Module shift8

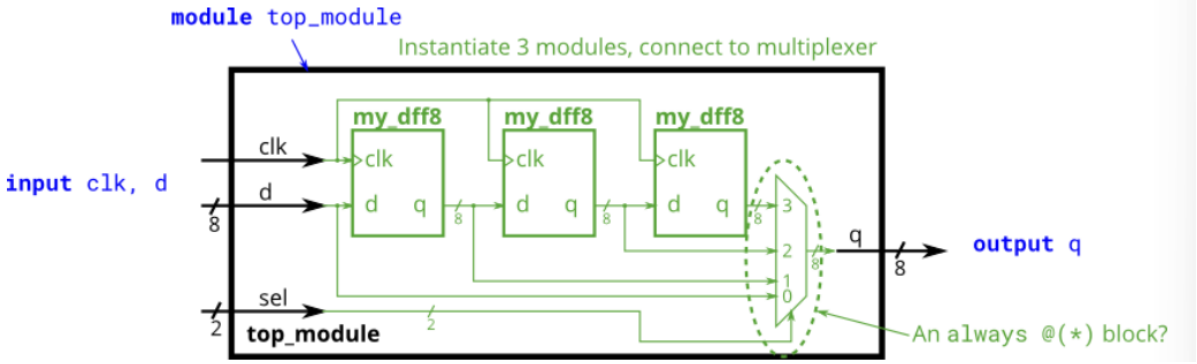
module_shift Previous Next module_add

This exercise is an extension of module_shift. Instead of module ports being only single pins, we now have modules with vectors as ports, to which you will attach wire vectors instead of plain wires. Like everywhere else in Verilog, the vector length of the port does not have to match the wire connecting to it, but this will cause zero-padding or trucation of the vector. This exercise does not use connections with mismatched vector lengths.

You are given a module `my_dff8` with two inputs and one output (that implements a set of 8 D flip-flops). Instantiate three of them, then chain them together to make a 8-bit wide shift register of length 3. In addition, create a 4-to-1 multiplexer (not provided) that chooses what to output depending on `sel[1:0]`: The value at the input d, after the first, after the second, or after the third D flip-flop. (Essentially, `sel` selects how many cycles to delay the input, from zero to three clock cycles.)

The module provided to you is: `module my_dff8 ( input clk, input [7:0] d, output [7:0] q );`

The multiplexer is not provided. One possible way to write one is inside an `always` block with a `case` statement inside. (See also: mux9to1v)



설명

- 이전에 했던 module_shift의 확장
- 이 모듈은 벡터를 모듈로써 가지고 있다.
- 포트의 벡터 길이는 연결된 와이어와 일치할 필요가 없지만, 이렇게 한다면 벡터의 zero padding 혹은 절단이 발생할 것이다.

조건

- 두 개의 입력과 한 개의 출력이 있는 모듈이 주어진다.
- 그 중 세개를 인스턴스화 후 체인으로 연결하여 길이가 3인 8-bit 폭의

시프트 레지스터를 만든다.

- 출력할 내용을 선택하기 위한 4:1 MUX를 만든다(MUX를 쓸 수 있는 한 가지 방법은 블록 안에 always 문자을 넣는 것).

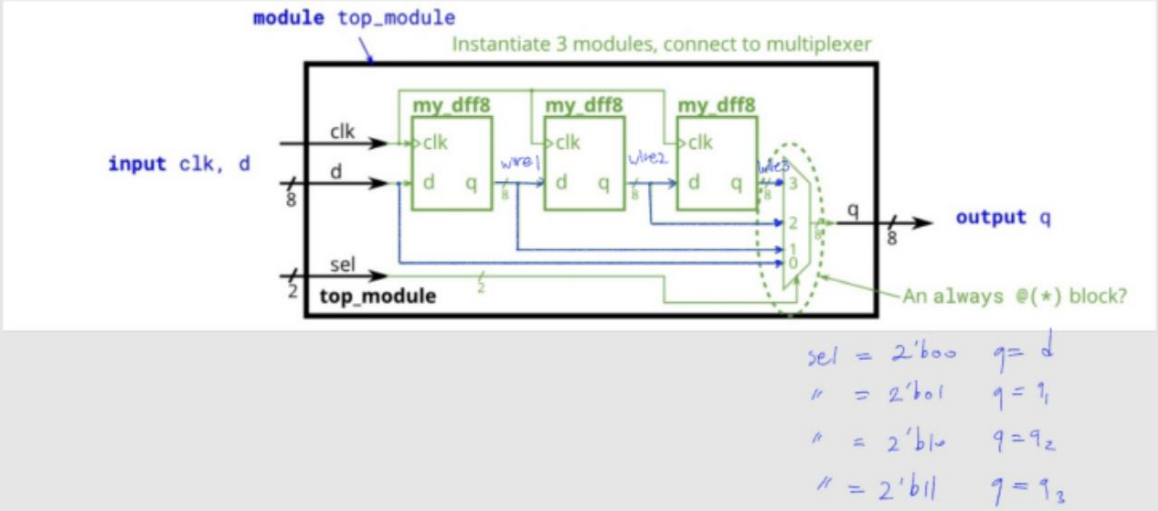
코드

5_Module shift8.v

```
1   module top_module (
2       input clk,
3       input [7:0] d,
4       input [1:0] sel,
5       output reg [7:0] q
6   );
7       wire [7:0] wire1, wire2, wire3;
8
9   // ( input clk, input [7:0] d, output [7:0] q );
10      my_dff8 U0 (.q(wire1), .clk(clk), .d(d));
11      my_dff8 U1 (.q(wire2), .clk(clk), .d(wire1));
12      my_dff8 U2 (.q(wire3), .clk(clk), .d(wire2));
13
14      always @(*) begin
15          case (sel)
16              0 : q = d;
17              1 : q = wire1;
18              2 : q = wire2;
19              3 : q = wire3;
20              default: q = 4'bx;
21          endcase
22      end
23
24  endmodule
```

코드 설명

- 이전에 풀었던 문제와 비슷하게 내부 wire를 3개 선언하여 연결하였다.



결과

module_shift8 — Compile and simulate

Running Quartus synthesis. Show Quartus messages...
Running ModelSim simulation. Show Modelsim messages...

Status: Success!

You have solved 24 problems. See my progress...

Timing diagrams for selected test cases

These are timing diagrams from some of the test cases we used. They may help you debug your circuit. The diagrams show inputs to the circuit, outputs from your circuit, and the expected reference outputs. The "Mismatch" trace shows which cycles your outputs don't match the reference outputs (0 = correct, 1 = incorrect).

Shift register