



디지털논리회로실습 Lab 5.

2017. 10. 25

HANYANG University

Contents

기본문법 실습

실습 설명

- Part I : R-S latch(Flip-flop)
- Part II : D-latch(Flip-flop)
- Part III : Master-Slave D Flip-flop
- Part IV : Gated D-Latch, edge triggered D Flip-flop

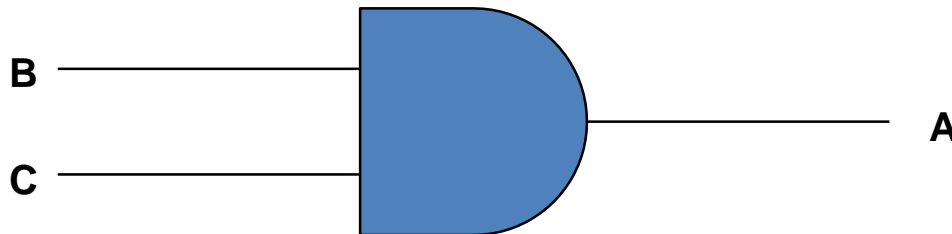
기본 문법 실습

wire

- 표현하고 싶은 node들을 단지 물리적인 연결 기본문법

➡ wire <변수명>;


```
1 wire      A, B, C, D, E; // simple 1-bit wide wires
2 wire [8:0] Wide;        // a 9-bit wide wire
3 reg I;
4
5 assign A = B & C;        // using a wire with an assign statement
```



기본 문법 실습

reg

- wire하고는 달리 그 신호에 새로운 값이 할당되기 전 까지 현재의 값을 그대로 유지
- 기본문법

 reg <변수명>;

```
module muxx(SW1, SW2, SW3, LED);  
    input SW1, SW2, SW3;  
    output LED;  
  
    wire a, b, c;  
    reg d;  
  
    assign a=SW1;  
    assign b=SW2;  
    assign c=SW3;  
  
    always @(posedge c) begin  
        d = a & b;  
    end  
  
    assign LED=d;  
  
endmodule
```

기본 문법 실습

always

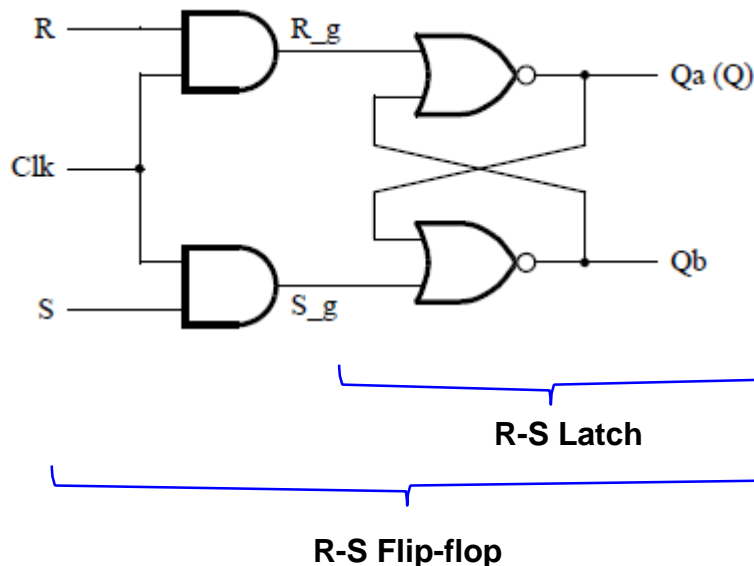
- 단독으로 쓰이면 '항상' 실행, always뒤에 @이 붙으면 @뒤에 조건이 '참'일 때만 실행
- 기본문법
 - ✎ always @ (<이벤트 리스트>) begin
... end

```
□ always @ (A or B) begin
  | if(A>B)
  |   Y = A;
  | else
  |   Y = B;
  | end
  |
```

실습설명

Part I : R-S latch(Flip-flop)

- Latch와 Flip-flop은 1bit 의 신호를 저장하기 위해 사용
- Flip-flop은 Latch에 Clock 신호가 연결되어있는 형태로 Clock신호를 사용하는 이유는 입력신호의 동기화를 위해 사용함



입력	출력
S R	Q
0 0	Q(변화없음)
0 1	0
1 0	1
1 1	x

실습설명

Part I : R-S latch(Flip-flop)

Verilog code

```
module part1 (Clk, R, S, Q);
    input Clk, R, S;
    output Q;

    wire R_g, S_g, Qa, Qb /* synthesis keep */;

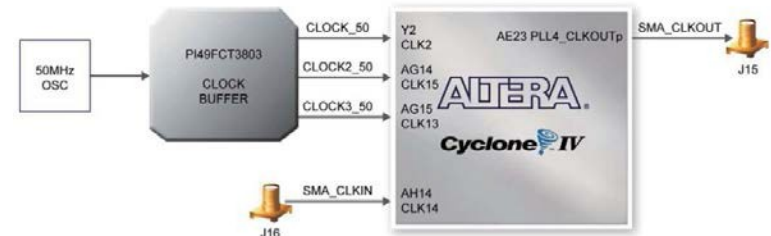
    assign R_g = R & Clk;
    assign S_g = S & Clk;
    assign Qa = ~(R_g | Qb);
    assign Qb = ~(S_g | Qa);

    assign Q = Qa;

endmodule
```

NOTE: Clk

- DE2-115보드에는 50 Mhz clock
신호기가 있음



Signal Name	FPGA Pin No.	Description	I/O Standard
CLOCK_50	PIN_Y2	50 MHz clock input	3.3V
CLOCK2_50	PIN_AG14	50 MHz clock input	3.3V
CLOCK3_50	PIN_AG15	50 MHz clock input	Depending on JP6
SMA_CLKOUT	PIN_AE23	External (SMA) clock output	Depending on JP6
SMA_CLKIN	PIN_AH14	External (SMA) clock input	3.3V

실습설명

Clk

- 50Mhz의 신호를 발생



- 기본문법

```
module muxx(SW1, Clk, LED);  
  
    input Clk;  
    input [1:0]SW1;  
    output [3:0]LED;  
    reg [3:0]a,b;  
  
    always @(posedge Clk) begin  
        if(SW1==2'b00) begin  
            a=4'b0000;  
            b=4'b1111;  
        end  
        else if(SW1==2'b01)  
            a=4'b0001;  
        else if(SW1==2'b10)  
            a=4'b0010;  
        else  
            a=4'b0111;  
    end  
  
    assign LED=a;  
  
endmodule
```


실습설명

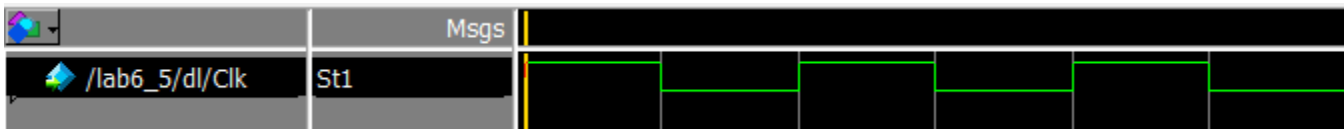
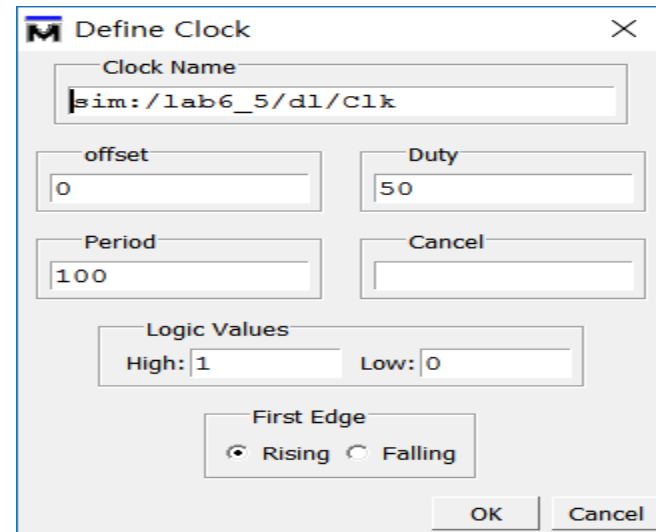
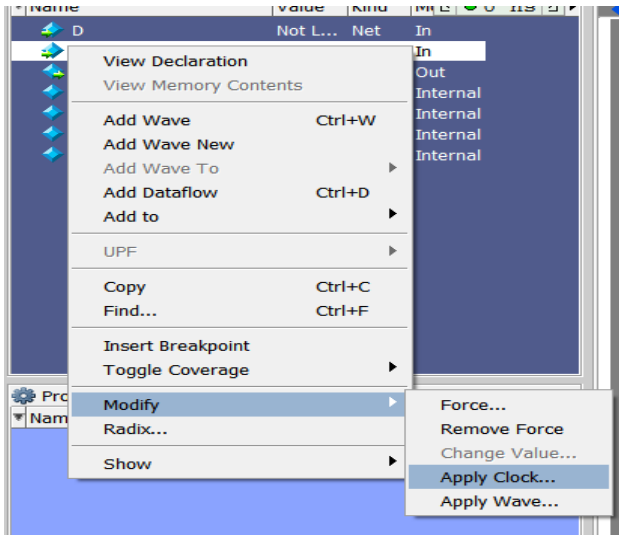
Part I : R-S latch(Flip-flop)

■ Clk in ModelSim

➤ Offset -> delay

➤ Duty -> ratio between 1 and 0

➤ Period 100 -> 0.1 ns

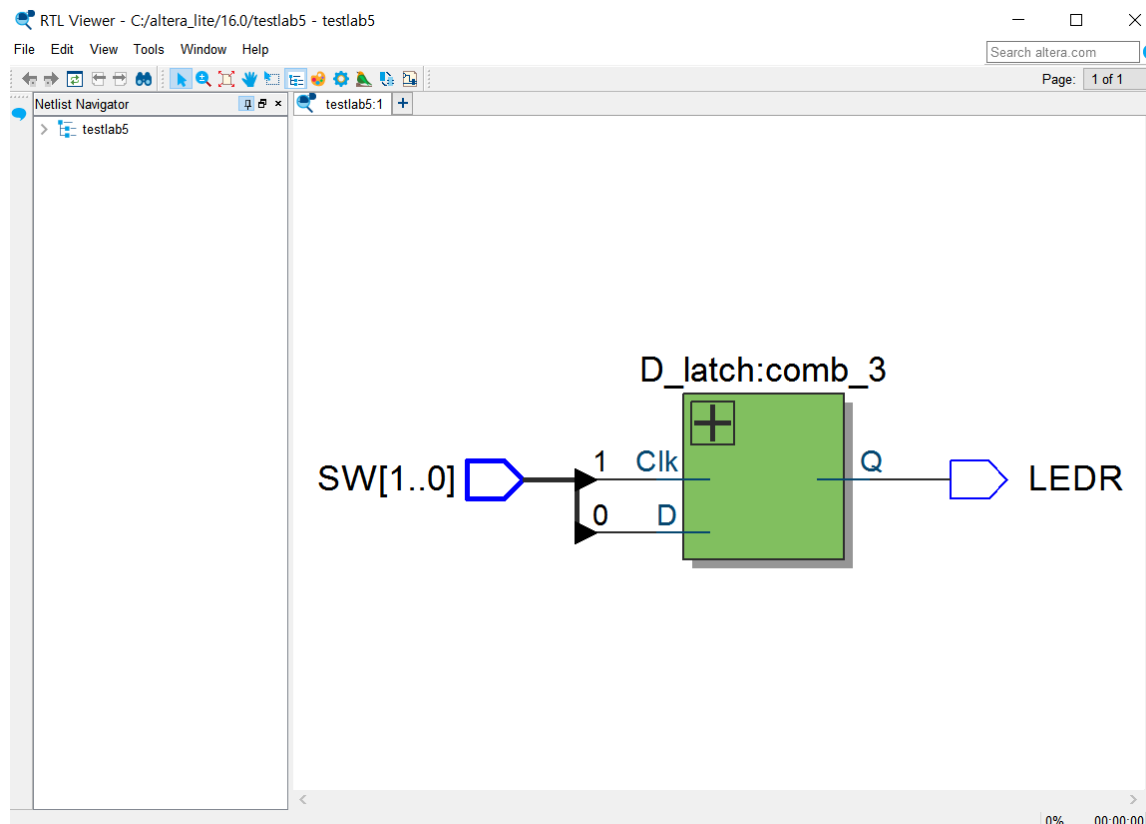


실습설명

Part I : R-S latch(Flip-flop)

RTL (Register Transistor Level) viewer

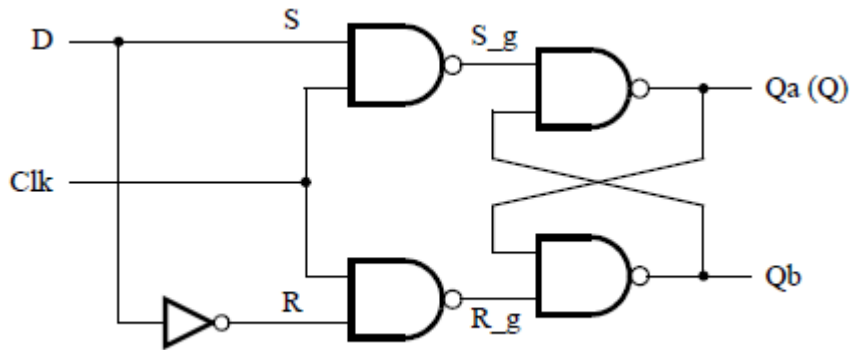
 Tools – Netlist Viewers – RTL Viewer



실습설명

Part II : D latch(Flip-flop)

- 현재상태와 관계없이 입력신호(D)를 그대로 출력(Q) 시키는 회로
- RS Flip-flop의 문제점을 해결하기 위해 R,S에 동시에 1을 넣을 수 없게 만들었음
- 데이터 전송용으로 많이 쓰임

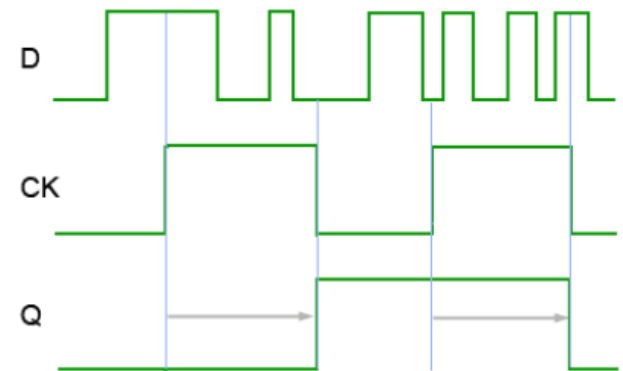
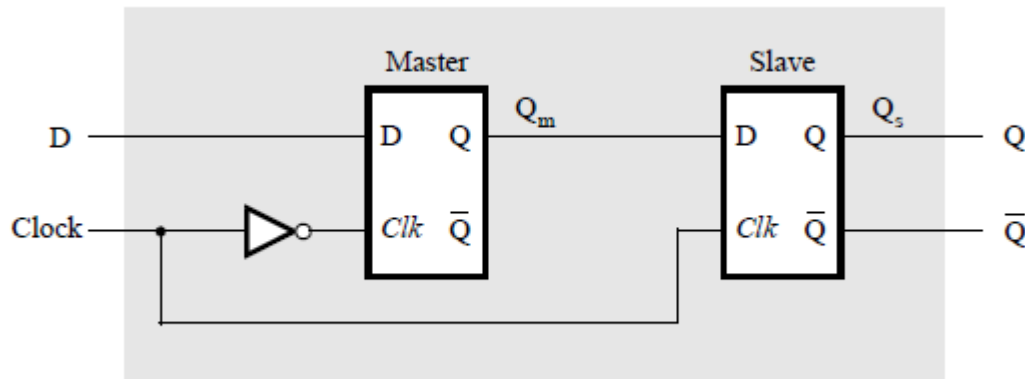


입력	출력
D	Q(t+1)
0	0 (Reset)
1	1 (Set)

실습설명

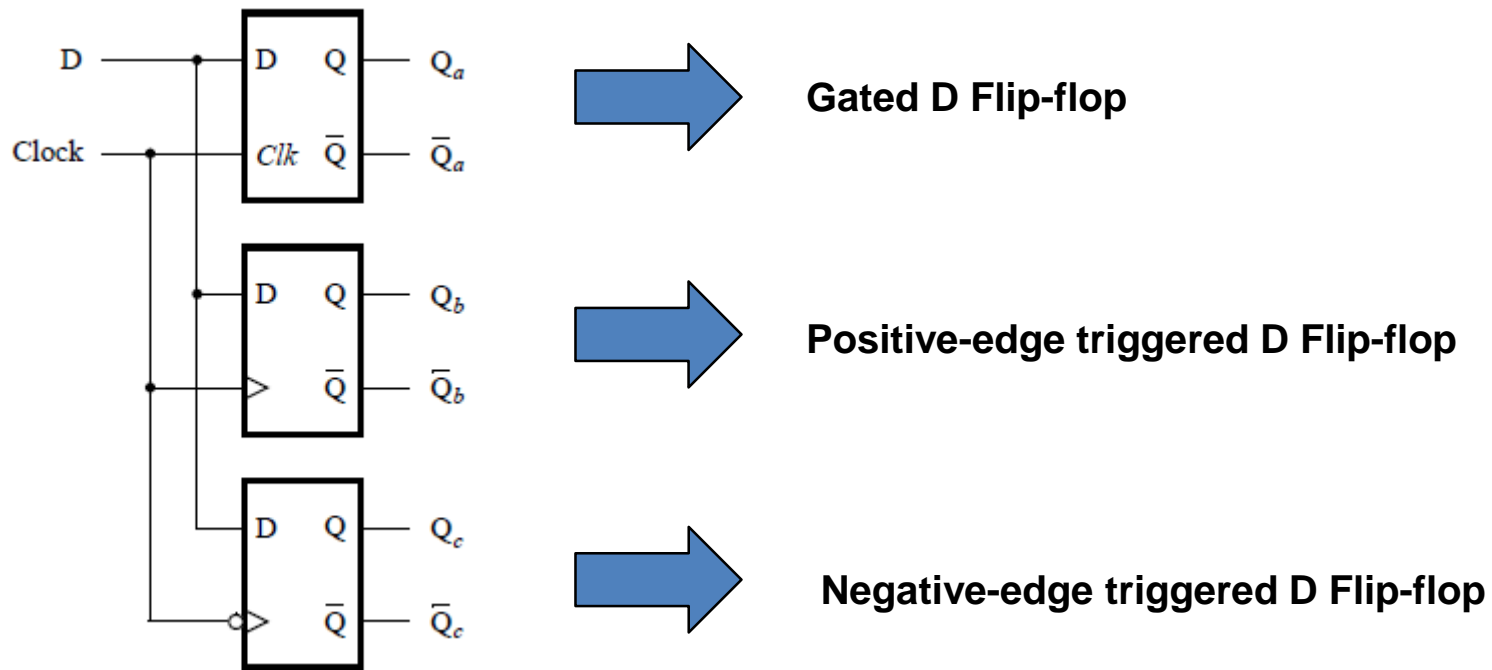
Part III : Master-Slave D Flip-flop

- Clock이 상승할때 Master Flip-flop이 동작하고 clock이 하강할때 Slave Flip-flop이 동작하여 Master의 값을 받아 출력하는 Flip-flop



실습설명

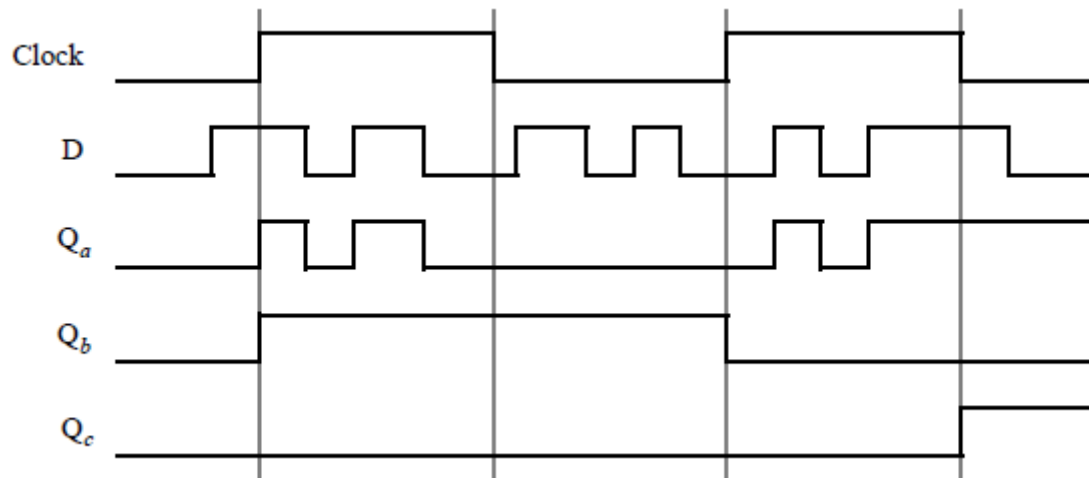
Part IV : Gated D-Latch, edge triggered D Flip-flop



실습설명

Part IV : Gated D-Latch, edge triggered D Flip-flop

Result : 시뮬레이션으로 확인



Part IV : Gated D-Latch, edge triggered D Flip-flop

■ Edge (Verilog)

☞ Verilog 의 'always' 문을 사용하여 Clock의 positive, negative edge를 실행함

- Positive edge -> posedge , Negative edge -> negedge

☞ always문은 논리회로의 동기화를 위해 사용됨

☞ 사용예

```
module pos_D_FF(D,Clk,Q);  
input D,Clk;  
output reg Q;  
always @(posedge Clk) begin  
Q=D;  
end  
endmodule
```