

# Lab 3. FSM

Digital System Design and Experiment

Graduate School of Convergence Science and Technology

Seoul National University



**Mobile Multimedia Systems Group**

## Goal of Lab 3

- Verilog를 이용하여 simple FSM을 설계하고 구현해본다.
- Vivado에서 구현한 RTL 코드의 synthesis와 implementation를 어떻게 수행하는지 알아본다.

# 1.

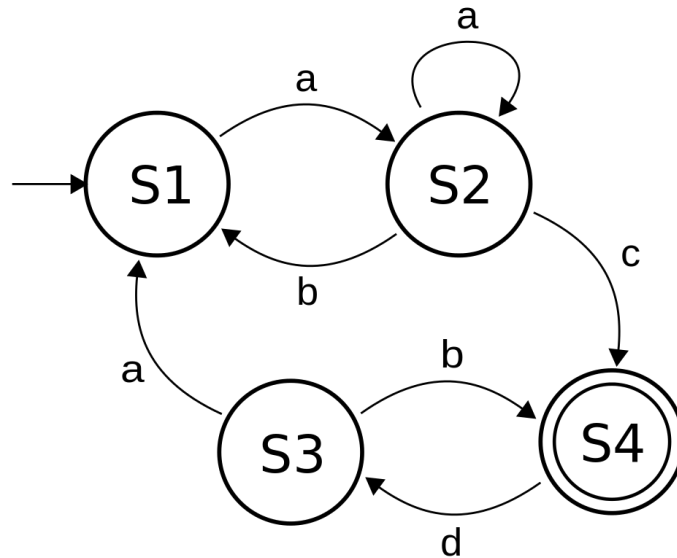
---

Background

---

# Finite State Machine

- 논리회로를 설계하기 위한 **design methodology** 중 하나.
- 논리회로 내부의 정보는 **하나의 state** 로 표현.
- 만약 시스템이 제한된 수의 state 간의 전환으로 표현된다면, 해당 시스템은 FSM을 이용하여 설계가 가능.



# Synthesis and Implementation

- Digital System Design Flow  
: SW Implementation → RTL Code → Gate Level Code → Physical Layout  

The diagram illustrates the Digital System Design Flow. It starts with 'SW Implementation' leading to 'RTL Code'. From 'RTL Code', a red bracket labeled 'Synthesis' spans to 'Gate Level Code'. From 'Gate Level Code', another red bracket labeled 'Implementation' spans to 'Physical Layout'.
- **Synthesis:** Compile RTL code into network of logic gates, and also optimize the design further.  
**Implementation:** Arrange physical layouts of logic gates in the design and connect them together using FPGA resource.
- Synthesis 와 implementation 를 하고 나면, gate 간의 delay 정보를 이용하여 보다 현실적인 simulation 수행 가능.
- **Synthesizable and implementable RTL code** 로 회로를 설계하는 것이 매우 중요!!

# 2.

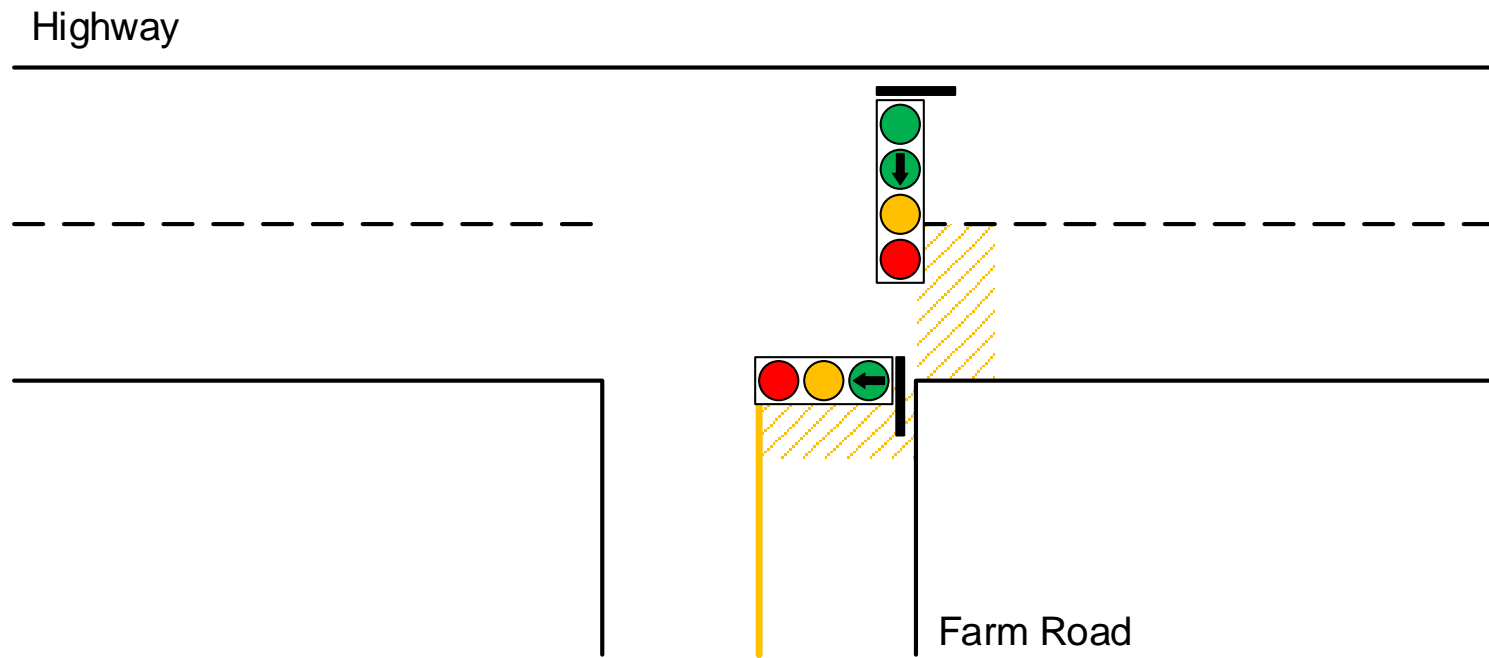
---

Traffic Light Controller Model

---

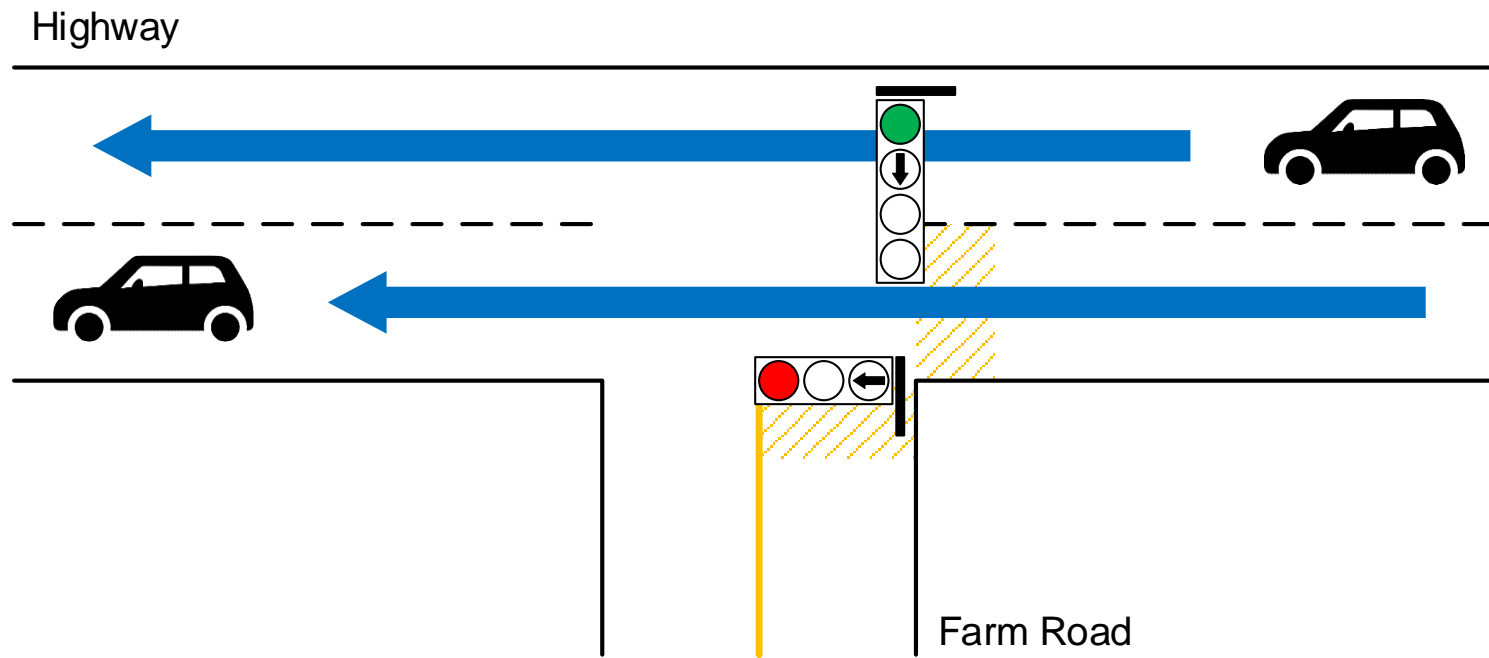
# Traffic Light Controller Model

- Highway 와 farm road, 두 가지 도로가 존재.
- 각각 farm road 와 highway 의 1차선에는 좌회전 차량에 대한 센서가 존재.
- 센서에 의해 차가 감지되면, 신호등 동작.
- 청신호, 좌회전 신호 후엔 반드시 황신호를 거쳐야 한다.



## TLC Model: Basic State

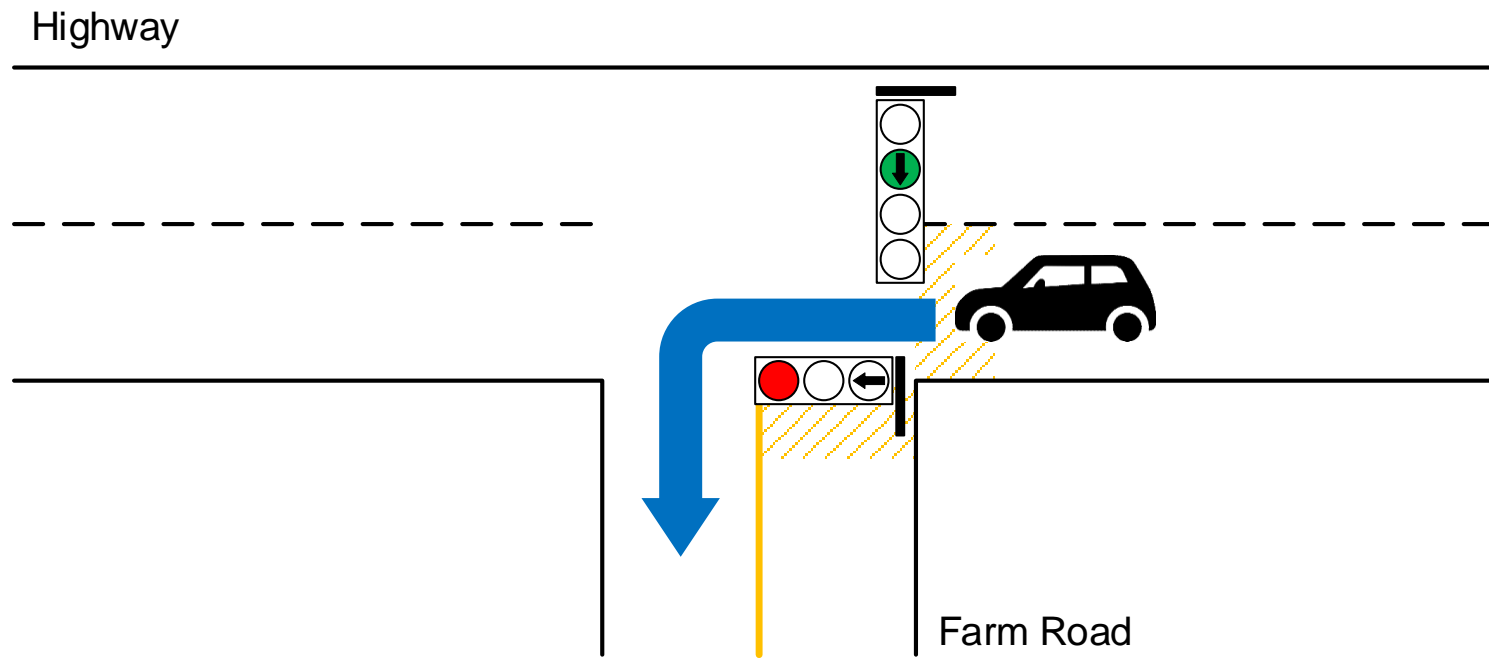
- 차가 감지 되지 않으면, highway 의 신호등은 청신호 유지. → **Basic state**
- 신호등이 일련의 동작을 마치고 basic state 로 돌아오면, basic state를 **1 clock cycle** 반드시 유지.





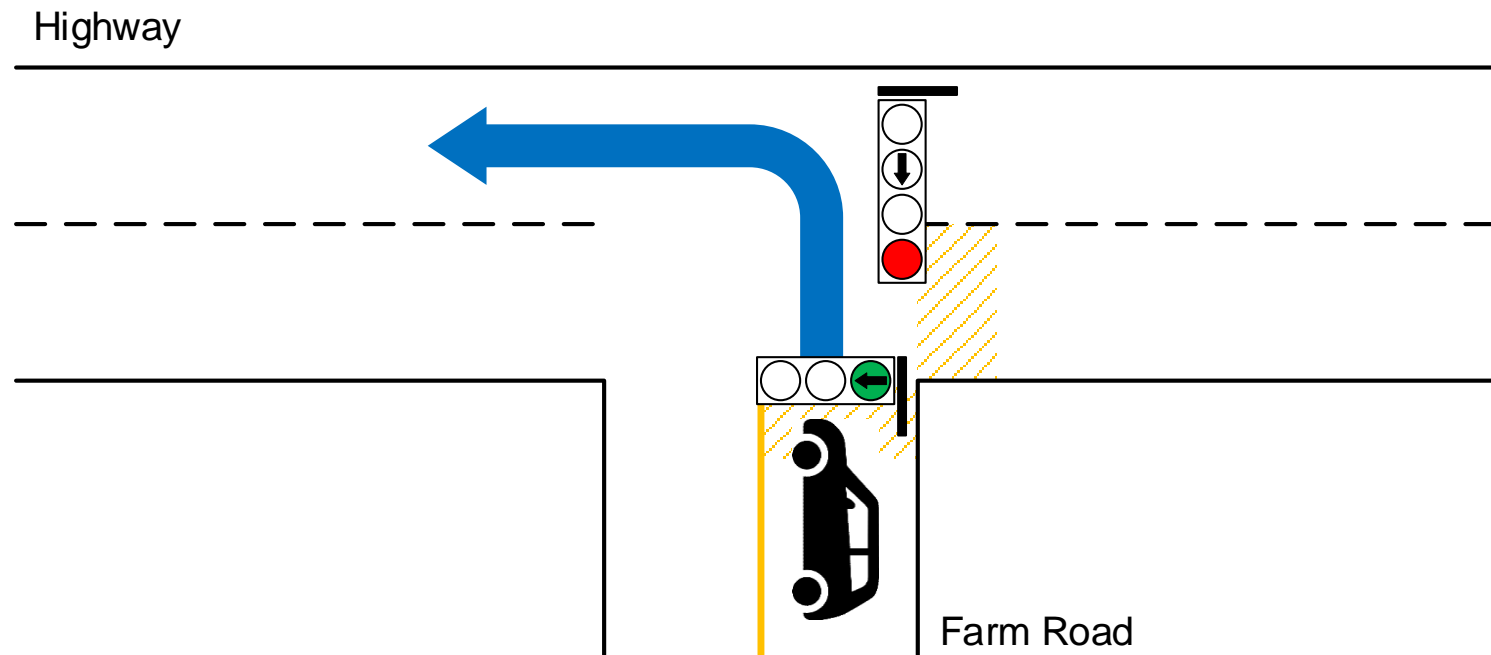
## TLC Model: Turn Left on Highway

- Highway 에서 차가 감지되면, highway 의 신호등은 청 → 황 → 좌회전 신호 순으로 동작.
- 좌회전 신호에서 timer 가 시작되며, 신호는 반드시 **5 clock cycle** 동안 유지.
- 이후, highway 의 신호등은 좌회전 → 황 → 청신호 순으로 동작.



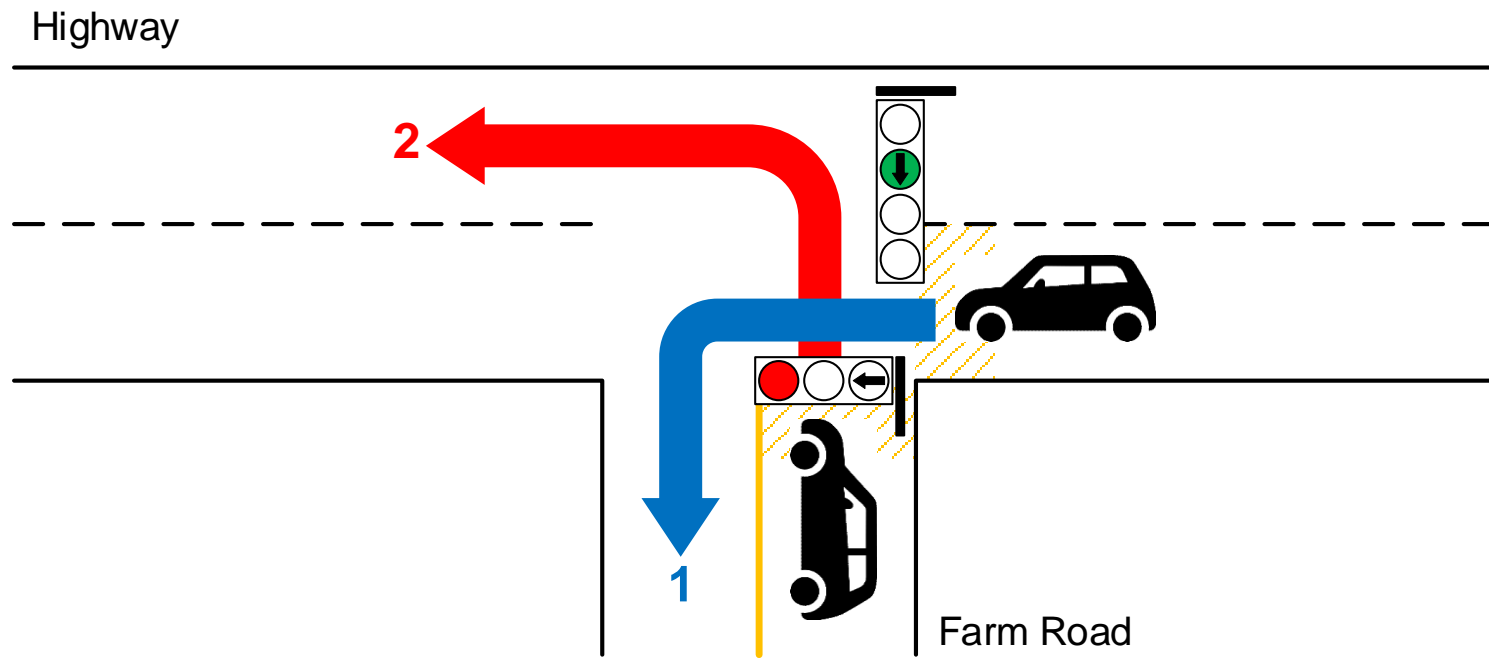
## TLC Model: Turn Left on Farm Road

- Farm road 에서 차가 감지되면, highway 의 신호등은 청 → 황 → 적신호 순으로 동작하고 farm road 의 신호등은 적 → 좌회전 신호 순으로 동작.
- 마찬가지로, 좌회전 신호는 **5 clock cycle** 동안 유지.
- 이후, farm road 의 신호등은 좌회전 → 황 → 적신호 순으로 동작하고 highway 의 신호등은 적 → 청신호 순으로 동작.



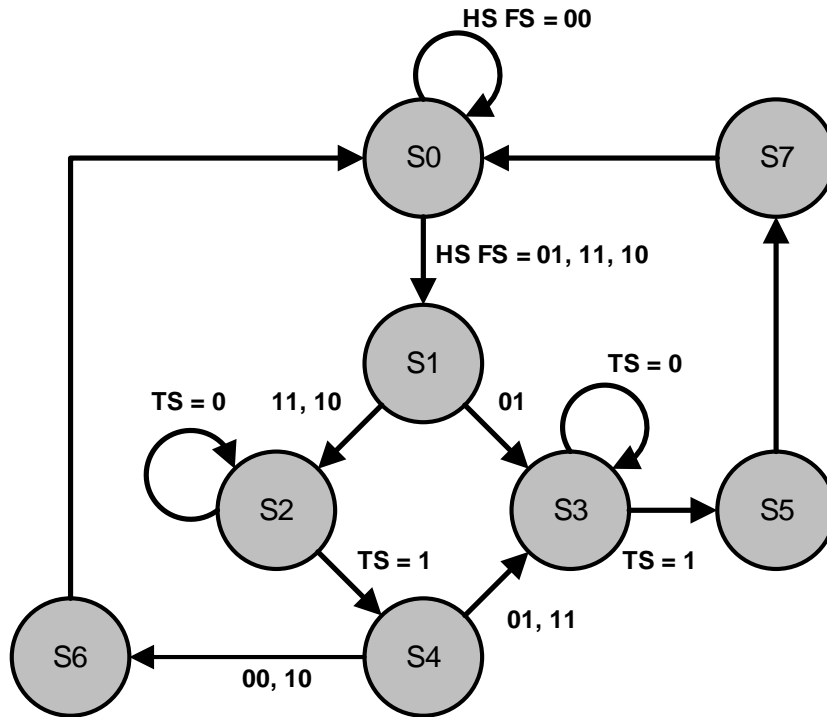
## TLC Model: Turn Left on Both Roads

- 두 도로에서 모두 차가 감지되는 경우, highway 의 신호등부터 먼저 동작.
- Highway 의 좌회전 신호 후에, highway 는 좌회전 → 황 → 적신호 순으로 동작하고 farm road 의 신호는 적 → 좌회전 신호 순으로 동작한다.



# FSM of Traffic Light Controller Model

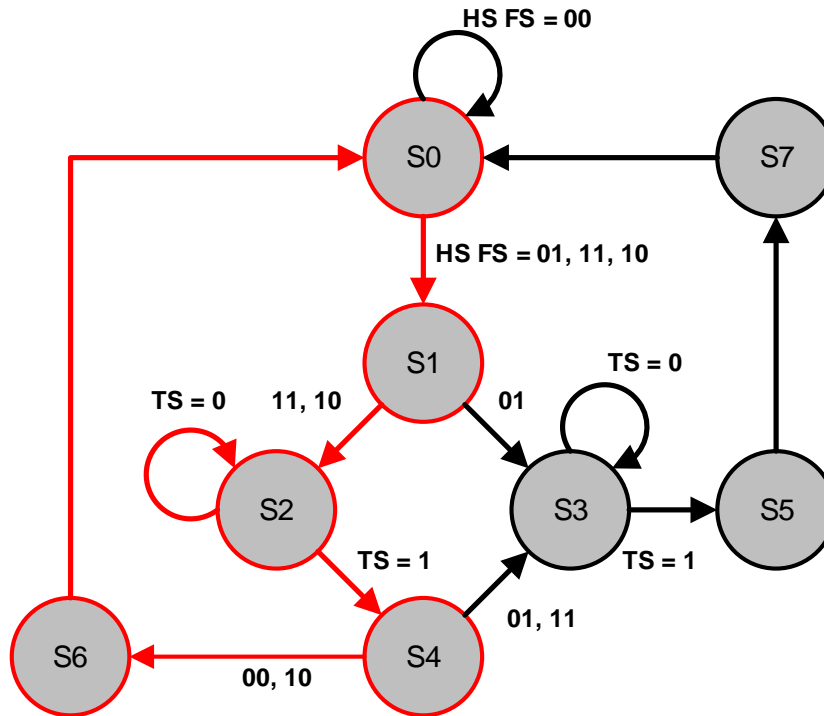
- HS: highway censor detection / FS: farm road censor detection
- TS 는 S2, S3가 5 clock cycle을 유지해야함을 의미.
- S6, S7 : 신호등이 일련의 동작을 마쳤을 때, basic state를 1 clock cycle 유지하는 것을 표현.



S0: H\_GREEN & F\_RED  
S1: H\_YELLOW & F\_RED  
S2: H\_LEFT & F\_RED  
S3: H\_RED & F\_LEFT  
S4: H\_YELLOW & F\_RED  
S5: H\_RED & F\_YELLOW  
S6: H\_GREEN & F\_RED  
S7: H\_GREEN & F\_RED

# FSM Flow: Turn Left on Highway

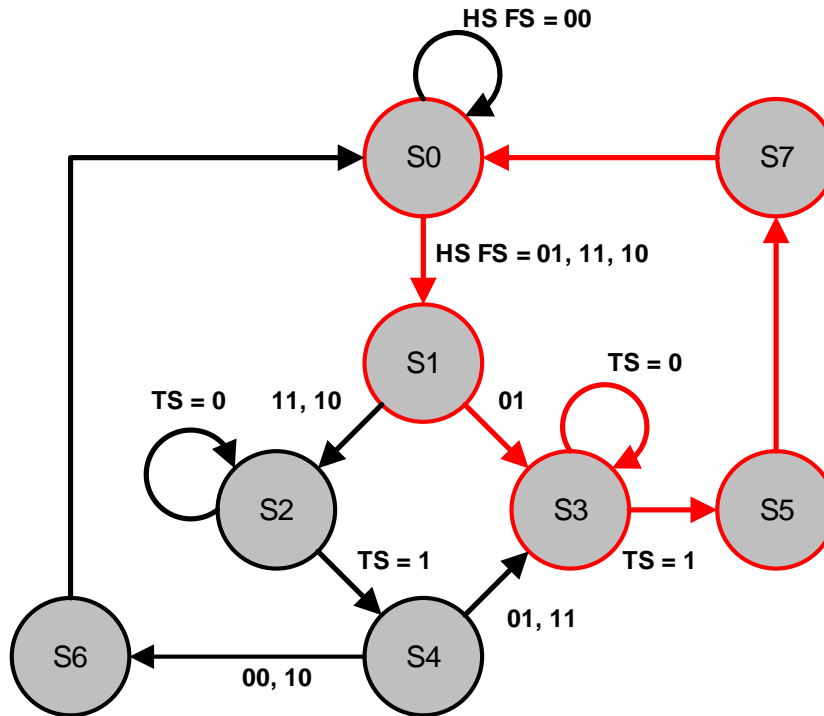
- $(HS, FS) = (1, 0)$



S0: H\_GREEN & F\_RED  
S1: H\_YELLOW & F\_RED  
S2: H\_LEFT & F\_RED  
S4: H\_YELLOW & F\_RED  
S6: H\_GREEN & F\_RED

# FSM Flow: Turn Left on Farm Road

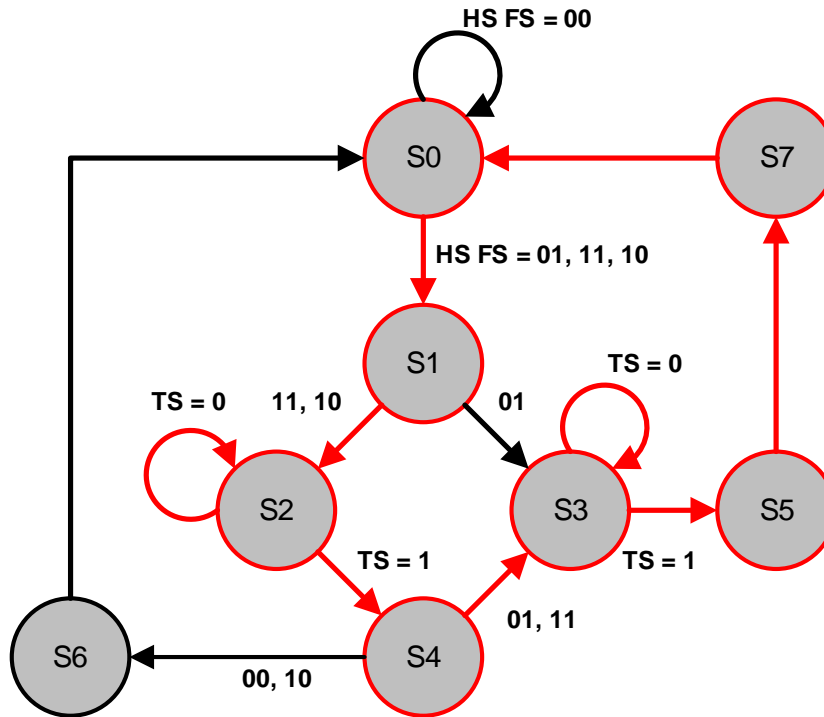
- $(HS, FS) = (0, 1)$



S0: H\_GREEN & F\_RED  
S1: H\_YELLOW & F\_RED  
S3: H\_RED & F\_LEFT  
S5: H\_RED & F\_YELLOW  
S7: H\_GREEN & F\_RED

# FSM Flow: Turn Left on Both Roads

- $(HS, FS) = (1, 1)$



S0: H\_GREEN & F\_RED  
S1: H\_YELLOW & F\_RED  
S2: H\_LEFT & F\_RED  
S4: H\_YELLOW & F\_RED  
S3: H\_RED & F\_LEFT  
S5: H\_RED & F\_YELLOW  
S7: H\_GREEN & F\_RED

# 3.

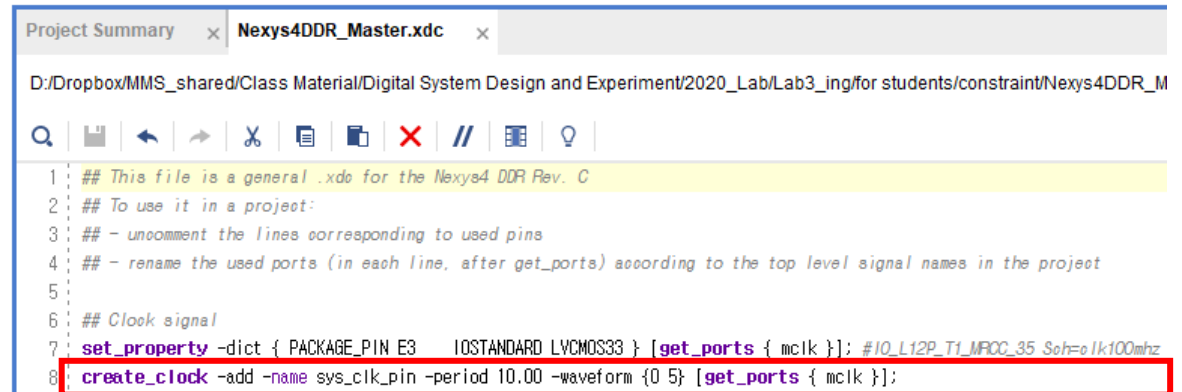
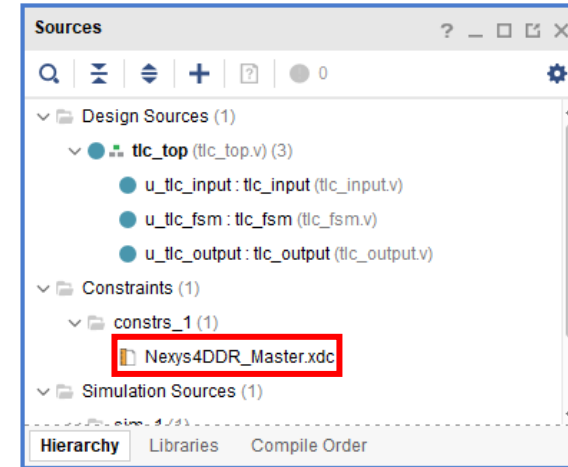
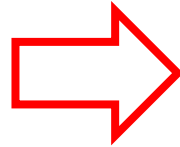
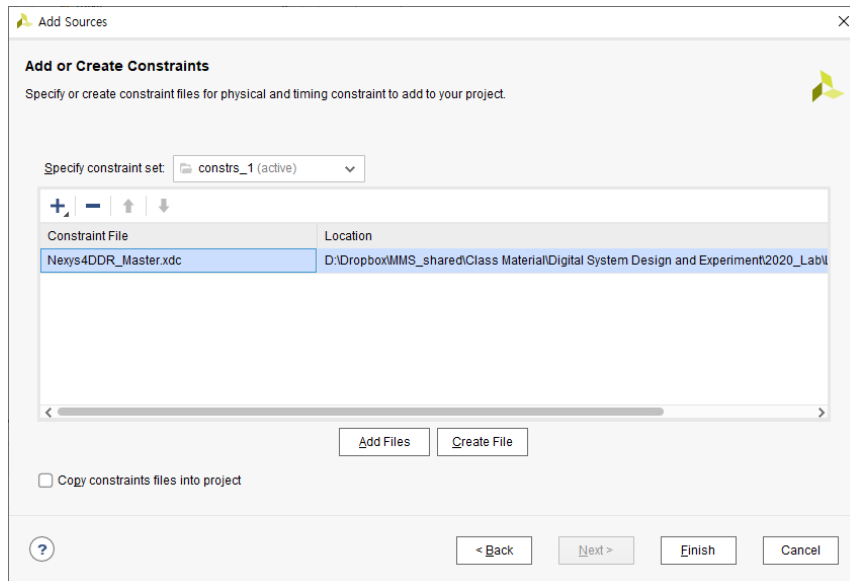
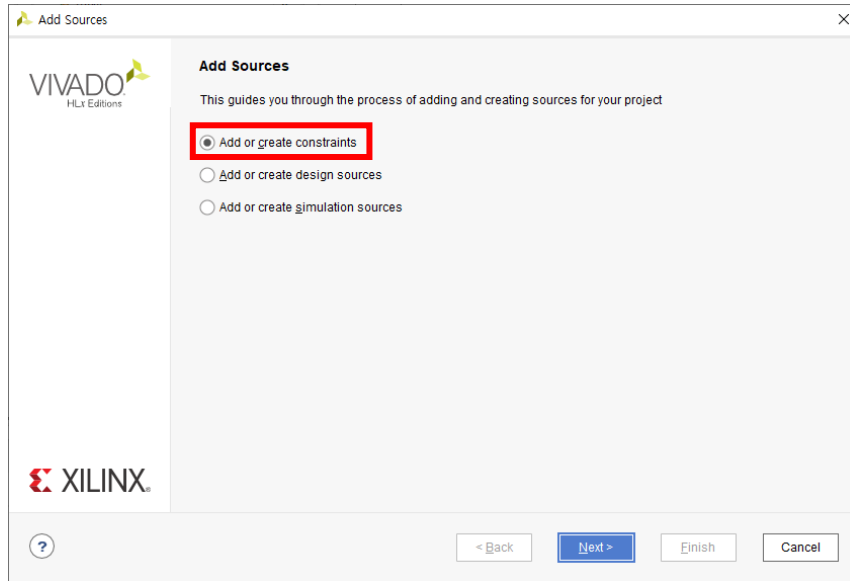
---

How to Run Synthesis and Implementation

---

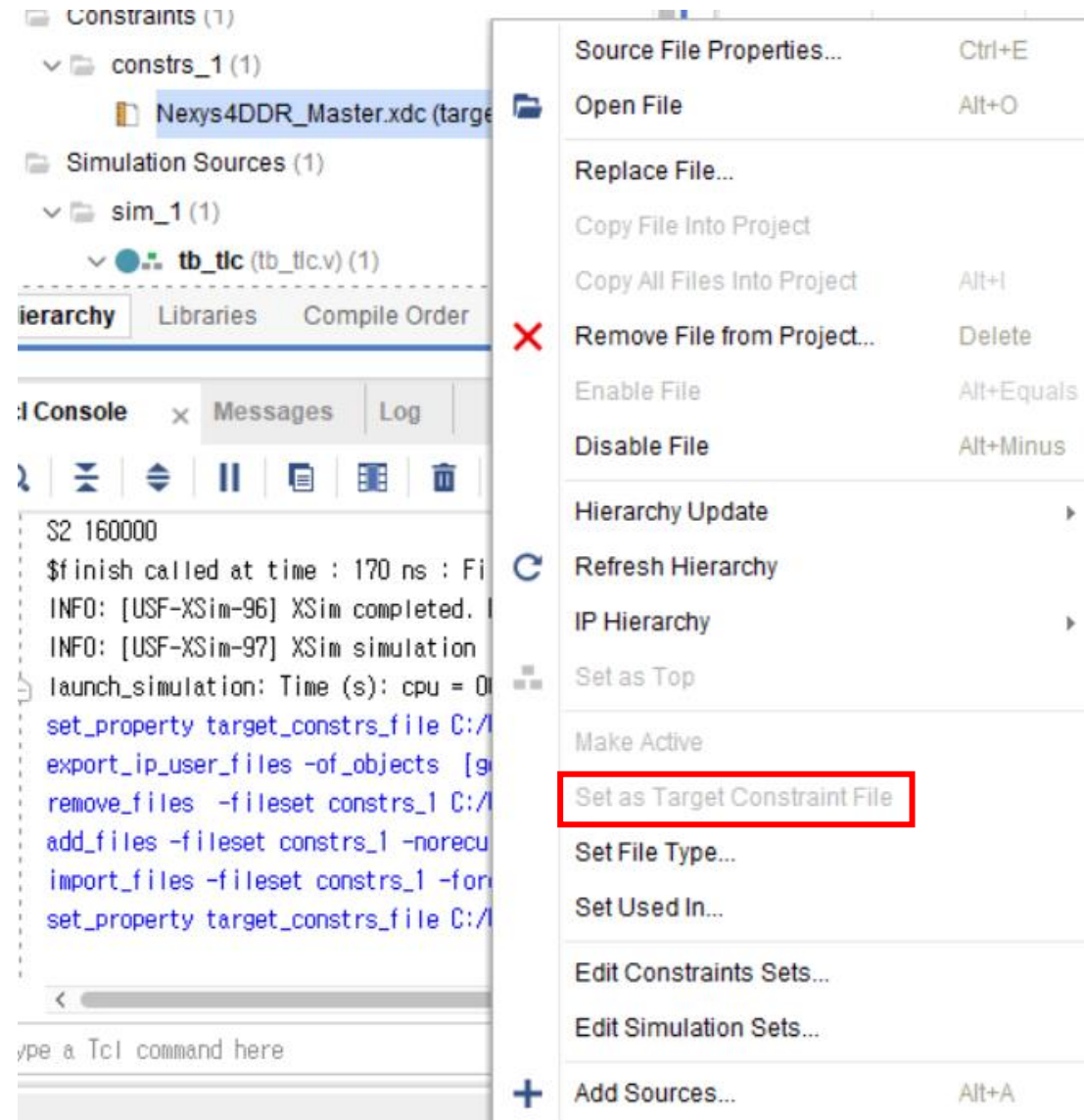


# Add Constraint File To Project

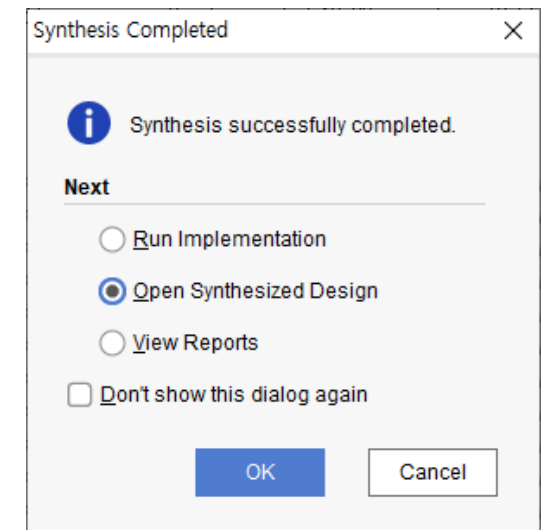
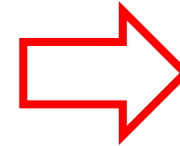
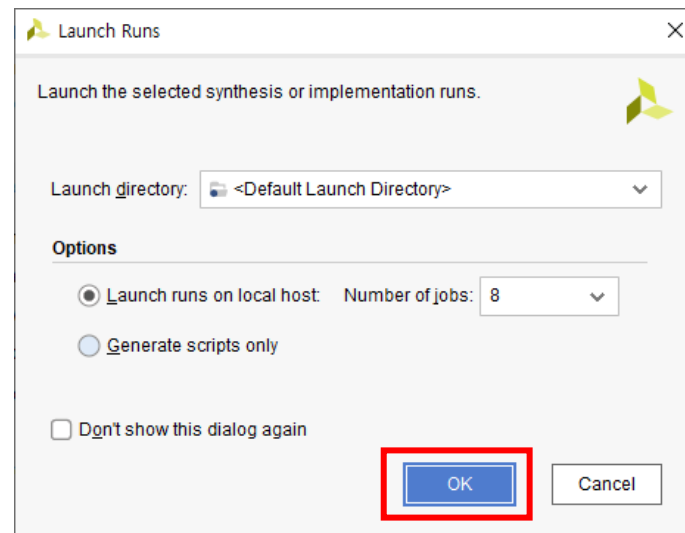
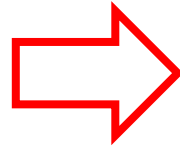
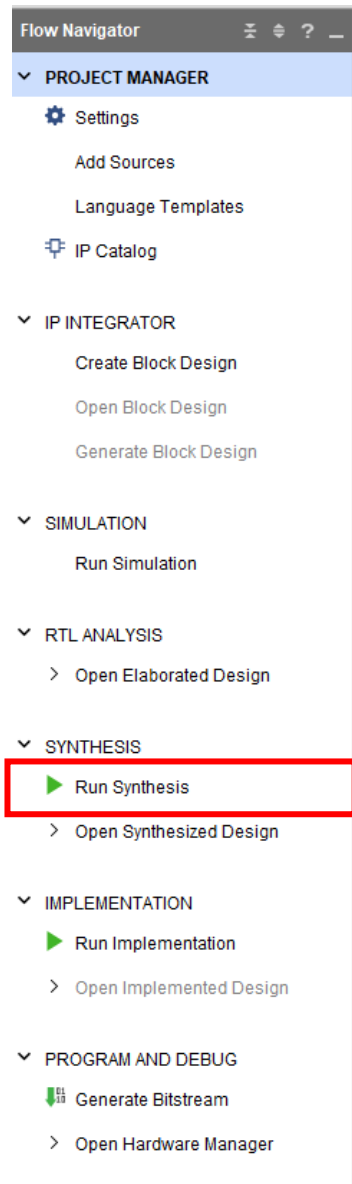


**Make sure that this part is correct!!!**  
**It gives information about generating clock signal.**

# Set Target Constraint



# How to Run Synthesis in Vivado



Tcl Console Messages Log Reports Design Runs x			
🔍 ⚙️ ⏮ ⏪ ⏩ ⏭ + %			
Name	Constraints	Status	WNS
▼ synth_1	constrs_1	Running synth_design...	

# Synthesis Result

- Open Synthesized Design 을 통해, synthesized design 에 대한 다양한 정보 확인 가능.  
(Timing, Utilization, Power, ...)

The screenshot displays the Synthesis tool interface. On the left, under the 'SYNTHESIS' tab, the 'Open Synthesized Design' section contains several options. Two options are highlighted with red boxes and red arrows pointing to summary windows:

- Report Timing Summary**: This option points to the 'Design Timing Summary' window. The window shows a table of timing metrics:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 7.533 ns	Worst Hold Slack (WHS): 0.137 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 8	Total Number of Endpoints: 8	Total Number of Endpoints: 9

Below the table, it states: "All user specified timing constraints are met."

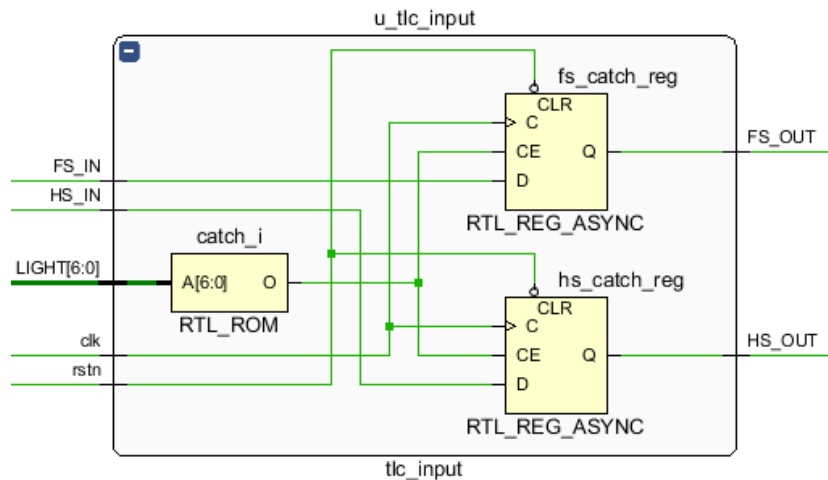
- Report Utilization**: This option points to the 'Summary' window. The window shows a table of resource utilization:

Resource	Utilization	Available	Utilization %
LUT	10	63400	0.02
FF	8	126800	0.01
IO	12	210	5.71

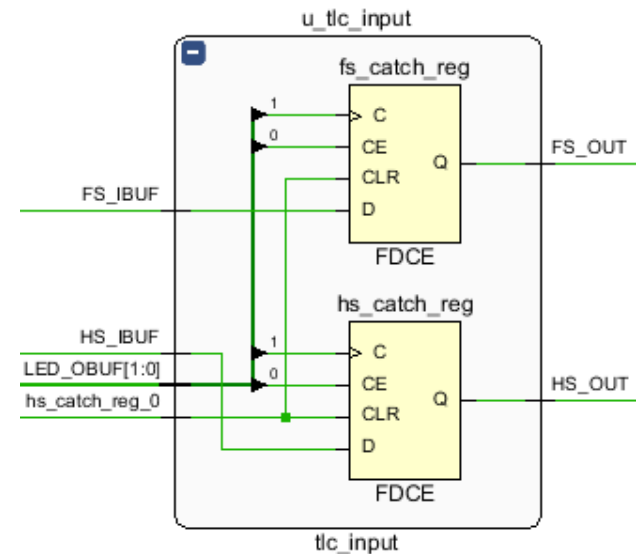
Below the table, there is a bar chart showing utilization percentages for LUT (1%), FF (1%), and IO (6%).

# Synthesis Result

- RTL design 과 synthesized design 의 Schematic을 비교해보면, 모듈의 구성과 핀 이름 등이 변경된 것을 확인할 수 있음.
- Synthesized design 은 FPGA 에서 사용 가능한 cell 만을 이용한다.

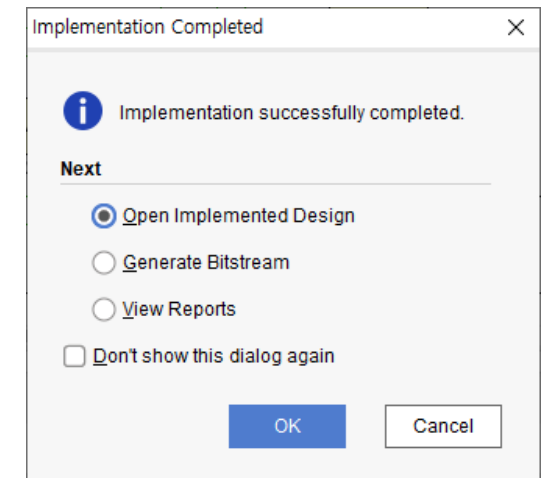
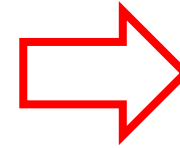
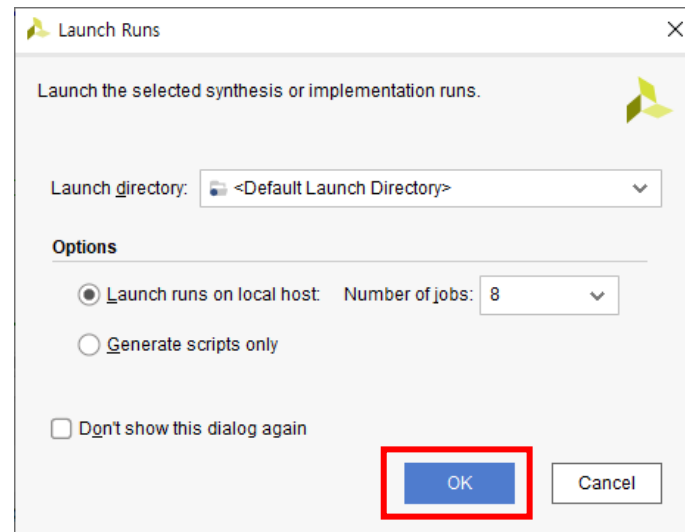
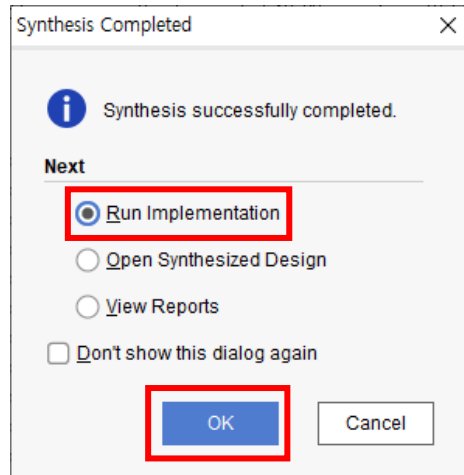
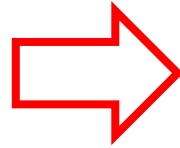


Schematic of RTL Design



Schematic of Synthesized Design

- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager



Tcl Console		Messages	Log	Reports	Design Runs
<div> <span>🔍</span> <span>🔍</span> <span>🔍</span> <span>⏮</span> <span>⏪</span> <span>⏩</span> <span>⏭</span> <span>+</span> <span>%</span> </div>					
Name	Constraints	Status	WNS		
✓ synth_1	constrs_1	synth_design Complete!			
🔄 impl_1	constrs_1	Running Design Initialization...			

# Implementation Result

- 마찬가지로 implemented design 에 대한 정보 확인 가능.
- 보다 실제 FPGA의 동작과 가까운 정보.

The screenshot displays the Xilinx IDE interface with the Implementation Results. The left sidebar shows the 'IMPLEMENTATION' menu with 'Report Timing Summary' and 'Report Utilization' highlighted. The main content area shows the 'Design Timing Summary' and 'Summary' reports.

**Design Timing Summary**

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 6.882 ns	Worst Hold Slack (WHS): 0.163 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 8	Total Number of Endpoints: 8	Total Number of Endpoints: 9

All user specified timing constraints are met.

**Summary**

Resource	Utilization	Available	Utilization %
LUT	10	63400	0.02
FF	8	126800	0.01
IO	12	210	5.71

Utilization (%)

LUT 1%  
FF 1%  
IO 6%

# 4.

---

## Overview of Lab 3

---



## Description of Lab 3

- tlc\_input.v: Module for capturing the input
- **tlc\_fsm.v: Skeleton code for FSM module of TLC model**
- tlc\_output.v: Module for output LED
- tlc\_top.v: Top module for combining all modules
- tb\_tlc.v: Testbench for testing implemented TLC model
  
- Only implement **TODO** part in “tlc\_fsm.v”
  - **다른 부분은 수정하지 말 것.**

## Description of Lab 3

- 매 clock 의 posedge에서 state 변경.
- Total 8 state: S0~S7
- 주어진 'timer' 시그널을 이용해서 S2 and S3 state 에서의 5 clock cycle 를 count.



- Output of top module:

LED[7]	LED[6]	LED[5]	LED[4]	LED[3]	LED[2]	LED[1]	LED[0]
CLOCK	H_RED	H_YELLOW	H_LEFT	H_GREEN	F_RED	F_YELLOW	F_LEFT

# Simulation Result Check

- Test for 3 cases: Car at F / Car at H / Car at H and F
- Tcl Console 을 통해 simulation 결과를 확인 가능.
- 어떤 state 에서 실패했는지 또한 확인 가능.

Case 3: Car at H and F

S1 48000  
S2 50000  
S2 52000  
S2 54000  
S2 56000  
S2 58000  
S4 60000  
S3 62000  
S3 64000  
S3 66000  
S3 68000  
S3 70000  
S5 72000  
S7 74000

[Result] All cars passed w/o crashing. Good job!

**ALL CASE PASS**

Case 3: Car at H and F

S1 fail 48000  
S2 fail 50000  
S2 fail 52000  
S2 fail 54000  
S2 fail 56000  
S2 fail 58000  
S4 fail 60000  
S3 62000  
S3 64000  
S3 66000  
S3 fail 68000  
S3 70000  
S5 fail 72000  
S7 fail 74000

[Result] Cars crashed :(, try again.

**CASE FAIL**

# Synthesis and Implementation Result Check

- 문제 없이 synthesis 와 implementation 을 통과했다면, 아래와 같은 결과를 확인 가능.

Tcl Console Messages Log Reports Design Runs x Timing Utilization																	
Q Z ⇅ ⏮ ⏪ ⏩ ⏭ + %																	
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	
✓ synth_1	constrs_1	synth_design Complete!								10	8	0.0	0	0	8/17/20, 2:51 PM	00:00:21	
✓ impl_1	constrs_1	route_design Complete!	6.882	0.000	0.163	0.000	0.000	0.098	0	10	8	0.0	0	0	8/17/20, 2:52 PM	00:00:47	

- 만약 Timing summary 의 결과가 NA 로 나타난다면, constraints file 의 clock 부분을 잘못 설정한 것. 반드시 수정해주어야 한다.

Tcl Console Messages Log Reports Design Runs x																	
Q Z ⇅ ⏮ ⏪ ⏩ ⏭ + %																	
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	
✓ synth_1	constrs_1	synth_design Complete!								10	8	0.0	0	0	8/17/20, 3:57 PM	00:00:21	
✓ impl_1	constrs_1	route_design Complete!	NA	NA	NA	NA	NA	2.783	0	10	8	0.0	0	0	8/17/20, 3:58 PM	00:00:45	

# Submission

- 구현한 source file(*tlc\_fsm.v*) 과 *Design Runs* 탭의 스크린샷을 압축하여 eTL에 제출

Tcl ConsoleMessagesLogReportsDesign Runs ×TimingUtilization

QZDIIKPP+%

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed
✓ synth_1	constrs_1	synth_design Complete!								10	8	0.0	0	0	8/17/20, 2:51 PM	00:00:21
✓ impl_1	constrs_1	route_design Complete!	6.882	0.000	0.163	0.000	0.000	0.098	0	10	8	0.0	0	0	8/17/20, 2:52 PM	00:00:47

- File name: “학번\_이름\_lab3.zip”
  - ex) 2021-12345\_홍길동\_lab3.zip
- Deadline: 9월 30일 금요일 23:59분까지