

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

#78

학생 : 김시윤

강사 : Innova Lee(이 상훈)

## Verilog 연습 - PWM

어제는 주파수 분주를 통해 업카운터를 설계하여 LED 를 제어하였다.

오늘은 주파수 분주를 통해 LED PWM 제어를 하고자 한다.

오늘의 문제를 먼저 정리하고 최종 완성된 베릴로그를 정리하고자 한다.

오늘의 문제는 베릴로그 배열선언이었다. 잘못된 배열선언인지 잘 모르겠다.

인터넷 쿼터스 베릴로그 자료로 하였는데 비바도에서는 인식이 안됐다.

그래서 배열을 지우고 변수의 값을 계속적으로 바꿔주는식으로 구현하였다.

잘못된 배열부터 적어본다.

선언

```
reg [17:0]pwm_cnt[5:0];
```

초기값 설정

```
initial begin
```

```
pwm_cnt[0] = 18'd0;  
pwm_cnt[1] = 18'd3000;  
pwm_cnt[2] = 18'd6000;  
pwm_cnt[3] = 18'd9000;  
pwm_cnt[4] = 18'd12000;  
pwm_cnt[5] = 18'd15000;  
end
```

조건문

```
if(cnt == pwm_cnt[i])  
이런식으로 했는데 배열을 인식하지 못하였다.  
뭐가 문제인지는 잘 모르겠다.  
그래서 배열을 포기하고 변수로 하였다.
```

## 최종 소스코드

### make\_180KHz.v

```
module make180KHz(clk_125MHz,clk_180KHz);

input clk_125MHz;
output clk_180KHz;
reg clk_pll;
reg [8:0]cnt;

initial cnt =0;

always @(posedge clk_125MHz) begin

    if(cnt == 9'd346) begin
        clk_pll <= ~clk_pll;
        cnt <= 9'b0;
    end
    else
        cnt = cnt + 9'b1;

end

assign clk_180KHz = clk_pll;

endmodule
```

### pwm\_make.v

```
module pwm_make(clk,sig_pwm);

input clk;
output reg sig_pwm;

reg [17:0] cnt;
reg [17:0] i;
//reg [31:0] max_cnt;
//integer max_cnt = 180000;

//
always @(posedge clk) begin
    // max_cnt <= 32'd180000;

    if(cnt == 17999)
        begin
```

```

        i = i+18'd200;
        cnt <= 18'b0;
        if(i >=18'd17999) begin
            i = 18'b0;
            end
        end

        else if(cnt <= i) begin
            sig_pwm = 1;
            end

        else begin
            sig_pwm = 0;
            end

        cnt = cnt + 18'b1;
    end

endmodule

```

topmodule.v

```

module topmodule(input wire orig_125MHz,output wire pwm_signal );

wire mod_180KHz;

make180KHz(.clk_125MHz(orig_125MHz),.clk_180KHz(mod_180KHz));
pwm_make(.clk(mod_180KHz),.sig_pwm(pwm_signal));

endmodule

```

