

LAB 6. Latch, Flip-Flop and Shift Register

2017 Fall Logic Design LAB

Department of Computer Science and Engineering

Seoul National University

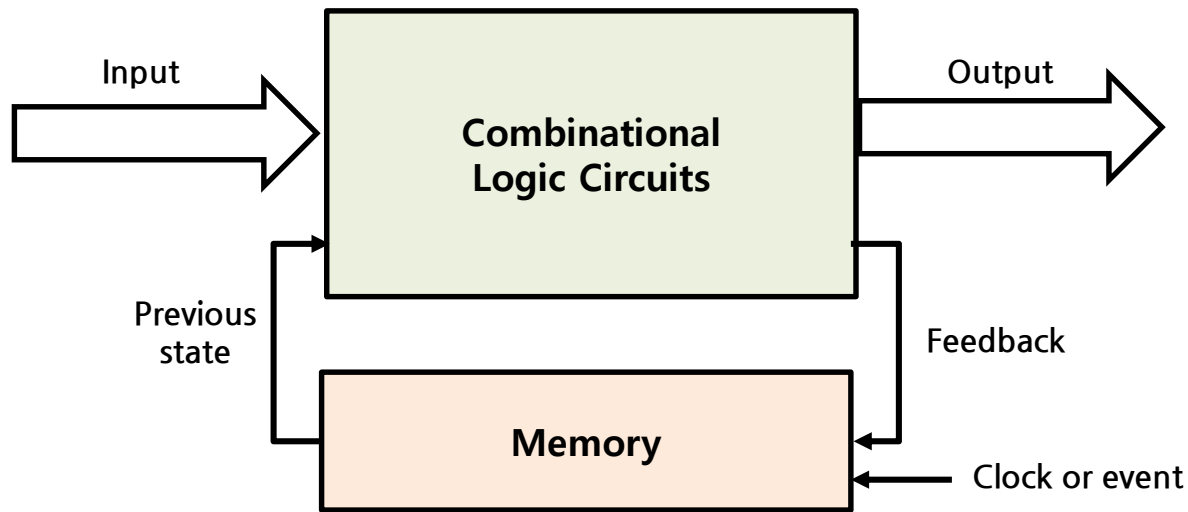
Outline

1. 순차 논리 회로 (Sequential Logic Circuits)
2. Latch 회로도 및 Verilog 구현
3. Flip-Flop 회로도 및 Verilog 구현
4. Shift Register 회로도 및 Verilog 구현

순차 논리 회로 (Sequential Logic Circuits)

- 순차 논리 회로란?

- 출력이 입력에 의해서만 결정되지 않고 기존에 들어왔던 입력과 출력에 영향을 받는 논리 회로

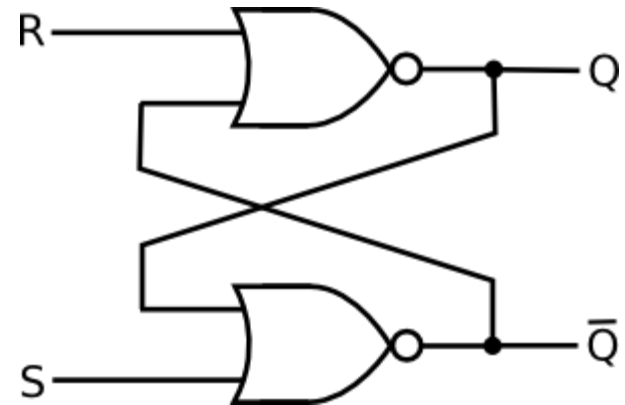


- 회로의 입력 및 출력, 내부 상태 등 저장하기 위해서는 bit를 저장할 수 있는 회로(storage element)가 필요
 - Latches, edge-triggered Flip-Flops
 - Registers

RS-Latch

- RS(Reset-Set) Latch Verilog 구현

```
module RS_Latch(  
    input R,  
    input S,  
    output Q,  
    output Qn  
);  
    //nor nor1(Q, R, Qn);  
    //nor nor2(Qn, S, Q);  
    assign Q = ~ (R | Qn);  
    assign Qn = ~ (S | Q);  
  
endmodule
```



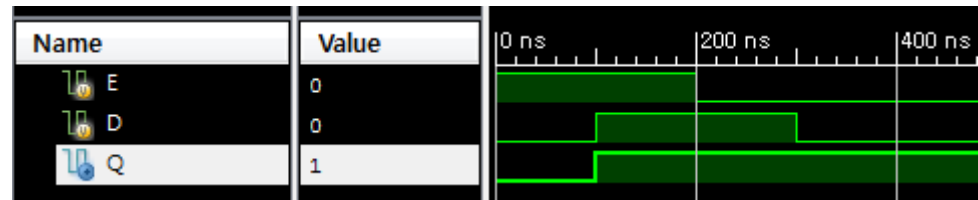
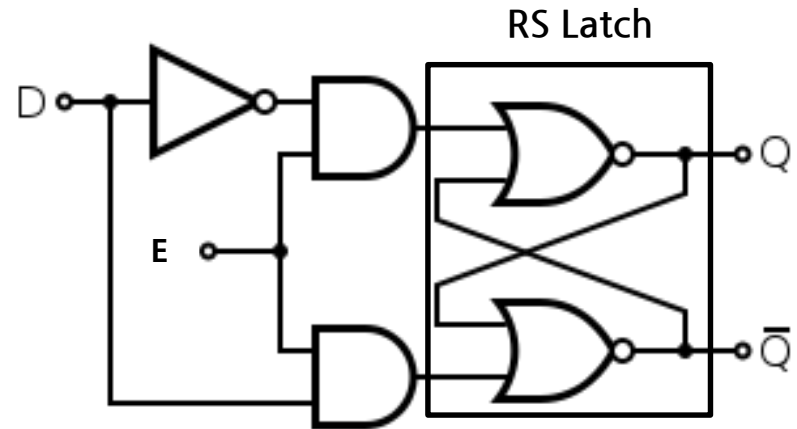
R	S	Q _{next}
0	0	Q
1	0	0
0	1	1
1	1	X

- Gated D Latch Verilog 구현

```

module D_Latch(
    input D,
    input E,
    output reg Q
);

always @(D or E)
begin
    if (E) Q <= D;
end
endmodule
    
```

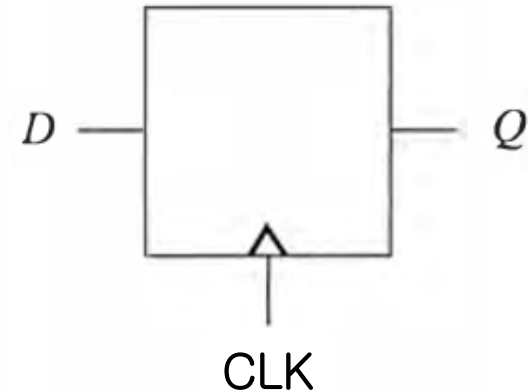


D	E	Q _{next}
0	1	0
1	1	1
X	0	Q

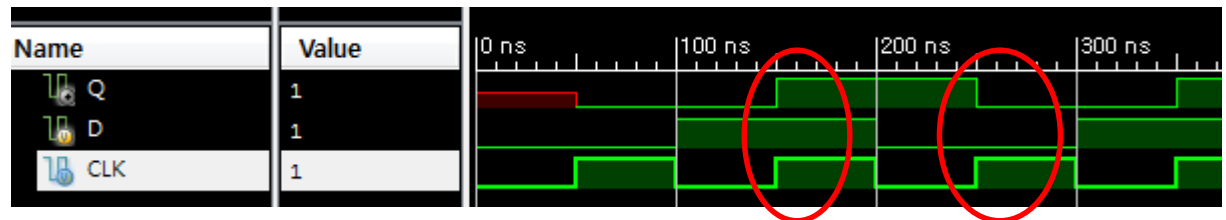
D Flip-Flop

- Positive edge-triggered D Flip-Flop Verilog 구현

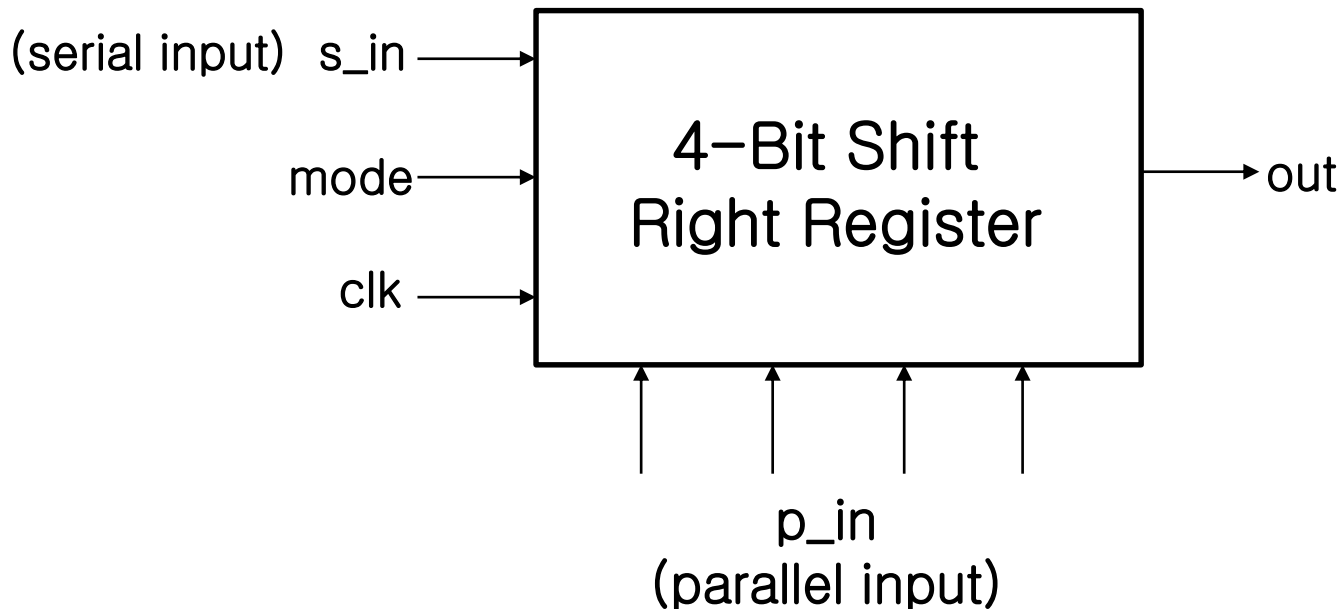
```
module D_FlipFlop(  
    input D,  
    input CLK,  
    output reg Q  
);  
  
always @ (posedge CLK)  
begin  
    Q <= D;  
end  
endmodule
```



D Flip-Flop의 schematic symbol



- Shift Register 란?
 - Flip-Flop으로 구성되며 데이터를 저장 및 각각 저장 소자 사이에서 shift 할 수 있는 register
- ex) 4-bit shift right register



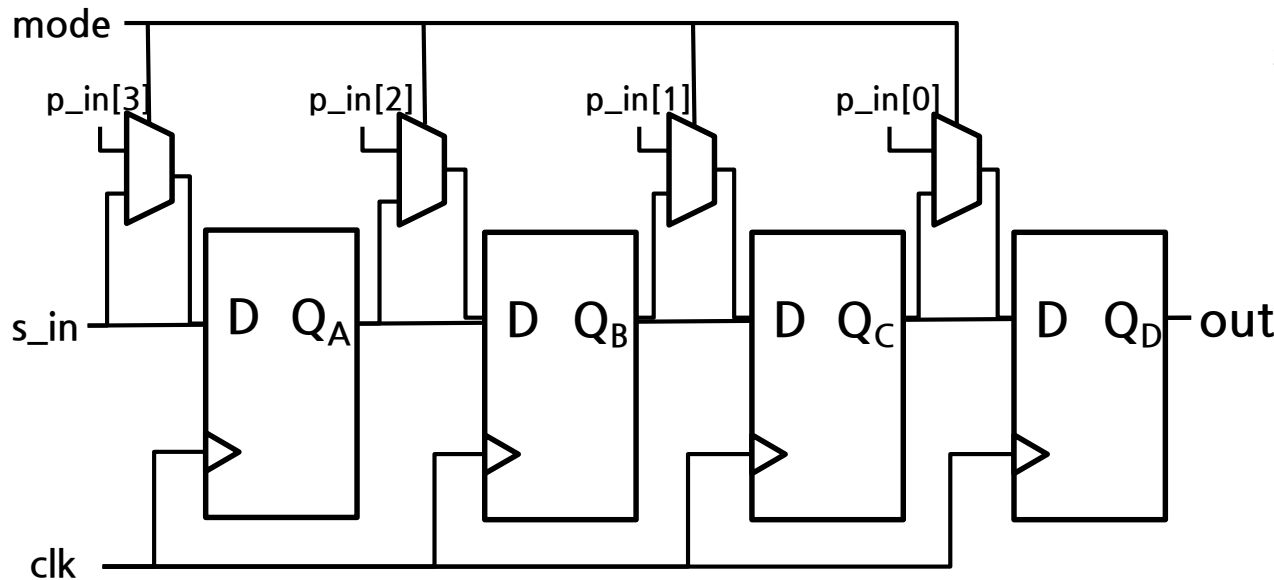
4-Bit Shift Right Register

[실습4]

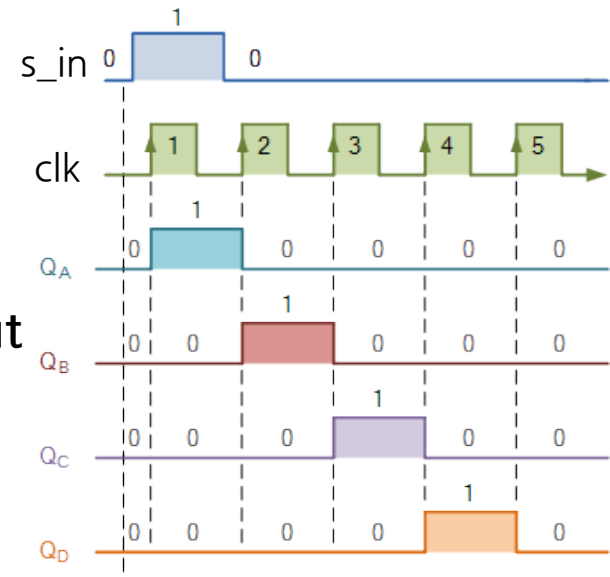
■ 기능

- Parallel input
- Serial input / output
- Shift one direction
- Positive edge-triggered

mode	operation
0	shift right
1	load new input



D Flip-Flop을 이용한 Shift Right Register



clk 에 따른 Shift Register 출력

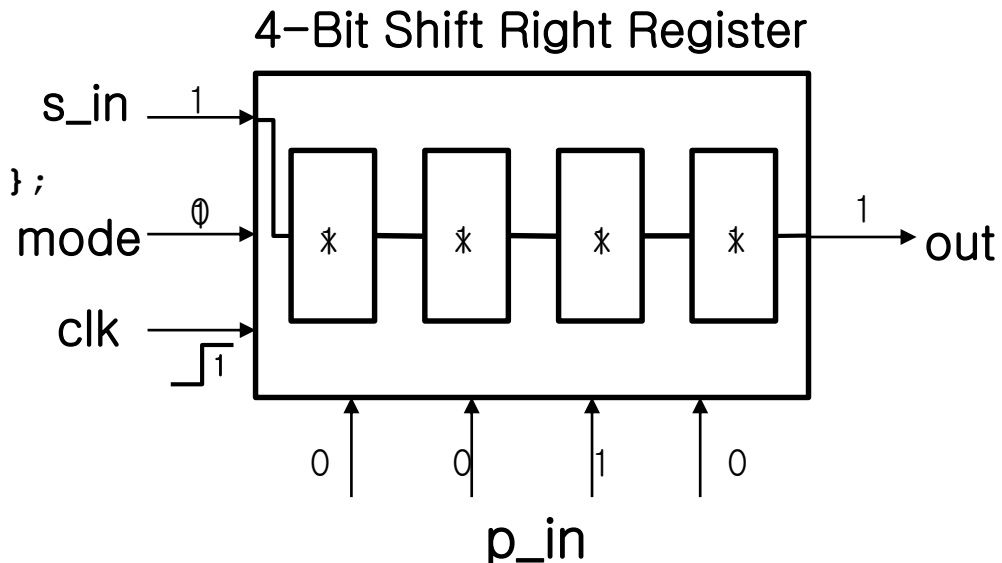
4-Bit Shift Right Register

[실습4]

■ Shift Right Register Verilog 구현

```
module shift_4bit(  
    input clk,  
    input s_in,  
    input mode,  
    input [3:0] p_in,  
    output reg out  
);  
reg [3:0] sreg;  
  
always @ (posedge clk)  
begin  
    if (mode == 1'b0)  
        begin  
            sreg <= {s_in, sreg[3:1]};  
        end  
    else  
        begin  
            sreg <= p_in;  
        end  
    out <= sreg[0];  
end  
endmodule
```

mode	operation
0	shift right
1	load new input



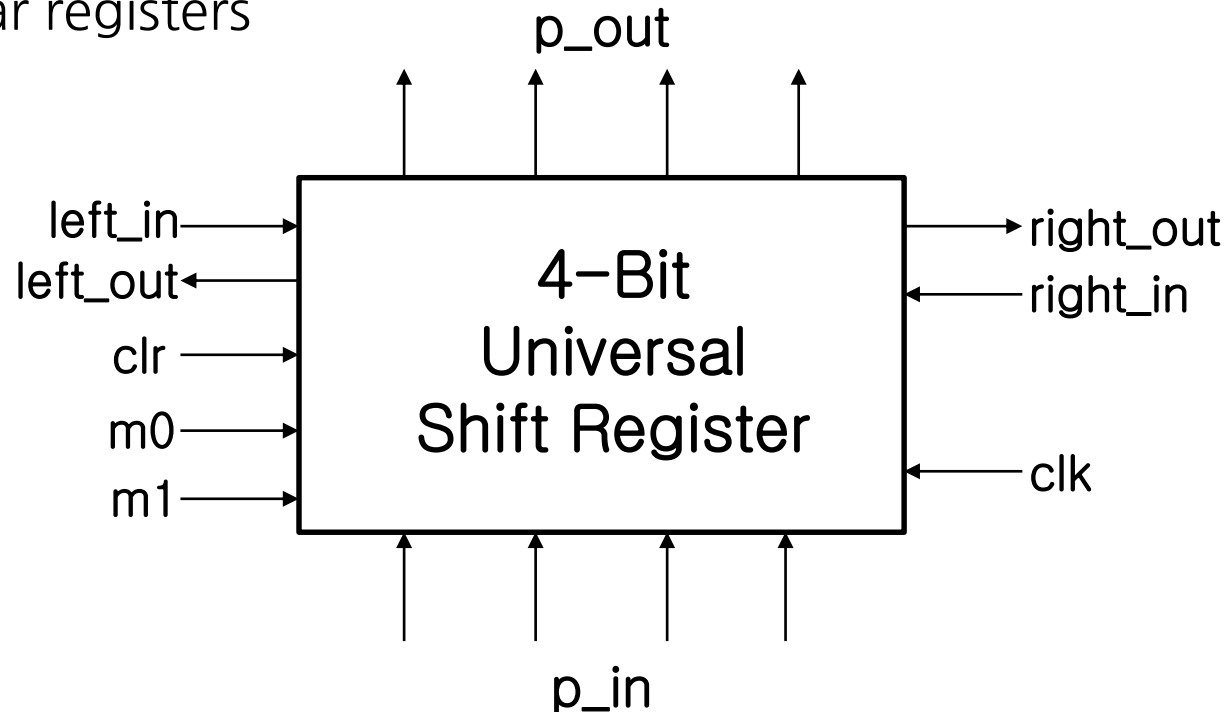
4-Bit Universal Shift Register

[실습4]

■ 기능

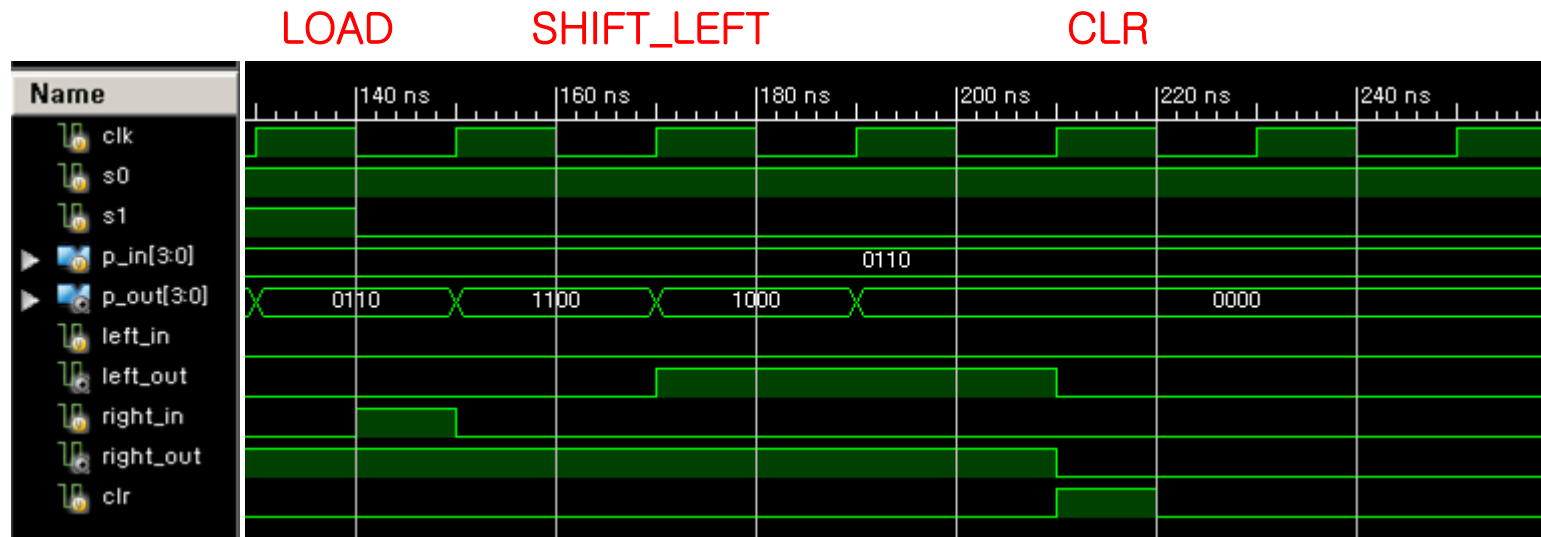
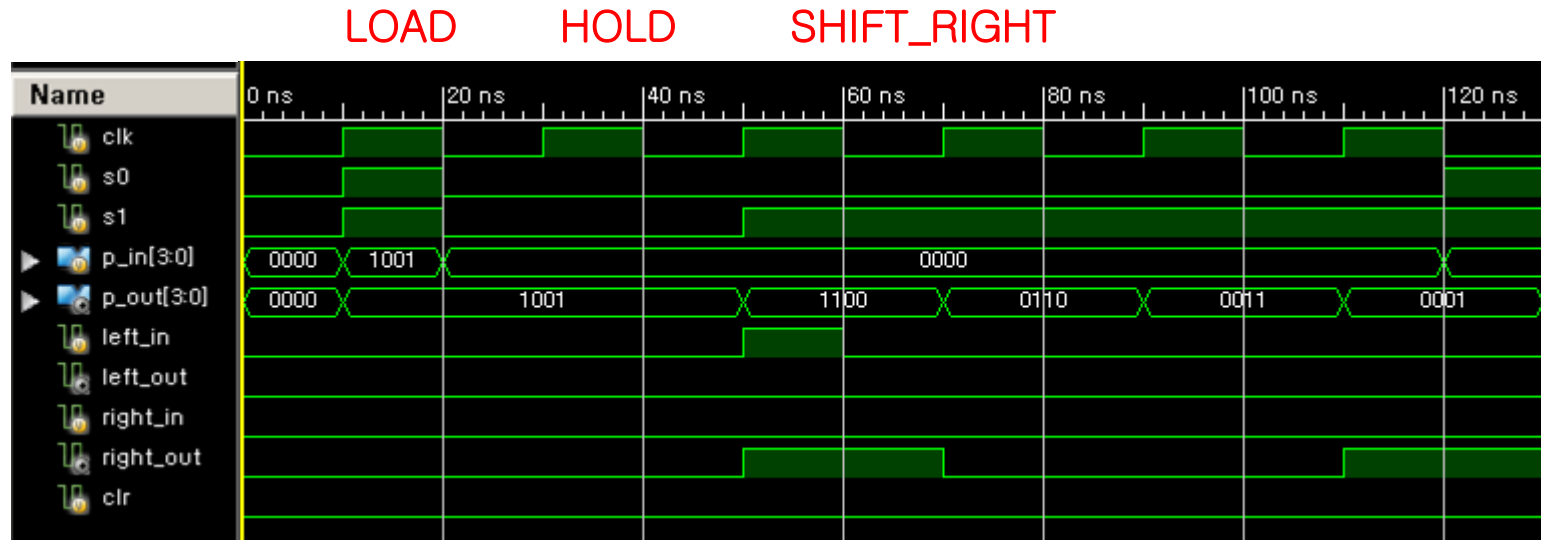
- Parallel input / output
- Serial input / output
- Shift left / right
- Clear registers

m0	m1	operation
0	0	hold state
0	1	shift right
1	0	shift left
1	1	load new input



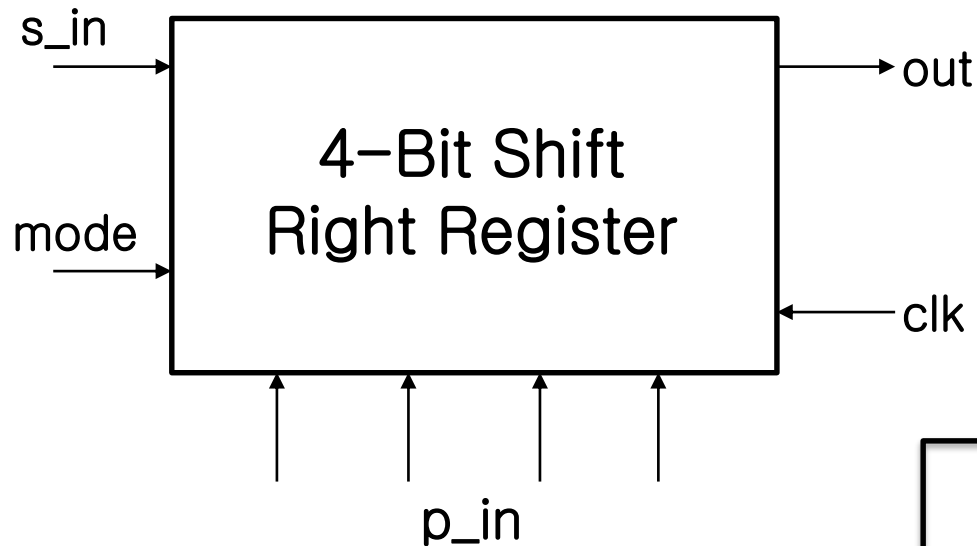
4-Bit Universal Shift Register 시뮬레이션

[실습4]



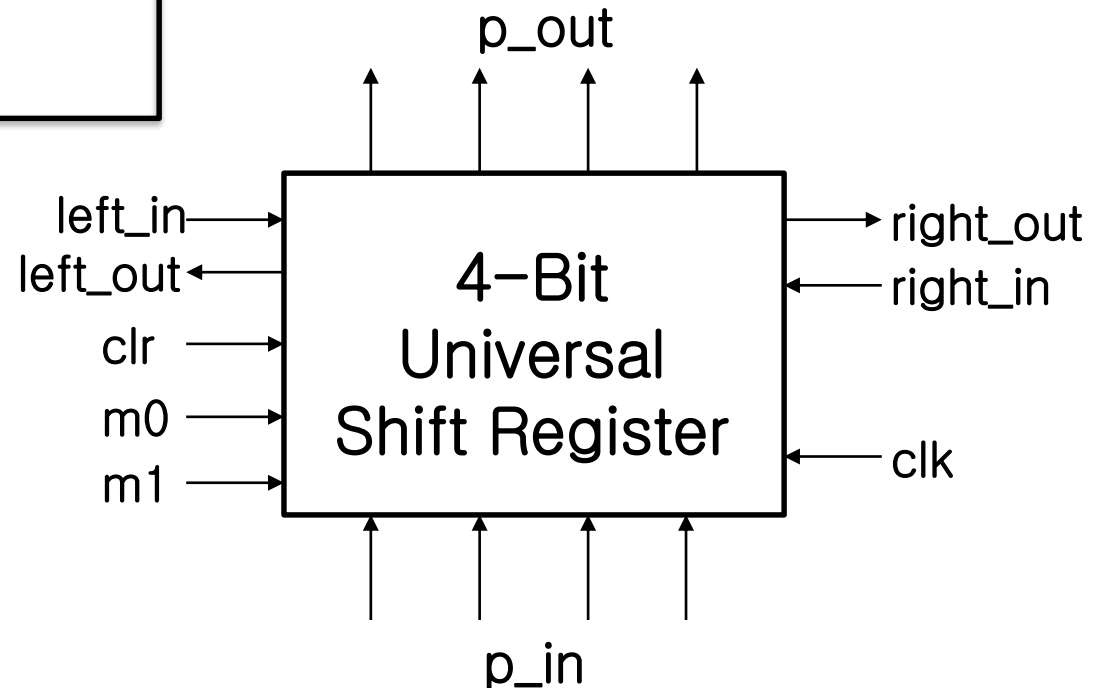
Shift Right Register / Universal Shift Register

[실습4]



mode	operation
0	shift right
1	load new input

m0	m1	operation
0	0	hold state
0	1	shift right
1	0	shift left
1	1	load new input



실습 1. RS Latch를 이용하여 Gated D Latch 구현

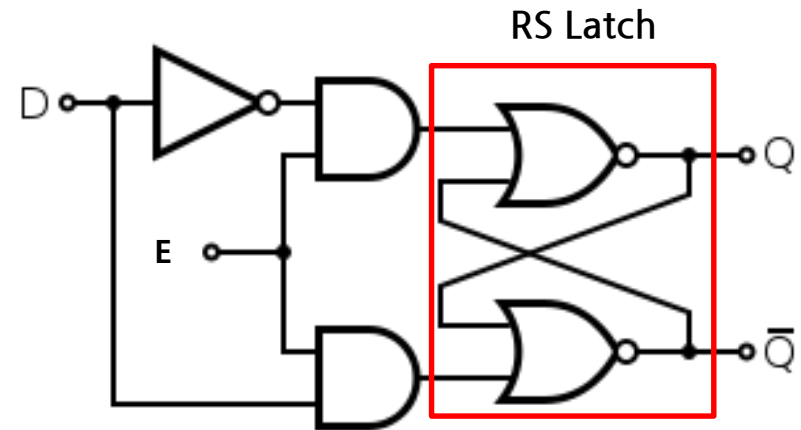
- 목표
 - RS Latch를 이용해서 Gated D Latch 구현
- 실습 내용
 - 먼저, 주어진 코드를 이용해 RS Latch를 구현한다.
 - RS Latch를 이용해서 Gated D Latch를 구현한다.
 - 시뮬레이션을 통해 동작이 예상한 결과와 같은 지 확인해본다.
- 제출 사항
 - 소스코드 및 시뮬레이션 결과 실습지 양식으로 제출

실습 1. RS Latch Verilog code 및 D Latch block diagram

- RS Latch Verilog code 및 RS Latch를 이용하여 D Latch를 구성한 모습

```
module RS_Latch(  
    input R,  
    input S,  
    output Q,  
    output Qn  
);  
  
nor nor1(Q, R, Qn);  
nor nor2(Qn, S, Q);  
  
endmodule
```

RS Latch Verilog code



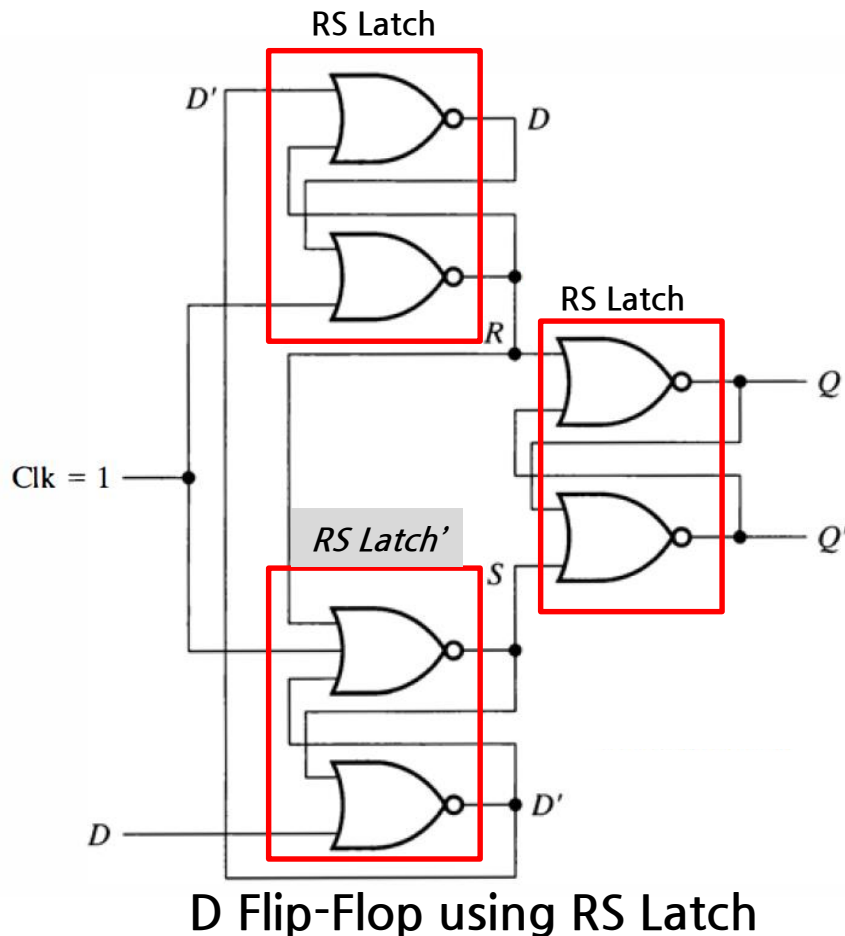
D Latch

실습 2. RS Latch를 이용하여 D Flip-Flop 구현

- 목표
 - RS Latch를 이용하여 D Flip-Flop 구현
- 실습 내용
 - 실습 1에서 구현한 RS Latch와 비슷하지만 input이 3개인 RS Latch'를 구현한다.
 - RS Latch 2개와 RS Latch'를 이용하여 negative edge-triggered D Flip-Flop을 구현해본다.
 - 시뮬레이션을 통해 동작이 예상한 결과와 같은 지 확인해본다.
- 제출 사항
 - 소스코드 및 시뮬레이션 결과 실습지 양식으로 제출

실습 2. RS Latch를 이용한 D Flip-Flop block diagram

- RS Latch 2개와 R 입력이 3-input nor gate인 RS Latch'를 이용하여 negative edge-triggered D Flip-Flop을 구성한 모습



실습 3. D Latch와 D Flip-Flop 차이 비교

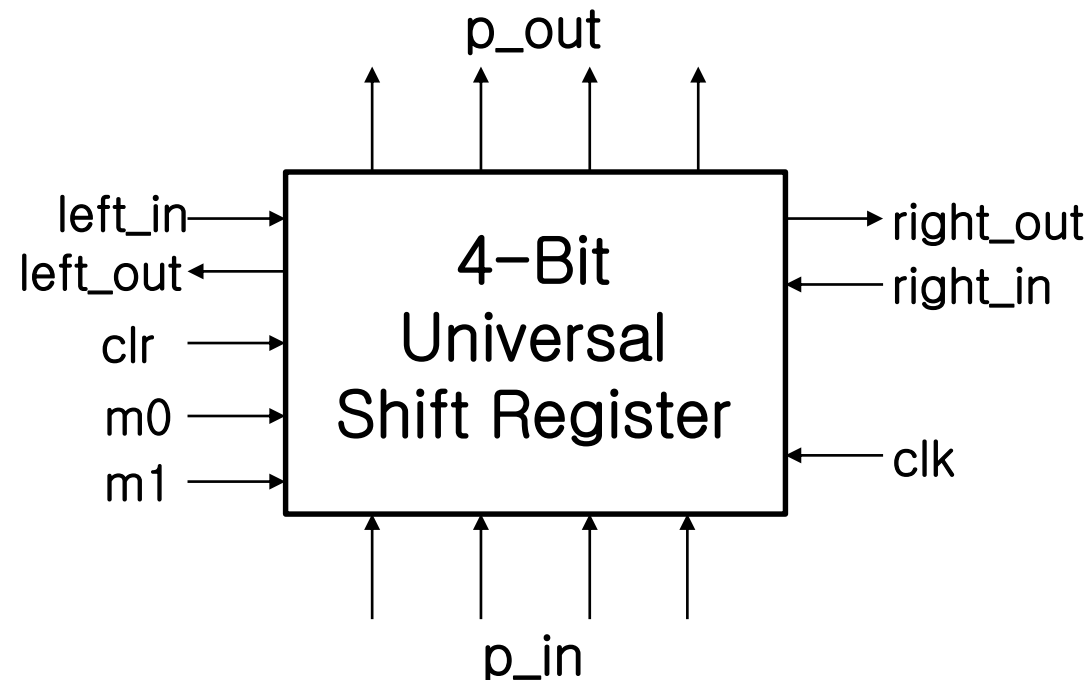
- 목표
 - 실습1과 2에서 구현해 본 D Latch와 D Flip-Flop의 차이를 비교
- 실습 내용
 - 구현한 D Latch와 D Flip-Flop의 차이를 확인해 볼 수 있도록 모듈을 설계한다.
 - 시뮬레이션을 통해 동작이 예상한 결과와 같은 지 확인해본다.
 - 구현한 D Flip-Flop은 negative edge-triggered 이므로, 모듈 내에서 CLK 입력을 invert 시켜 D Flip-Flop의 입력으로 만들어 비교해야 한다.
- 제출 사항
 - 소스코드 및 실습지에 제공된 입력을 사용하여 시뮬레이션한 결과

실습 4. 4-Bit Universal Shift Register 구현

- 목표
 - 현재 저장중인 상태를 출력하는 4-Bit Universal Shift Register 구현
- 실습 내용
 - 이전 슬라이드에서 다룬 내용을 바탕으로 4-Bit Universal Shift Register를 구현해본다.
 - 시뮬레이션을 통해 동작이 예상한 결과와 같은 지 확인해본다.
- 제출 사항
 - 소스코드 및 실습지에 제공된 입력을 사용하여 시뮬레이션한 결과

실습 4. 4-Bit Universal Shift Register에 대한 정보

- 현재 저장된 상태를 출력하는 4-Bit Universal Shift Register의 block diagram과 4가지 operation에 대한 control signal encoding



4-Bit Universal Shift Register

m0	m1	operation
0	0	hold state
0	1	shift right
1	0	shift left
1	1	load new input

Control signal encoding

실습 제출 안내

- 제출 항목

- 실습지 참고하여 보고서 작성
- 각 실습에서 작성한 Verilog 소스 코드

- 제출 방법 및 기한

- 작성한 보고서(PDF) 및 소스코드를 압축하여 하나의 파일로 제출
- ETL 과제 게시판에 **팀 별로 제출**
- **일요일 오후 6시**까지