



# 컴퓨터공학 기초 실험2

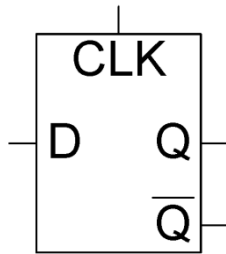
## Lab #4

Latch & flip-flop design with/without reset/set

# Latch & Flip-Flop

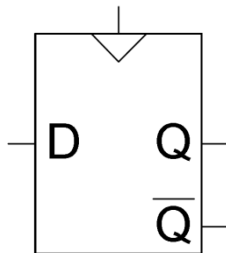
## ➤ D Latch & D Flip-Flop

- ✓ Latch와 flip-flop은 이전에 값을 유지하고 있는 저장 소자 역할을 한다.
- ✓ D latch는 clock이 enable 상태를 유지하는 동안 입력 D 값의 변화를 출력한다.



CLK	Q
0	이전 Q
1	D

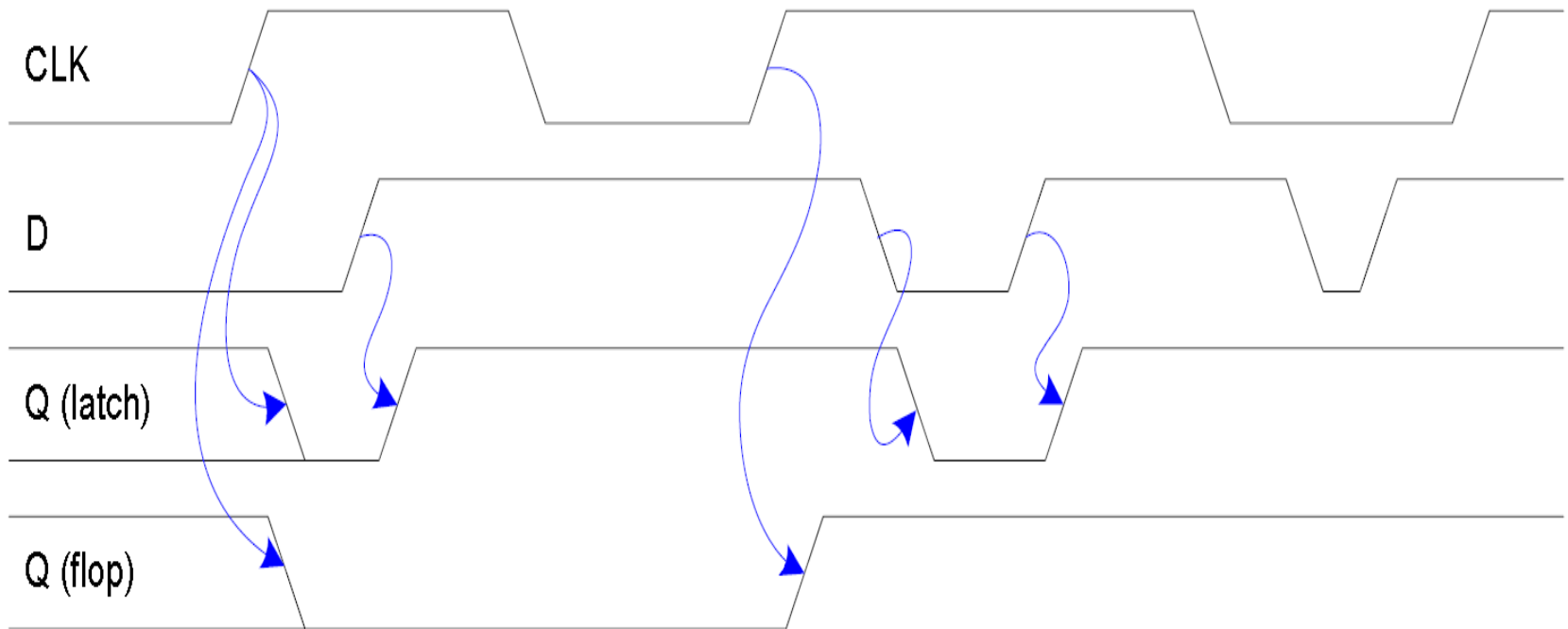
- ✓ D flip-flop는 clock의 rising edge나 falling edge에서만 D 값으로 출력이 바뀌게 된다. 해당 실습에서는 rising edge를 사용한다. 다른 경우에는 D 값이 바뀌더라도 이전 Q 값을 그대로 유지한다.



CLK	Q
↑	D
Other case	이전 Q

## Latch & Flip-Flop(Cont.)

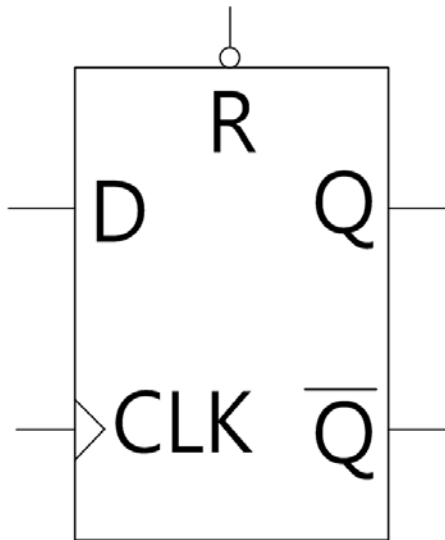
- ▶ 다음의 그림을 통해 latch와 flip-flop의 차이를 알 수 있다.



# Resettable D Flip-Flop

## ➤ Resettable D Flip-Flop

- ✓ D flip-flop에 reset이 기능이 추가된 D flip-flop이다. 실습에서 구현하는 resettable D flip-flop에서 reset은 active low에 동작한다.



Input			Output
R	D	CLK	Q
0	X	X	0
1	0	↑	0
1	1	↑	1
1	x	↓ or 0 or 1	이전 Q

# PRACTICE

# D Latch (1/3)

## ➤ New Project Wizard

- ✓ Project name : \_dlatch
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

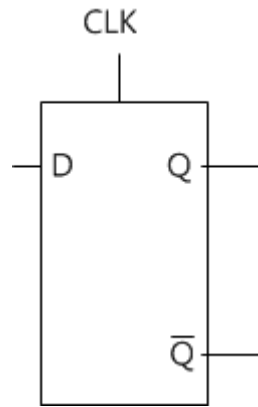
## ➤ Verilog file

- ✓ Add files : gates.v
- ✓ New files : \_dlatch.v, tb\_dlatch.v

파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 4 – ALU32에 사용한 file이다.  
\_srlatch.v는 앞서 생성한 file이다.

# D Latch (2/3)

## ➤ D Latch



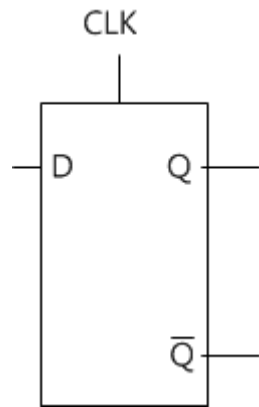
Symbol

Input		Output	
$CLK$	$D$	$Q$	$\bar{Q}$
0	X	$Q_{prev}$	$\bar{Q}_{prev}$
1	0	0	1
1	1	1	0

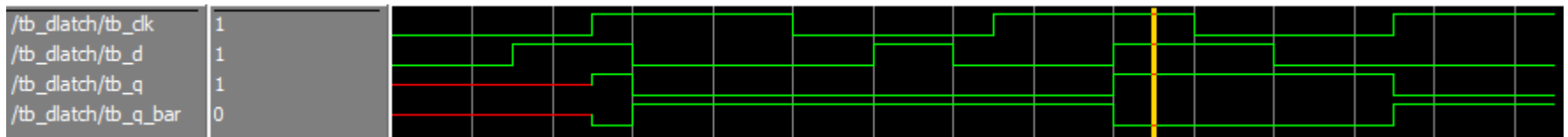
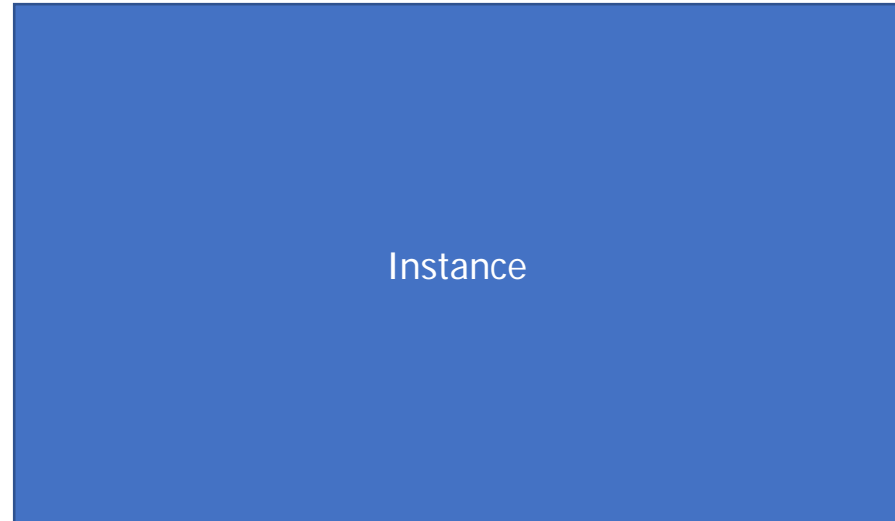
Truth table

# D Latch (3/3)

## ➤ Implementation



symbol



Waveform



# D Flip-Flop (1/3)

## ➤ New Project Wizard

- ✓ Project name : \_dff
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

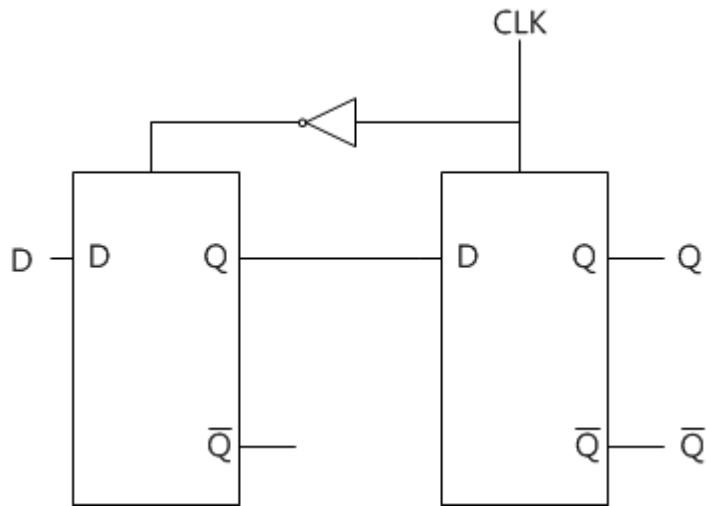
## ➤ Verilog file

- ✓ Add files : gates.v, \_dlatch.v
- ✓ New files : \_dff.v, tb\_dff.v

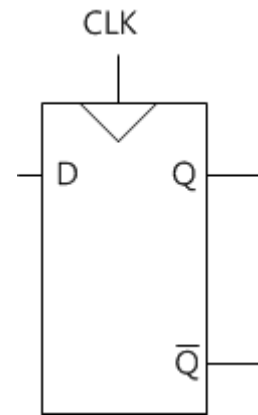
파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 4 – ALU32에 사용한 file이다.  
\_srlatch.v와 \_dlatch.v는 앞서 생성한 file이다.

## D Flip-Flop (2/3)

### ➤ D Flip-Flop



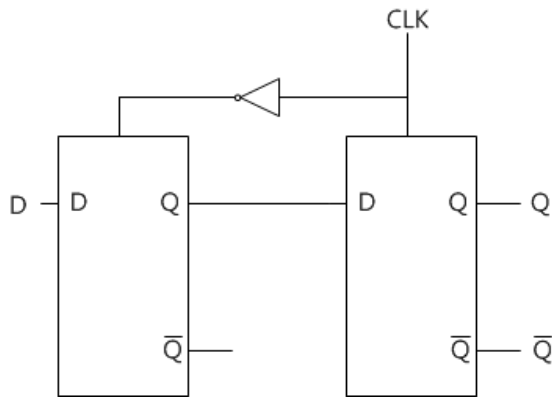
Schematic



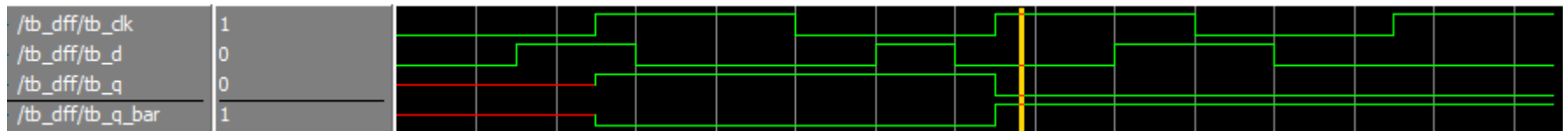
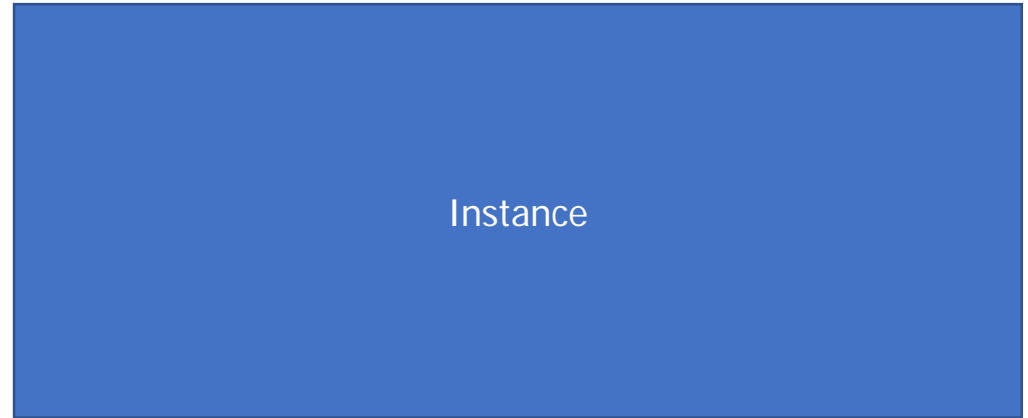
Symbol

# D Flip-Flop (3/3)

## ➤ Implementation



Schematic



Waveform

# Enabled D Flip-Flop (1/2)

## ➤ New Project Wizard

- ✓ Project name : \_dff\_en
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

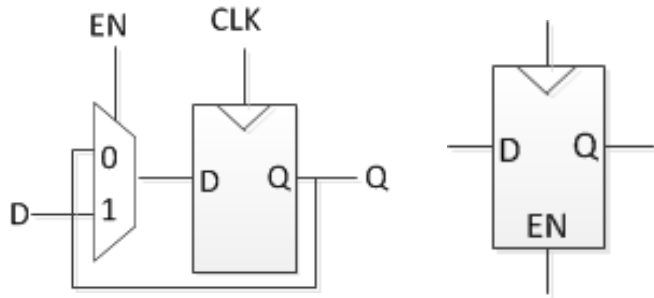
## ➤ Verilog file

- ✓ Add files : gates.v, \_dlatch.v, \_dff.v, mx2.v
- ✓ New files : \_dff\_en.v, tb\_dff\_en.v

파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 4 – ALU32에 사용한 file이다.  
mx2.v는 기존 Lab 4 – ALU4에 사용한 file이다.  
\_srlatch.v, \_dlatch.v와 \_dff.v는 앞서 생성한 file이다.

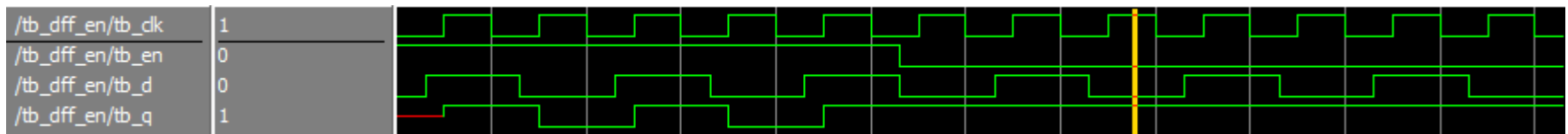
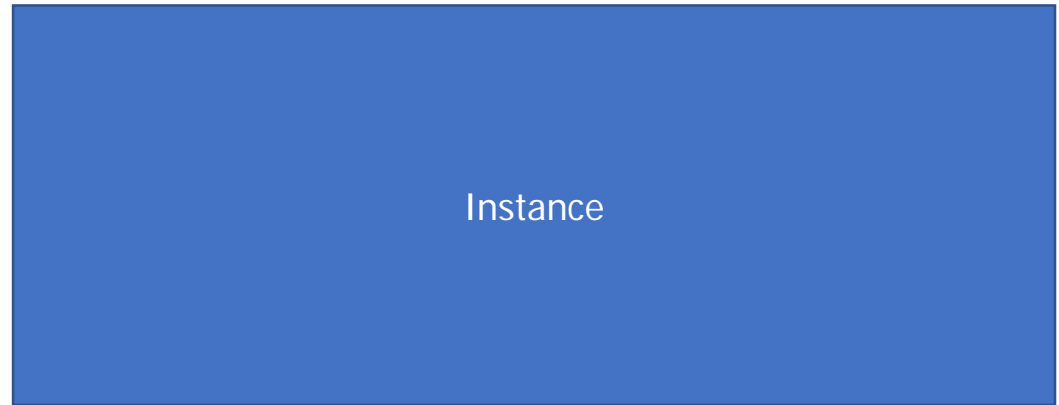
# Enabled D Flip-Flop (2/2)

## ➤ Enabled D Flip-Flop



Schematic

Symbol



Waveform

# Resettable D Flip-Flop (1/2)

## ➤ New Project Wizard

- ✓ Project name : \_dff\_r
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

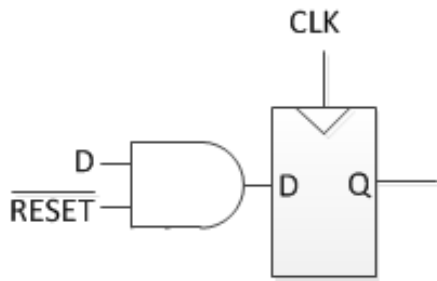
## ➤ Verilog file

- ✓ Add files : gates.v, \_dlatch.v, \_dff.v
- ✓ New files : \_dff\_r.v, tb\_dff\_r.v

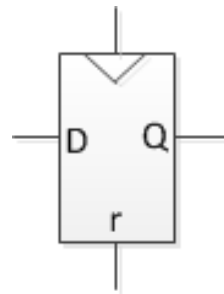
파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 3 – ALU32에 사용한 file이다.  
mx2.v는 기존 Lab 3 – ALU4에 사용한 file이다.  
\_srlatch.v, \_dlatch.v와 \_dff.v는 앞서 생성한 file이다.

# Resettable D Flip-Flop (2/2)

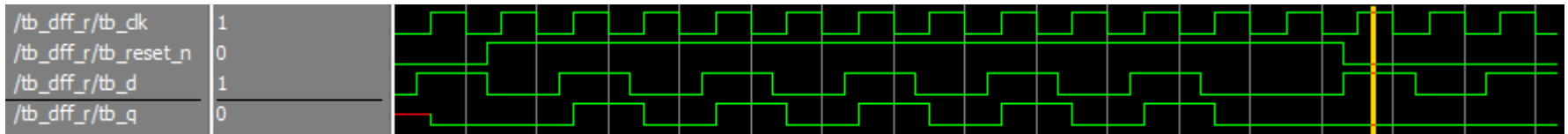
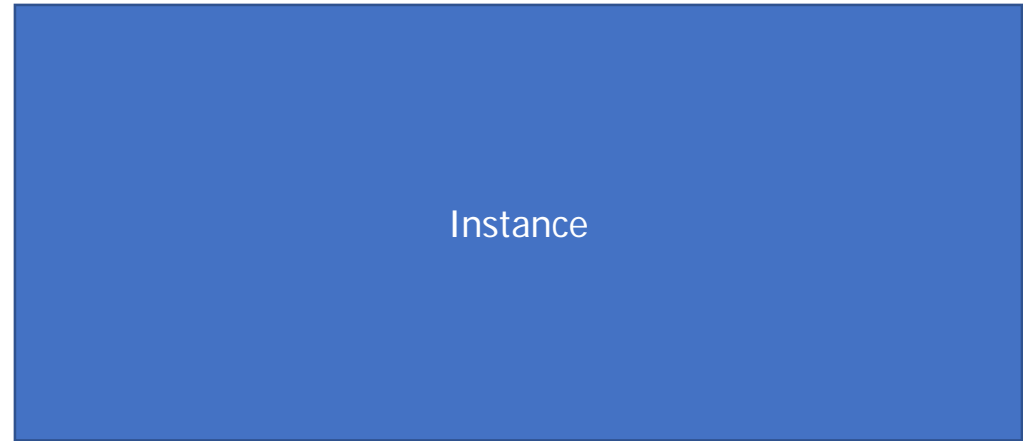
- D flip-flop with active-low synchronous reset



Schematic



Symbol



Waveform

# Synchronous Set/Resettable D Flip-Flop (1/2)

## ➤ New Project Wizard

- ✓ Project name : \_dff\_rs
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

## ➤ Verilog file

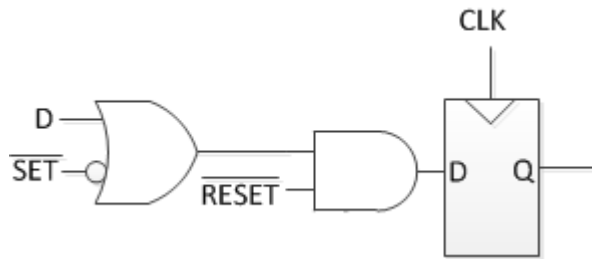
- ✓ Add files : gates.v, \_dlatch.v, \_dff.v
- ✓ New files : \_dff\_rs.v, tb\_dff\_rs.v

파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 4 – ALU32에 사용한 file이다.  
mx2.v는 기존 Lab 4 – ALU4에 사용한 file이다.  
\_srlatch.v, \_dlatch.v와 \_dff.v는 앞서 생성한 file이다.

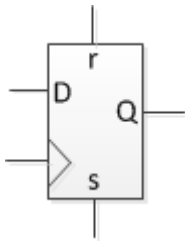


# Synchronous Set/Resettable D Flip-Flop (2/2)

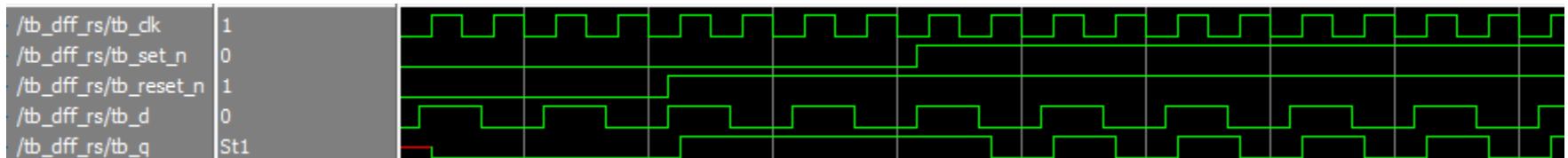
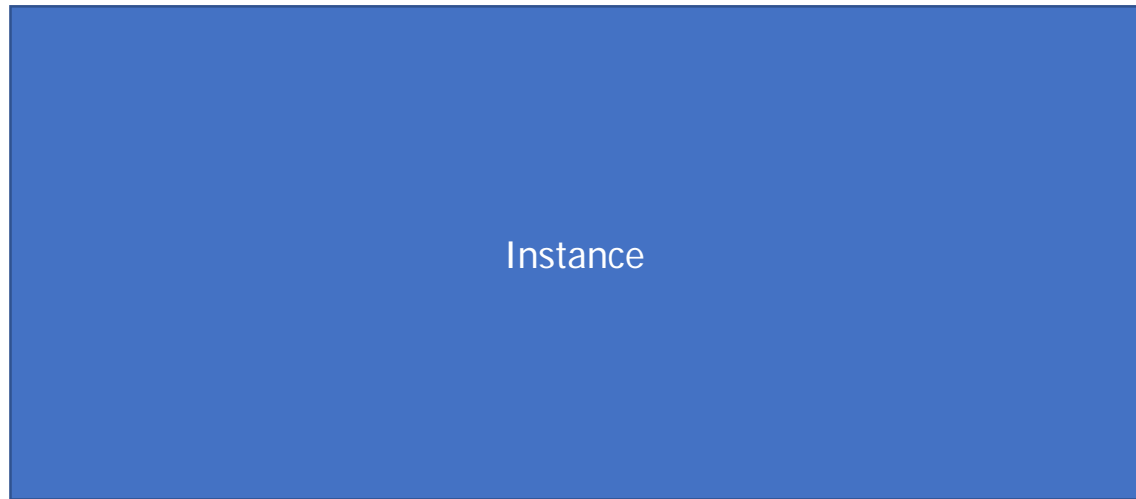
- D flip-flop with active-low synchronous reset and set



Schematic



Symbol



Waveform

# Register (1/3)

## ➤ New Project Wizard

- ✓ Project name : \_register32
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

## ➤ Verilog file

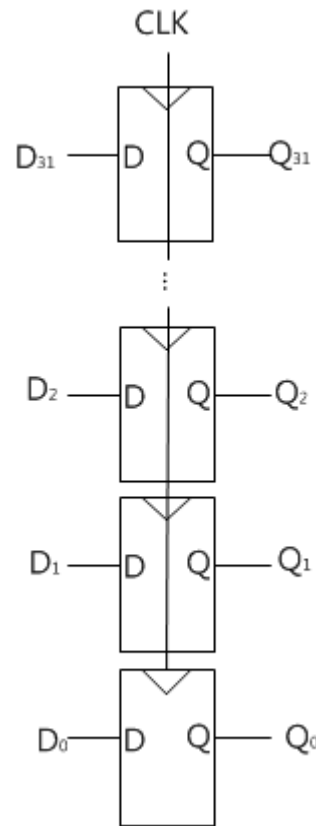
- ✓ Add files : gates.v, \_dlatch.v, \_dff.v
- ✓ New files : \_register8.v, \_register32.v, tb\_register32.v

파일을 추가할 때 해당 프로젝트 폴더에 복사하여 집어넣는다.  
gates.v는 기존 Lab 4 – ALU32에 사용한 file이다.  
mx2.v는 기존 Lab 4 – ALU4에 사용한 file이다.  
\_srlatch.v, \_dlatch.v와 \_dff.v는 앞서 생성한 file이다.

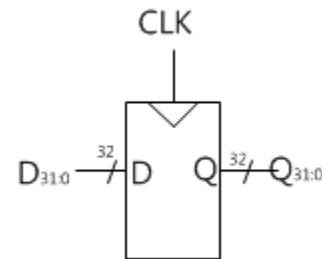
# Register (2/3)

## ➤ N-bits Register

- ✓ N-bits register는 N개의 flip-flop을 한 줄로 늘어놓음으로써 구현



Schematic

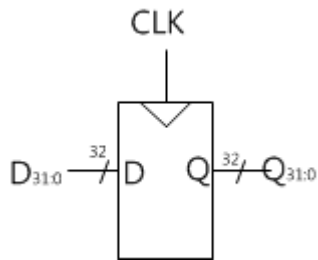


Symbol

# Register (3/3)

## ➤ Implementation

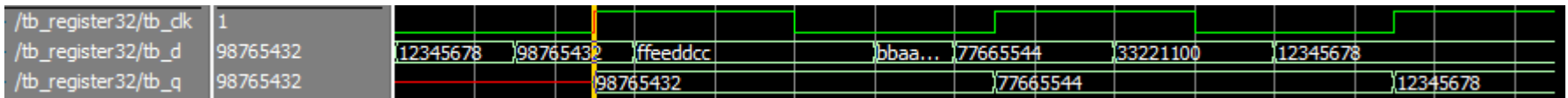
- ✓ 32-bits register



Symbol

Instance of 8 bits register

Instance of D FF



Waveform

# Async/Sync Set/Resettable D Flip-Flop (1/4)

## ➤ New Project Wizard

- ✓ Project name : \_dff\_rs\_sync\_async
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

## ➤ Verilog file

- ✓ Add files : -
- ✓ New files : \_dff\_rs\_sync.v, \_dff\_rs\_async.v, \_dff\_rs\_sync\_async.v  
tb\_dff\_rs\_sync\_async.v

해당 프로젝트는 기존의 structural implementation과 다르게 **behavioral implementation**을 통하여 **asynchronous/synchronous** 차이를 확인하는 것으로 별도의 파일 추가는 하지 않는다.

## Async/Sync Set/Resettable D Flip-Flop (2/4)

- D flip-flop with active-low synchronous reset and set

```
module _dff_rs_sync(clk, set_n, reset_n, d, q);  
    input      clk, set_n, reset_n, d;  
    output     q;  
    reg q;  
  
    always@(posedge clk)
```

Instance

# Async/Sync Set/Resettable D Flip-Flop (3/4)

- D flip-flop with active-low asynchronous reset and set
  - ✓ What is difference with previous D flip-flop?

```
module _dff_rs_async(clk, set_n, reset_n, d, q);  
    input      clk, set_n, reset_n, d;  
    output     q;  
    reg q;  
  
    always@(posedge clk or negedge set_n or negedge reset_n)
```

Instance

# Async/Sync Set/Resettable D Flip-Flop (4/4)

- 앞서 작성한 두 개의 set/resettable D FF를 추가하고 이에 대한 testbench를 작성하여 두 개의 차이점을 확인하여 본다.

✓ Hint : set\_n과 reset\_n이 모두 active low에 동작

```
module _dff_rs_sync_async(clk, set_n, reset_n, d, q_sync, q_async);  
input clk, set_n, reset_n, d;  
output q_sync, q_async;
```

Instance



# Assignment 4

## ➤ Report

- ✓ 자세한 사항은 lab document 참고

## ➤ Submission

### ✓ Soft copy

- 월요일 반 : 10월 19일 밤 23시 59분까지
- 금요일 오전반 : 10월 23일 밤 23시 59분까지
- 금요일 오후반 : 10월 23일 밤 23시 59분까지
- 실습 미수강은 디지털 논리2 조교 공지에 따름

# 채점기준

세부사항		점수	최상	상	중	하	최하
소스코드	Source code가 잘 작성 되었는가? (Structural design으로 작성되었는가?)	10	10	8	5	3	0
	주석을 적절히 달았는가? (반드시 영어로 주석 작성)	20	20	15	10	5	0
설계검증 (보고서)	보고서를 성실히 작성하였는가? (보고서 형식에 맞추어 작성)	30	30	20	10	5	0
	합성결과를 설명하였는가?	10	10	8	5	3	0
	검증을 제대로 수행하였는가? (모든 입력 조합, waveform 설명)	30	30	20	10	5	0
총점		100					

# References

- Altera Co., [www.altera.com/](http://www.altera.com/)
- D. M. Harris and S. L. Harris, Digital Design and Computer Architecture, Morgan Kaufmann, 2007
- 이준환, 디지털논리회로2 강의자료, 광운대학교, 컴퓨터 공학과, 2020

Q&A

**THANK YOU**