

합성(Synthesis)

Introduction to Synthesis

```
always @ (posedge CLK in_XTAL1 or posedge in_Rom_select)
begin
    if (in_Rom_select == 1'b1)
        reset_connect <= 1'b0;
    else
        reset_connect <= 1'b1;
    :
end
```

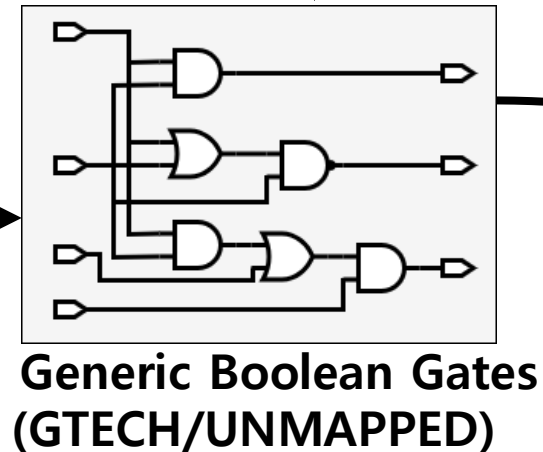
RTL source

① **Translate** (read_Verilog)

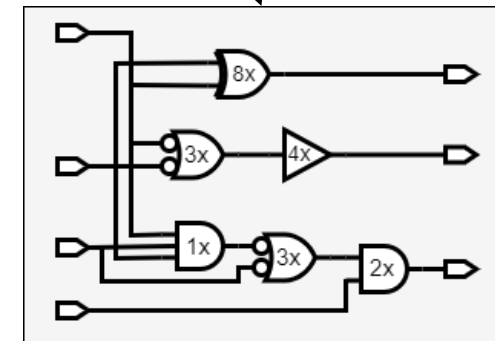
Constraints

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

② **Constraint** (source)



③ **Map + Optimize**



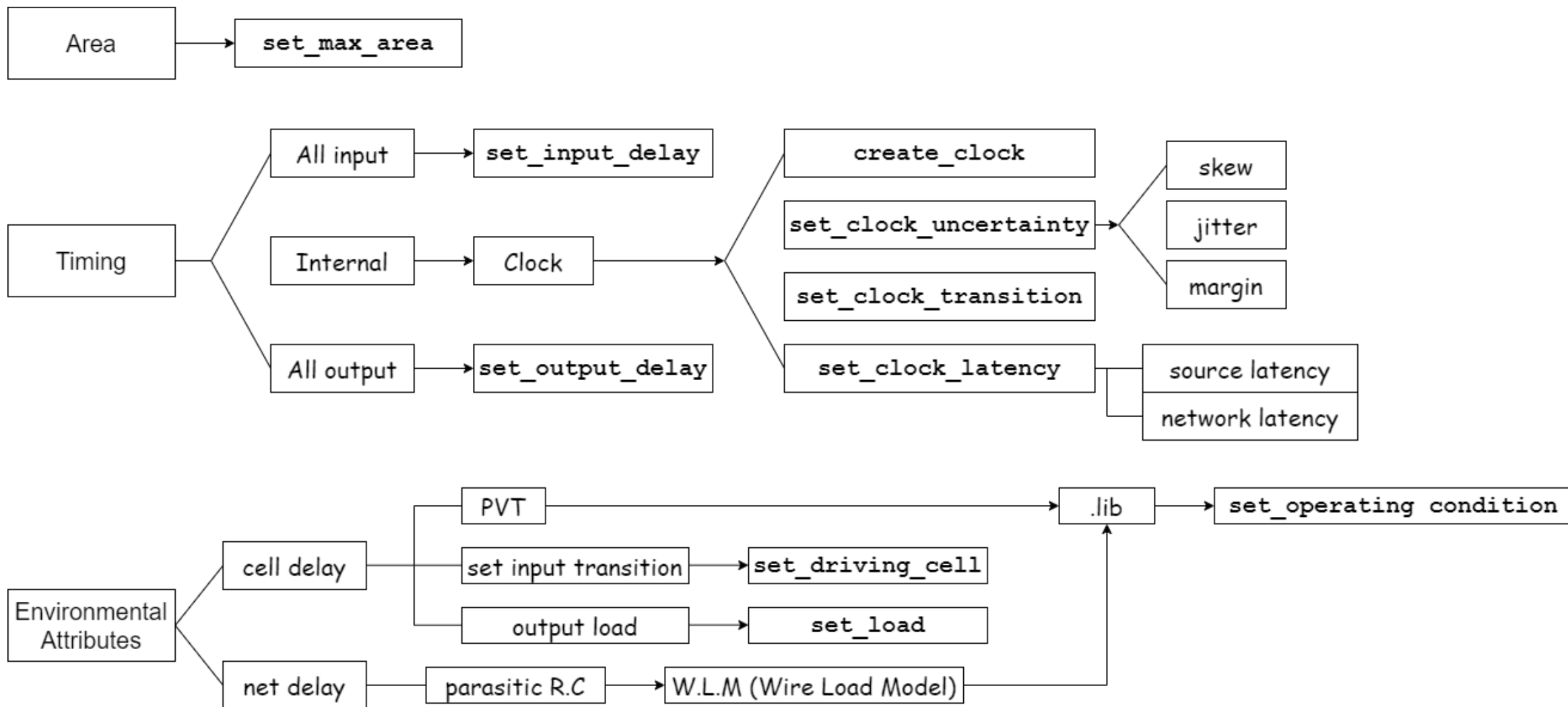
④ **Save** (Write out Gate Level Netlist)

Technology specific Gates (MAPPED)

Constraints

Synthesis

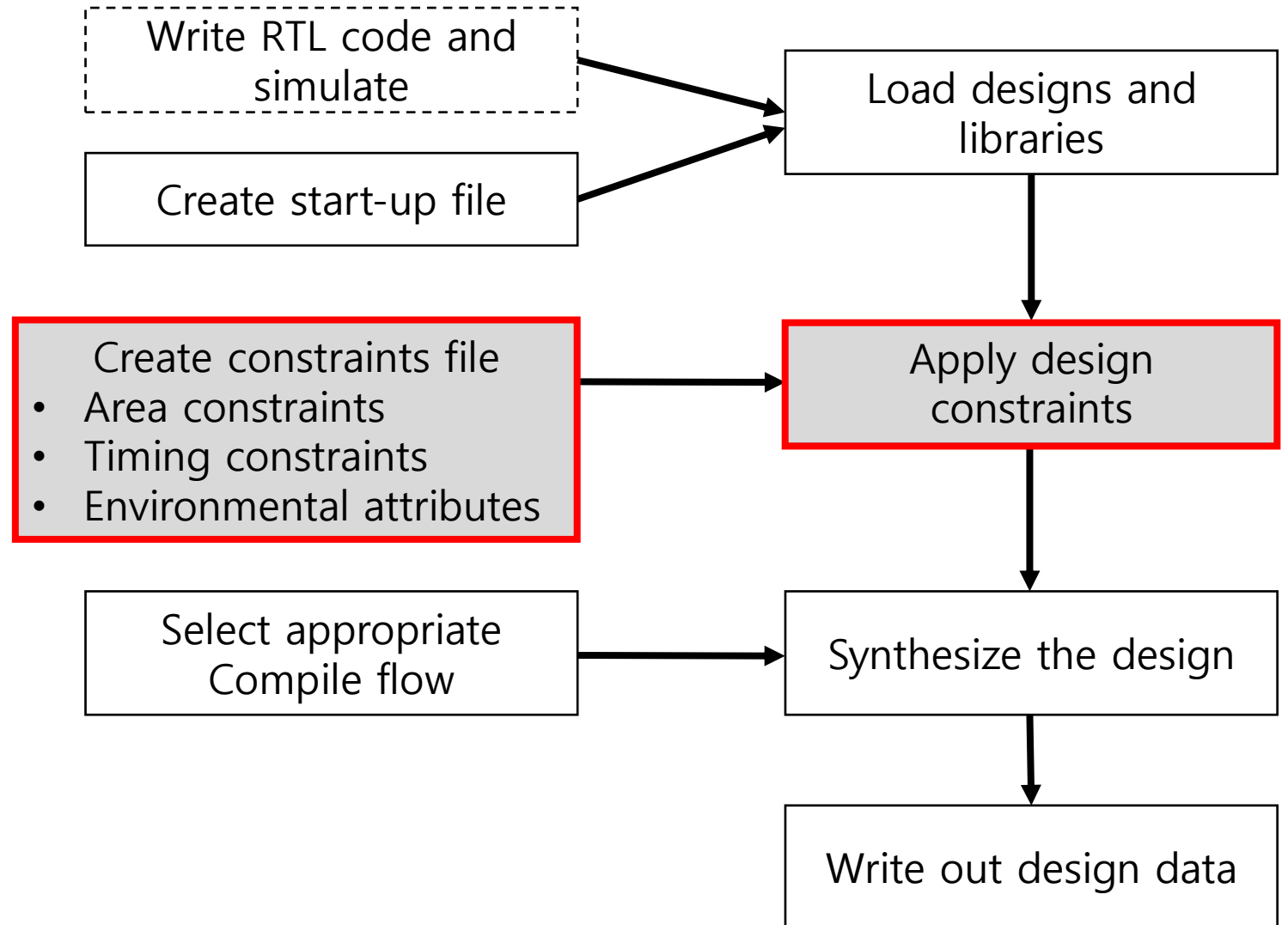
Constraint



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
 - mils², mm² or μm^2
- **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

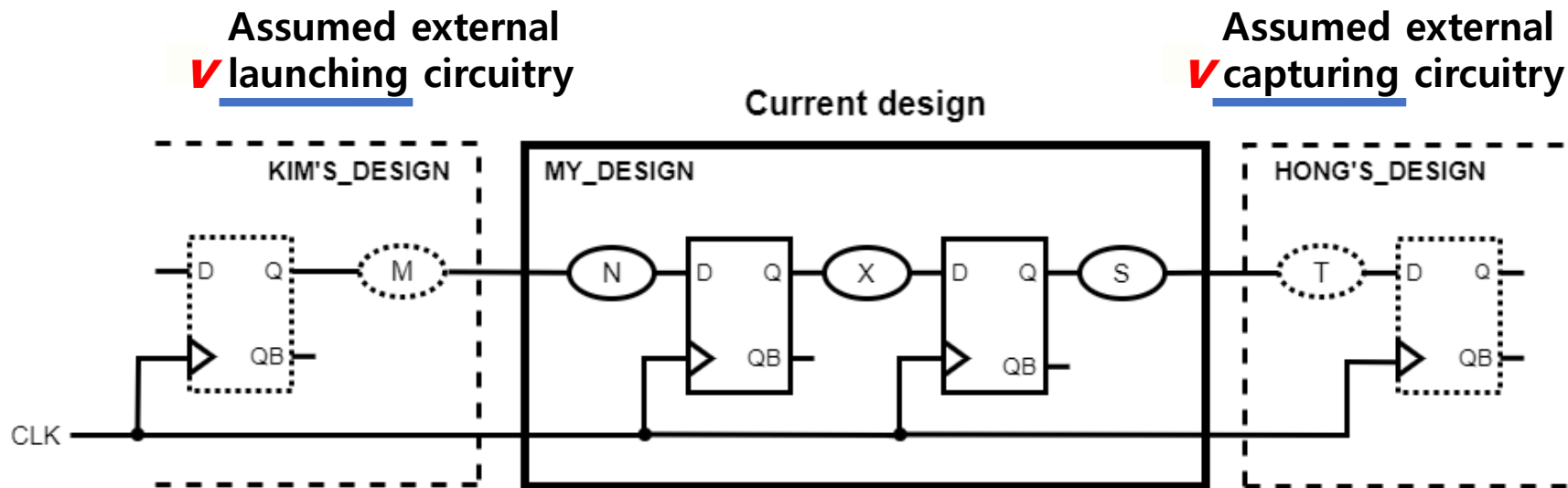
셋업 타임 조건 설정

- **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

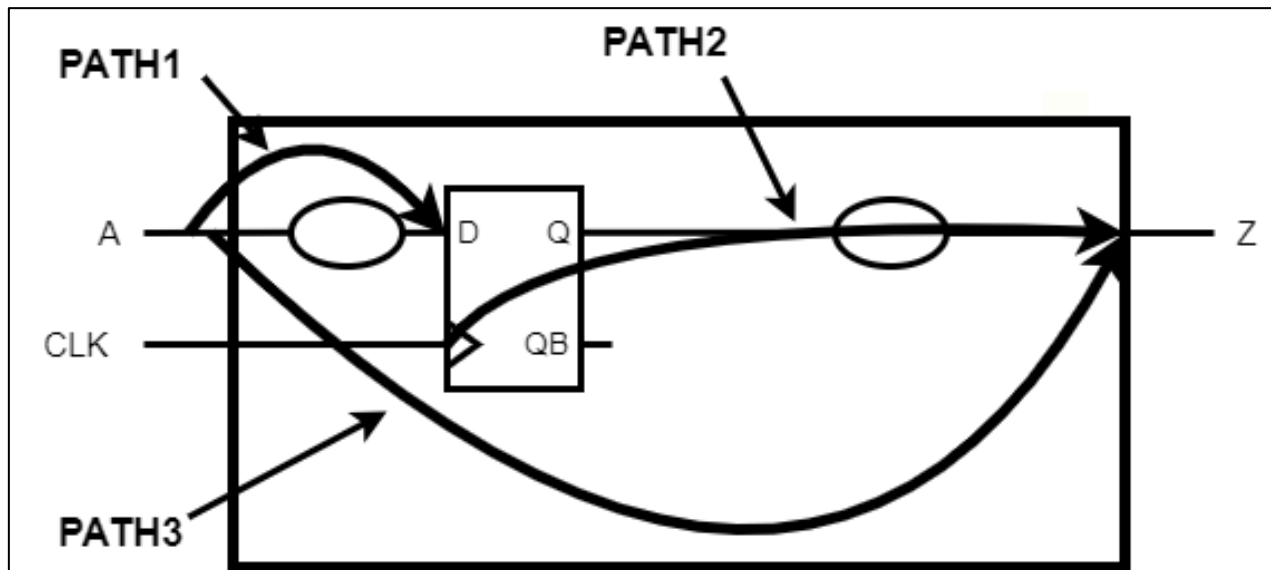
기본적인 디자인(예시)



Constraints

Timing Constraints

합성 진행 시 타이밍 분석

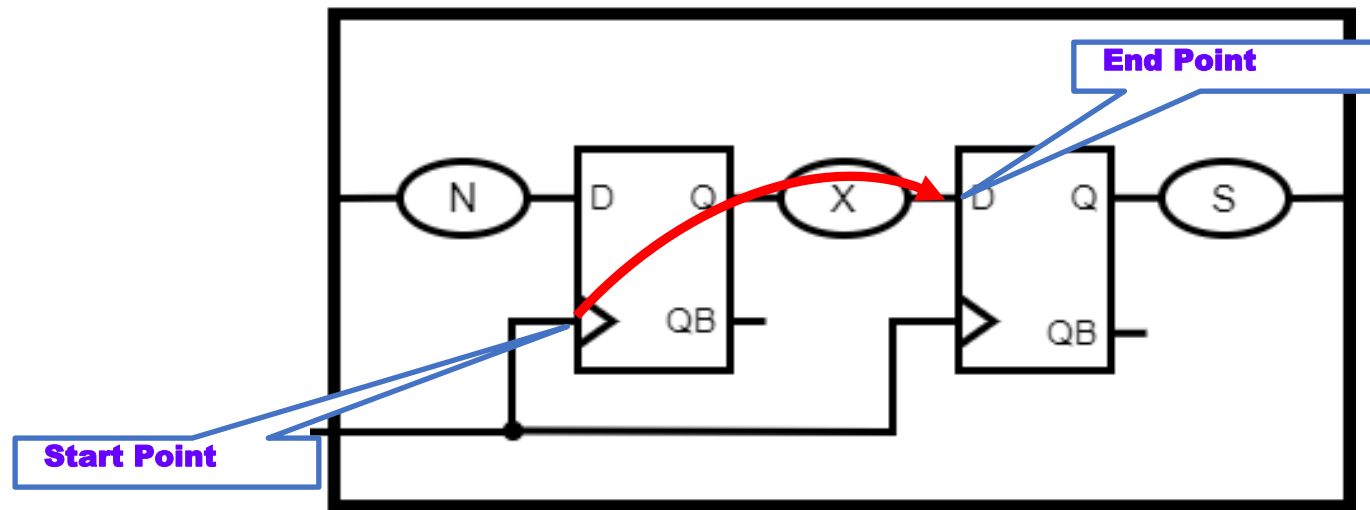


- Internal paths (Reg-to-Reg): 디자인 내부 레지스터 간의 신호 경로를 정의한다.
- All input logic paths: 외부 입력에서 내부 회로로 이어지는 신호 경로를 설정한다.
- All output paths: 내부 회로에서 외부 출력으로 이어지는 신호 경로를 설정한다.

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.

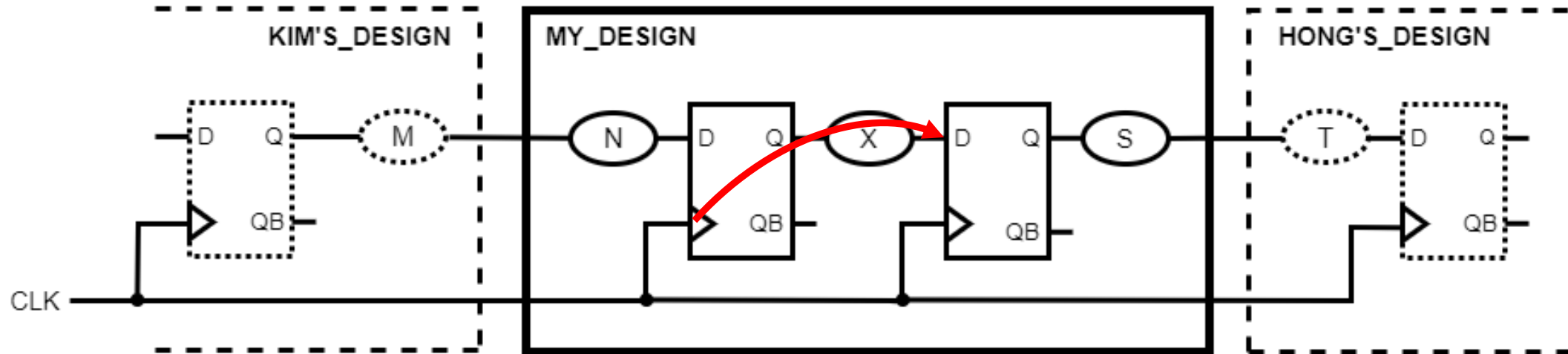
- All input logic paths (starting at input ports)
- The internal (register to register) paths
- All output paths (ending at output ports)

1. 입력 단
2. 디자인 내부
3. 출력 단

CLK 및 아래 3가지 중요

- **Uncertainty(skew)**
- **Transition**
- **Latency**

Timing Constraints



Constraints

Timing Constraints

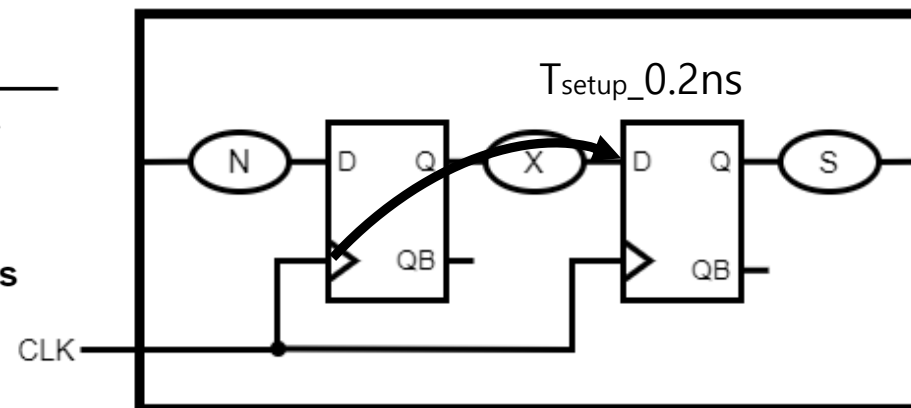
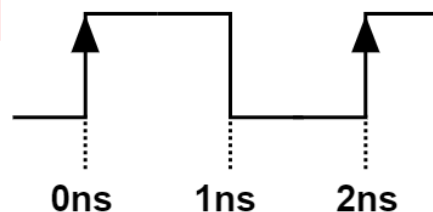
Constraining Reg-to-Reg Path Example

Spec:

Clock Period = 2ns

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

**2ns
→ 500Mhz**



- 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

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

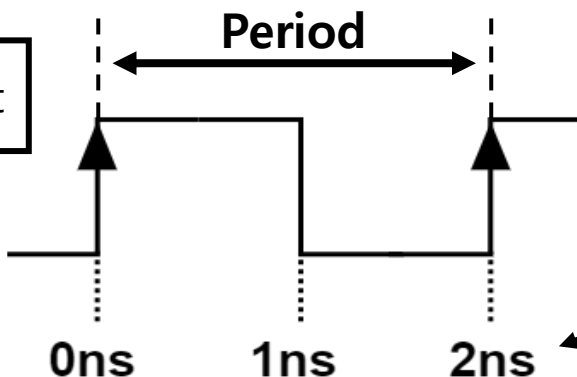
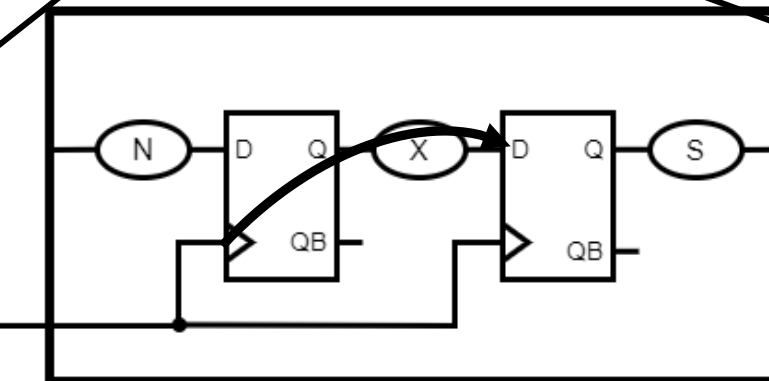
TCL: Command option

TCL: Embedded command

Port object **Clk**

Same name by default

Clock object **Clk**



Unit of time is 1ns in this example.
Defined in the technology library

Rise at 0ns with 50% duty cycle, by default

Constraints

Timing Constraints

기본적인 클럭 동작

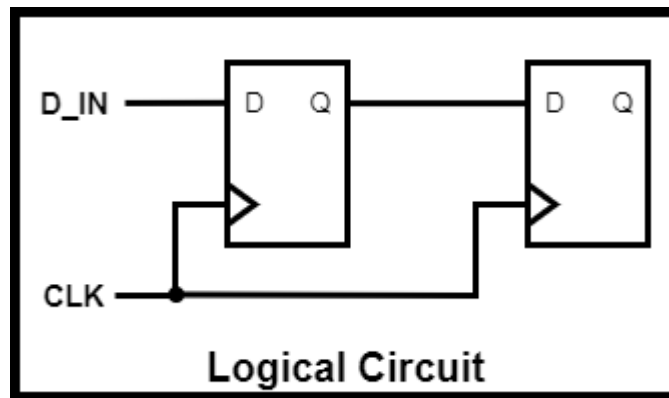
- 단일 클럭 디자인에서 클럭을 정의하면 레지스터 간 모든 타이밍 path(Reg to Reg)들이 단일 주기, 셋업 타임에 의해 결정된다.
기본적으로 클럭은 0ns에서 상승, Duty cycle은 50%
- By default DC will not "buffer up" the clock network, even when connected to many clock/enable pins of flip-flop/latches
 - 클럭 신호는 "이상적(ideal)"이라고 가정한다. (infinite drive capability)
 - ◆ 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

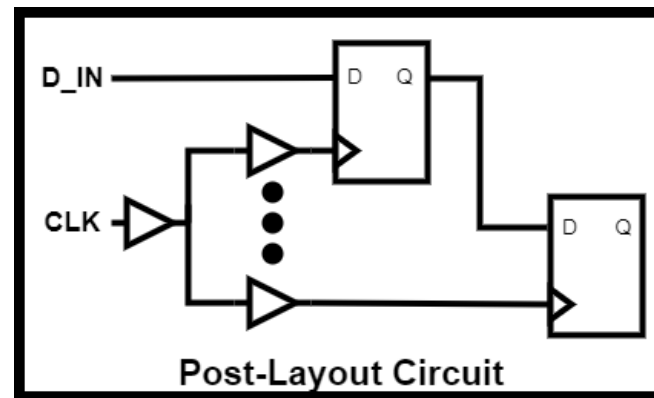
Timing Constraints

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

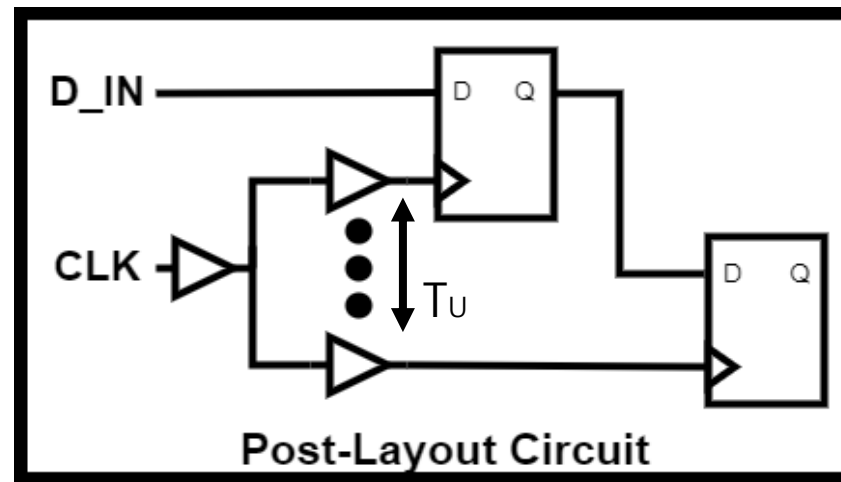
- Back End 단계 Post-CTS STA Constraints에서는 CTS(Clock Tree Synthesis)를 통해 실제 클럭 트리가 생성되고, Skew, Network Latency, Transition의 실제 값이 계산된다.
- 이 단계에서는 Front End에서 설정했던 추정값을 제거하고, 실제 값을 기반으로 타이밍 제약 조건을 재설정한다.

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

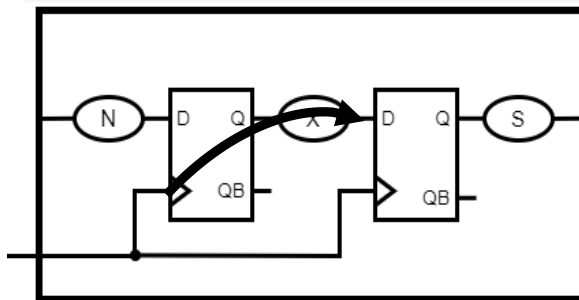
Set_clock_uncertainty and Setup Timing

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

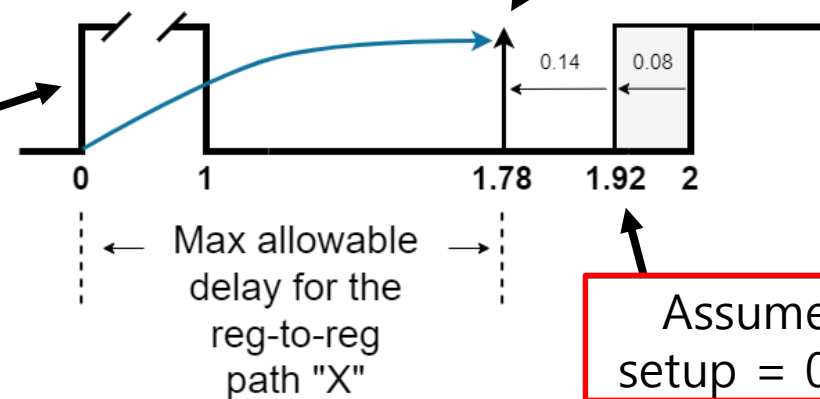
Example:

```
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)

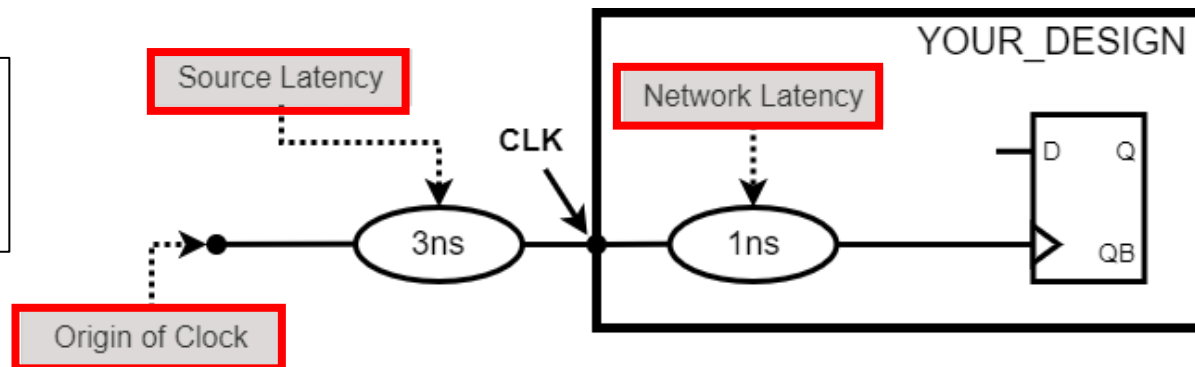
```
Genus> set_clock_latency -source -max 3 [get_clocks CLK]
```

: 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.

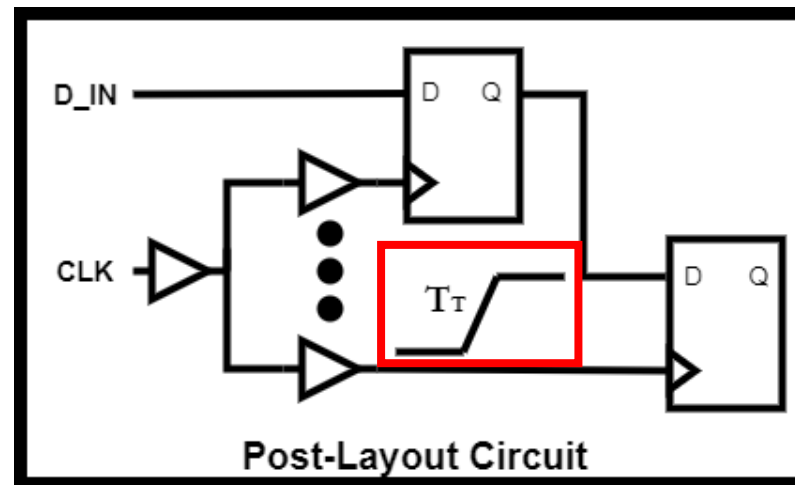


- Latency는 클럭 신호가 전달되는 과정에서 발생하는 시간적 지연(delay)을 의미한다.
- 오실레이터(Oscillator)와 같은 클럭 소스에서 시작하여 내부의 레지스터까지 신호가 도달하는 동안 발생하는 모든 지연 시간을 포함한다.

Constraints

Timing Constraints

Transition 시간 모델링



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

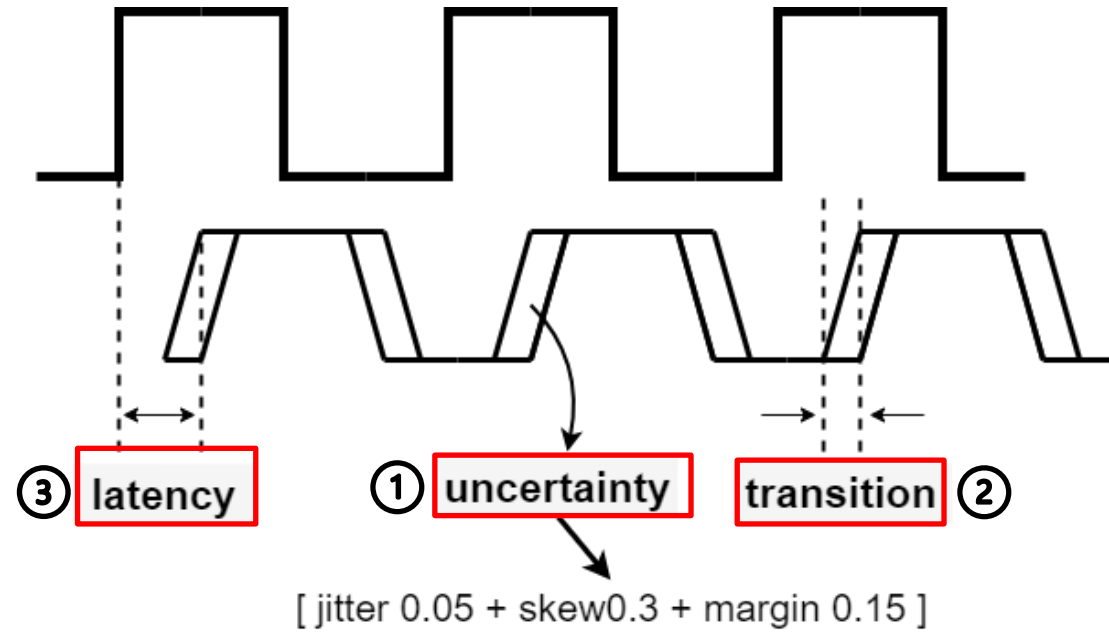
: 클럭에 0.08ns의 transition 설정

- 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

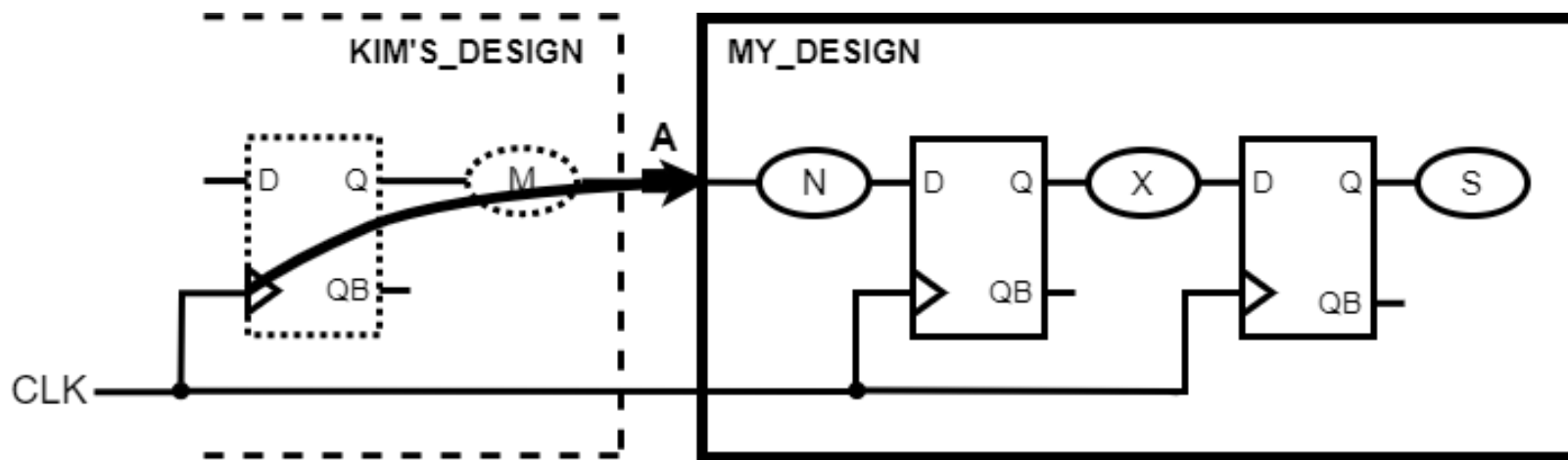


- timing constraint의 3가지 변수를 종합하여 선언하는 예시이다.
- 이러한 선언은 Front End와 Back End 단계에서 사용하는 값과 방식에 차이가 있으며, 각각의 단계에서 적절한 설정이 필요하다.

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

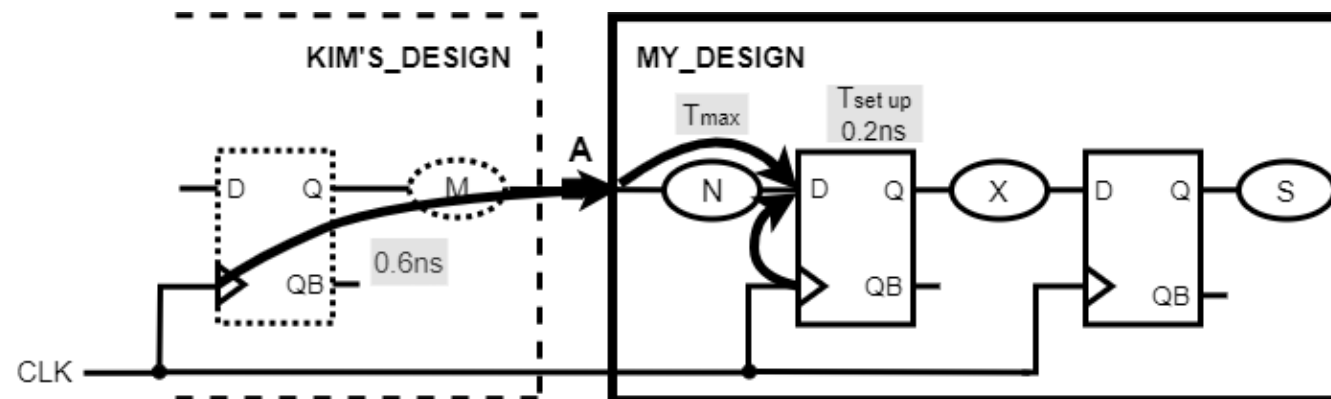
Constraining Input Paths: Example1

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

mydesign.con

```
set_clock_uncertainty -setup 0.3 [get_clocks Clk]
```

```
set_input_delay -max 0.6 -clock Clk [get_ports A]
```

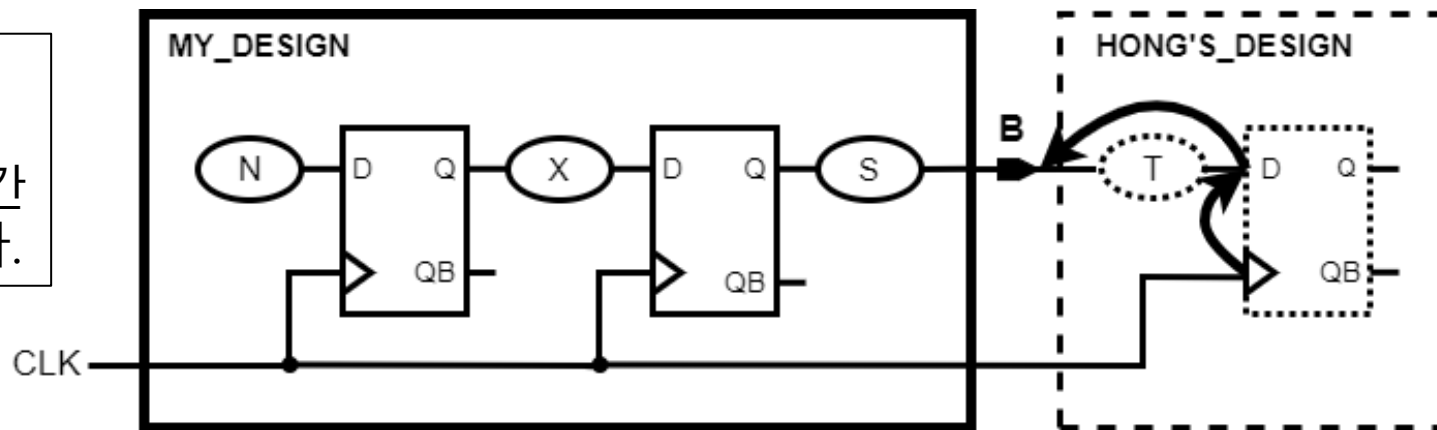


Constraints

Timing Constraints

Constraining output Paths

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

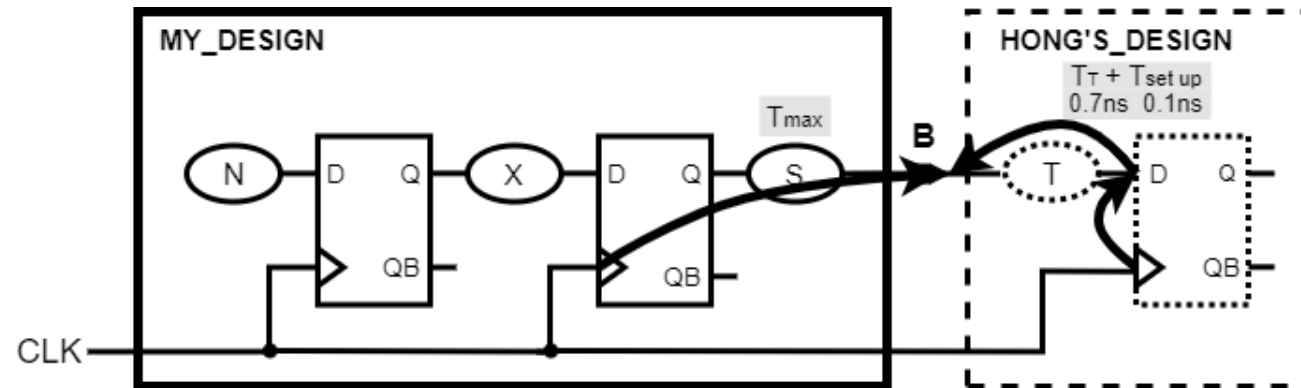


- Output path는 input path와 반대로 내부 신호가 외부 신호로 전달되는 경로이다.
- Input path와 마찬가지로 각 디자인들은 같은 클럭 신호를 공유하고 있기에 현재 디자인의 내부 레지스터의 출력부터 다음 디자인의 레지스터 입력까지 하나의 path를 형성하게 된다.

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
 - 일일이 딜레이를 기술하기 어렵기 때문에 Vclk을 사용하게 됨

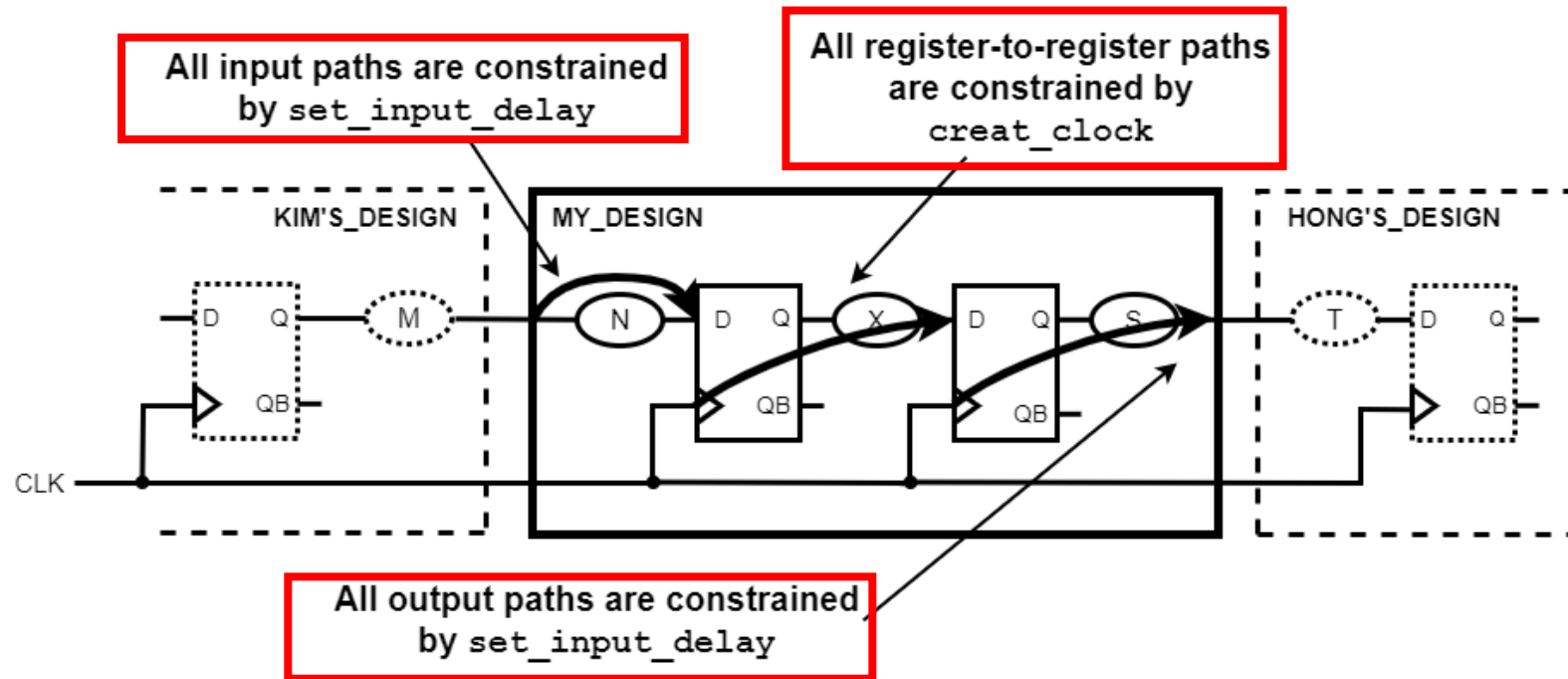
```
# Apply timing constraint
create_clock -period 10 -name CLK [get_port clk]

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

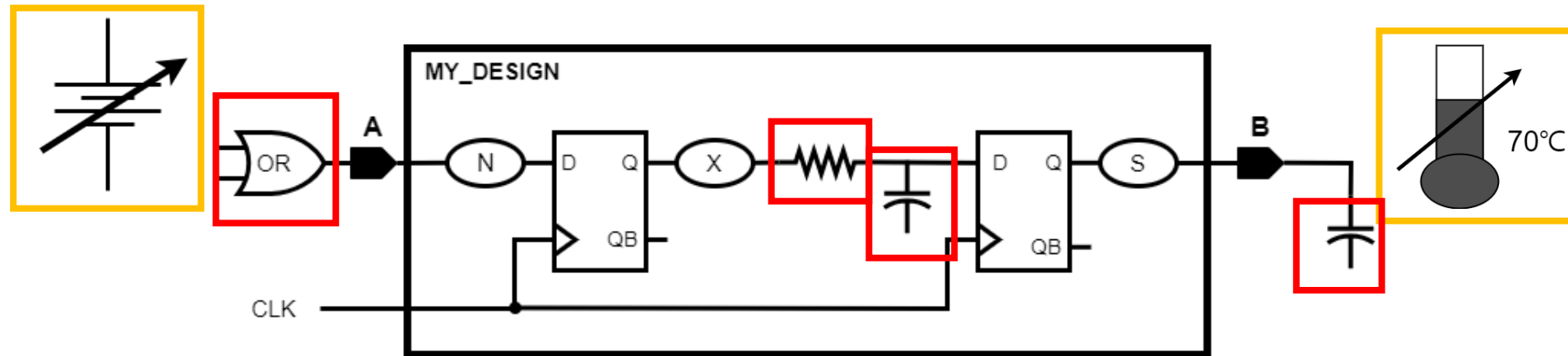
Timing Constraint Summary



Environmental Attributes

Environmental Attributes

Timing에 영향을 주는 요인들



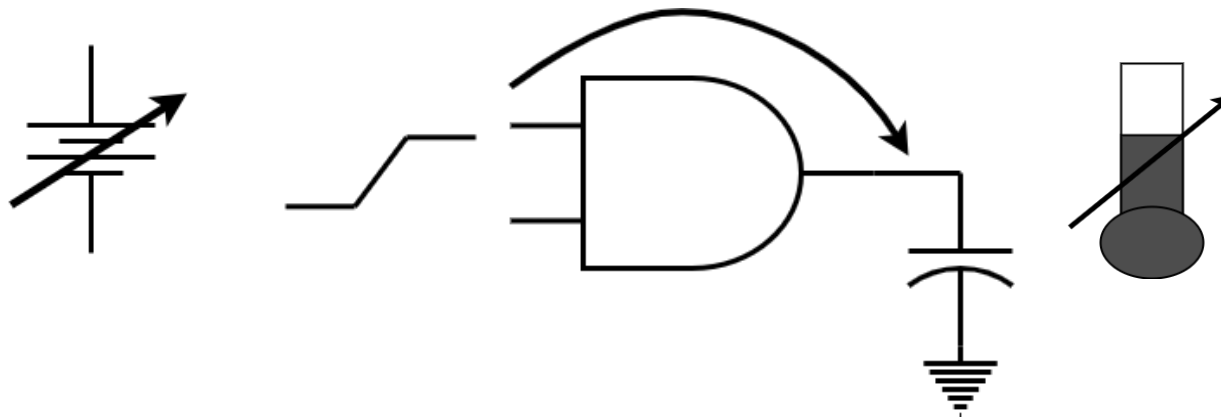
- 환경적 요인(Environmental Attributes)은 설계 및 검증 과정에서 타이밍과 성능에 영향을 미치는 물리적 요인들을 의미한다.
- 이러한 요인들은 설계 및 동작 환경에 항상 존재하기 때문에 신뢰성과 안정성을 위해 반드시 고려되어야 한다.
- 환경적 요인은 Cell Delay와 Net Delay가 있다. Cell Delay에는 PVT corner와 Output Load(출력 부하) 및 Input Transition(입력 기울기)이 있으며 Net Delay에는 Parasitic RC(기생 RC)가 있다.

Environmental Attributes

Cell Delay

Cell Delay에 영향을 미치는 요인

높은 전압에서는 속도가 빨라
지고 전력 소모가 증가하며,
낮은 전압에서는 속도가 느려
지고 전력 소모가 감소한다



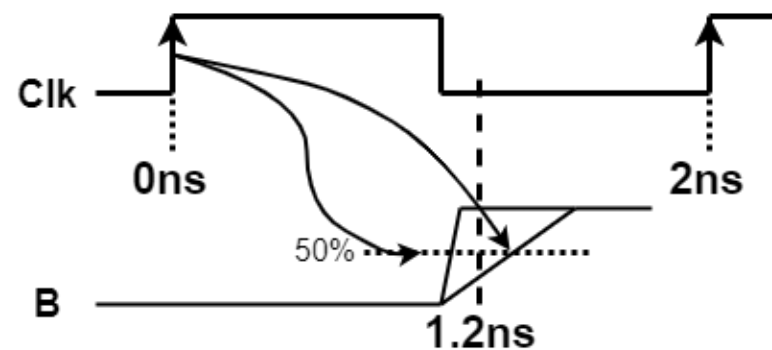
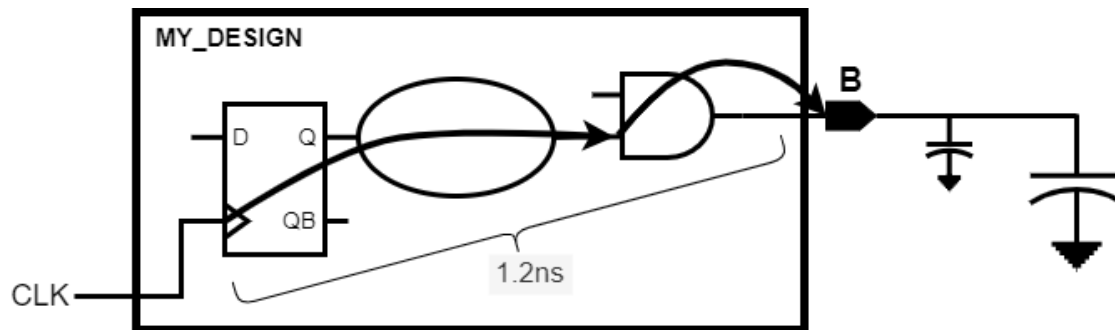
온도는 칩의 동작 성능에 영향을
준다. 일반적으로 반도체는 낮은
온도(-40°C)에서는 빠른 성능을
보이며, 높은 온도(125°C)에서는
느린 성능을 보인다.

- 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

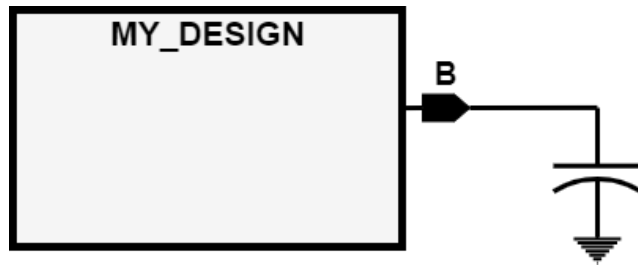


- Output load는 디자인의 출력 포트에 연결된 부하(Load) 용량으로 신호의 전달 속도와 타이밍에 영향을 준다.
- 출력 부하가 클수록 신호가 전달되는 속도가 느려지고 타이밍에 오류가 발생할 가능성이 높아진다.

Environmental Attributes

Cell Delay

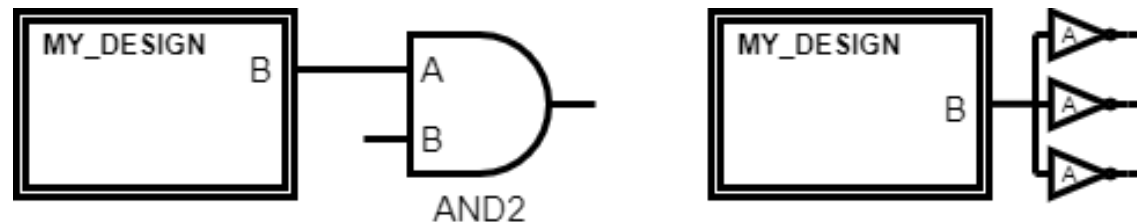
Modeling Output Capacitive Load: Example1



Environmental Attributes

Cell Delay

Modeling Output Capacitive Load: Example 2



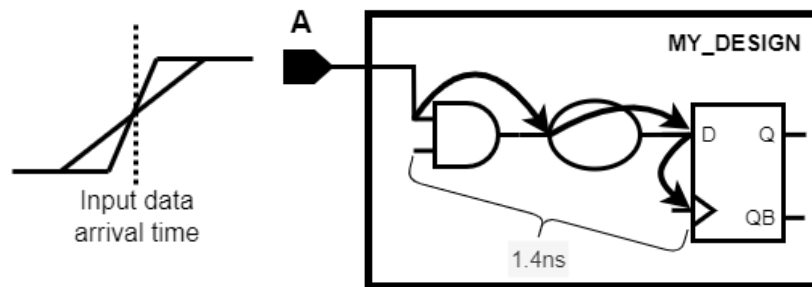
- set_load 명령어를 사용하여 부하를 불러올 라이브러리 이름과 셀의 이름을 선언하여 포트와 연결한다.
- 최대한 pessimistic한 조건을 부여해야 하므로 큰 셀을 사용하는 것이 일반적이다.

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

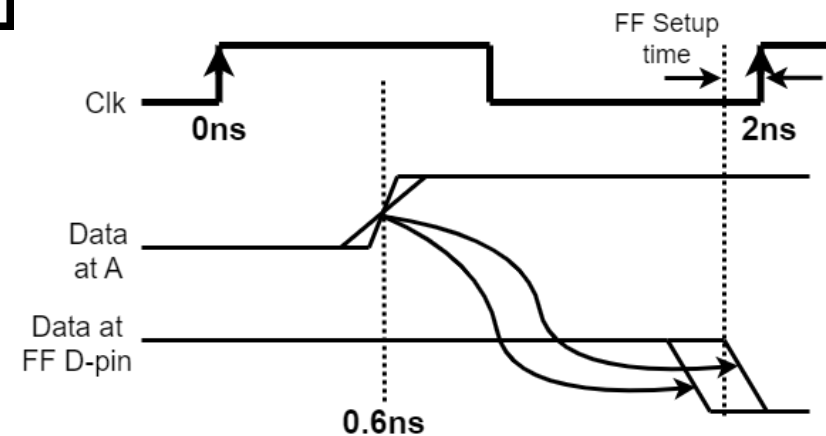
Environmental Attributes

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.

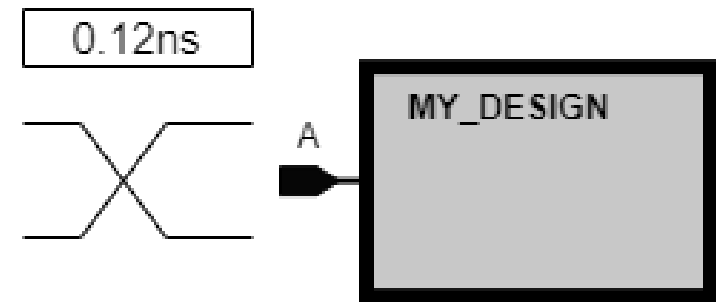


Environmental Attributes

Cell Delay

Modeling Input Transition: Example 1

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

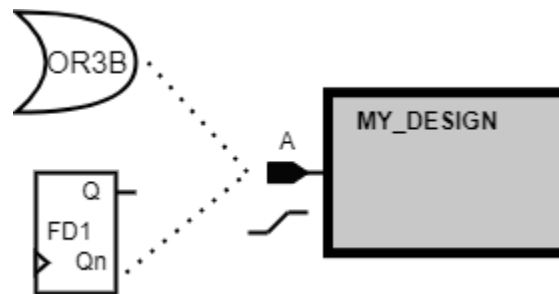


- Transition으로 발생하는 지연시간은 Output load와 마찬가지로 입력 포트에 걸리는 transition 수치를 직접 설정하는 방법과 라이브러리 내의 특정 셀이 신호를 보내주고 있다고 가정하는 방법이 있다.
- 일반적으로 입력포트에 특정 셀이 있다고 가정하는 경우는 다음과 같은 명령어를 사용한다. 이런 방법을 Driving Cell이라고 한다.

Environmental Attributes

Cell Delay

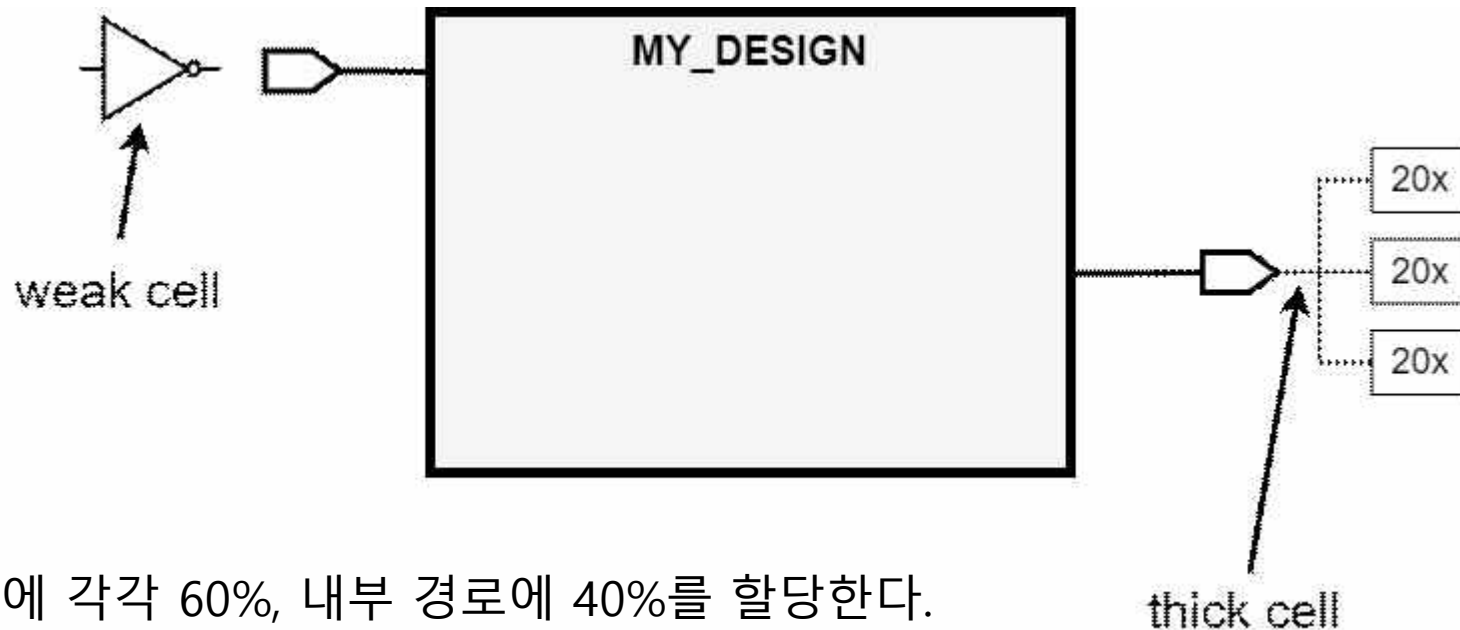
Modeling Input Transition: Example 2



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

Environmental Attributes

Load Budgeting



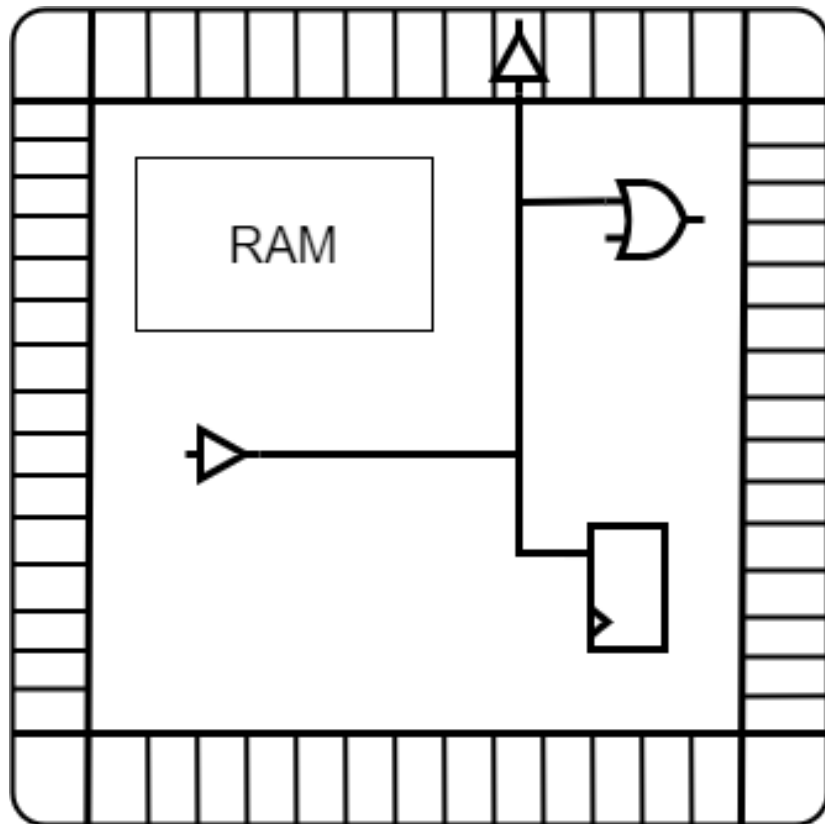
- 보통 입력과 출력에 각각 60%, 내부 경로에 40%를 할당한다.
- Load Budgeting 적용 시 입력 경로에서는 약한 셀(Weak Cell)이 신호를 드라이빙한다고 가정하고 출력 경로에서는 큰 부하가 걸린다고 가정한다.

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

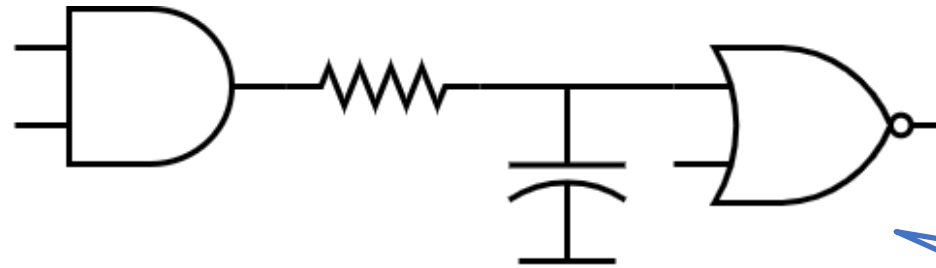


- Net Delay는 배선에 존재하는 저항 (Resistance)과 커패시턴스(Capacitance)에 의해 발생하며, 이를 Parasitic RC라고 한다.

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 지연 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 확인가능

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 D

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)

```
dc_shell> report_lib 90nm_com
```

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

- 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 parasitic이 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)

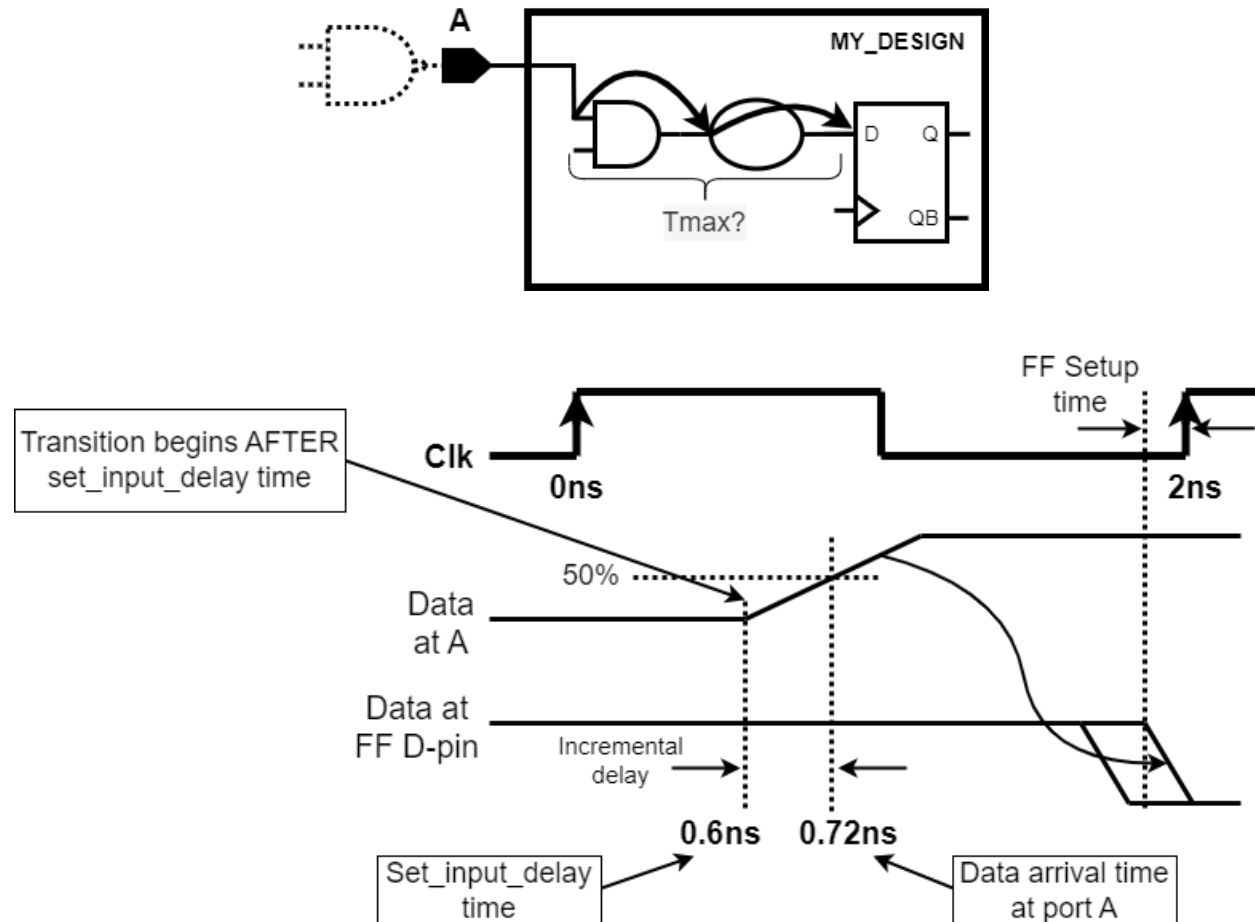
- Topographical mode eliminates the use of WLMs
- Uses placement algorithms under-the-hood to estimate wire lengths and parasitics
- Provides much better timing correlation to that of the actual physical layout

Topographical Mode는
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 pessimis

1. **Zero output load**를 가정하고 지정
2. **Set_driving_cell** 로 실재를 지정

Spec:

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]
```

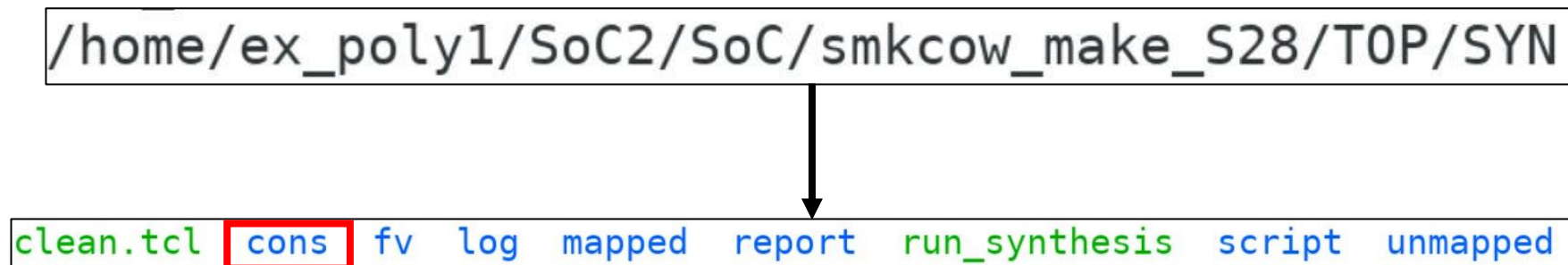
Constraint
cons 폴더 확인

Synthesis

genus

- 합성에 사용할 Constraint가 담긴 cons 폴더 확인

\$> cd cons



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

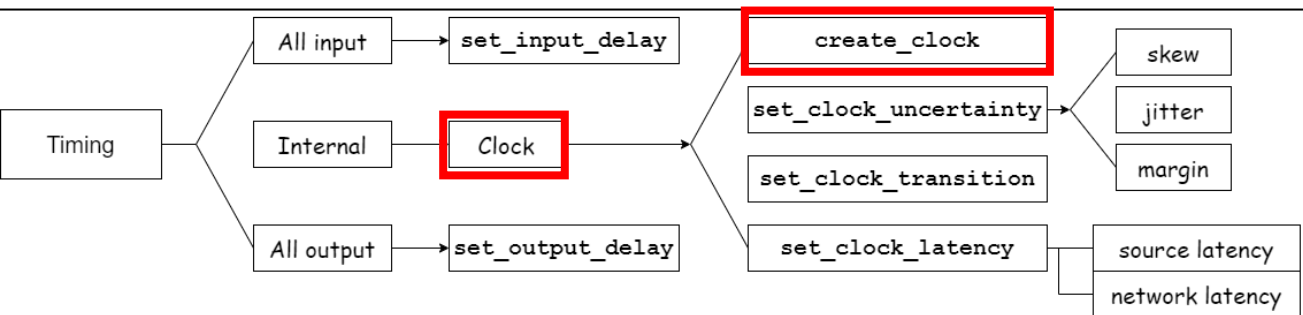
```
1 #####
2
3 # Created by write_sdc on Sat Mar 15 14:01:37 2014
4
5 #####
6 # STD_LIB  slow_vdd1v0_basicCells.lib:slow_vdd1v0
7 #   operating_conditions (PVT_0P9V_125C) {
8 #     process : 1;
9 #     temperature : 125;
10 #     voltage : 0.9;
11 #   }
12
13 set STD_LIB  slow_vdd1v0_basicCells.lib:slow_vdd1v0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



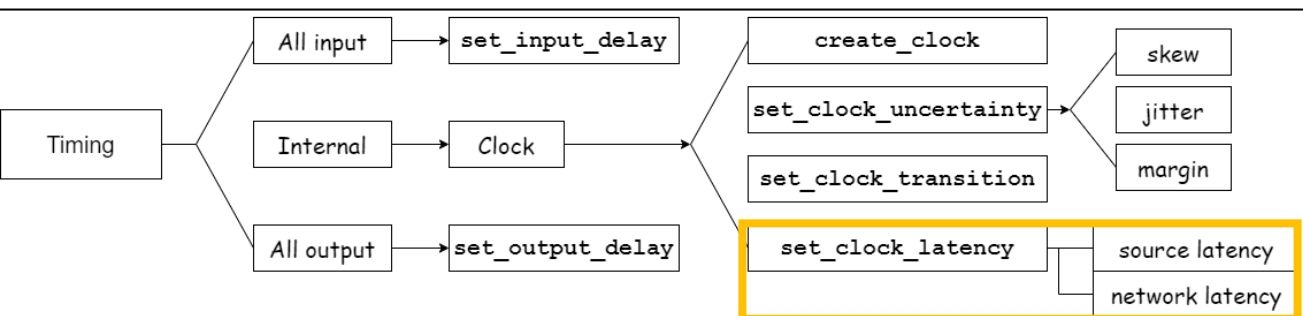
10ns를 사용하지만 목표는
7ns로 더 어렵게 설정함

```
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

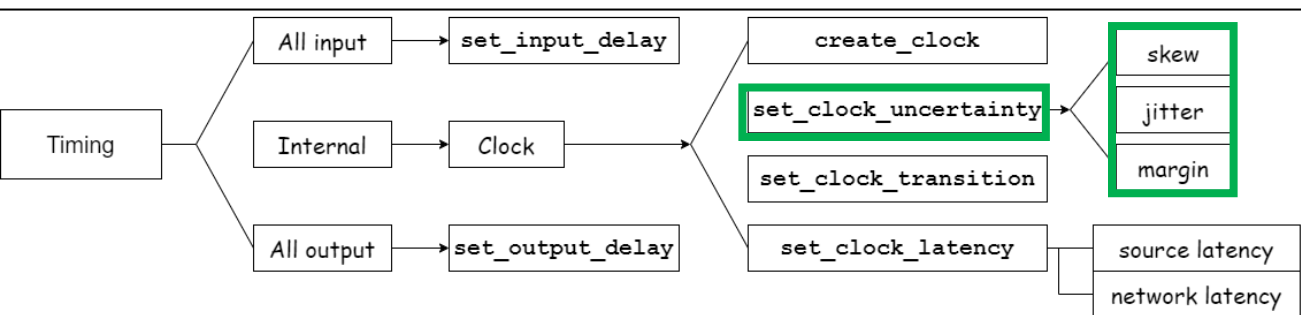


```
18 #####
19 reset_design
20 #####
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]
22
23 # External clock source and network latency is 0.2ns
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]
26
27 # 60ps skew (+300ps and -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

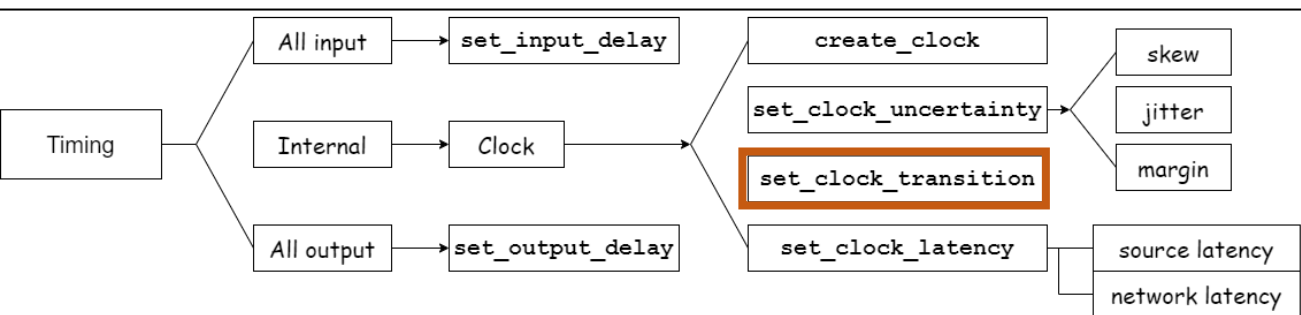


```
18 #####
19 reset_design
20 #####
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]
22
23 # External clock source and network latency is 0.2ns
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]
26
27 # 60ps skew (+300ps and -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

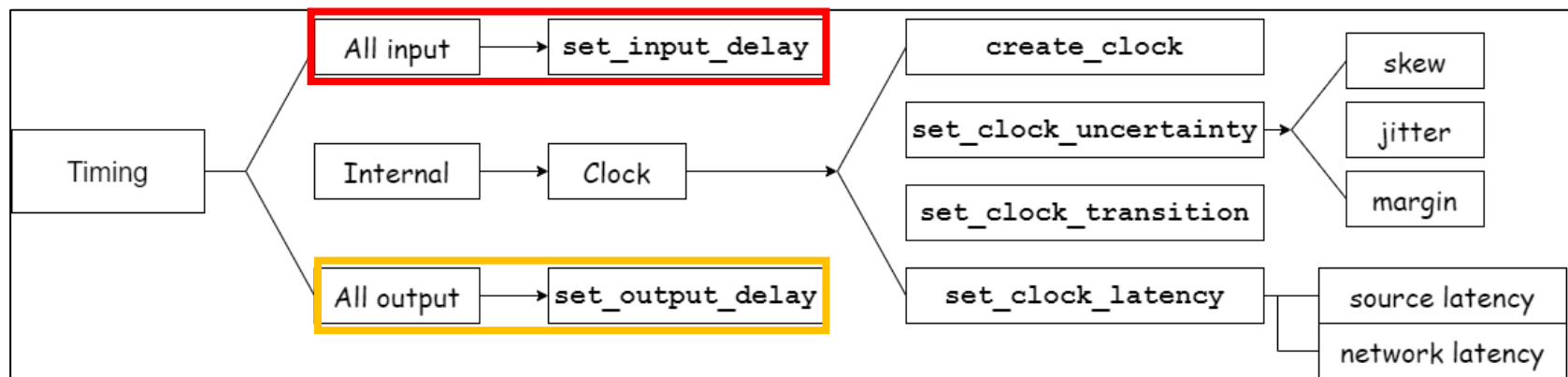


```
18 #####
19 reset_design
20 #####
21 create_clock -period 7 -name MAIN_CLOCK [get_ports XTAL1]
22
23 # External clock source and network latency is 0.2ns
24 set_clock_latency -source -max 0.2 [get_clocks MAIN_CLOCK]
25 set_clock_latency -max 0.2 [get_clocks MAIN_CLOCK]
26
27 # 60ps skew (+300ps and -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



Clock은 제외시킴



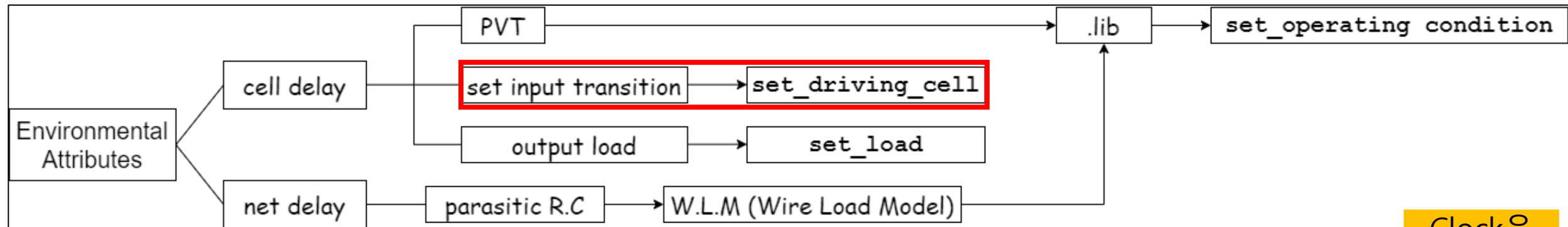
```
47 #####
48 ##### input, output delay #####
49 #####
50 set_input_delay -max 0.1 -clock MAIN_CLOCK [all_inputs]
51 set_input_delay -min 0.01 -clock MAIN_CLOCK [all_inputs]
52
53 remove_input_delay [get_ports "XTAL1"]
54
55 set_output_delay -max 0.1 -clock MAIN_CLOCK [all_outputs]
56 set_output_delay -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

Synthesis

genus

- Environmental Attributes 내용 확인



Clock은
제외시킴



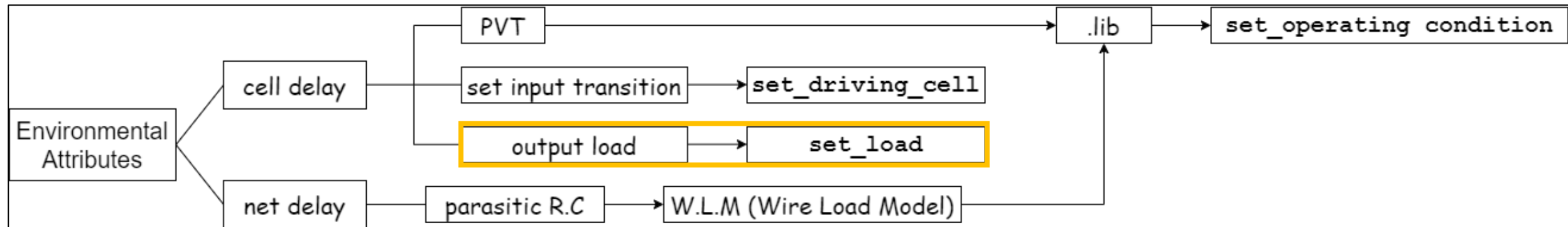
```
64 #####
65 ##### Select this point! #####
66 #####
67 #set_driving_cell -lib_cell DFFHQX1 -pin Q [remove_from_collection [all_inputs] [get_ports clk]]
68
69 set_driving_cell -lib_cell PADDI -pin Y -library [get_libs $STD_LIB] [remove_from_collection [all_inputs] [get_ports XTAL1]]
70
71 #set_load [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
```

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



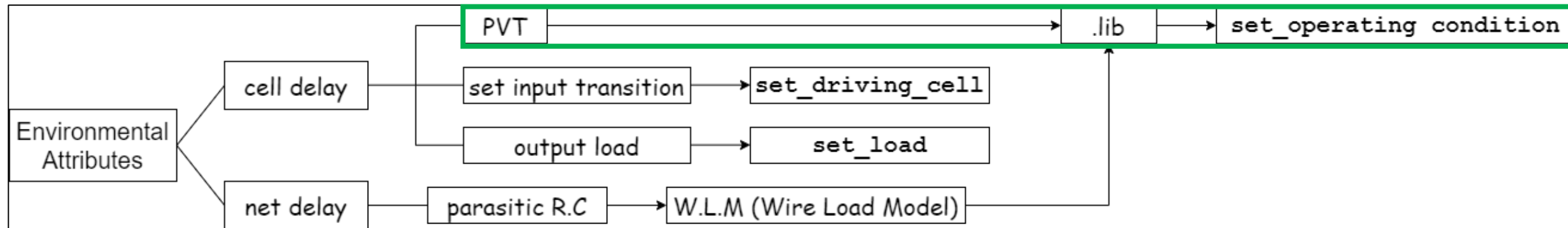
```
64 #####
65 ##### Select this point! #####
66 #####
67 #set_driving_cell -lib_cell DFFHQX1 -pin Q [remove_from_collection [all_inputs] [get_ports clk]]
68
69 set_driving_cell -lib_cell PADDI -pin Y -library [get_libs $STD_LIB] [remove_from_collection [all_inputs] [get_ports XTAL1]]
70
71 #set_load [load of $STD_LIB/PADD0/A] [get_ports {TD0 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_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/A] [get_ports {TD0 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*]
```

genus

- genus

