

Xilinx Zynq FPGA, TI DSP,
MCU 기반의
프로그래밍 및 회로 설계 전문가
과정

#77

학생 : 김시윤

강사 : Innova Lee(이 상훈)

Verilog 연습

저번에는 Verilog 를 익숙해지기 위해 D Flip Flop 을 설계해보았다.

이번에는 베릴로그를 이용하여 주파수 분주를 하고, 분주한 주파수로 4 비트 업 2 진카운터를 설계하여 LED 를 출력하는 것을 목표로 한다.

목표를 잡고 만드느라 고생을 많이했다.

패키지를 만들지 않고 한개의 디바이스에서 이 동작들을 하게 하고 싶었는데 결국 실패하였다.

그래서 잘못된 예를 먼저 적어보도록 한다.

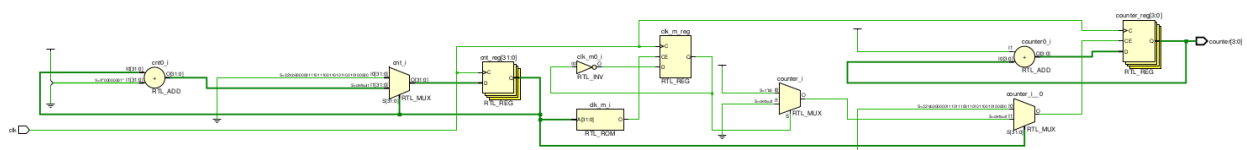
-잘못된 예시

```
module (clk,counter);  
  
    input clk;  
    output reg [3:0] counter;  
  
    reg clk_m;  
    integer cnt=0;  
  
    always @(posedge clk)    begin  
  
        cnt <=cnt +1;  
  
        if(cnt == 62500000) begin  
            clk_m <= ~clk_m;  
            cnt <=0;  
        end  
        else if(~clk_m) begin  
            counter <= counter + 1;  
        end  
    end  
end  
endmodule
```

위에서 목표한 것과 같은 결과가 아닌 순서가 엉킨 결과가 나왔다.

그이유를 생각해보면 우선 clk 문제 같다. 그리고 wire 로 연결해 줘야 하는데 reg 로 값을 저장해 인식해서 잘못된 결과가 나오는것 같다.

합성을 시킨 후 RTL 스케메틱을 확인하면 다음과 같은 이상한 논리도가 나온다.



그래서 패키지를 시켜 디바이스 두개를 만들어 연동시키기로 결정하였다.

-패키지 설정.

패키지 안에 디바이스는 총 2 개가 있다.

1. 클럭을 분주해주는 PLL 디바이스.
2. 4 비트 2 진 업 카운터 (카운터의 출력은 LED 에 묶음)

1. make_pll.v

```
module make_pll(clk_50MHz, clk_1Hz);

input clk_50MHz;
output clk_1Hz;

reg clk_pll;

reg [31:0]counter;

always @(posedge clk_50MHz)
begin
    if(counter == 32'd62499999)
        begin
            clk_pll <= ~clk_pll;
            counter <= 32'b0;
        end
    else
        counter <= counter + 32'b1;
    end
    assign clk_1Hz = clk_pll;
endmodule
```

2, counterled.v

```
module counterled(clk_counter,count);

input clk_counter;
output reg [3:0] count;

always @(posedge clk_counter) begin

    count <= count + 4'b1;
end

endmodule
```

3.topmodule.v

```

module topmodule(input wire top_50MHz, output wire [3:0] top_count);

wire top_1Hz;

make_pll(.clk_50MHz(top_50MHz),.clk_1Hz(top_1Hz));
counterled(.clk_counter(top_1Hz),.count(top_count));

endmodule

```



여기서 클럭을 분주해주는 counter 를 62499999 로 한 이유는
1 초에 1 씩 증가를 시키기위해 동작 클럭이 1Hz 짜리 클럭이 필요하기 때문이다.\

우리가 사용하는 보드는 125MHz 의 클럭을 갖고있으며 그 클럭은 L16 포트에 설정이 되어있다.
여기서 듀티비가 50 인클럭을 만들어주면 위와같은 카운터 값이 나온다.

출력사진은 0,1,2,3 만 찍어서 올렸다.

