

컴퓨터 공학 기초 실험 2

Assignment 5. Latch & flip-flop design with/without reset/set

1. D Latch

이번 실습에서는 이전에 값을 유지하고 있는 저장 소자 역할을 하는 latch 와 flip-flop 을 설계하고, 더불어 reset 과 set 기능을 구현한다. 또한, 구현한 flip-flop 을 사용하여 N-bits register 를 구현하는 방법에 대하여 살펴보도록 한다

➤ Description

- ✓ D latch 는 clock 이 enable 상태를 유지하는 동안 입력 D 값의 변화를 출력한다.
- ✓ 다음은 D latch 의 symbol 과 truth table 이다.

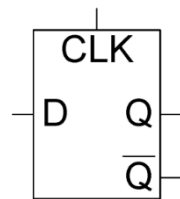


Figure 1 - D latch 의 symbol

CLK	Q
0	이전 Q
1	D

Table 1- D latch 의 truth table

➤ Structural Description

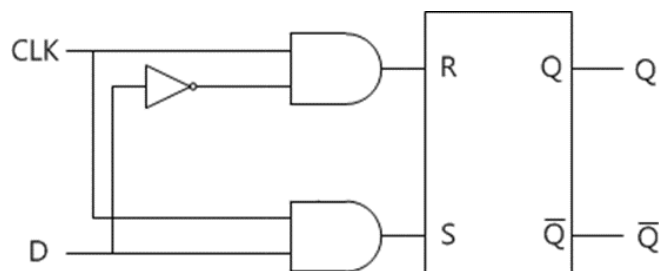


Figure 2 . D-latch Schematic

2. D Flip-Flop

➤ Description

- ✓ D flip-flop 은 clock 의 rising edge 나 falling edge 에서만 D 값으로 출력이 바뀌게 된다. 다른 경우에는 D 값이 바뀌더라도 이전 Q 값을 그대로 유지한다. 해당 실습에서는 clock 의 rising edge 를 사용한다.
- ✓ 다음은 D flip-flop 의 symbol 과 truth table 이다.

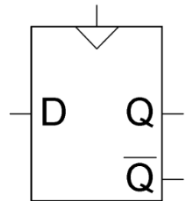


Figure 3 - D flip-flop 의 symbol

CLK	Q
↑	D
in other case	이전 Q

Table 2 - D flip-flop 의 truth table

➤ Structural Description

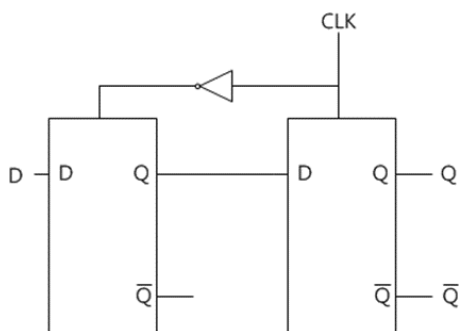


Figure 4 . D-latch Schematic

3. Comparison of D Latch and D Flip-Flop

- 다음 그림을 통해 D latch 와 D flip-flop 의 차이를 알 수 있다.

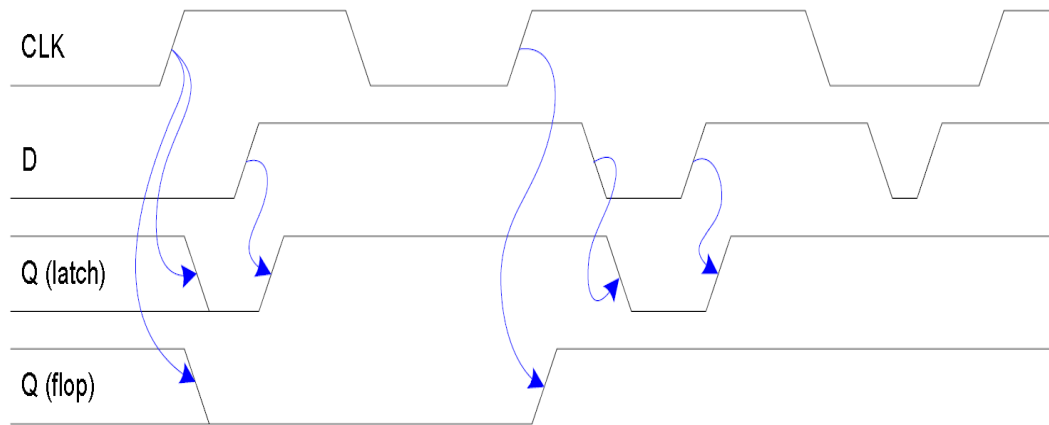


Figure 5 - Waveform of D Latch and D Flip-Flop

4. Resettable D Flip-Flop

➤ Description

- ✓ Resettable D flip-flop 은 D flip-flop 에 reset 기능이 추가된 D flip-flop 이다. 실습에서 구현하는 resettable D flip-flop 은 active low 에 동작한다. Active low 에 동작한다는 reset 의 값이 0 일 때, reset 기능을 수행한다는 의미이다.
- ✓ 다음은 Asynchronous resettable D flip-flop 의 symbol 과 truth table 이다.

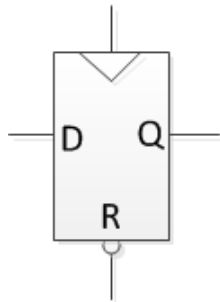


Figure 6 – Resettable D FF 의 symbol

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

Table 3 - Resettable D FF 의 truth table

5. Set/Resettable D Flip-Flop

➤ Description

- ✓ Set/Resettable D flip-flop 은 D flip-flop 에 reset 과 set 기능이 추가된 D flip-flop 이다. 실습에서 구현할 set/resettable D flip-flop 에서 set 과 reset 은 active low 에 동작하며, reset 신호가 set 신호보다 우선순위를 가지고 있다. 즉, reset 과 set 둘 다 모두 0 이면, reset 이 우선순위를 가져 D flip-flop 의 결과는 0 이 된다.
- ✓ 다음은 Asynchronous set/resettable D flip-flop 의 symbol 과 truth table 이다.

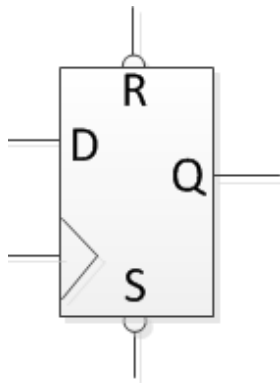


Figure 7 - Set/resettable D FF 의 symbol

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

Table 4 – Set/resettable D FF 의 truth table

6. Verilog 구현

➤ D Latch

✓ Implementation

앞서 설명한 D latch 의 schematic 은 다음 그림과 같다.

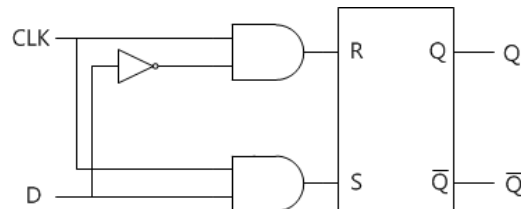


Figure 8 - D latch 의 schematic

_dlatch 라는 이름으로 project 와 top module 을 만들고, D latch 를 구현 후 검증하여 본다.

- Top module 의 이름을 dlatch 로 할 경우, verilog 에서 사전에 정의된 primitive 이기 때문에 사용할 수 없다.

➤ D Flip-Flop

✓ Implementation

앞서 설명한 D flip-flop 은 D latch 2 개를 이용하여 구조적으로 구현할 수 있다. 다음 그림은 D flip-flop 의 schematic 이다.

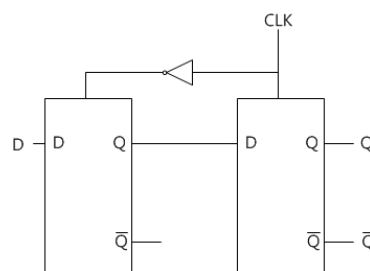


Figure 9 - D flip-flop 의 schematic

_dff 라는 이름으로 project 와 top module 을 만들고, D flip-flop 을 구현 후 검증한다.

➤ Enabled D Flip-Flop

✓ Implementation

Enabled D FF 는 D FF 앞에 multiplexer 를 추가하여 입력 data 를 사용할 지를 결정한다.

다음 그림은 enabled D FF 의 schematic 이다.

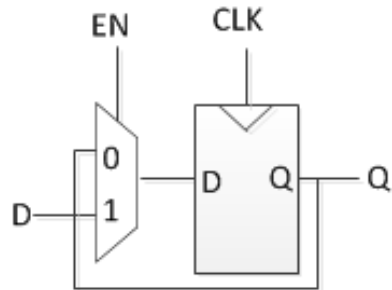


Figure 10 – Enabled D flip-flop 의 schematic

_dff_en 라는 이름으로 project 와 top module 을 만들고, enable D FF 를 구현 후 검증한다.

➤ Resettable D Flip-Flop

✓ Implementation

Resettable D FF 는 D FF 앞에 and gate 를 추가하여 입력 data 를 초기화할 지를 결정한다. 다음 그림은 resettable D FF 의 schematic 이다.

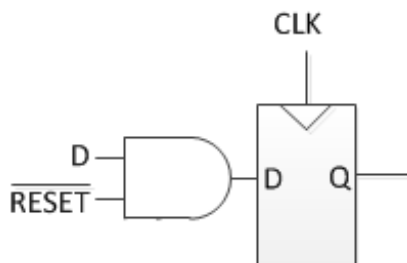


Figure 11 – Resettable D flip-flop 의 schematic

_dff_r 라는 이름으로 project 와 top module 을 만들고, resettable D FF 를 구현 후 검증한다.

➤ Synchronous Set/Resettable D Flip-Flop

✓ Implementation

해당 실습에서는 reset 과 set 기능을 가지는 D flip-flop 을 구현하여 본다. 여기서 active low 의 의미는 reset 이나 set 이 각각 0 이 되었을 때, reset 또는 set 의 기능을 함을 의미하며, synchronous 는 clock 에 동기화가 맞춰지는 것을 의미한다. 이를 구조적으로 구현하기 위하여 앞서 구현한 D flip-flop 과 and, or, not gate 를 이용한다. 다음 그림은 이에 대한 schematic 이다.

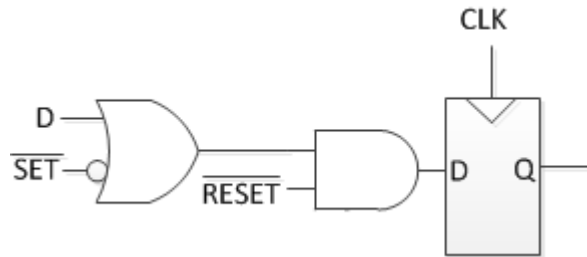


Figure 12 – Set/resettable D flip-flop 의 schematic

_dff_rs 라는 이름으로 project 와 top module 을 만들고, set/resettable D FF 를 구현 후 검증한다.

➤ Register

✓ Description

N-bits register 는 N 개의 flip-flop 을 한 줄로 늘어놓음으로써 구현한다. 다음 그림은 32 bits register 의 symbol 을 나타낸다.

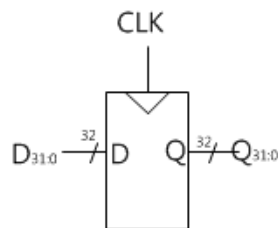


Figure 13 – 32bits register 의 symbol

✓ Implementation

다음 그림은 32 bits register 의 schematic 이다.

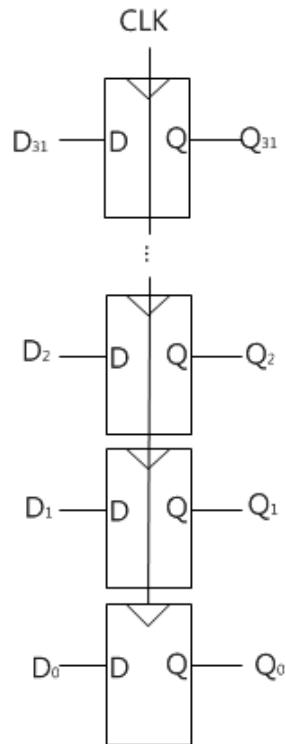


Figure 14 - 32bit register 의 schematic

(32-bit register 를 구현 시 D FF 를 32 개 instantiation 하여도 되고, 8-bit register 를 구현 후 이를 4 개 instantiation 하여 32-bit register 를 구현하여도 무관하다. 하지만, 이번 실습에서는 8-bit register 를 instance 사용하는 것을 기준으로 한다.)

➤ Async/Sync Set/Resettable D Flip-Flop

- ✓ 앞선 flip-flop(또는 register)의 구현은 structural implementation 을 하였다. 이번에는 behavioral implementation 을 통하여 asynchronous/synchronous 의 차이를 확인하는 것이 목적이다.
- ✓ Implementation of Synchronous Set/Resettable D Flip-Flop
다음 그림은 synchronous set/resettable D FF 의 code 이다.

```
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)
    begin
        if(reset_n == 0)      q <= 1'b0;
        else if(set_n == 0)   q <= 1'b1;
        else                  q <= d;
    end

endmodule
```

Figure 15 – Implementation of synchronous set/resettable D FF

- ✓ Implementation of Asynchronous Set/Resettable D Flip-Flop
다음 그림은 asynchronous set/resettable D FF 의 code 이다.

```
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)
    begin
        if(reset_n == 0)      q <= 1'b0;
        else if(set_n == 0)   q <= 1'b1;
        else                  q <= d;
    end

endmodule
```

Figure 16– Implementation of asynchronous set/resettable D FF

✓ Implementation of Top Module

다음 그림은 앞선 두 module 을 instance 한 top module 이다.

```
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;  
  
_dff_rs_sync    U0_dff_rs_sync(.clk(clk), .set_n(set_n), .reset_n(reset_n), .d(d), .q(q_sync));  
_dff_rs_async   U1_dff_rs_async(.clk(clk), .set_n(set_n), .reset_n(reset_n), .d(d), .q(q_async));  
  
endmodule
```

Figure 17 - Implementation of top module

✓ Top module 에 대한 testbench 를 작성하여 두 개의 차이점을 확인하여 본다.

(Hint : set_n 과 reset_n 이 모두 active low 에 동작한다.)

7. Report

레포트는 공지사항에 올린 보고서 양식에 맞추어 작성하고, 다음의 사항에 대하여서도 추가적으로 작성한다.

- D latch, D flip-flop, enabled D flip-flop, resettable D flip-flop, set/resettable D flip-flop, 32 bits register, async/sync set/resettable D flip-flop 에 대하여 구현하고 검증한다.
- ✓ RTL viewer 는 포함한다.
- 원리(배경지식)에 nand gate 4 개를 이용하여 D latch 을 구현하는 방법에 대하여 조사하고 설명한다.
- 원리(배경지식)에 실습에서 구현한 enabled D flip-flop 이 아닌 다른 방법으로 enabled D flip-flop 을 구현하는 방법에 대하여 조사하고 설명한다.
- 고찰 및 결론에 D FF with active-low synchronous reset and set 과 D FF with active-low asynchronous reset and set 의 차이를 설명하여라.
- 채점기준

세부사항		점수	최상	상	중	하	최하
소스코드	Source code 가 잘 작성 되었는가? (Structural design 으로 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					

8. Submission

➤ 제출기한

- 자세한 제출기한은 KLAS 와 일정을 참고

➤ 과제 업로드

- ✓ Source code 와 report 를 같이 ZIP 파일로 압축하여 KLAS(종합정보서비스) 과제 제출에 해당 과제 upload
 - ✓ 업로드 파일명은 (요일#)_(학번)_Assignment_#.zip
 - 요일번호
 - 실습 미수강은 0
 - 월요일 0, 1, 2 교시 1
 - 화요일 0, 1, 2 교시 2
 - 수요일 0, 1, 2 교시 3
 - Ex) 월요일 반 수강, 2019110609, Assignment 1 제출 시
1_2019110609_Assignment_01.zip 으로 제출
 - ✓ Report 명은 (요일#)_(학번)_Assignment_#.pdf
 - 요일 번호는 위의 업로드 파일명과 동일하게 진행
 - ✓ Ex) 수요일 반 수강, 2019110609, Assignment 1 제출 시
3_2019110609_Assignment_01.pdf 으로 제출
 - ✓ Report 는 PDF 로 변환해 제출 (미수행시 감점)
- Source code 압축 시 db, incremental_db, simulation 폴더는 삭제 (미수행시 감점)
- Source code 압축 시 ~.bak 파일 삭제 (미수행시 감점)
- 제출할 프로젝트
- ✓ 해당 실습은 프로젝트 폴더가 아닌 **Verilog file** 만을 하나의 폴더에 모아서 제출한다.
제출할 file list 는 다음과 같다. 그 외에 필요하다고 생각되는 파일을 추가하여도 된다.
 - ✓ Module file
gates.v, mx2.v, _dlatch.v, _dff.v, _dff_en.v, _dff_r.v, _dff_rs.v, _register8.v, _register32.v, _dff_rs_sync.v, _dff_rs_async.v, _dff_rs_sync_async.v
 - ✓ Testbench file
tb_dlatch.v, tb_dff.v, tb_dff_en.v, tb_dff_r.v, tb_dff_rs.v, tb_register32.v, tb_dff_rs_sync_async.v