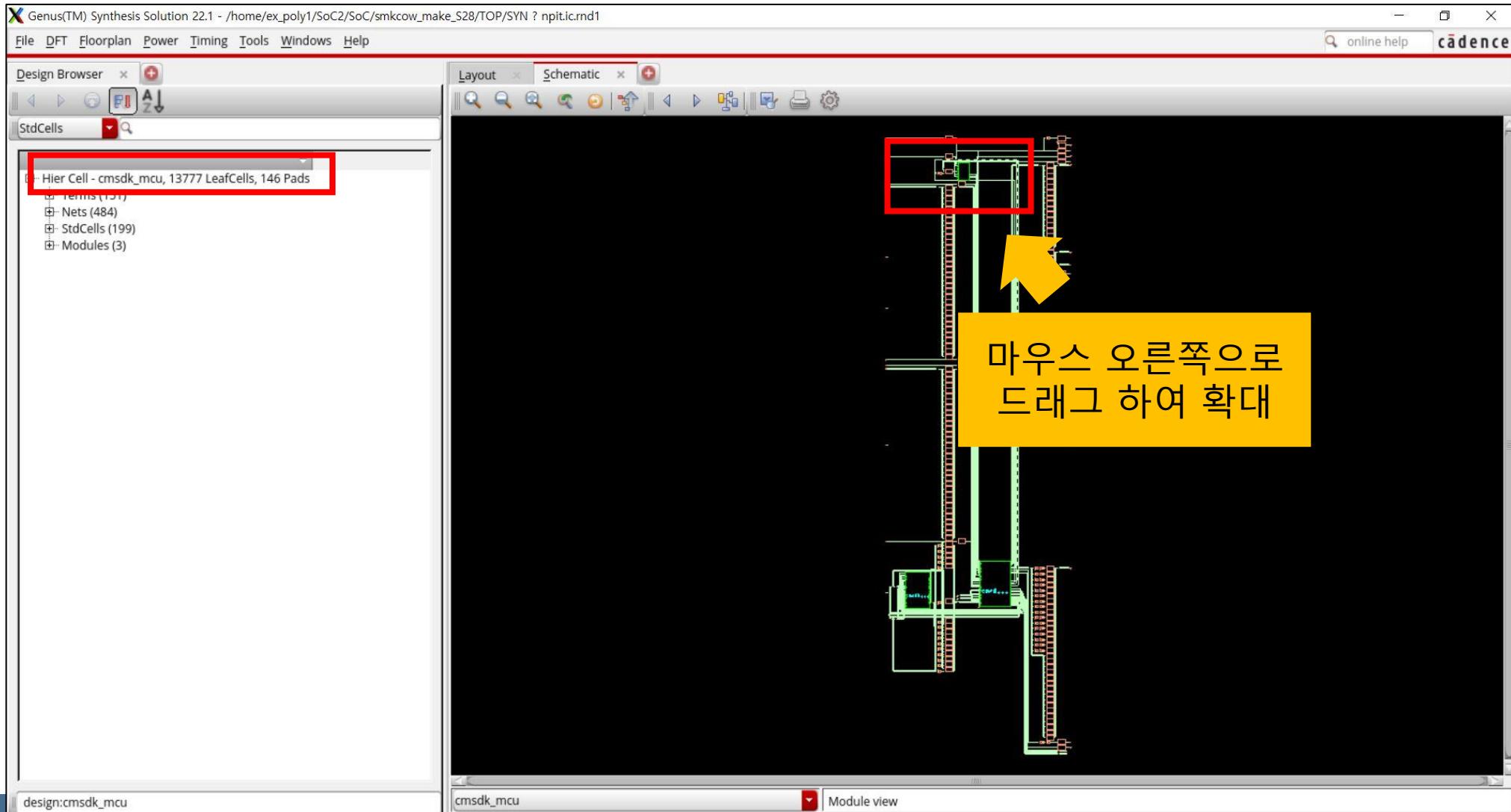
- Hier Cell 우 클릭
- schematic view 클릭
- In Main 클릭



# Synthesis

genus

\$> gui\_show

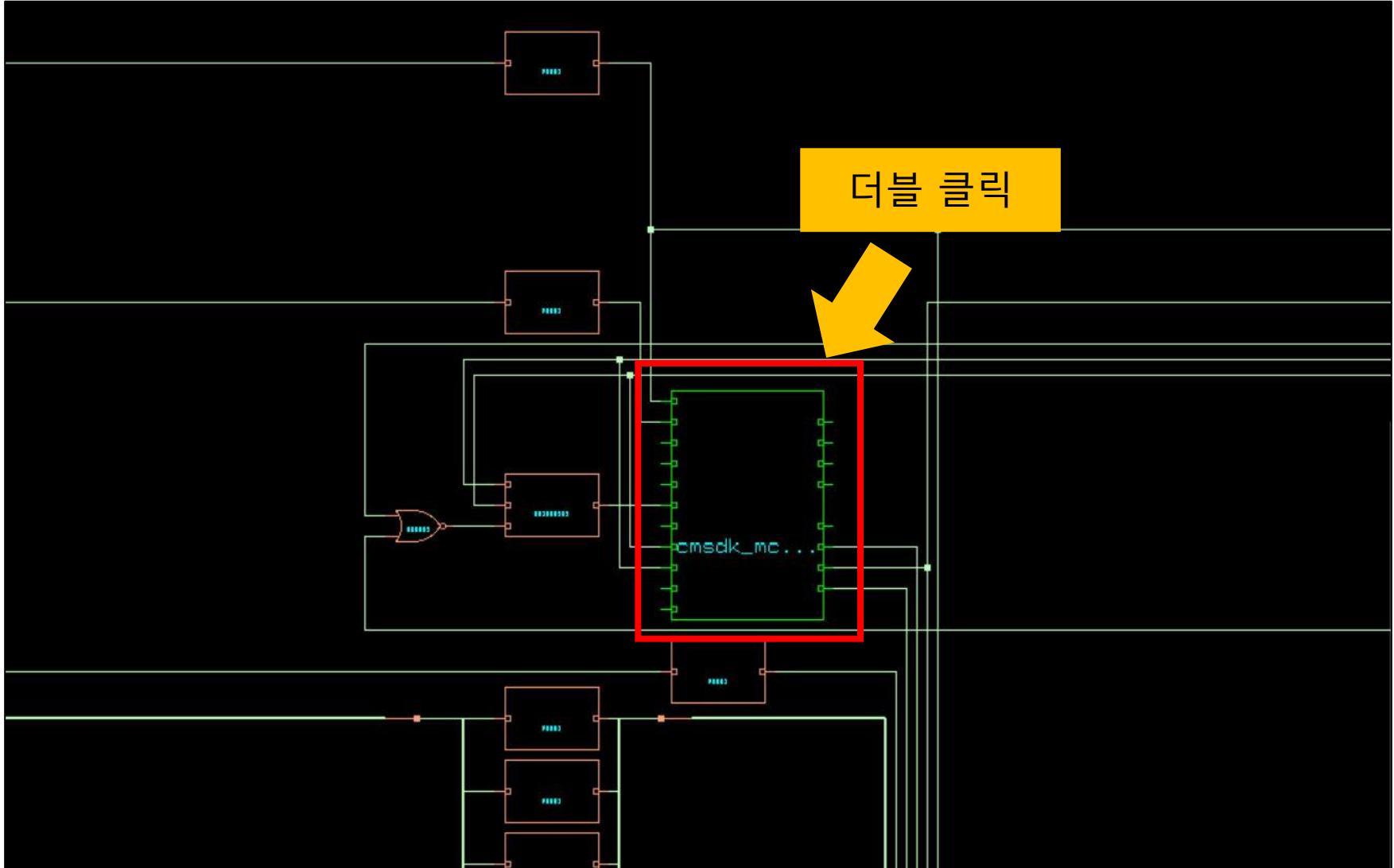


# Synthesis

genus

\$> gui\_show

- syn\_opt 실행 후 schematic 확인

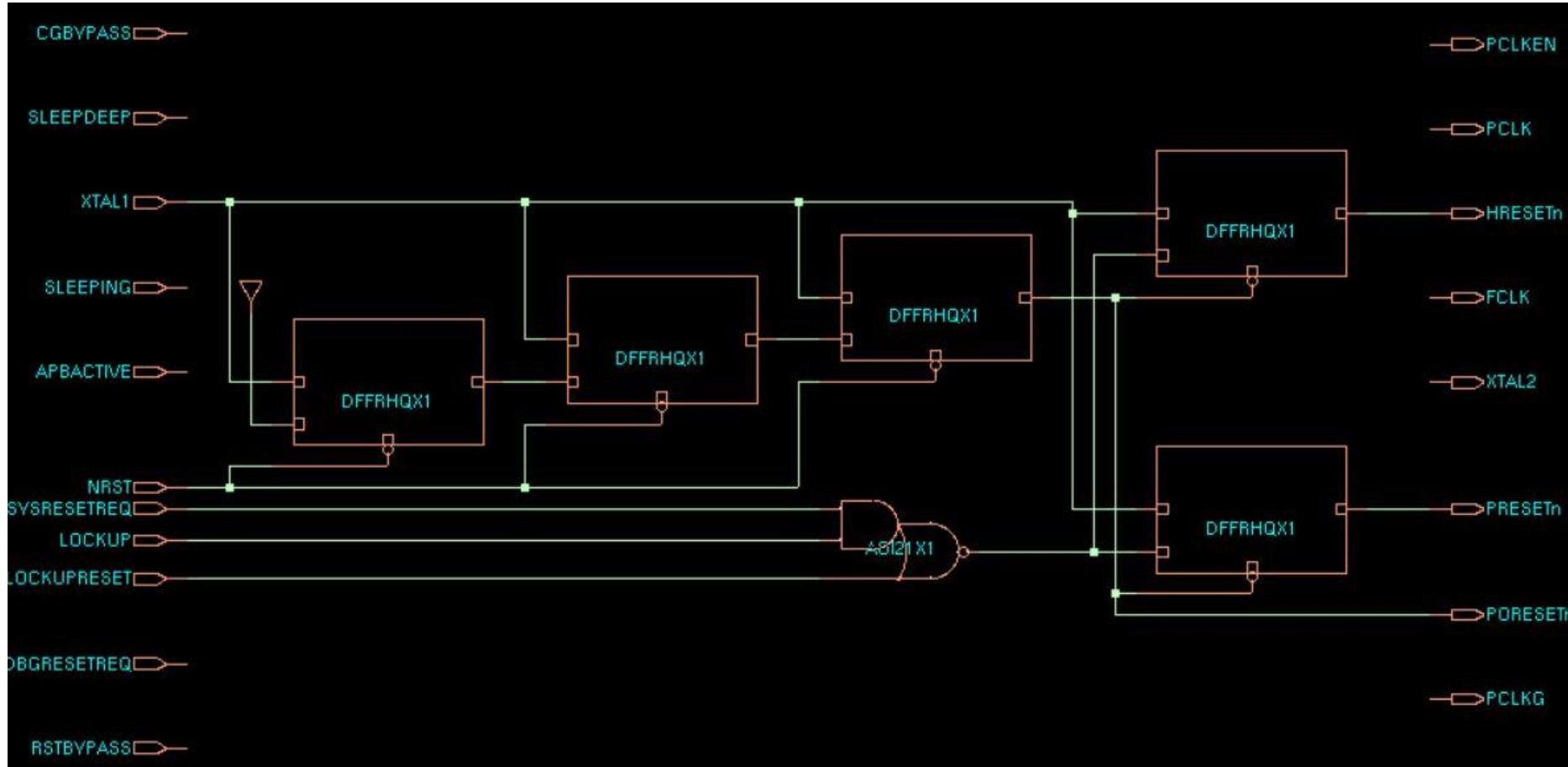


# Synthesis

genus

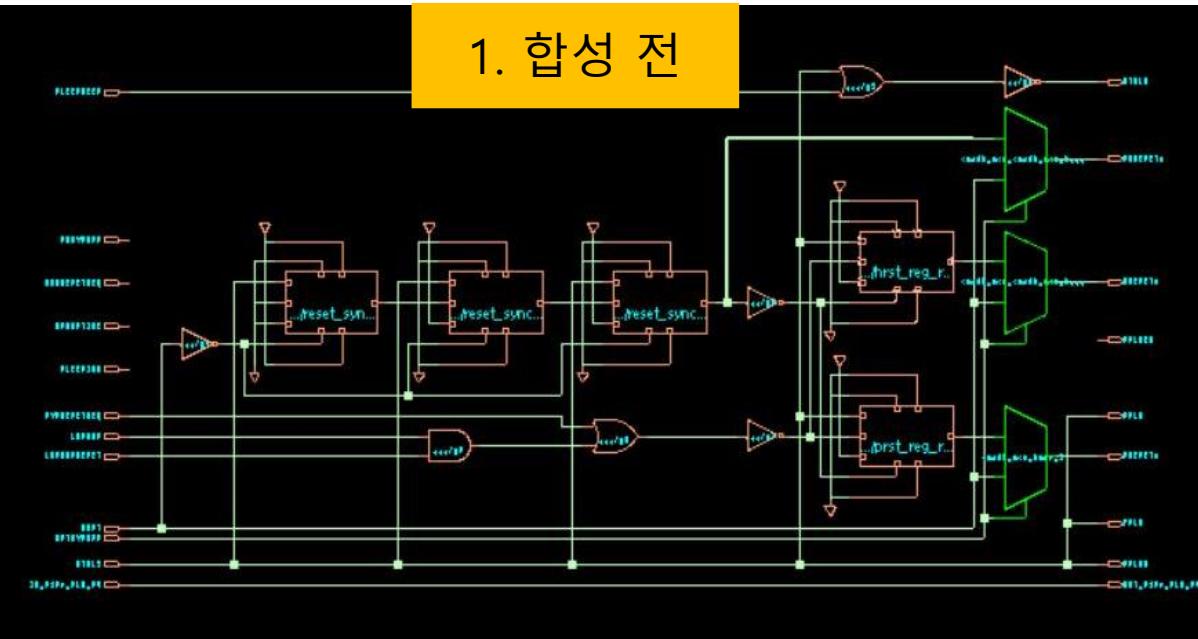
\$> gui\_show

- syn\_opt 실행 후 schematic 확인
- syn\_map과 차이가 없음  
→ 이 모듈에는 최적화할 곳이 없기 때문임

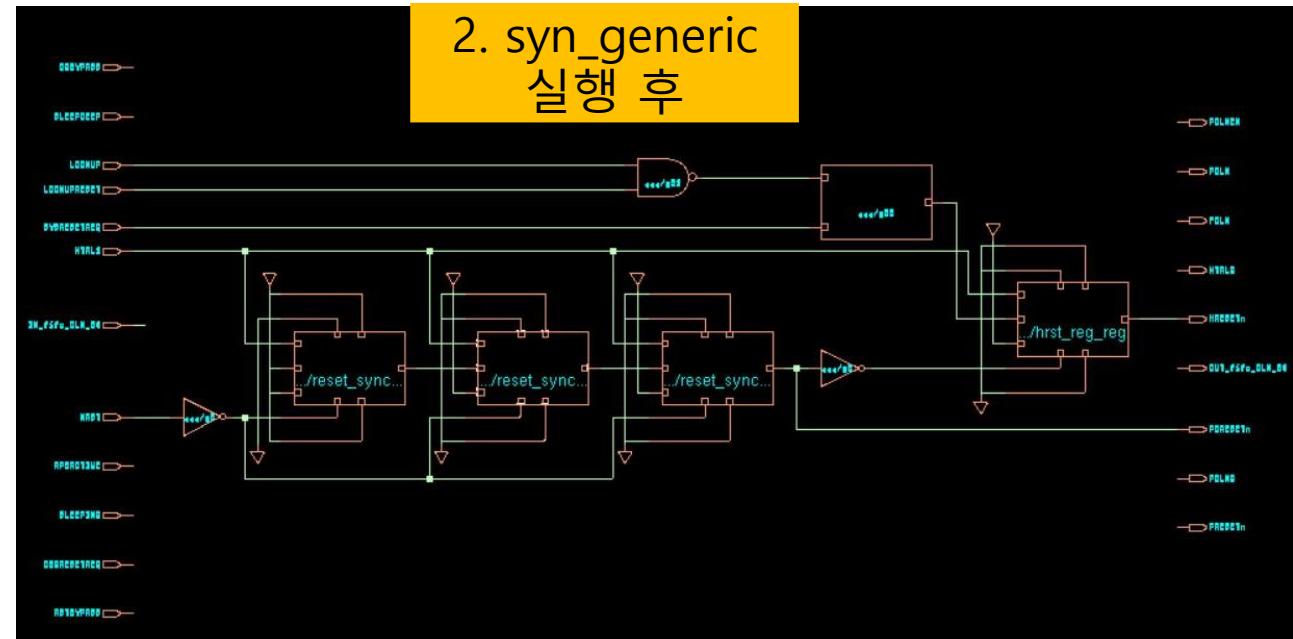


# Synthesis

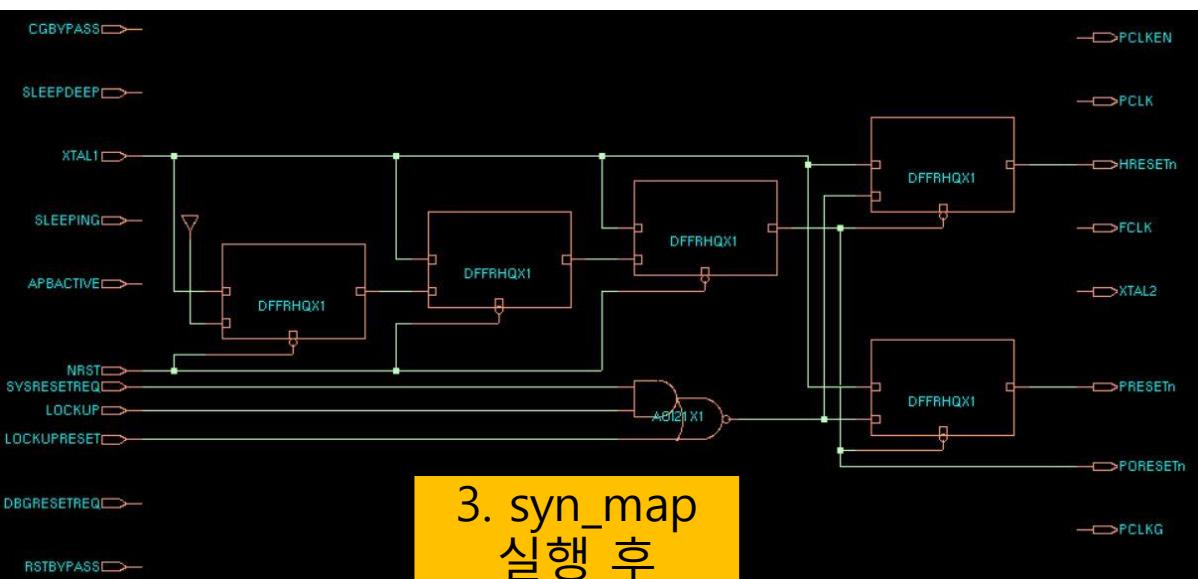
1. 합성 전



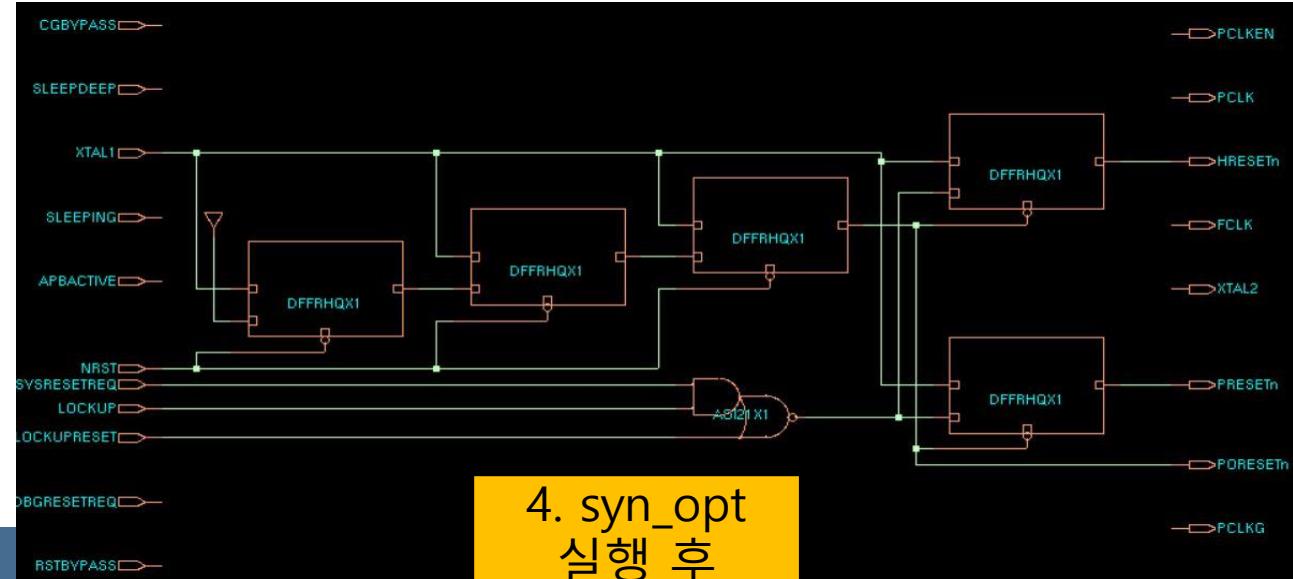
2. syn\_generic  
실행 후



3. syn\_map  
실행 후



4. syn\_opt  
실행 후



# Synthesis

## Genus 툴 실행 결과

genus

```
Warning: Library 'fast_vdd1v0' found in multiple domains.  
Info: Use -domain to uniquify.  
Info: Selecting library 'fast_vdd1v0' in domain '-1'  
Warning: Library 'giolib045' found in multiple domains.  
Info: Use -domain to uniquify.  
Info: Selecting library 'giolib045' in domain '-1'  
Warning: Library 'slow_vdd1v0' found in multiple domains.  
Info: Use -domain to uniquify.  
Info: Selecting library 'slow_vdd1v0' in domain '-1'  
Info : Joules engine is used. [RPT-16]  
      : Joules engine is being used for the command report_power.  
Output file: ./report/18_cmsdk_mcu_SYN_pwr.rpt  
@file(cortexm0_45nm.tcl) 178: report_qor  
@file(cortexm0_45nm.tcl) 179: report_timing -nworst 100 -net  
@file(cortexm0_45nm.tcl) 180: report_summary  
> ${RPT_DIR}/19_${TOP DESIGN}_SYN_qor.rpt  
> ${RPT_DIR}/20_${TOP DESIGN}_SYN_timing.rpt  
> ${RPT_DIR}/21_${TOP DESIGN}_SYN_summary.rpt
```

- report 폴더에 여러가지 report를 저장함

```
161 #####  
162 ## REPORTS  
163 #####  
164  
165 report_analysis_views -type all      > ${RPT_DIR}/08_${TOP DESIGN}_SYN_analysis_view.rpt  
166 report_area                          > ${RPT_DIR}/09_${TOP DESIGN}_SYN_area.rpt  
167 report_clocks                        > ${RPT_DIR}/10_${TOP DESIGN}_SYN_clock.rpt  
168 report_clocks -generated           >> ${RPT_DIR}/10_${TOP DESIGN}_SYN_clock.rpt  
169 report_datapath                      > ${RPT_DIR}/11_${TOP DESIGN}_SYN_datapath.rpt  
170 report_design_rules                 > ${RPT_DIR}/12_${TOP DESIGN}_SYN_design_rules.rpt  
171 report_gates                         > ${RPT_DIR}/13_${TOP DESIGN}_SYN_gate.rpt  
172 report_gates -power                 >> ${RPT_DIR}/13_${TOP DESIGN}_SYN_gate.rpt  
173 report_hierarchy                     > ${RPT_DIR}/14_${TOP DESIGN}_SYN_hier.rpt  
174 report_memory_cells                 > ${RPT_DIR}/15_${TOP DESIGN}_SYN_mem_cell.rpt  
175 report_messages -all                > ${RPT_DIR}/16_${TOP DESIGN}_SYN_message.rpt  
176 report_nets -hierarchical          > ${RPT_DIR}/17_${TOP DESIGN}_SYN_net.rpt  
177 report_power                         > ${RPT_DIR}/18_${TOP DESIGN}_SYN_pwr.rpt  
178 report_qor                           > ${RPT_DIR}/19_${TOP DESIGN}_SYN_qor.rpt  
179 report_timing -nworst 100 -net       > ${RPT_DIR}/20_${TOP DESIGN}_SYN_timing.rpt  
180 report_summary                      > ${RPT_DIR}/21_${TOP DESIGN}_SYN_summary.rpt
```

# Synthesis

genus

## Genus 툴 실행 결과

```
@genus:root: 102> write_design -basename ${MAPPED_DIR}/${TOP DESIGN}_mapped  
(write_design): Writing Genus content. Constraint interface is 'smsc'  
Exporting design data for 'cmsdk_mcu' to ./mapped/cmsdk_mcu_mapped...  
%# Begin write_design (01/26 23:33:34, mem=1909.54M)  
      flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot  
UM:*                                         write_design  
Setting attribute of root '/': 'set_boundary_change' =  
No scan chains were found.  
File ./mapped/cmsdk_mcu_mapped.mmmc.tcl has been written.  
Finished SDC export (command execution time mm:ss (real) = 00:00).  
Info: file ./mapped/cmsdk_mcu_mapped.default_emulate_constraint_mode.sdc has been written  
Info: file ./mapped/cmsdk_mcu_mapped.default_emulate_constraint_mode.sdc has been written  
** To load the database source ./mapped/cmsdk_mcu_mapped.genus_setup.tcl in an Genus(TM) Synthesis Solution session.  
Finished exporting design data for 'cmsdk_mcu' (command execution time mm:ss cpu = 00:01, real = 00:05).  
. .  
%# End write_design (01/26 23:33:39, total cpu=17:00:01, real=17:00:05, peak res=1016.40M, current me m=1909.54M)
```

- mapped 폴더에 TOP 디자인의 이름을 활용한 디자인 폴더를 생성

```
182 #####  
183 ## write Innovus file set (verilog, SDC, config, etc.)  
184 #####  
185  
186 #source -echo ./script/report syn.tcl  
187 write_design -basename ${MAPPED_DIR}/${TOP DESIGN}_mapped  
188 write_hdl > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.v  
189 write_sdc > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.sdc  
190 write_sdf -timescale ns -nonegchecks -recrrem split -edges check_edge  
-setuphold split > ${MAPPED_DIR}/${TOP DESIGN}_delays.sdf
```

# Synthesis

genus

## Genus 툴 실행 결과

```
@genus:root: 103> write_hdl > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.v
@genus:root: 104> write_sdc > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.sdc
Finished SDC export (command execution time mm:ss (real) = 00:01).
@genus:root: 105> write_sdf -timescale ns -nonegchecks -recrem split -edges check_edge
                     -setuphold split > ${MAPPED_DIR}/${TOP DESIGN}_delays.sdf
Warning : No delay description exists for an instance. [WSDF-201]
          : Cell u_cmsdk_mcu_system/u_apb_subsystem/u_ahb_to_apb/g1363.
          : The timing arc of the instance does not exist or is disabled by disable_t
imeing or constant value. The delay information will not be generated for the instance
. The instance could be a loop breaker or its inputs could be driven by constant
Warning : No delay description exists for an instance. [WSDF-201]
          : Cell pad5.
Warning : No delay description exists for an instance. [WSDF-201]
          : Cell pad71.
```

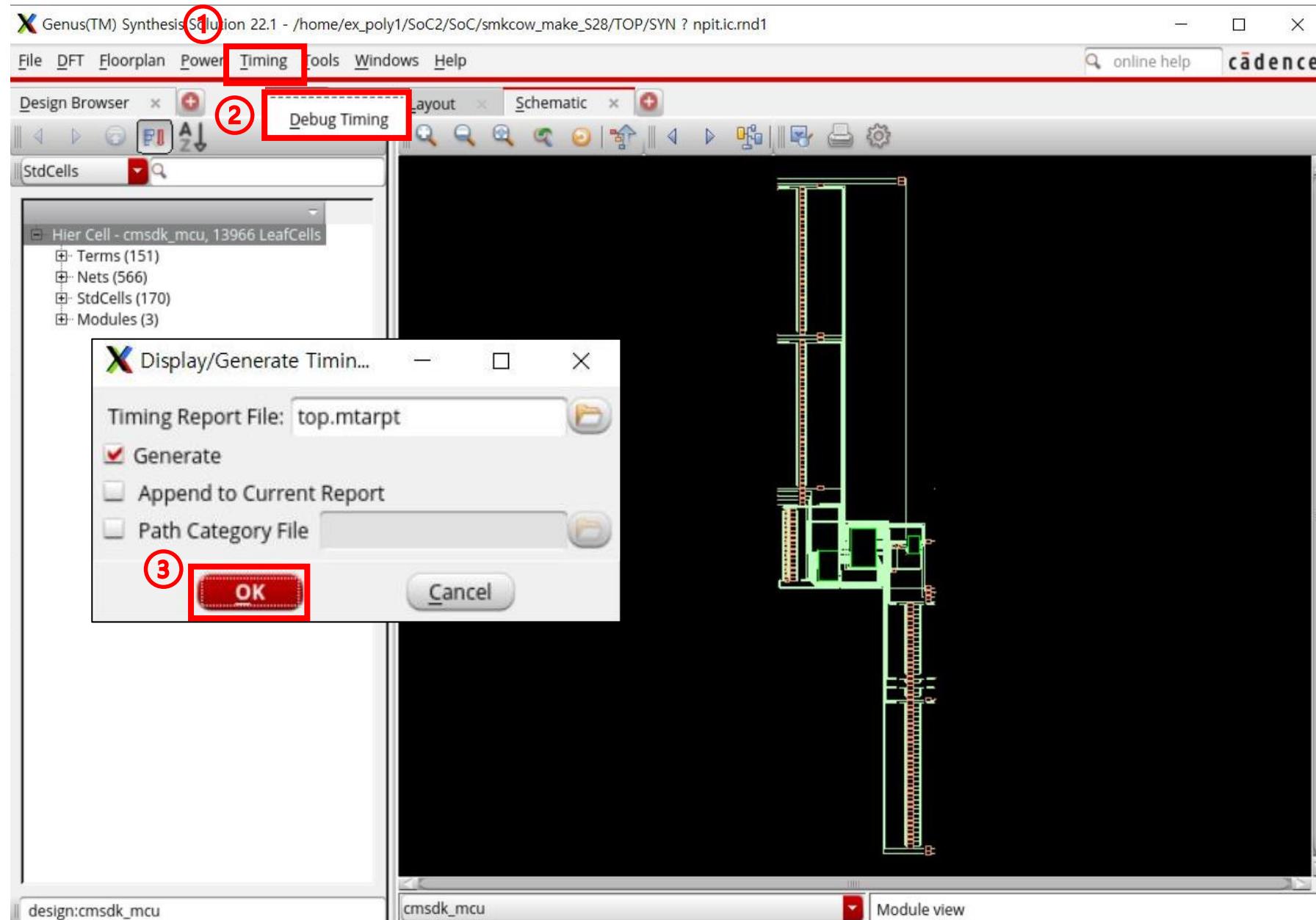
- 188행은 Gate\_Level\_Netlist를 최종적으로 뽑아내며, 이 파일은 다음 단계 툴에 사용 됨
- 189행은 툴이 constrain를 기억한 후 다음 툴을 위해 write out
- 190행의 sdf는 Standard Delay Format의 약자로 cell이 동작하는데 걸린 시간을 저장

```
182 #####
183 ## write Innovus file set (verilog, SDC, config, etc.)
184 #####
185
186 #source -echo ./script/report_syn.tcl
187 write_design -basename ${MAPPED_DIR}/${TOP DESIGN}_mapped
188 write_hdl > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.v
189 write_sdc > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.sdc
190 write_sdf -timescale ns -nonegchecks -recrem split -edges check_edge
           -setuphold split > ${MAPPED_DIR}/${TOP DESIGN}_delays.sdf
```

# Synthesis

genus

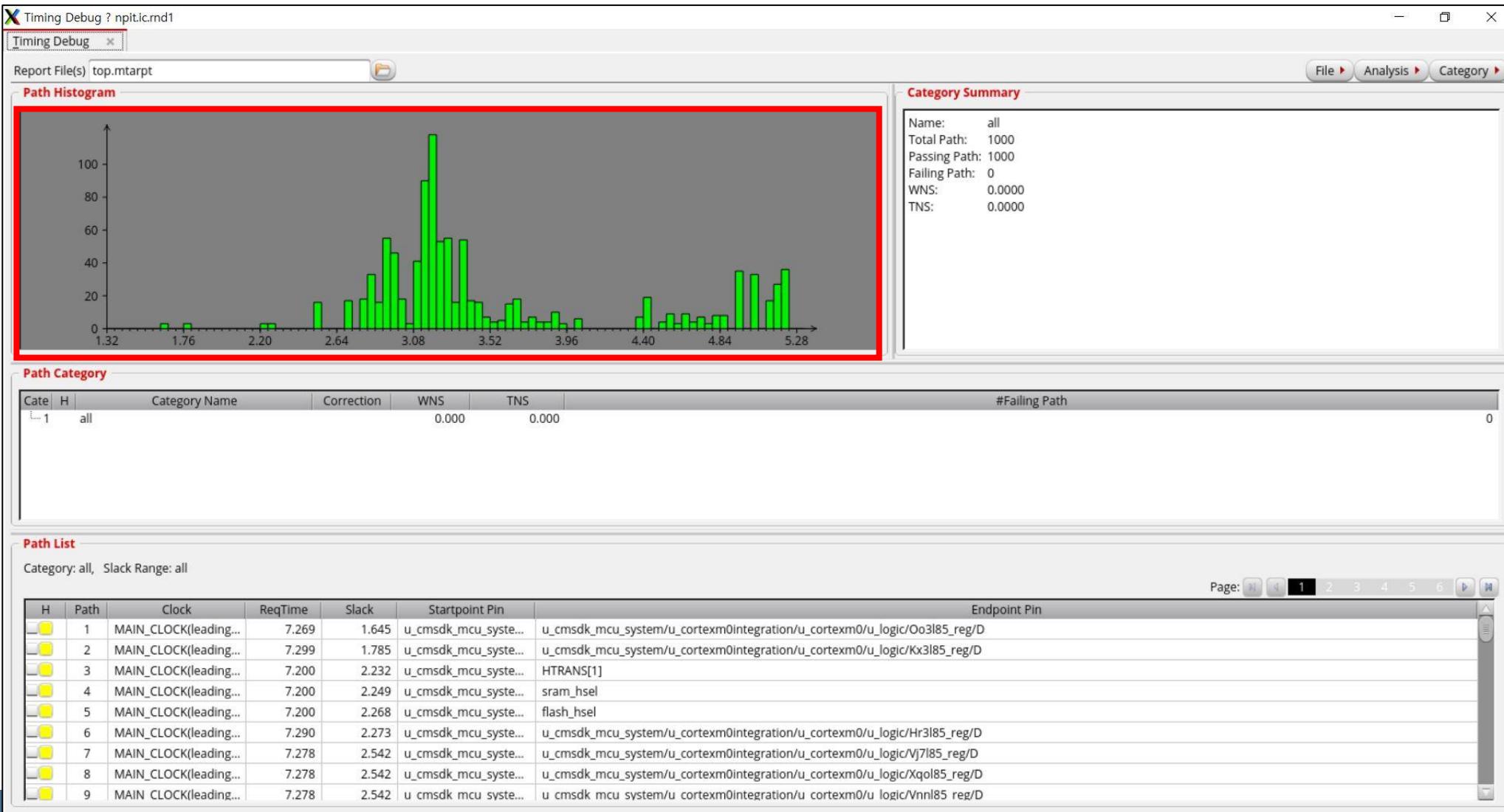
- Debug Timing



# Synthesis

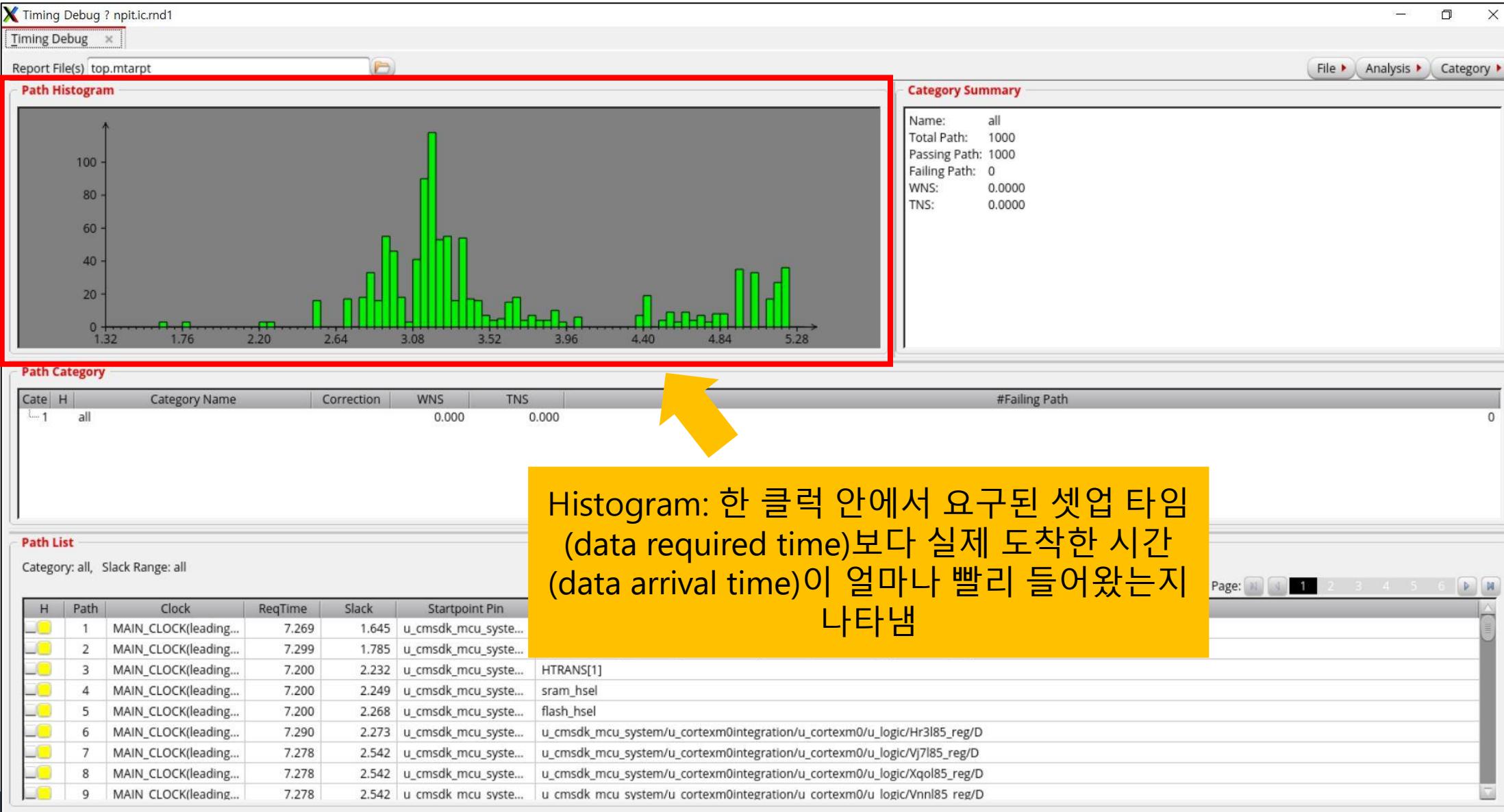
genus

- Debug Timing: 텍스트로 확인했던 정보를 그래픽으로 확인
- Path가 모두 녹색이므로 이상 없음(문제 발생 시 빨간색으로 표시됨)



# Synthesis

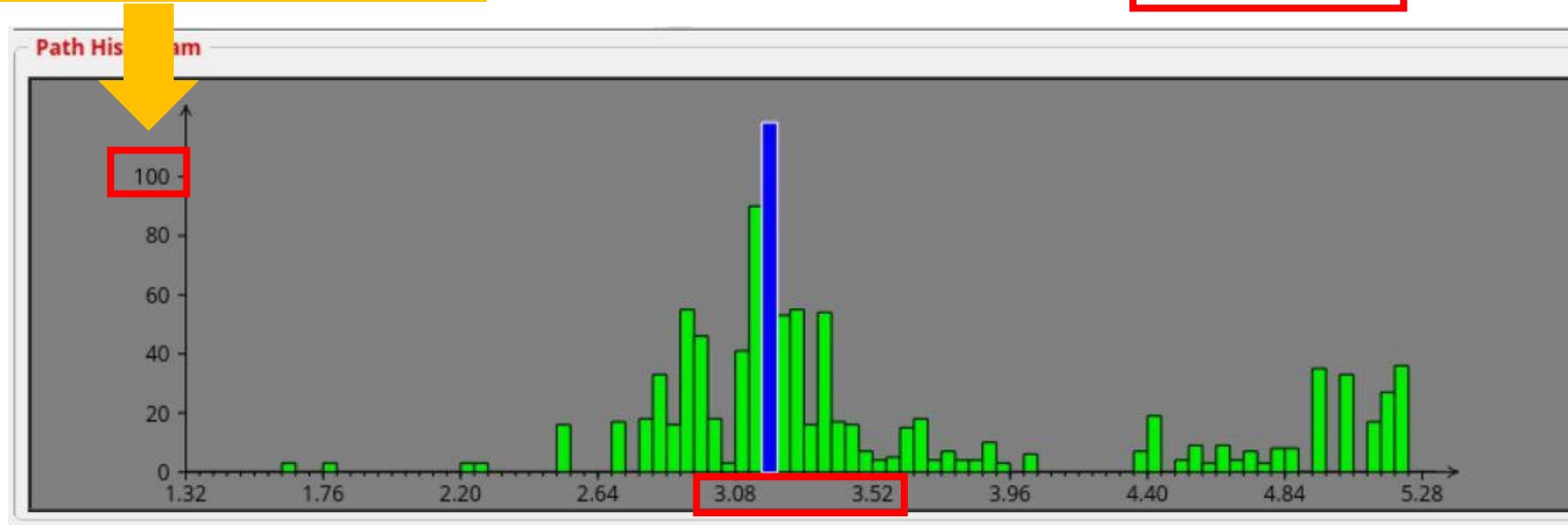
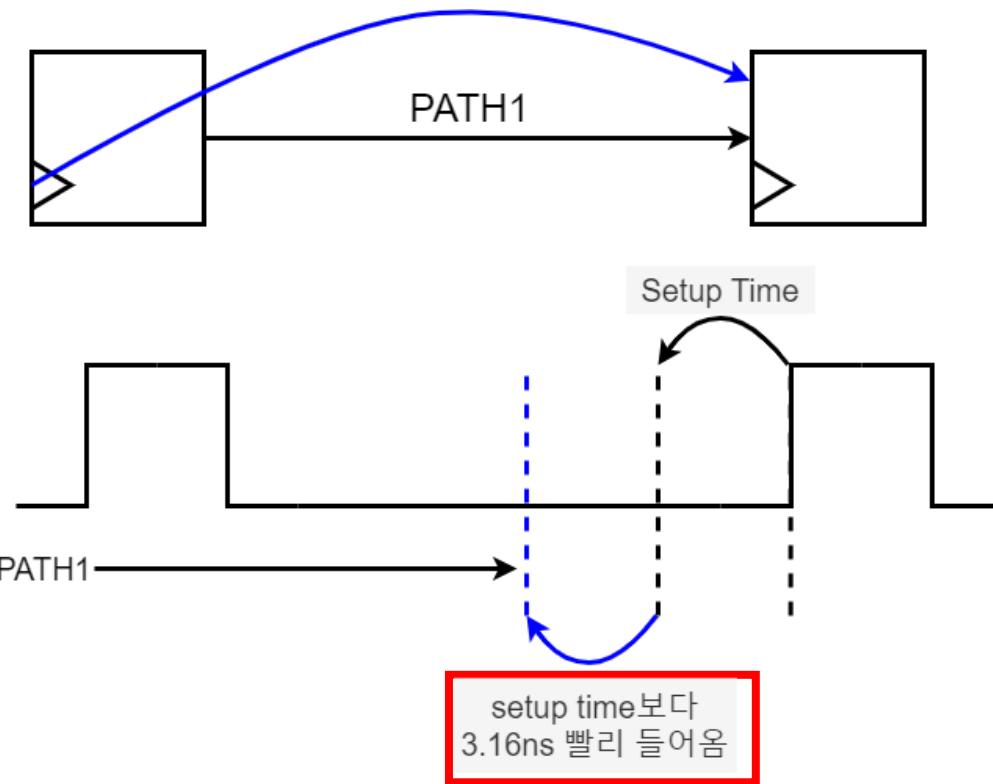
genus



# Synthesis

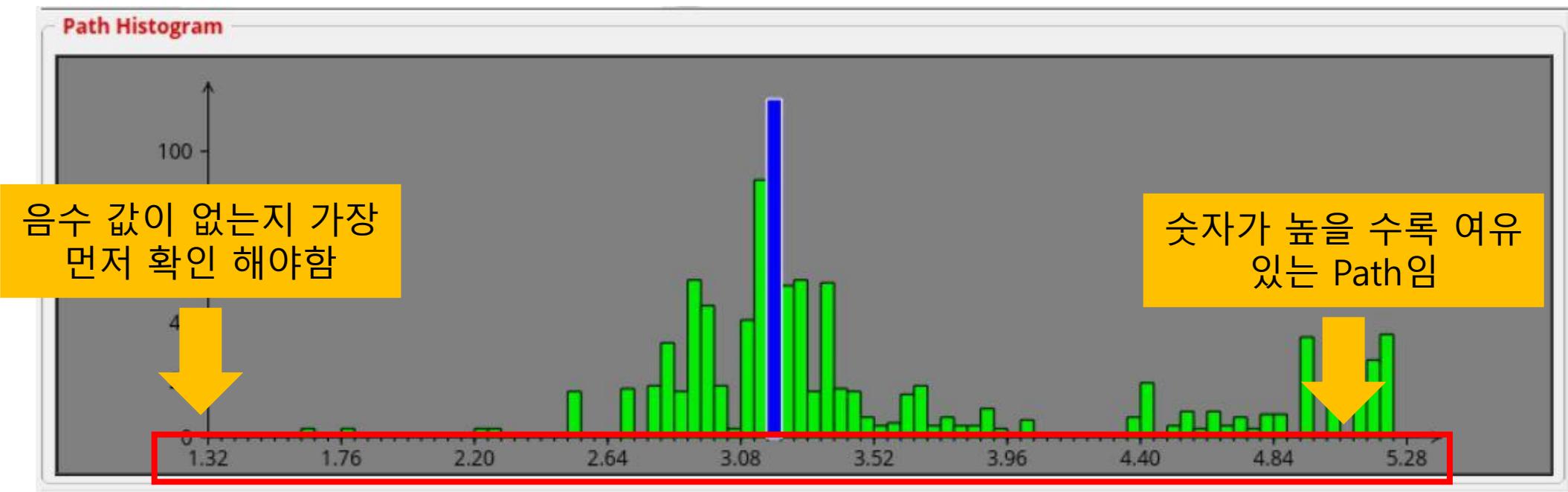
genus

3.16ns 일찍 도착한 Path가 100개 이상 된다는 것을 알 수 있음



# Synthesis

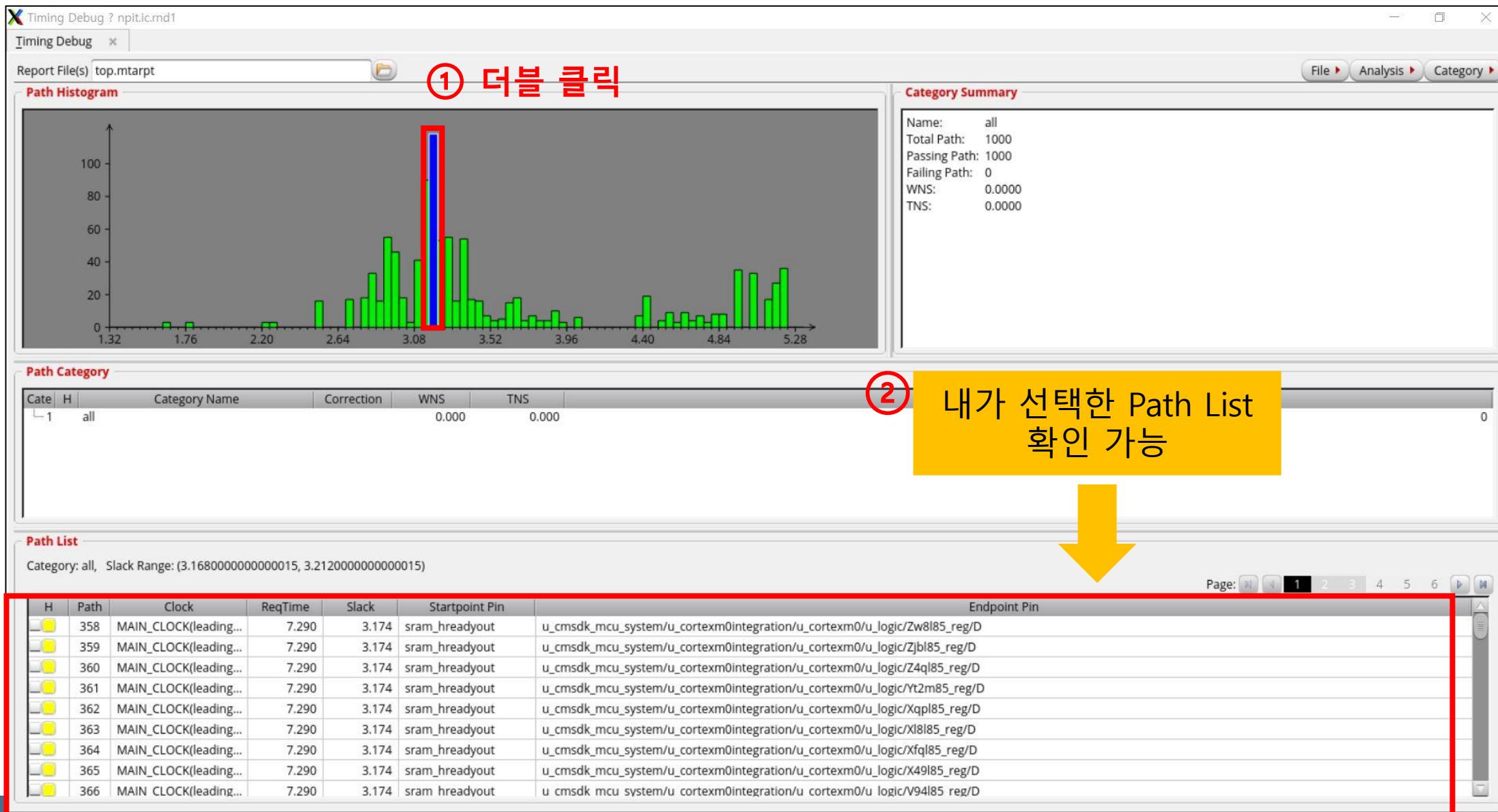
genus



# Synthesis

- Histogram의 세부내용 확인 방법

genus



# Synthesis

## genus

- Histogram의 세부내용 확인 방법
- Start point와 Endpoint 확인 가능함

Path List

Category: all, Slack Range: (3.1680000000000015, 3.2120000000000015)

Page: 1 2 3 4 5 6

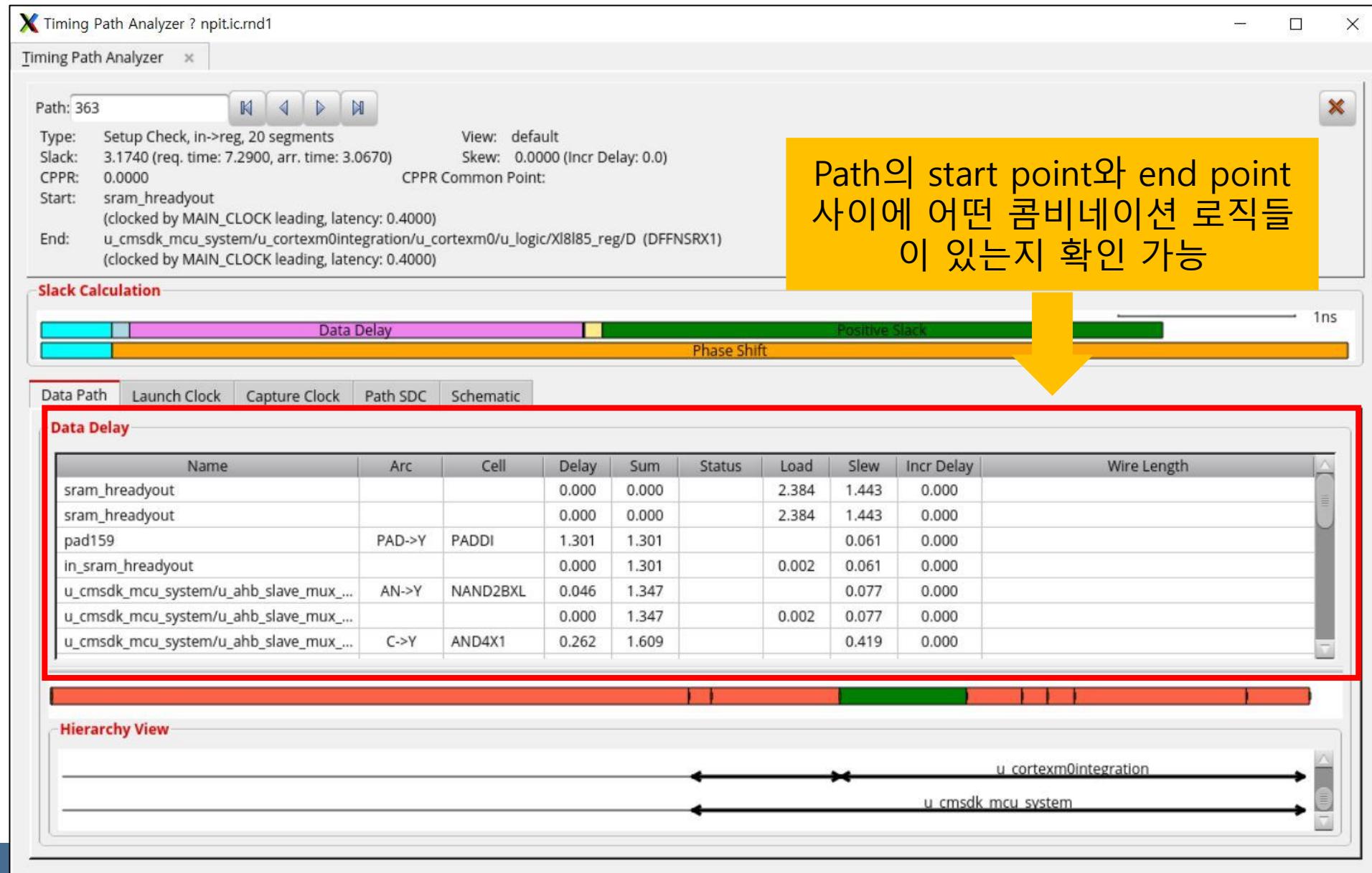
H	Path	Clock	ReqTime	Slack	Startpoint Pin	Endpoint Pin
358	MAIN_CLOCK(leading...		7.290	3.174	sram_nreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Zw8l85_reg/D
359	MAIN_CLOCK(leading...		7.290	3.174	sram_hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Zjbl85_reg/D
360	MAIN_CLOCK(leading...		7.290	3.174	sram_hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Z4ql85_reg/D
361	MAIN_CLOCK(leading...		7.290	3.174	sram_hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Yt2m85_reg/D
362	MAIN_CLOCK(leading...		7.290	3.174	sram_breadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Vqpl85_reg/D
363	MAIN_CLOCK(leading...		7.290	3.174	sram_hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Xl8l85_reg/D
364	MAIN_CLOCK(leading...		7.290	3.174	sram_nreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Xlql85_reg/D
365	MAIN_CLOCK(leading...		7.290	3.174	sram_hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/X49l85_reg/D
366	MAIN CLOCK(leading...		7.290	3.174	sram hreadyout	u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/V94l85_reg/D

① 더블 클릭

# Synthesis

- Histogram의 세부내용 확인 방법

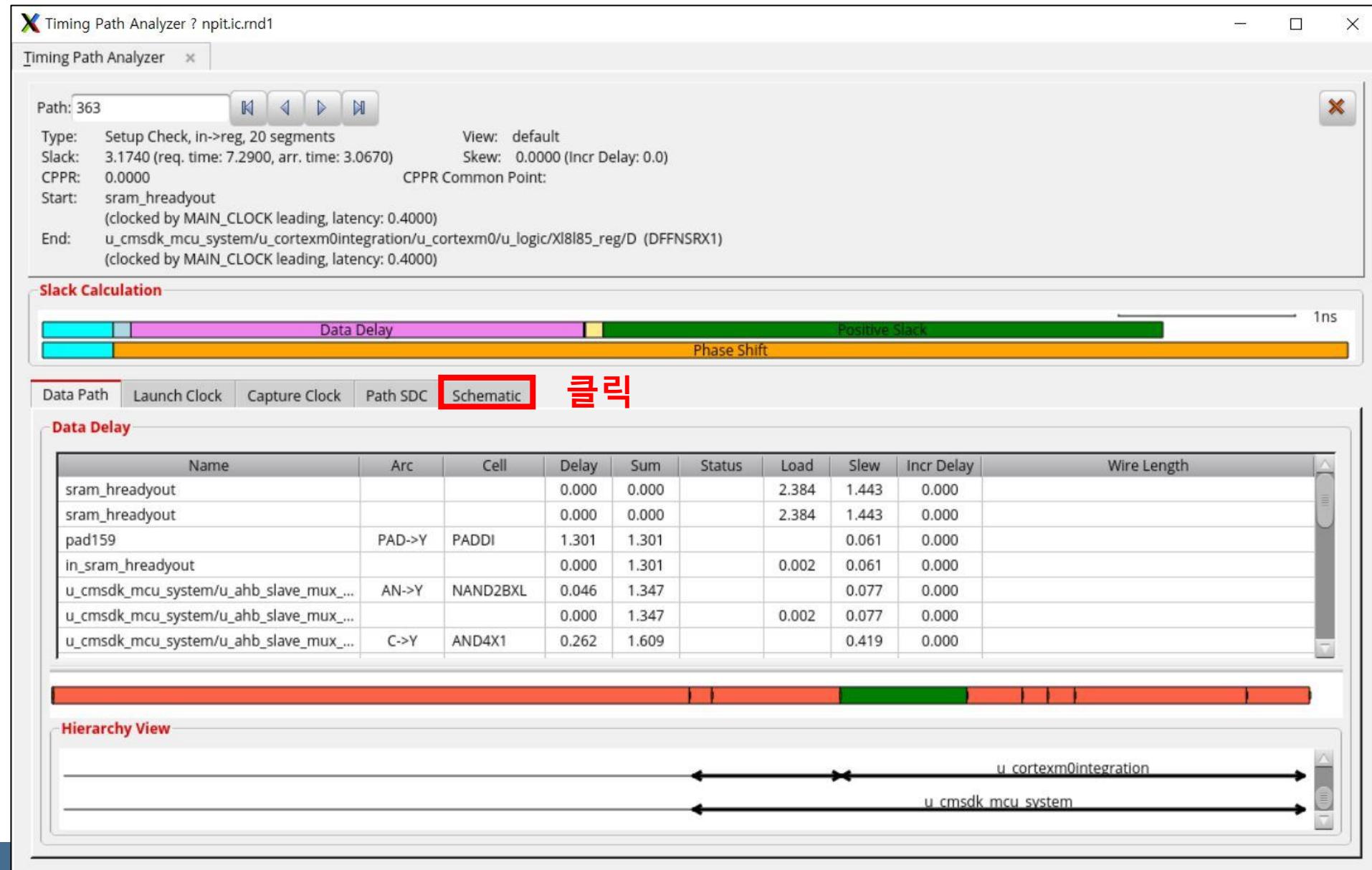
genus



# Synthesis

- Histogram의 세부내용 확인 방법

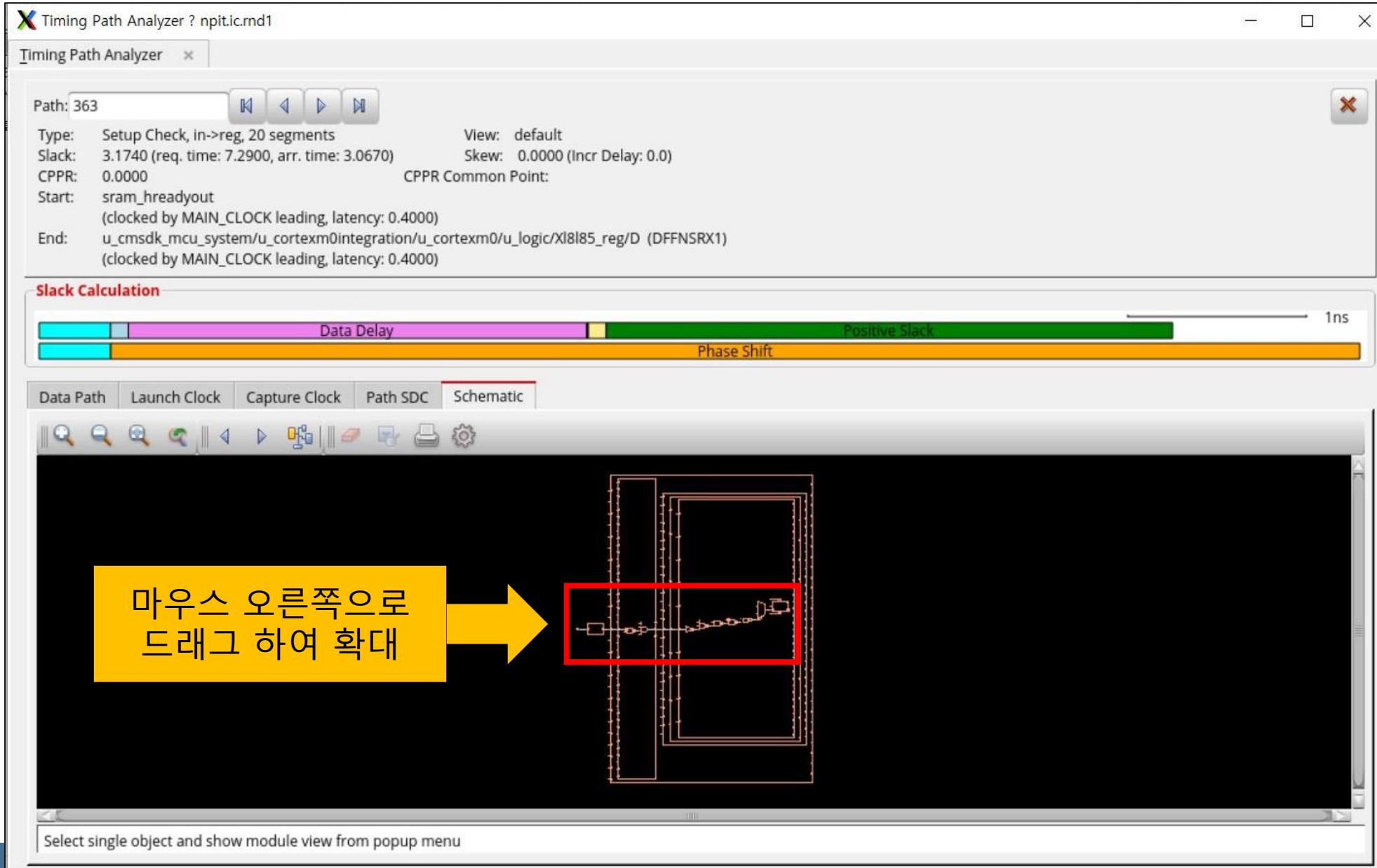
genus



# Synthesis

genus

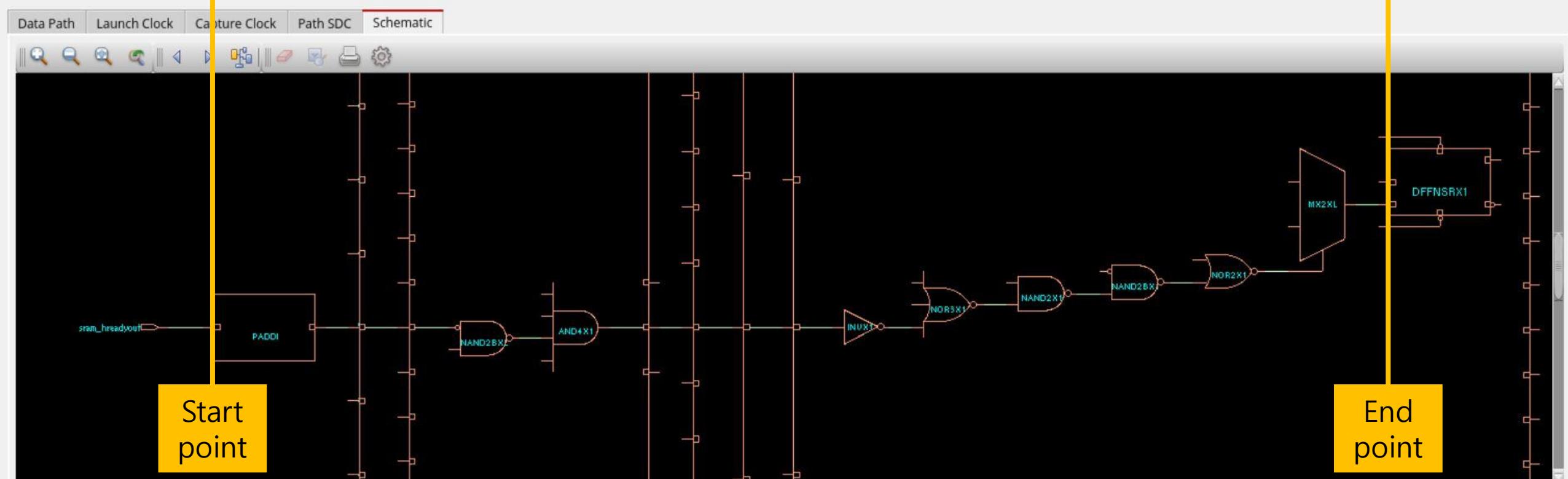
- Schematic 확인 가능



# Synthesis

genus

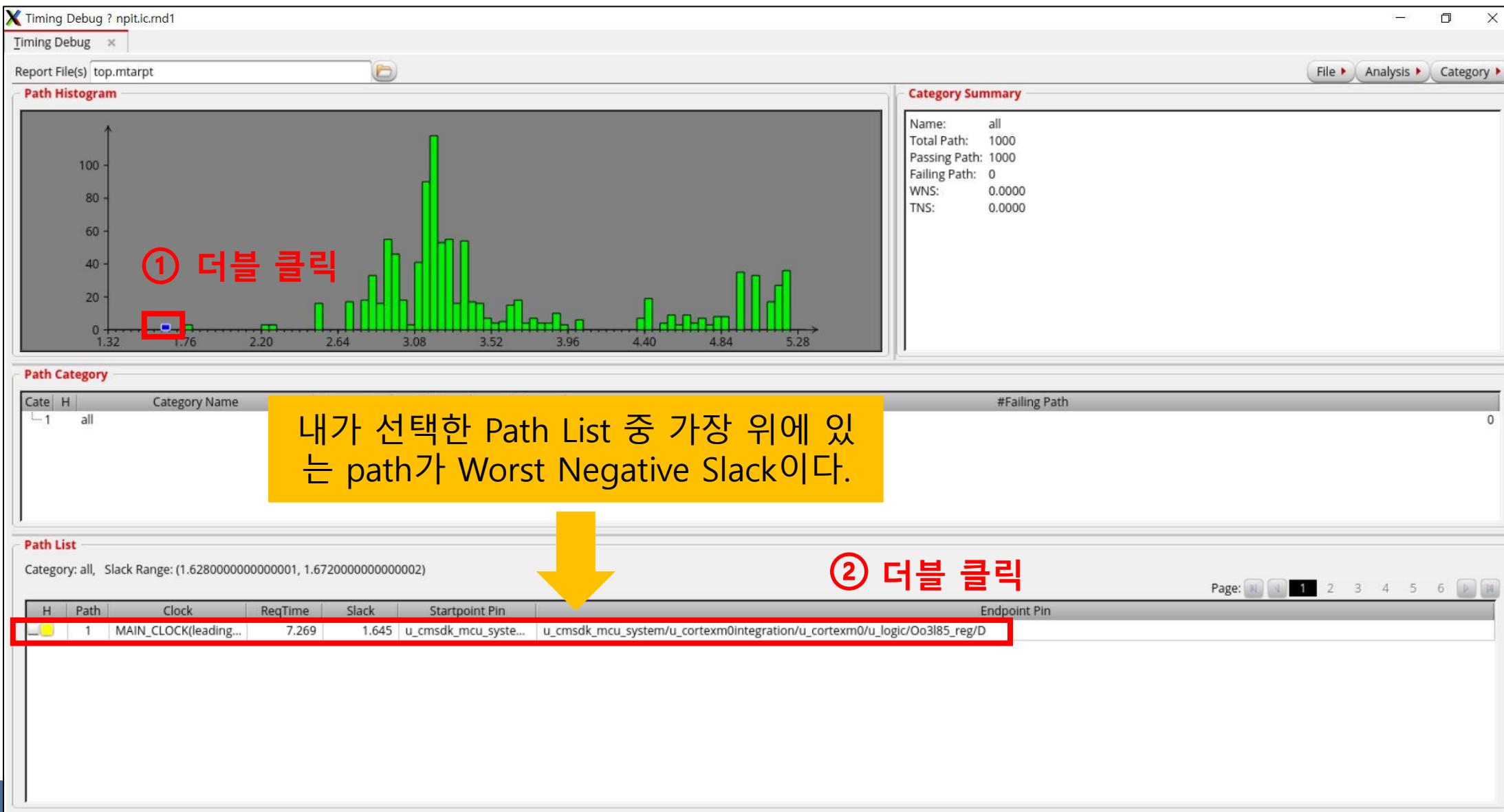
Path와 Path 사이의 start point, end point, 콤비네이션 로직을 schematic으로 확인할 수 있음



# Synthesis

genus

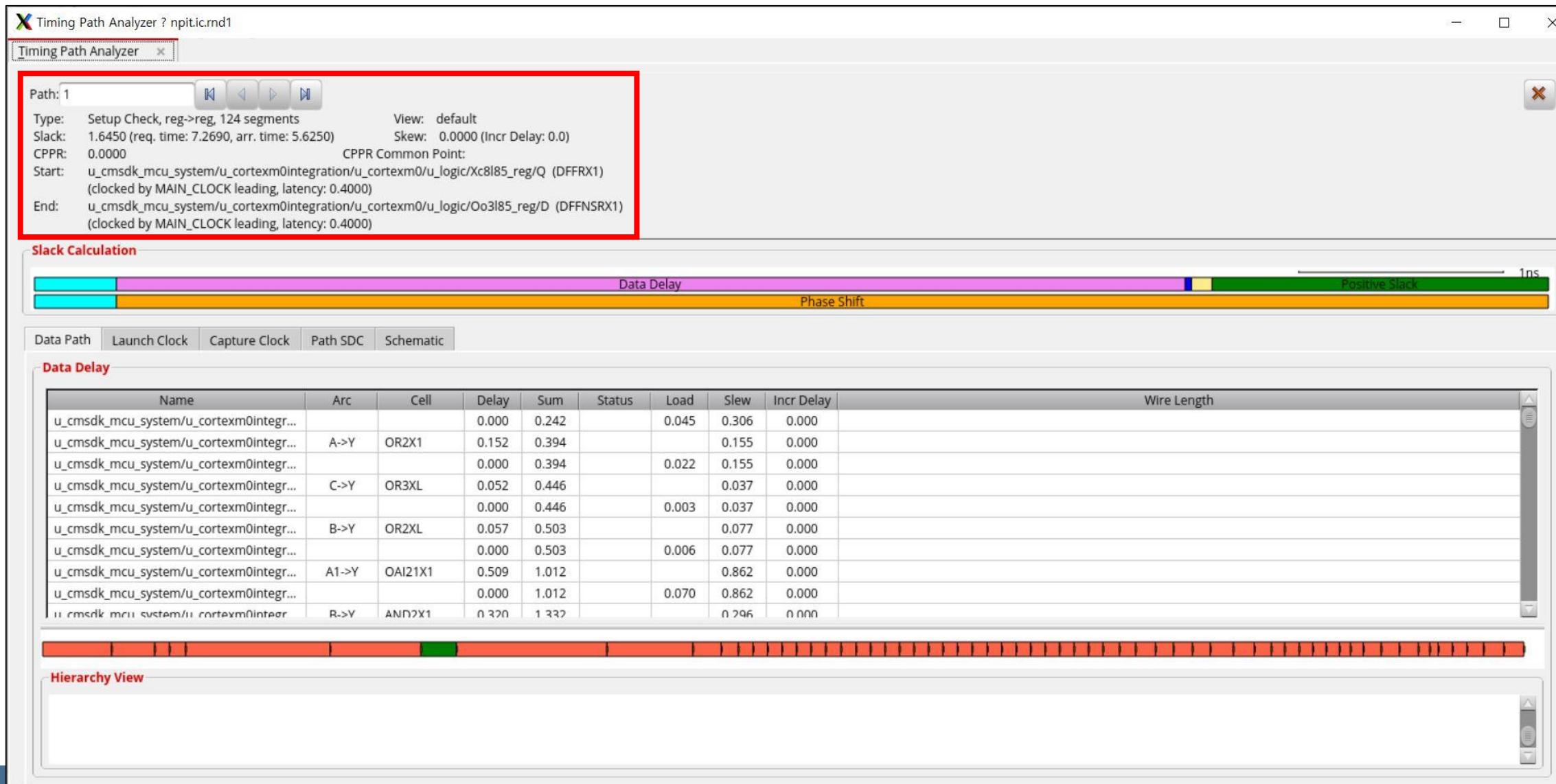
- Histogram을 이용한 worst negative Slack 확인



# Synthesis

genus

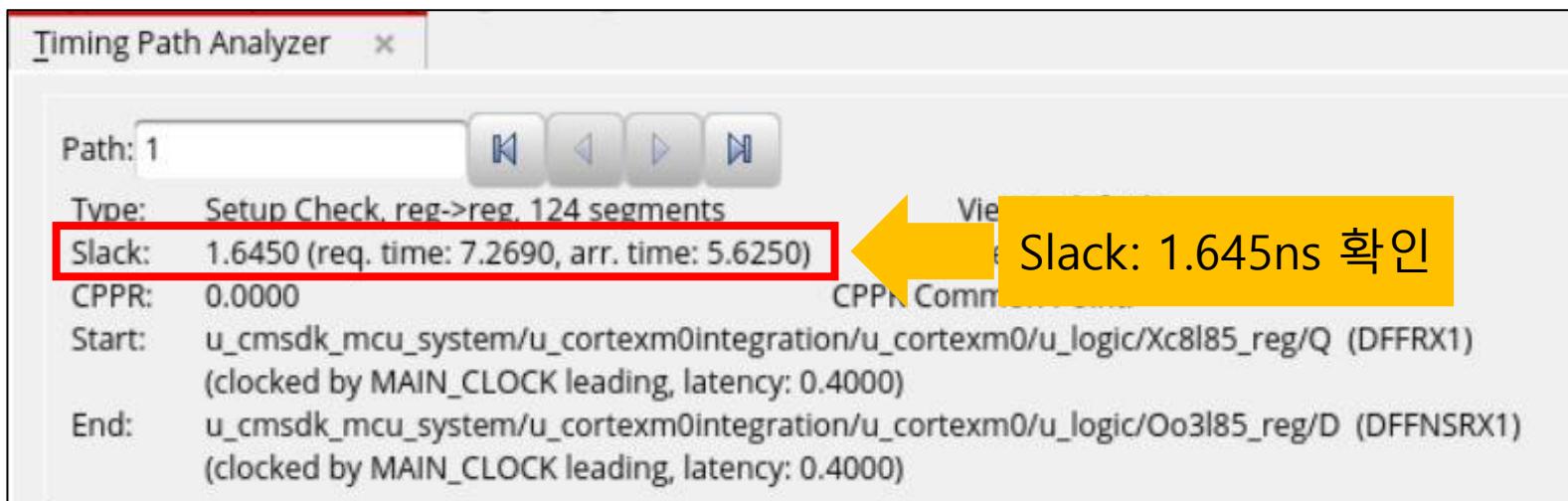
- Histogram을 이용한 worst negative Slack 확인



# Synthesis

## genus

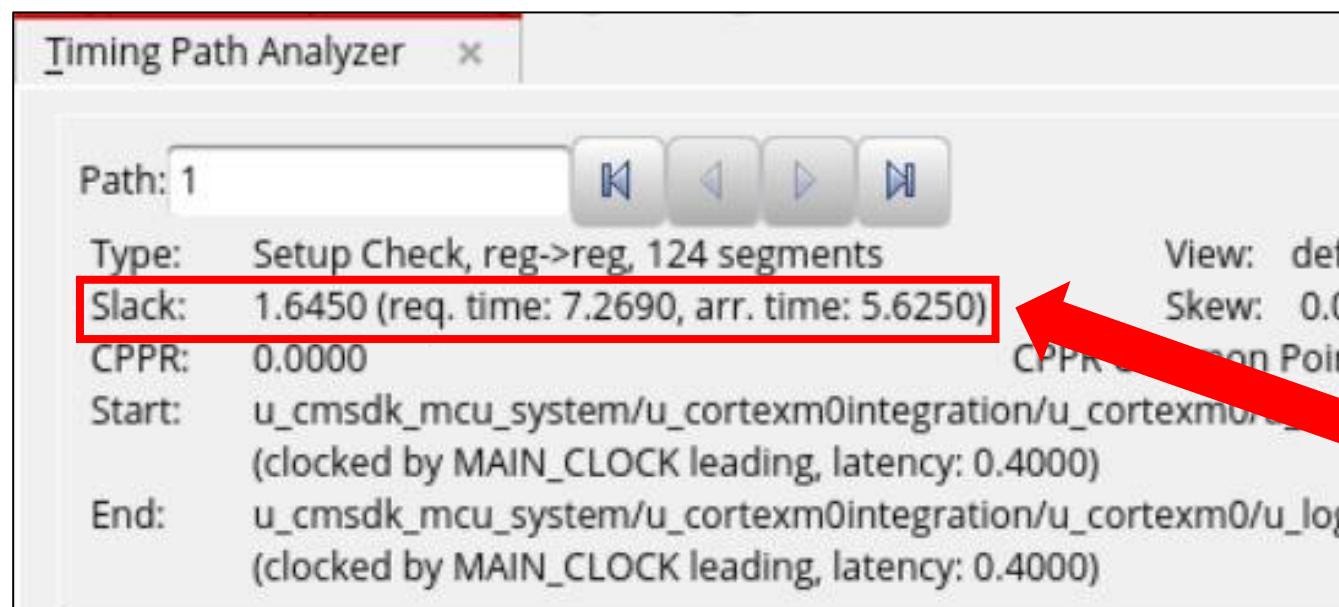
- Histogram을 이용한 worst negative Slack 확인



# Synthesis

genus

- Histogram을 이용해 total negative Slack을 확인한 결과 report timing 결과와 같음
- Slack이 양수이므로 문제없음



report timing 결과

Path 1: MET (1645 ps) Setup Check with Pin u\_cmse

Group: MAIN\_CLOCK

Startpoint: (R) u\_cmsdk\_mcu\_system/u\_cortexm0/integration/u\_cortexm0/u\_log

Clock: (R) MAIN\_CLOCK

Endpoint: (F) u\_cmsdk\_mcu\_system/u\_cortexm0/integration/u\_cortexm0/u\_log

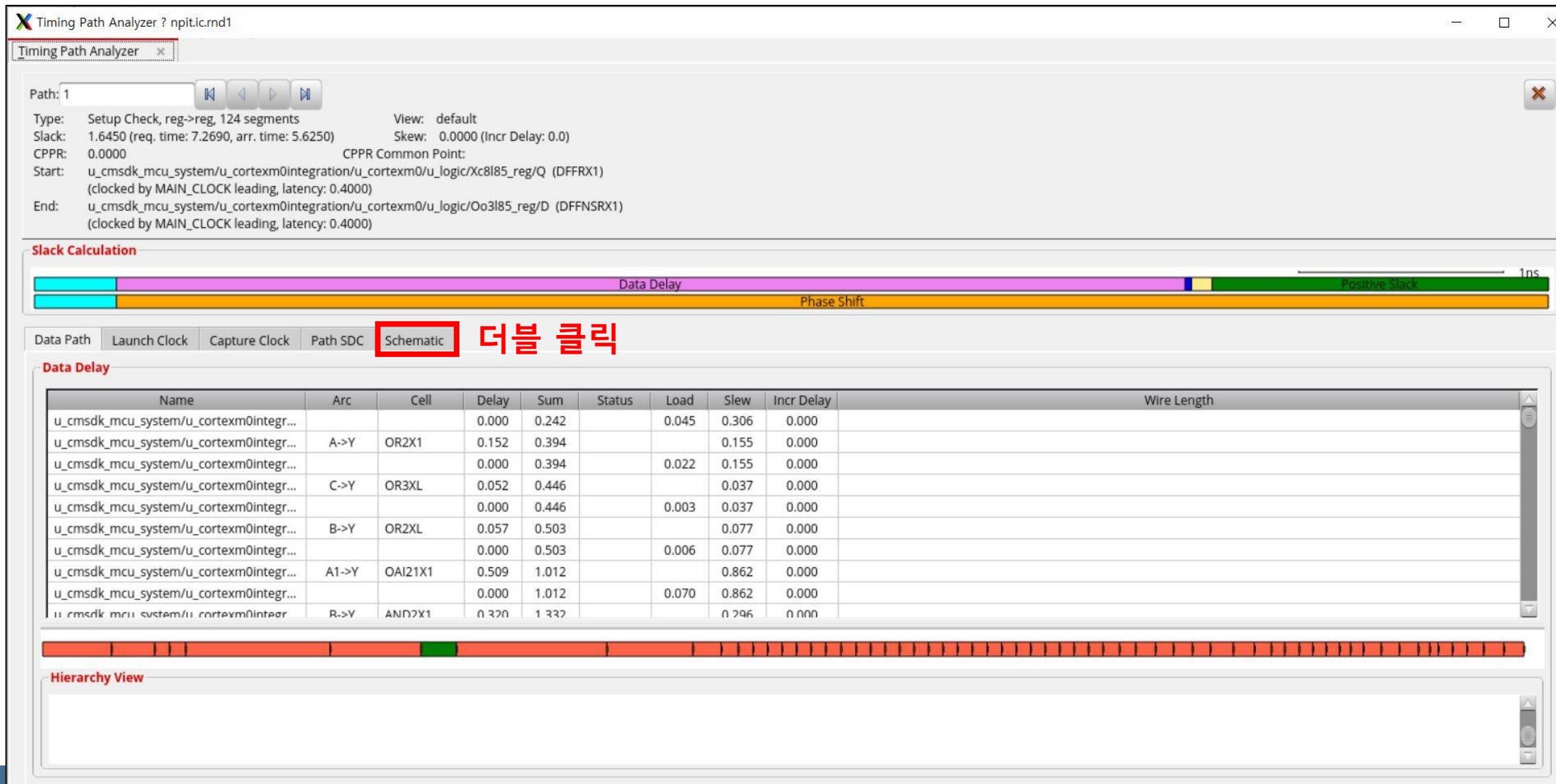
Clock: (R) MAIN\_CLOCK

	Capture	Launch
Clock Edge:+	7000	0
Src Latency:+	200	200
Net Latency:+	200 (I)	200 (I)
Arrival:=	7400	400
Setup:-	31	
Uncertainty:-	100	
Required Time:=	7269	
Launch Clock:-	400	
Data Path:-	5225	
Slack:=	1645	

# Synthesis

genus

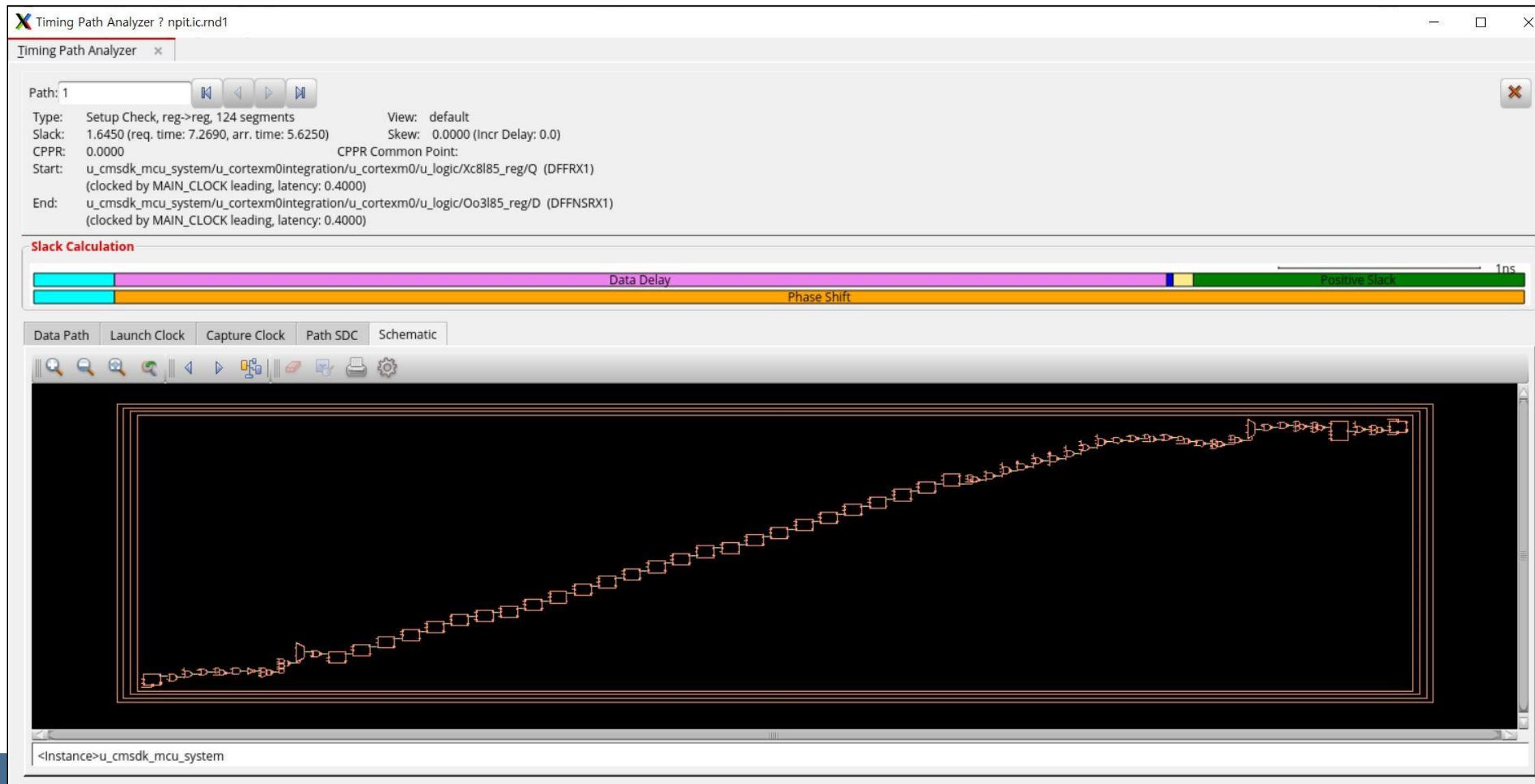
- Histogram을 이용한 worst negative Slack 확인



# Synthesis

genus

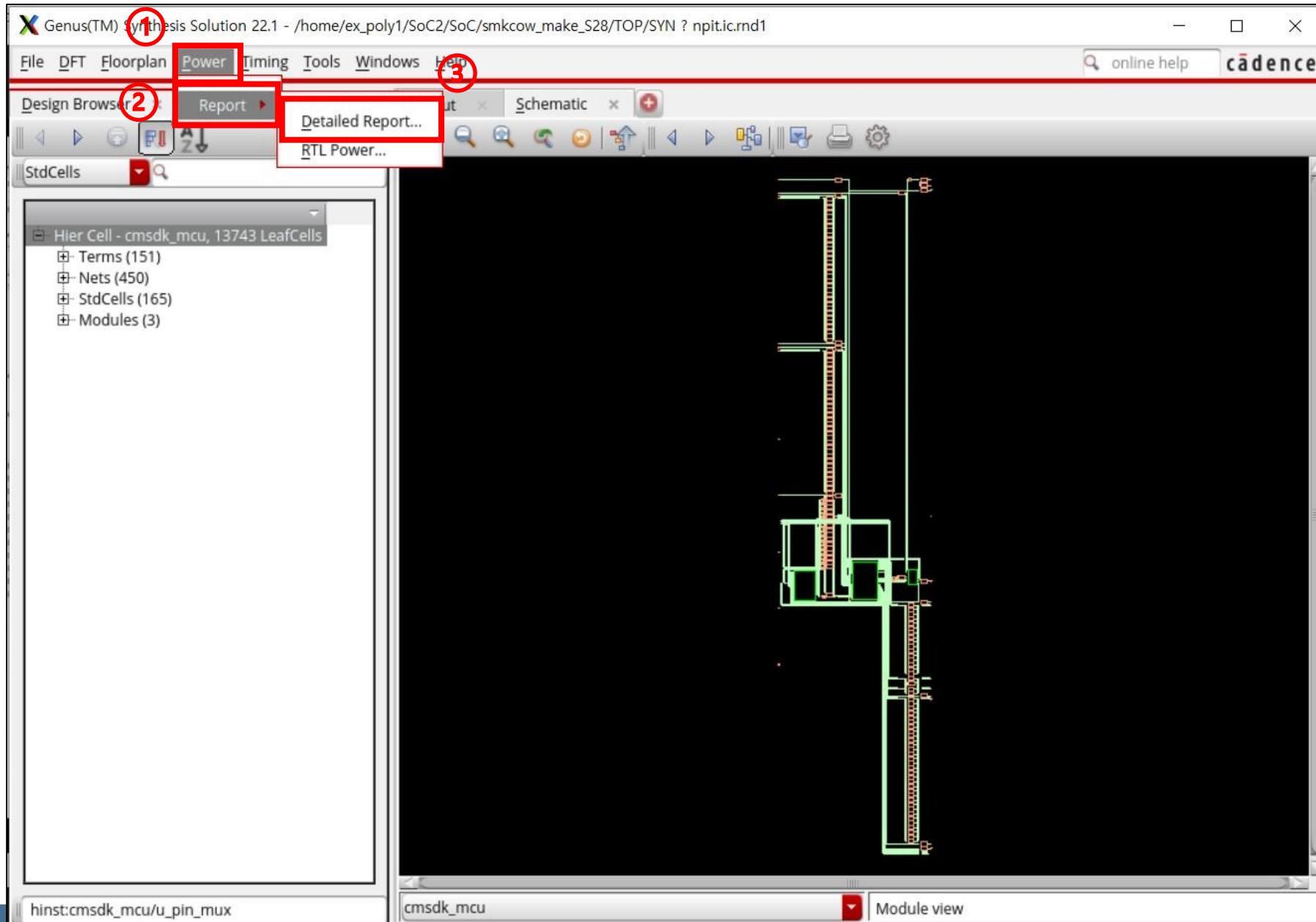
- Histogram을 이용해 worst negative Slack의 schematic 확인 가능



# Synthesis

genus

- Power를 통해 전력 소모량을 알 수 있음



# Synthesis

genus

각 instance의 전력 소모량을 알 수 있음

TOP 모듈을 보면 총 누적 전력 소모량을 알 수 있음

Instance	Cells	Leakage (nW)	Internal (nW)	Net (nW)	Switching (nW)
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_test_slave.u_apb_test_slave	170	21.198	23.65.246	411.187	411.6.478
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_timer_0.u_apb_timer_0	383	59.581	62674.151	3337.701	66011.852
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_timer_1.u_apb_timer_1	383	59.589	62679.337	3336.089	66015.426
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_uart_0.u_apb_uart_0	481	80.187	88276.954	5172.004	93448.958
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_uart_1.u_apb_uart_1	481	80.322	88257.579	5172.971	93430.550
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_uart_2.u_apb_uart_2	480	80.331	88267.739	5168.302	93436.041
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog	469	84.528	82159.548	9836.187	91995.734
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog_frc	342	63.021	58141.882	9572.026	67713.908
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/u_ahb_to_apb	140	30.587	40682.143	3066.842	43748.985
cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/u_apb_slave_mux	16	16.303	317.111	383.158	700.269
cmsdk_mcu/u_cmsdk_mcu_system/u_cmsdk_mcu_stclkctrl	5.830	7558.270	254.603	7812.873	
cmsdk_mcu/u_cmsdk_mcu_system/u_cmsdk_mcu_sysctrl	12.999	12803.600	571.532	13375.132	
cmsdk_mcu/u_cmsdk_mcu_system/u_cortexm0integration	1193.780	932151.657	303284.851	1235436.508	
cmsdk_mcu/u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0	7790	1193.780	932151.657	303284.851	1235436.508
cmsdk_mcu/u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic	7790	1193.780	932151.657	303284.851	1235436.508
cmsdk_mcu/u_cmsdk_mcu_system/u_system_rom_table	42	6.357	4884.568	203.913	5088.480
cmsdk_mcu/u_pin_mux	6	0.892	248.572	260.794	509.366
cmsdk_mcu	13743	5817237.115	91854993.074	2949871.304	94804864.378

# Synthesis

genus

```
$> ./clean.tcl
```

```
$> ./run_synthesis
```

지금부터는 run\_synthesis를 실행하여 전체 스크립트를 자동 배치모드로 한번에 진행

①

```
[ex_poly1@npit SYN]$ ./clean.tcl
```

②

```
[ex_poly1@npit SYN]$ ./run_synthesis
```

# Synthesis

genus를 사용 준비

```
/home/ex_poly1/SoC2/SoC/smkcov_make_S28/TOP/SYN
```

- 두개의 터미널 사용 (위의 경로에서 진행)
- 하나는 `./run_synthesis` 명령실행, 나머지는 report 파일의 생성을 확인

The image shows two terminal windows side-by-side. The left window, titled 'ex\_poly1@npit:SYN', displays the command history for running synthesis:

```
[ex_poly1@npit SYN]$ pwd  
/home/ex_poly1/SoC2/SoC/smkcov_make_S28/TOP/SYN  
[ex_poly1@npit SYN]$ ./clean.tcl  
[ex_poly1@npit SYN]$ ./run_synthesis
```

The right window, also titled 'ex\_poly1@npit:SYN', shows the directory structure and the list of generated report files:

```
File Edit View Search Terminal Help  
[ex_poly1@npit SYN]$ ls  
lean.tcl cons fv log mapped report run_synthesis script unmapped  
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
0_check_design.rpt  
1_check_timing.rpt  
2_report_port.rpt  
3_report_datapath_generic.rpt  
4_report_datapath_map.rpt  
5_default_post_syn_map.rpt  
5_MAIN_CLOCK_post_syn_map.rpt  
final.rpt  
[ex_poly1@npit report]$
```

On the right, the report files are grouped into columns:

syn_generic_cmsdk_mcu.db	syn_generic_qor.rpt	syn_generic_time.rpt
syn_map_area.rpt	syn_map_cmsdk_mcu.db	syn_map_gates.rpt
syn_map_qor.rpt	syn_map_time.rpt	

# Synthesis

## genus

\$> ./run\_synthesis

- ./run\_synthesis 명령으로 합성의 모든 과정이 진행됨

### Genus 툴 실행 결과

```
-----+-----+-----+-----+-----+-----+-----+-----+
00:01:28(00:01:32) | 00:00:21(00:00:25) | 25.7( 27.2) | 23:36:34 (Feb01) | 991
.4 MB | FINAL
-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+
Number of threads: 2 * 1 (id: default, time_info v1.57)
Info: (*N*) indicates data that was populated from previously saved time_info database
Info: CPU time includes time of parent + longest thread
@file(cortexm0_45nm.tcl) 203: puts "====="
=====
@file(cortexm0_45nm.tcl) 204: puts "Synthesis Finished ....."
Synthesis Finished .....
@file(cortexm0_45nm.tcl) 205: puts "====="
=====
#@ End verbose source ./script/cortexm0_45nm.tcl
WARNING: This version of the tool is 236 days old.
```

# Synthesis

genus

\$> cd report

- 합성 진행상황을 report 폴더에서 실시간으로 확인

```
[ex_poly1@npit SYN]$ ls  
clean.tcl  cons  fv  log  mapped  report  run_synthesis  script  unmapped
```

# Synthesis

genus

\$> //

- 합성 //
- 명령어를 활용하여 report가 실시간으로 생성되는지 확인함
- 각 report의 결과가 정상적으로 나왔는지 확인함

합성 전

```
[ex_poly1@npit report]$ ll  
total 0
```

합성 후

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_men_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt  
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

# Synthesis

genus

\$> vi 00\_check\_design.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt  
  
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

큰 문제가 없음을  
확인함

- RTL소스의 무결성을 확인함

Name	Total
2 Check Design Report (c)	
3 -----	
4 -----	
5 Summary	
6 -----	
7 -----	
8 -----	
9 -----	
10 Unresolved References	0
11 Empty Modules	0
12 Unloaded Port(s)	0
13 Unloaded Sequential Pin(s)	0
14 Unloaded Combinational Pin(s)	11
15 Assigns	137
16 Undriven Port(s)	0
17 Undriven Leaf Pin(s)	0
18 Undriven hierarchical pin(s)	1241
19 Multidriven Port(s)	0
20 Multidriven Leaf Pin(s)	0
21 Multidriven hierarchical Pin(s)	0
22 Multidriven unloaded net(s)	0
23 Constant Port(s)	0
24 Constant Leaf Pin(s)	880
25 Constant hierarchical Pin(s)	56
26 Preserved leaf instance(s)	13566
27 Preserved hierarchical instance(s)	30
28 Feedthrough Modules(s)	0
29 Libcells with no LEF cell	4
30 Physical (LEF) cells with no libcell	107
31 Subdesigns with long module name	0
32 Physical only instance(s)	0
33 Logical only instance(s)	0
34 -----	
35 Done Checking the design.	

# Synthesis

genus

\$> vi 01\_check\_timing.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00 check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

• 내가 입력한 constraint에 문제가 없음을 확인함

1	=	Generated by:	Genus(TM) Synthesis Solution 22.16-s078_1
2		Generated on:	Feb 01 2025 11:35:18 pm
3		Module:	cmsdk_mcu
4		Technology libraries:	fast_vdd1v0 1.0 slow_vdd1v0 1.0 giolib045 physical_cells
5		Operating conditions:	PVT_0P9V_125C
6		Interconnect mode:	global
7		Area mode:	physical library
8			=====
9			
10			
11			
12			
13			
14			
15		Lint summary	
16		Unconnected/logic driven clocks	0
17		Sequential data pins driven by a clock signal	0
18		Sequential clock pins without clock waveform	0
19		Sequential clock pins with multiple clock waveforms	0
20		Generated clocks without clock waveform	0
21		Generated clocks with incompatible options	0
22		Generated clocks with multi-master clock	0
23		Generated clocks with master clock not reaching the generated clock target	0
24		Paths constrained with different clocks	0
25		Loop-breaking cells for combinational feedback	0
26		Nets with multiple drivers	0
27		Timing exceptions with no effect	0
28		Suspicious multi_cycle exceptions	0
29		Pins/ports with conflicting case constants	0
30		Inputs without clocked external delays	0
31		Outputs without clocked external delays	0
32		Inputs without external driver/transition	0
33		Outputs without external load	0
34		Exceptions with invalid timing start-/endpoints	0
35			0
36			0
37			0
		Total:	0

# Synthesis

genus

\$> vi 02\_report\_port.rpt

- External Driver에 원하는 cell이 입력되어 있는지 확인하는 것이 중요

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

```
1 ======  
2 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on: Feb 01 2025 11:35:18 pm  
4 Module: cmsdk_mcu  
5 Technology libraries: fast_vdd1v0 1.0  
6 slow_vdd1v0 1.0  
7 giolib045  
8 physical_cells  
9 Operating conditions: PVT_0P9V_125C  
10 Interconnect mode: global  
11 Area mode: physical library  
12 ======  
13  
14  
15 External Delays & Exceptions  
16 -----  
17 Port Dir Clock Rise Fall Ext Delay Exception  
18 19 Object Object/Type  
20 XTAL1 in N/A N/A N/A N/A N/A N/A  
21 NRST in MAIN_CLOCK 100.0 100.0 in_del N/A N/A  
22 in MAIN_CLOCK no_value no_value in_del_89_1 N/A  
23 nTRST in MAIN_CLOCK 100.0 100.0 in_del_1_1 N/A  
24 in MAIN_CLOCK no_value no_value in_del_90_1 N/A  
25 SWCLKTCK in MAIN_CLOCK 100.0 100.0 in_del_2_1 N/A  
26 in MAIN_CLOCK no_value no_value in_del_91_1 N/A  
27 TDI in MAIN_CLOCK 100.0 100.0 in_del_3_1 N/A  
28 in MAIN_CLOCK no_value no_value in_del_92_1 N/A  
29 SWDIOTMS in MAIN_CLOCK 100.0 100.0 in_del_4_1 N/A
```

# Synthesis

genus

\$> vi 02\_report\_port.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

- /External Driver 명령어를 사용하여 검색
- PADDI와 같이 내가 원하는 셀이 입력되어 있음

External Driver/Slew							
Port	Dir	External	External	Slew	Slew	Timing Case	Timing Ideal
		Driver	Driver				
XTAL1	in			0.0	0.0		false
NRST	in	PADDI/Y	PADDI/Y	0.0	0.0		false
nTRST	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
SWCLKTCK	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
TDI	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
SWDIOTMS	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hreadyout	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[31]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[30]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[29]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[28]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[27]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[26]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[25]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[24]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[23]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[22]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[21]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[20]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
flash_hrdata[19]	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false

# Synthesis

genus

\$> vi 03\_report\_datapath\_generic.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
1 Command: report datapath > ./report/03_report_datapath_generic.rpt  
2 Warning: '-sort' is not specified. By default, 'report datapath' orders the components based on instance name  
3 =====  
4 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
5 Generated on: Feb 01 2025 11:35:31 pm  
6 Module: cmsdk_mcu  
7 Technology libraries: fast_vdd1v0 1.0  
8 slow_vdd1v0 1.0  
9 giolib045  
10 physical_cells  
11 fast_vdd1v0 1.0  
12 slow_vdd1v0 1.0  
13 giolib045  
14 physical_cells  
15 Operating conditions: PVT_0P9V_125C  
16 Interconnect mode: global  
17 Area mode: physical library  
18 =====  
19  
20  
21 Type CellArea Percentage  
22 -----  
23 datapath modules 0.00 0.00  
24 external muxes 0.00 0.00  
25 others 2139958.05 100.00  
26  
27 total 2139958.05 100.00
```

Total의 숫자가 작을수록 critical path가 적고 area가 작아짐

# Synthesis

genus

\$> vi 04\_report\_datapath\_map.rpt

- syn\_generic → syn\_map 진행 후 datapath 비교

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
1 Command: report datapath > ./report/04_report_datapath_map.rpt  
2 Warning: '-sort' is not specified. By default, 'report datapath' orders the components based on instance name  
3 =====  
4 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
5 Generated on: Feb 01 2025 11:35:53 pm  
6 Module: cmsdk_mcu  
7 Technology libraries: fast_vdd1v0 1.0  
8 slow_vdd1v0 1.0  
9 giolib045  
10 physical_cells  
11 fast_vdd1v0 1.0  
12 slow_vdd1v0 1.0  
13 giolib045  
14 physical_cells  
15 Operating conditions: PVT_0P9V_125C  
16 Interconnect mode: global  
17 Area mode: physical library  
18 =====  
19  
20  
21 Type CellArea Percentage  
22 -----  
23 datapath modules 0.00 0.00  
24 external muxes 0.00 0.00  
25 others 2139890.38 100.00  
26  
27 total 2139890.38 100.00
```

Total의 숫자가 작을수록 critical path가 적고 area가 작아짐

# Synthesis

genus

\$> vi 05\_MAIN\_CLOCK\_post\_syn\_map.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_svn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
1 =====  
2 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on: Feb 01 2025 11:35:54 pm  
4 Module: cmsdk_mcu  
5 Operating conditions: PVT_OP9V_125C  
6 Interconnect mode: global  
7 Area mode: physical library  
8 =====  
9  
10  
11 Path 1: MET (1644 ps) Setup Check with Pin u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/CKN->D  
12 Group: MAIN_CLOCK  
13 Startpoint: (R) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Xc8l85_reg/CK  
14 Clock: (R) MAIN_CLOCK  
15 Endpoint: (F) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/D  
16 Clock: (R) MAIN_CLOCK  
17  
18 Capture Launch  
19 Clock Edge:+ 7000 0  
20 Src Latency:+ 200 200  
21 Net Latency:+ 200 (I) 200 (I)  
22 Arrival:= 7400 400  
23  
24 Setup:- 31  
25 Uncertainty:- 100  
26 Required Time:= 7269  
27 Launch Clock:- 400  
28 Data Path:- 5225  
29 Slack:= 1644
```

Slack에 문제 없음

# Synthesis

genus

\$> vi 06\_report\_constraint.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MATN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

- Constraint에 만족하는 디자인이 나왔는지 검증

```
1 =====  
2 Generated by:          Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on:          Feb 01 2025 11:36:03 pm  
4 Module:                cmsdk_mcu  
5 Operating conditions: PVT A09V 125C  
Setup Timing, clock_period,  
pulse_width에 문제 없음  
=====  
11 Checking for violation type : Setup Timing Slack  
12 -----  
13 No timing slack violation found.  
14 -----  
15 Checking for violation type : clock_period  
16 -----  
17 No paths found.  
18 -----  
19 Checking for violation type : pulse_width  
20 -----  
21 No paths found.  
22 -----  
23 Checking for violation type : max_capacitance  
24 -----  
25 -----  
26 | Pin      | Required|Actual |Slack | Mode |  
27 |          | Load   | Load   | Load  |  
28 |          | (ff)  | (ff)  | (ff) |  
29 -----
```

# Synthesis

\$> vi 06\_report\_constraint.rpt

genus

- max\_capacitance에 Required Load보다 Actual Load가 크게 걸림
- Io셀을 이미 넣은 상황이기 때문에 **수정 불가**
- BE과정에서 개선의 여지가 있기 때문에 일단 유지

Checking for violation type : max\_capacitance

Pin	Required Load (ff)	Actual Load (ff)	Slack Load (ff)	Mode
port:cmsdk_mcu/P1[0]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[1]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[2]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[3]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[4]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[5]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[6]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[7]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[8]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[9]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[10]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[11]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[12]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[13]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[14]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/P1[15]	888.30	2388.40	-1500.10	default
port:cmsdk_mcu/flash_hrdraft[14]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[13]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[12]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[11]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[10]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[9]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[8]	888.30	2385.40	-1497.10	default
port:cmsdk_mcu/sram_hrdraft[7]	888.30	2385.40	-1497.10	default

# Synthesis

\$> vi 06\_report\_constraint.rpt

genus

- max\_transition에 Required slew보다 Actual slew가 크게 걸림
- 같은 셀 중 드라이브 스트레스가 높은 셀을 찾아 수정 가능

Checking for violation type : max_transition				
	Pin	Required Slew (ps)	Actual Slew (ps)	Slack Slew (ps)
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4850/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4851/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4852/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4853/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4854/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4855/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4856/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4857/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4858/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4859/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4860/B1	280	915	-635
	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsystem/gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g 4861/B1	280	915	-635

No max\_fanout rule violations.

fanout 문제 없음

# Synthesis

genus

\$> vi 07\_MAIN\_CLOCK\_post\_syn\_opt.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
1 =====  
2 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on: Feb 01 2025 11:36:09 pm  
4 Module: cmsdk_mcu  
5 Operating conditions: PVT_OP9V_125C  
6 Interconnect mode: global  
7 Area mode: physical library  
8 =====  
9  
10  
11 Path 1: MET (1645 ps) Setup Check with Pin u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/CKN->D  
12 Group: MAIN_CLOCK  
13 Startpoint: (R) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Xc8l85_reg/CK  
14 Clock: (R) MAIN_CLOCK  
15 Endpoint: (F) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/D  
16 Clock: (R) MAIN_CLOCK  
17  
18 Capture Launch  
19 Clock Edge:+ 7000 0  
20 Src Latency:+ 200 200  
21 Net Latency:+ 200 (I) 200 (I)  
22 Arrival:= 7400 400  
23  
24 Setup:- 31  
25 Uncertainty:- 100  
26 Required Time:= 7269  
27 Launch Clock:- 400  
28 Data Path:- 5225  
29 Slack:= 1645
```

Slack에 문제 없음

# Synthesis

genus

\$> vi 08\_cmsdk\_mcu\_SYN\_analysis\_view.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
1 Multi-Mode Analysis View Report  
2 _____  
3  
4 + ALL Views  
5 |  
6 + Analysis View: default_emulate_view  
7 |  
8 + Delay Calc Corner: default_emulate_delay_corner  
9 |  
10 + timing_condition: default_emulate_timing_cond_max  
11 |  
12 + library_sets: default_emulate_libset_max  
13 |  
14 + timing: /home/ex_poly1/SoC2/SoC/smkcowl_make_S28/TOP/SYN/../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/tim  
ing/fast_vdd1v0_basicCells.lib  
15 /home/ex_poly1/SoC2/SoC/smkcowl_make_S28/TOP/SYN/../../../../GPKD045/digital/gsclib045_all_v4.4/gsclib045/  
timing/slow_vdd1v0_basicCells.lib  
16 /home/ex_poly1/SoC2/SoC/smkcowl_make_S28/TOP/SYN/../../../../GPKD045/digital/giolib045_v3.5/timing/pads_SS  
_slvg.lib  
17 |  
18 |  
19 |  
20 + rc_corner: default_emulate_rc_corner  
21 |  
22 + postRoute_res: 1.0 1.0 1.0  
23 |  
24 + postRoute_cap: 1.0 1.0 1.0  
25 |  
26 + postRoute_xcap: 1.0 1.0 1.0  
27 |  
28 + postRoute_clkcap: 1.0 1.0 1.0  
29 |  
30 + postRoute_clkres: 1.0 1.0 1.0  
31 |
```

# Synthesis

genus

\$> vi 09\_cmsdk\_mcu\_SYN\_area.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

1	=====	Generated by:	Genus(TM) Synthesis Solution 22.16-s078_1	2	3	Generated on:	Feb 01 2025 11:36:11 pm	4	Module:	cmsdk_mcu	5	Operating conditions:	PVT_0P9V_125C	6	Interconnect mode:	global	7	Area mode:	physical library	8
10	Instance	Module	Cell Count	Cell Area	Net Area	Total Area														
11	cmsdk_mcu		13777	2139913.638	18759.351	2158672.989														
12	u_cmsdk_mcu_clockctrl	cmsdk_mcu_clockctrl_PRESENT0	6	52.450	2.303	55.073														
13	u_cmsdk_mcu_system	cmsdk_mcu_system	13566	37429.164	18046.327	55475.491														
14	u_addr_decode	cmsdk_mcu_addr_decode_BASEADDR_GPIO0032h40010000_BA	31	46.170	23.187	69.357														
15	u_ahb_default_slave_1	cmsdk_ahb_default_slave	5	18.810	1.844	20.654														
16	u_ahb_gpio_0	cmsdk_ahb_gpio_ALTERNATE_FUNC_MASK16h0000_ALTERNAT	625	1794.474	694.475	2488.949														
17	u_ahb_to_gpio	cmsdk_ahb_to_gpio	18	106.020	0.741	106.761														
18	u_iop_gpio	cmsdk_iop_gpio_ALTERNATE_FUNC_MASK16h0000_ALTERNAT	607	1688.454	647.549	2336.003														
19	u_ahb_gpio_1	cmsdk_ahb_gpio_ALTERNATE_FUNC_MASK16h002a_ALTERNAT	643	1832.094	731.175	2563.269														
20	u_ahb_to_gpio	cmsdk_mcu_system_cmsdk_ahb_to_iop_1	18	106.020	0.741	106.761														
21	u_iop_gpio	cmsdk_iop_gpio_ALTERNATE_FUNC_MASK16h002a_ALTERNAT	625	1726.074	683.527	2409.601														
22	u_ahb_slave_mux_sys_bus	cmsdk_ahb_slave_mux_PORT0_ENABLE1_PORT1_ENABLE1_PO	127	271.890	128.537	400.427														
23	u_apb_subsystem	cmsdk_apb_subsystem_APB_EXT_PORT12_ENABLE0_APB_EXT	4192	12309.606	5217.094	17526.700														
24	gen_apb_dualtimers_2.u_apb_dualtimers_2	cmsdk_apb_dualtimers	1095	3471.642	1389.577	4861.219														
25	u_apb_timer_frc_1	cmsdk_apb_dualtimers_frc	490	1549.602	621.340	2170.942														
26	u_apb_timer_frc_2	cmsdk_mcu_system_cmsdk_apb_dualtimers_frc_1	490	1548.576	620.979	2169.555														
27	gen_apb_test_slave.u_apb_test_slave	cmsdk_apb_test_slave	120	361.836	101.681	463.517														
28	gen_apb_timer_0.u_apb_timer_0	cmsdk_apb_timer	383	1052.676	462.774	1515.450														
29	gen_apb_timer_1.u_apb_timer_1	cmsdk_mcu_system_cmsdk_apb_timer_1	383	1053.018	462.774	1515.792														
30	gen_apb_uart_0.u_apb_uart_0	cmsdk_apb_uart	481	1341.324	546.210	1887.534														
31	gen_apb_uart_1.u_apb_uart_1	cmsdk_mcu_system_cmsdk_apb_uart_1	481	1341.324	546.210	1887.534														
32	gen_apb_uart_2.u_apb_uart_2	cmsdk_mcu_system_cmsdk_apb_uart_2	480	1340.982	545.469	1886.451														
33	gen_apb_watchdog.u_apb_watchdog	cmsdk_apb_watchdog	469	1535.238	546.761	2081.999														
34	u_apb_watchdog_frc	cmsdk_apb_watchdog_frc	342	1172.034	424.097	1596.131														
35	u_ahb_to_apb	cmsdk_ahb_to_apb_ADDRWIDTH16_REGISTER_RDATA1_REGIS	140	504.108	94.060	598.168														
36	u_apb_slave_mux	cmsdk_apb_slave_mux_PORT0_ENABLE1_PORT1_ENABLE1_PO	157	303.354	87.769	391.123														
37	u_cmsdk_mcu_stclkctrl	cmsdk_mcu_stclkctrl_DIV_RATIO18h003e8	30	93.708	31.626	125.334														
38	u_cmsdk_mcu_sysctrl	cmsdk_mcu_sysctrl_BE0	81	222.300	85.165	307.465														
39	u_cortexm0integration	CORTEXM0INTEGRATION	7790	20730.330	10901.048	31631.378														
40	u_cortexm0	CORTEXM0DS	7790	20730.330	10901.048	31631.378														
41	u_logic	cortexm0ds_logic	7790	20730.330	10901.048	31631.378														

디자인 전체의  
Area 확인

# Synthesis

genus

\$> vi 10\_cmsdk\_mcu\_SYN\_clock.rpt

- Clock에 대한 정보를 확인함

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

```
1 =====  
2 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on: Feb 01 2025 11:36:11 pm  
4 Module: cmsdk_mcu  
5 Operating conditions: PVT_0P9V_125C  
6 Interconnect mode: global  
7 Area mode: physical library  
8 =====  
9  
10 11 Clock Description  
12 -----  
13  
14 15 Clock  
15 Name Period Rise Fall Clock Source No of  
16 Domain Pin/Port Registers  
17 MAIN_CLOCK 7000.0 0.0 3500.0 domain_1 XTAL1 2285  
18  
19 20 Clock Network Latency / Setup Uncertainty  
21  
22 23 Network Network Source Source Setup Setup  
23 Clock Latency Latency Latency Latency Uncertainty Uncertainty  
24 Name Rise Fall Rise Fall Rise Fall  
25  
26 MAIN_CLOCK 200.0 200.0 200.0 200.0 100.0 100.0  
27  
28 29 Clock Relationship (with uncertainty & latency)  
29 -----
```

# Synthesis

genus

\$> vi 11\_cmsdk\_mcu\_SYN\_datapath.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock_rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

- syn\_map → syn\_opt 진행 후 datapath 비교

```
1 Command: report datapath > ./report/11_cmsdk_mcu_SYN_datapath.rpt  
2 Warning: '-sort' is not specified. By default, 'report datapath' orders the components based on instance name  
3 =====  
4 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
5 Generated on: Feb 01 2025 11:36:11 pm  
6 Module: cmsdk_mcu  
7 Operating conditions: PVT_0P9V_125C  
8 Interconnect mode: global  
9 Area mode: physical library  
10 =====  
11  
12  
13      Type          CellArea  Percentage  
14 -----  
15  datapath modules    0.00      0.00  
16  external muxes     0.00      0.00  
17  others             2139913.64  100.00  
18  
19  total              2139913.64  100.00  
20
```

Total의 숫자가 작을수록 critical path가 적고 area가 작아짐

# Synthesis

genus

\$> vi 12\_cmsdk\_mcu\_SYN\_design\_rulse.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

## DRC: Design Rule Check

- /Max 검색하여 DRC 관련 정보를 확인함
- Max capacitance, Max fanout, Max transition 확인

```
1 =====  
2 Generated by:          Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on:          Feb 01 2025 11:36:11 pm  
4 Module:                cmsdk_mcu  
5 Operating conditions: PVT_0P9V_125C  
6 Interconnect mode:    global  
7 Area mode:             physical library  
8 =====  
9  
10 Max_transition design rule (violation total = 1034288)  
11 Pin                      Slew      Max      Violation  
12 -----  
13 u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/g207479/B  
         912        280       632  
14   --> Other violations on net  
           40422  
15 u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/g207480/B  
         912        280       632  
16   --> Other violations on net  
           40422  
17 u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/g207481/B  
         912        280       632  
18   --> Other violations on net  
           40422  
19 u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/g207482/B  
         912        280       632
```

# Synthesis

genus

\$> vi 13\_cmsdk\_mcu\_SYN\_gate.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

1	=====	2	Generated by:	Genus(TM) Synthesis Solution 22.16-s078_1
2		3	Generated on:	Feb 01 2025 11:36:12 pm
4		5	Module:	cmsdk_mcu
6		7	Operating conditions:	PVT_0P9V_125C
8		9	Interconnect mode:	global
10		11	Area mode:	physical library
11	Gate	Instances	Area	Library
12	-----			
13	ADDFX1	215	1102.950	fast_vdd1v0
14	ADDHX1	7	26.334	fast_vdd1v0
15	AND2X1	134	183.312	fast_vdd1v0
16	AND2XL	164	224.352	fast_vdd1v0
17	AND3XL	15	30.780	fast_vdd1v0
18	AND4X1	5	11.970	fast_vdd1v0
19	AND4XL	11	26.334	fast_vdd1v0
20	A021X1	3	7.182	fast_vdd1v0
21	A021XL	12	28.728	fast_vdd1v0
22	A022X1	1	2.736	fast_vdd1v0
23	A022XL	173	473.328	fast_vdd1v0
24	AOI211X1	2	4.788	fast_vdd1v0
25	AOI211XL	22	45.144	fast_vdd1v0
26	AOI21X1	84	143.640	fast_vdd1v0
27	AOI21XL	141	241.110	fast_vdd1v0
28	AOI221X1	386	924.084	fast_vdd1v0
29	AOI222X1	169	520.182	fast_vdd1v0

# Synthesis

genus

\$> vi 14\_cmsdk\_mcu\_SYN\_hier.rpt

- 내 디자인의 계층 구조를 확인함

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt  
  
1 =====  
2 Generated by: Genus(TM) Synthesis Solution 22.16-s078_1  
3 Generated on: Feb 01 2025 11:36:13 pm  
4 Module: cmsdk_mcu  
5 Operating conditions: PVT_0P9V_125C  
6 Interconnect mode: global  
7 Area mode: physical library  
8 =====  
9  
10  
11 Hierarchy Report Format :  
12  
13 level instance ( module ) <status>  
14  
15 status : preserve_<value> -- indicating preserve hierarchy or inherited_preserve value  
16 : blackbox -- indicating unresolved instance  
17  
18 =====  
19  
20 0 cmsdk_mcu  
21 1 u_cmsdk_mcu_clkctrl ( cmsdk_mcu_clkctrl_CLKGATE_PRESENT0 )  
22 1 u_cmsdk_mcu_system ( cmsdk_mcu_system ) preserve_true  
23 2 u_addr_decode ( cmsdk_mcu_addr_decode_BASEADDR_GPIO032h40010000_BASEADDR_GPIO132h40011000_  
BOOT_LOADER_PRESENT1h0_BASEADDR_SYSROMTABLE32hf0000000 ) preserve_true
```

# Synthesis

genus

\$> vi 15\_cmsdk\_mcu\_SYN\_mem\_cell.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

- 메모리 관련 정보 확인 가능

```
1 =====  
2 # Generated by: Cadence Genus(TM) Synthesis Solution 22.16-s078_1  
3 # Generated on: Feb 1 2025 23:36:13  
4  
5 Library : fast_vdd1v0  
6 No memory cells found  
7  
8 Library : slow_vdd1v0  
9 No memory cells found  
10  
11 Library : giolib045  
12 No memory cells found
```

내 디자인에 메모리가  
없음을 확인함

# Synthesis

genus

\$> vi 16\_cmsdk\_mcu\_SYN\_message.rpt

```
[ex_poly1@npit SYN]$ cd report/  
[ex_poly1@npit report]$ ls  
00_check_design.rpt  
01_check_timing.rpt  
02_report_port.rpt  
03_report_datapath_generic.rpt  
04_report_datapath_map.rpt  
05_default_post_syn_map.rpt  
05_MAIN_CLOCK_post_syn_map.rpt  
06_report_constraint.rpt  
07_default_post_syn_opt.rpt  
07_MAIN_CLOCK_post_syn_opt.rpt  
08_cmsdk_mcu_SYN_analysis_view.rpt  
09_cmsdk_mcu_SYN_area.rpt  
10_cmsdk_mcu_SYN_clock.rpt  
11_cmsdk_mcu_SYN_datapath.rpt  
12_cmsdk_mcu_SYN_design_rules.rpt  
13_cmsdk_mcu_SYN_gate.rpt  
14_cmsdk_mcu_SYN_hier.rpt  
15_cmsdk_mcu_SYN_mem_cell.rpt  
16_cmsdk_mcu_SYN_message.rpt
```

```
17_cmsdk_mcu_SYN_net.rpt  
18_cmsdk_mcu_SYN_pwr.rpt  
19_cmsdk_mcu_SYN_qor.rpt  
20_cmsdk_mcu_SYN_timing.rpt  
final.rpt  
syn_generic_cmsdk_mcu.db  
syn_generic_qor.rpt  
syn_generic_time.rpt  
syn_map_area.rpt  
syn_map_cmsdk_mcu.db  
syn_map_gates.rpt  
syn_map_qor.rpt  
syn_map_time.rpt  
syn_opt_area.rpt  
syn_opt_cmsdk_mcu.db  
syn_opt_gates.rpt  
syn_opt_qor.rpt  
syn_opt_time.rpt
```

- 합성과정에서 나온 정보를 메시지로 확인할 수 있음

한 개의 사용하지 않는 register를 삭제한다는 메시지

```
1 ======  
2 Message S  
3 ======  
4 -----  
5 |0 ...| Sev  
6 -----  
7 |CDFG-508|Warning| 1|Removing unused register.  
8 ||||| Genus removes the flip flop or  
9 ||||| latch inferred for an unused  
10 ||||| signal or variable. To preserve  
11 ||||| the flip-flop or latch, set the  
12 ||||| hdl_preserve_unused_registers  
13 ||||| attribute to true or use a  
14 ||||| pragma in the RTL.  
15 |CDFG-818|Warning| 1|Using default parameter value  
16 ||||| for module elaboration.  
17 |CFM-1|Info | 1|Wrote dofile.  
18 |CFM-5|Info | 1|Wrote formal verification  
19 ||||| information.  
20 |DPOPT-5|Info | 1|Skipping datapath optimization.  
21 |DPOPT-6|Info | 1|Pre-processed datapath logic.  
22 |ELAB-1|Info | 1|Elaborating Design.  
23 |ELAB-3|Info | 1|Done Elaborating Design.  
24 |ELABUTL-123|Warning| 20|Undriven module output port.  
25 |ELABUTL-127|Warning| 20|Undriven module input port.  
26 ||||| Run check_design to check  
27 ||||| 'Undriven Port(s)/Pin(s)  
28 ||||| ' section for all undriven  
29 ||||| module input ports. It is
```

# Synthesis

genus

\$> vi 17\_cmsdk\_mcu\_SYN\_net.rpt

- Wire에 대한 정보를 담고 있음

```
17_cmsdk_mcu_SYN_net.rpt
18_cmsdk_mcu_SYN_pwr.rpt
19_cmsdk_mcu_SYN_qor.rpt
20_cmsdk_mcu_SYN_timing.rpt
final.rpt
syn_generic_cmsdk_mcu.db
syn_generic_qor.rpt
syn_generic_time.rpt
syn_map_area.rpt
syn_map_cmsdk_mcu.db
syn_map_gates.rpt
syn_map_qor.rpt
syn_map_time.rpt
syn_opt_area.rpt
syn_opt_cmsdk_mcu.db
syn_opt_gates.rpt
syn_opt_qor.rpt
syn_opt_time.rpt
```

Net	Loads	Drivers	Wire Cap(fF)	Wire Res(k-ohm)	Wireload Model
APBACTIVE	0	1	0.0	0.000	
FCLK	8	1	0.0	0.000	
LOCKUP	2	1	5.5	0.024	
LOCKUPRESET	2	1	4.8	0.021	
PMUENABLE	0	1	0.0	0.000	
PORESETn	1	1	0.0	0.000	
PRESETn	1	1	0.0	0.000	
SLEEPING	0	1	0.0	0.000	
SYSRESETREQ	1	1	3.2	0.014	
WDGRESETREQ	1	1	2.4	0.010	
cmsdk_SYSRESETREQ	1	1	1.6	0.007	
i_swclkck	0	1	0.0	0.000	
i_swditms	0	1	0.0	0.000	
i_tdi	0	1	0.0	0.000	
i_trst_n	0	1	0.0	0.000	
in_NRST	1	1	0.0	0.000	
in_flash_hrdata[0]	1	1	1.6	0.007	
in_flash_hrdata[1]	1	1	1.6	0.007	
in_flash_hrdata[2]	1	1	1.6	0.007	
in_flash_hrdata[3]	1	1	1.6	0.007	

# Synthesis

genus

\$> vi 18\_cmsdk\_mcu\_SYN\_pwr.rpt

- Power 소모 관련 정보를 담고 있음

```
17_cmsdk_mcu_SYN_net.rpt
18_cmsdk_mcu_SYN_pwr.rpt
19_cmsdk_mcu_SYN_qor.rpt
20_cmsdk_mcu_SYN_timing.rpt
final.rpt
syn_generic_cmsdk_mcu.db
syn_generic_qor.rpt
syn_generic_time.rpt
syn_map_area.rpt
syn_map_cmsdk_mcu.db
syn_map_gates.rpt
syn_map_qor.rpt
syn_map_time.rpt
syn_opt_area.rpt
syn_opt_cmsdk_mcu.db
syn_opt_gates.rpt
syn_opt_qor.rpt
syn_opt_time.rpt
```

Category	Leakage	Internal	Switching	Total	Row%
	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
memory	9.44624e-07	1.52379e-03	3.23107e-04	1.84784e-03	1.82%
register	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
latch	1.23094e-06	4.65504e-04	1.02382e-03	1.49055e-03	1.47%
logic	4.37853e-10	7.33379e-07	3.20881e-04	3.21615e-04	0.32%
bbox	5.81506e-03	8.98785e-02	2.26011e-03	9.79537e-02	96.40%
clock	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	5.81724e-03	9.18686e-02	3.92791e-03	1.01614e-01	100.01%
Subtotal	5.72%	90.41%	3.87%	100.00%	100.00%
Percentage					

# Synthesis

genus

\$> vi 19\_cmsdk\_mcu\_SYN\_qor.rpt

```
17_cmsdk_mcu_SYN_net.rpt
18_cmsdk_mcu_SYN_pwr.rpt
19_cmsdk_mcu_SYN_qor.rpt
20_cmsdk_mcu_SYN_timing.rpt
final.rpt
syn_generic_cmsdk_mcu.db
syn_generic_qor.rpt
syn_generic_time.rpt
syn_map_area.rpt
syn_map_cmsdk_mcu.db
syn_map_gates.rpt
syn_map_qor.rpt
syn_map_time.rpt
syn_opt_area.rpt
syn_opt_cmsdk_mcu.db
syn_opt_gates.rpt
syn_opt_qor.rpt
syn_opt_time.rpt
```

qor: quality of results

- 전체적인 통계를 확인할 수 있음

```
1 =====
2 Generated by:           Genus(TM) Synthesis Solution 22.16-s078_1
3 Generated on:          Feb 01 2025 11:36:16 pm
4 Module:                cmsdk_mcu
5 Operating conditions: PVT_0P9V_125C
6 Interconnect mode:    global
7 Area mode:             physical library
8 =====
9
10 Timing
11 -----
12
13   Clock   Period
14 -----
15 MAIN_CLOCK 7000.0
16
17
18   Cost      Critical      Violating
19   Group     Path Slack   TNS    Paths
20 -----
21 default      No paths   0.0
22 MAIN_CLOCK   1644.7    0.0      0
23 -----
24 Total         0.0        0
25
26 Instance Count
27 -----
28 Leaf Instance Count           13777
29 Physical Instance count       0
30 Sequential Instance Count     2285
31 Combinational Instance Count 11492
32 Hierarchical Instance Count   32
33
34 Area
```

# Synthesis

genus

\$> vi 20\_cmsdk\_mcu\_SYN\_timing.rpt

```
17_cmsdk_mcu_SYN_net.rpt
18_cmsdk_mcu_SYN_pwr.rpt
19_cmsdk_mcu_SYN_qor.rpt
20_cmsdk_mcu_SYN_timing.rpt
final.rpt
syn_generic_cmsdk_mcu.db
syn_generic_qor.rpt
syn_generic_time.rpt
syn_map_area.rpt
syn_map_cmsdk_mcu.db
syn_map_gates.rpt
syn_map_qor.rpt
syn_map_time.rpt
syn_opt_area.rpt
syn_opt_cmsdk_mcu.db
syn_opt_gates.rpt
syn_opt_qor.rpt
syn_opt_time.rpt
```

```
1 =====
2 Generated by:      Genus(TM) Synthesis Solution 22.16-s078_1
3 Generated on:      Feb 01 2025 11:36:17 pm
4 Module:           cmsdk_mcu
5 Operating conditions: PVT_0P9V_125C
6 Interconnect mode: global
7 Area mode:        physical library
8 =====
9
10
11 Path 1: MET (1645 ps) Setup Check with Pin u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/CKN->D
12     Group: MAIN_CLOCK
13     Startpoint: (R) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/Xc8l85_reg/CK
14     Clock: (R) MAIN_CLOCK
15     Endpoint: (F) u_cmsdk_mcu_system/u_cortexm0integration/u_cortexm0/u_logic/0o3l85_reg/D
16     Clock: (R) MAIN_CLOCK
17
18     Capture          Launch
19     Clock Edge:+    7000          0
20     Src Latency:+   200           200
21     Net Latency:+  200 (I)       200 (I)
22     Arrival:=       7400          400
23
24     Setup:-          31
25     Uncertainty:-   100
26     Required Time:= 7269
27     Launch Clock:-  400
28     Data Path:-     5225
29     Slack:=         1645
30
31 #
32 #                                     Timing Point
33 #                                         Flags Arc Edge Cell Fanout Load Trans Delay Arrival Instance
34 #                                         (fF) (ps) (ps) (ps) Location
35 u_cmsdk_mcu_system/u_cortexm0/u_logic/Xc8l85_reg/CK - - R (arrival) 1445 - 100 0 400 (-,-)
36 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/Xc8l85_reg/Q (P) CK->Q R DFFRX1 37 45.0 306 242 642 (-,-)
37 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/Xc8l85 - - - (net) - - - - - (-,-)
38 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/g215877/Y (P) A->Y R OR2X1 19 22.5 155 152 794 (-,-)
39 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/n_5666 - - - (net) - - - - - (-,-)
40 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/g207746/Y (P) C->Y R OR3XL 2 3.0 38 52 847 (-,-)
41 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/n_5665 - - - (net) - - - - - (-,-)
42 u_cmsdk_mcu_system/u_cortexm0/integration/u_cortexm0/u_logic/g207696/Y (P) B->Y R OR2XL 5 6.5 77 57 904 (-,-)
```

Slack에 문제 없음

# Synthesis

genus

\$> cd log

- log 폴더에 툴의 시작부터 종료까지의 메시지가 모두 저장되어 있음

```
[ex_poly1@npit SYN]$ ls  
clean.tcl  cons  fv  log  mapped  report  run_synthesis  script  unmapped
```

# Synthesis

## genus

```
$> vi all.log
```

- all.log 파일은 툴의 시작부터 종료까지의 메시지가 모두 저장되어 있는 파일

```
[ex_poly1@npit log]$ ls  
all.cmd all.log
```

# Synthesis

genus

\$> vi all.log

```
Cadence Genus(TM) Synthesis Solution.  
Copyright 2024 Cadence Design Systems, Inc. All rights reserved worldwide.  
Cadence and the Cadence logo are registered trademarks and Genus is a trademark  
of Cadence Design Systems, Inc. in the United States and other countries.  
  
[00:12:47.440177] Configured Lic search path (22.01-s003): 35266@npit-service1.ptime.org  
  
Version: 22.16-s078_1, built Sun Jun 09 22:32:57 PDT 2024  
Options: -files ./script/cortexm0_45nm.tcl -log ./log/all.log  
Date: Mon Feb 03 00:12:47 2025  
Host: npit.ic.rnd1 (x86_64 w/Linux 3.10.0-1160.119.1.el7.x86_64) (20cores*80cpus*  
2physical cpus*Intel(R) Xeon(R) Silver 4316 CPU @ 2.30GHz 30720KB) (395262912KB)  
PID: 33843  
OS Linux Server release 7.9 (Maipo)  
Che 2937줄이 생성된 것을 알 수 있음 Genus_Synthesis  
[00:12:47.028131] Lic Lic check successful  
[00:12:47.028131] Feature usage summary:  
"all.log" 2937L, 199957C
```

# Synthesis

genus

- **/Error** 명령어를 통해 툴 사용 전반의 에러유무를 확인함
- 이전 report에서 확인했던 사항들을 다시 확인할 수 있음

\$> vi all.log

```
Info    : PWRA-0002 Finished power computation.
Info    : PWRA-0007 [PwrInfo] Completed successfully.
          : Info=6, Warn=2, Error=0, Fatal=0
Finished exporting design database to file './report/syn_map_cmsdk_mcu.db' for 'cmsdk_mcu' (command execution time mm:ss cpu = 00:01, real = 00:02).
Finished generating snapshot at stage syn_map (command execution time mm:ss cpu = 00:00, real = 00:04).
%# End write_snapshot (02/03 00:13:51, total cpu=17:00:00, real=17:00:04, peak res=962.80M, current mem=1837.57M)
@file(cortexm0_45nm.tcl) 132: write_db -to_file ${MAPPED_DIR}/syn_map.db
Finished exporting design database to file './mapped/syn_map.db' for 'cmsdk_mcu' (command execution time mm:ss cpu = 00:01, real = 00:01).
@file(cortexm0_45nm.tcl) 138: syn_opt ; #This command can't write report by using the mark '>'
Current PLE settings:

Aspect ratio      : 1.000
Shrink factor     : 1.000
Scale of res/length : 1.000
search hit BOTTOM, continuing at TOP
```

검색 결과를 전부  
확인했다는 뜻임

# Synthesis

## genus

- mapped 폴더에서 합성 최종 결과 파일들을 확인할 수 있음

```
$> cd mapped
```

```
[ex_poly1@npit SYN]$ ls  
clean.tcl  cons  fv  log  mapped  report  run_synthesis  script  unmapped
```

# Synthesis

genus

\$> //

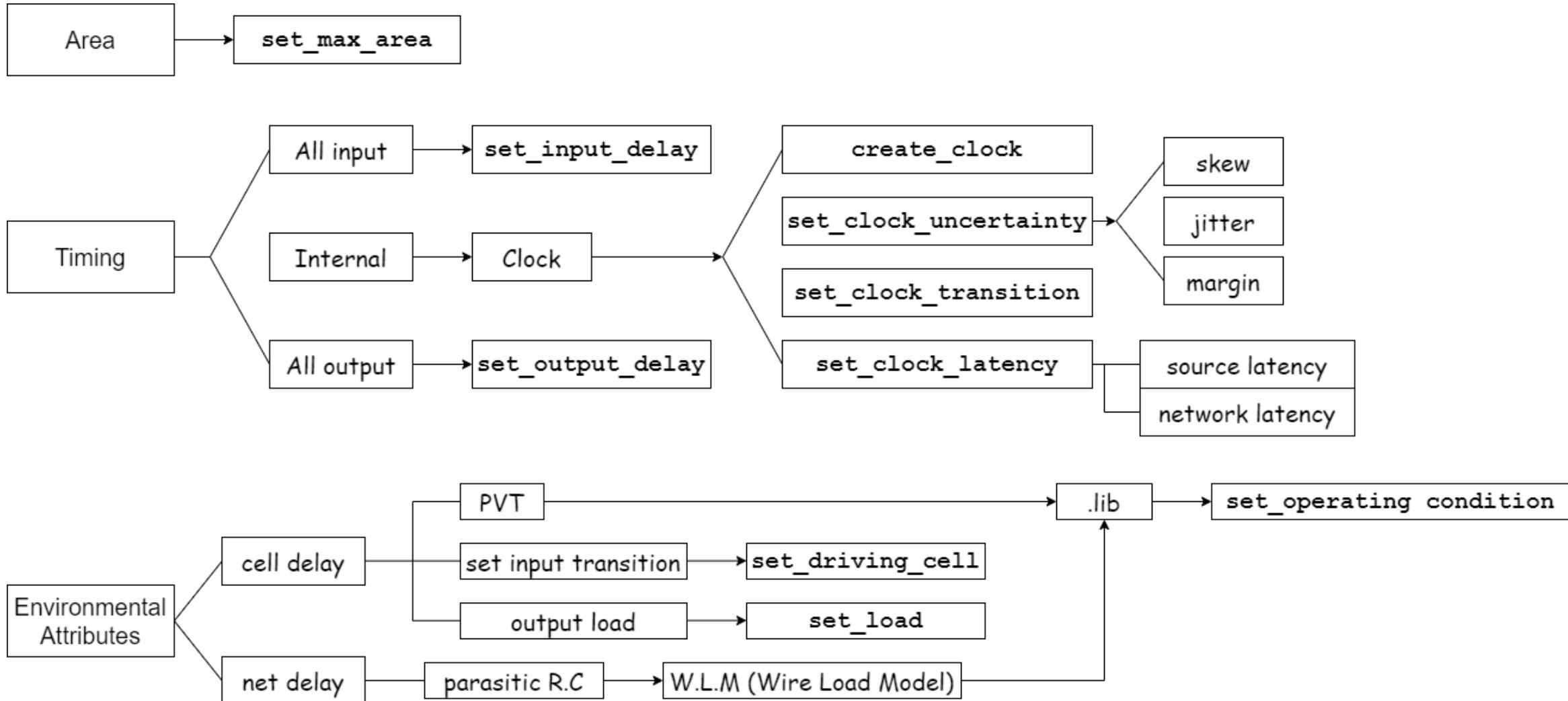
- mapped 내용 확인
- 합성 후 마지막에 write out했던 파일을 확인할 수 있음
- mapped의 파일들을 다음 단계 툴에서 입력으로 사용함

```
182 #####
183 ## write Innovus file set (verilog, SDC, config, etc.)
184 #####
185
186 #source -echo ./script/report_syn.tcl
187 write design -basename ${MAPPED_DIR}/${TOP DESIGN} mapped
188 write_hdl > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.v
189 write_sdc > ${MAPPED_DIR}/${TOP DESIGN}_syn_final.sdc
190 write_sdf -timescale ns -nonegchecks -recrem split -edges check_edge
    -setuphold split > ${MAPPED_DIR}/${TOP DESIGN}_delays.sdf
```

```
[ex_poly1@npit mapped]$ ll
total 12468
-rw-r--r-- 1 ex_poly1 rnd 6276538 Feb 3 15:42 cmsdk_mcu_delays.sdf
-rw-r--r-- 1 ex_poly1 rnd 54926 Feb 3 15:42 cmsdk_mcu_mapped.default_emulate_constraint_mode.sdc
-rw-r--r-- 1 ex_poly1 rnd 3949 Feb 3 15:42 cmsdk_mcu_mapped.dont_touch.sdc
-rw-r--r-- 1 ex_poly1 rnd 2102131 Feb 3 15:42 cmsdk_mcu_mapped.g
-rw-r--r-- 1 ex_poly1 rnd 783 Feb 3 15:42 cmsdk_mcu_mapped.genus_init.tcl
-rw-r--r-- 1 ex_poly1 rnd 4414 Feb 3 15:42 cmsdk_mcu_mapped.genus_setup.tcl
-rw-r--r-- 1 ex_poly1 rnd 27106 Feb 3 15:42 cmsdk_mcu_mapped.lec.taf.gz
-rw-r--r-- 1 ex_poly1 rnd 56638 Feb 3 15:42 cmsdk_mcu_mapped.metrics.json
-rw-r--r-- 1 ex_poly1 rnd 2399 Feb 3 15:42 cmsdk_mcu_mapped.mmmc.tcl
-rw-r--r-- 1 ex_poly1 rnd 5379 Feb 3 15:42 cmsdk_mcu_mapped.preserve.tcl
-rw-r--r-- 1 ex_poly1 rnd 11002 Feb 3 15:42 cmsdk_mcu_mapped.root.g
-rw-r--r-- 1 ex_poly1 rnd 20 Feb 3 15:42 cmsdk_mcu_mapped.safety.taf.gz
-rw-r--r-- 1 ex_poly1 rnd 1475673 Feb 3 15:42 cmsdk_mcu_mapped.v
-rw-r--r-- 1 ex_poly1 rnd 54926 Feb 3 15:42 cmsdk_mcu_syn_final.sdc
-rw-r--r-- 1 ex_poly1 rnd 1475673 Feb 3 15:42 cmsdk_mcu_syn_final.v
-rw-r--r-- 1 ex_poly1 rnd 586856 Feb 3 15:42 syn_map.ab
-rw-r--r-- 1 ex_poly1 rnd 588392 Feb 3 15:42 syn_opt.db
```

# Synthesis(정리 및 요약)

## Constraint



# Synthesis(정리 및 요약)

## 합성 검증 4단계

1. **check\_timing –intent**
2. **report\_port**
3. **report\_constraint**
4. **report\_timing**

합성 후 반드시 확인해야 할 결과임

①

Total: 0

②

External Driver/Slew							
Port	Dir	External Driver Max-Rise	External Driver Max-Fall	External Slew Max-Rise	External Slew Max-Fall	Timing Case Value	Timing Ideal Driver
XTAL1	in			0.0	0.0		false
NRST	in	PADDI/Y	PADDI/Y	0.0	0.0		false
nTRST	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false
SWCLKTCK	in	PADDI/Y	PADDI/Y	3317.1	1443.3		false

# Synthesis(정리 및 요약)

## 합성 검증 4단계

1. check\_timing\_intent
2. report\_port
3. report\_constraint
4. report\_timing

③

Checking for violation type : Setup Timing Slack	←																				
No timing slack violation found.	←																				
Checking for violation type : clock_period	←																				
No paths found.	←																				
Checking for violation type : pulse_width	←																				
No paths found.	←																				
Checking for violation type : max_transition	←																				
<table border="1"><thead><tr><th>Pin</th><th>Required Slew (ps)</th><th>Actual Slew (ps)</th><th>Slack (ps)</th><th>Mode</th></tr></thead><tbody><tr><td>pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1</td><td>280</td><td>915</td><td>-635</td><td>default</td></tr><tr><td>pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1</td><td>280</td><td>915</td><td>-635</td><td>default</td></tr></tbody></table>	Pin	Required Slew (ps)	Actual Slew (ps)	Slack (ps)	Mode	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1	280	915	-635	default	pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1	280	915	-635	default	←					
Pin	Required Slew (ps)	Actual Slew (ps)	Slack (ps)	Mode																	
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1	280	915	-635	default																	
pin:cmsdk_mcu/u_cmsdk_mcu_system/u_apb_subsyste...m_gen_apb_watchdog.u_apb_watchdog/u_apb_watchdog_frc/g4850/B1	280	915	-635	default																	
Checking for violation type : max_capacitance	←																				
<table border="1"><thead><tr><th>Pin</th><th>Required Load (ff)</th><th>Actual Load (ff)</th><th>Slack Load (ff)</th><th>Mode</th></tr></thead><tbody><tr><td>port:cmsdk_mcu/P1[0]</td><td>888.30</td><td>2388.40</td><td>-1500.10</td><td>default</td></tr><tr><td>port:cmsdk_mcu/P1[1]</td><td>888.30</td><td>2388.40</td><td>-1500.10</td><td>default</td></tr><tr><td>port:cmsdk_mcu/P1[2]</td><td>888.30</td><td>2388.40</td><td>-1500.10</td><td>default</td></tr></tbody></table>	Pin	Required Load (ff)	Actual Load (ff)	Slack Load (ff)	Mode	port:cmsdk_mcu/P1[0]	888.30	2388.40	-1500.10	default	port:cmsdk_mcu/P1[1]	888.30	2388.40	-1500.10	default	port:cmsdk_mcu/P1[2]	888.30	2388.40	-1500.10	default	←
Pin	Required Load (ff)	Actual Load (ff)	Slack Load (ff)	Mode																	
port:cmsdk_mcu/P1[0]	888.30	2388.40	-1500.10	default																	
port:cmsdk_mcu/P1[1]	888.30	2388.40	-1500.10	default																	
port:cmsdk_mcu/P1[2]	888.30	2388.40	-1500.10	default																	
No max_fanout rule violations.	←																				

합성 후 반드시 확인해야 할 결과임

# Synthesis(정리 및 요약)

## 합성 검증 4단계

1. **check\_timing –intent**
2. **report\_port**
3. **report\_constraint**
4. **report\_timing**

④

Setup:-	31
Uncertainty:-	100
Required Time:=	7269
Launch Clock:-	400
Data Path:-	5225
<b>Slack:=</b>	<b>1645</b>

합성 후 반드시 확인해야 할 결과임