

4190.301A Hardware System Design  
Spring 2020

# Hardware System Design Lab8 Report

Kim Bochang  
2014-16757

## 1. <lab7> Introduce

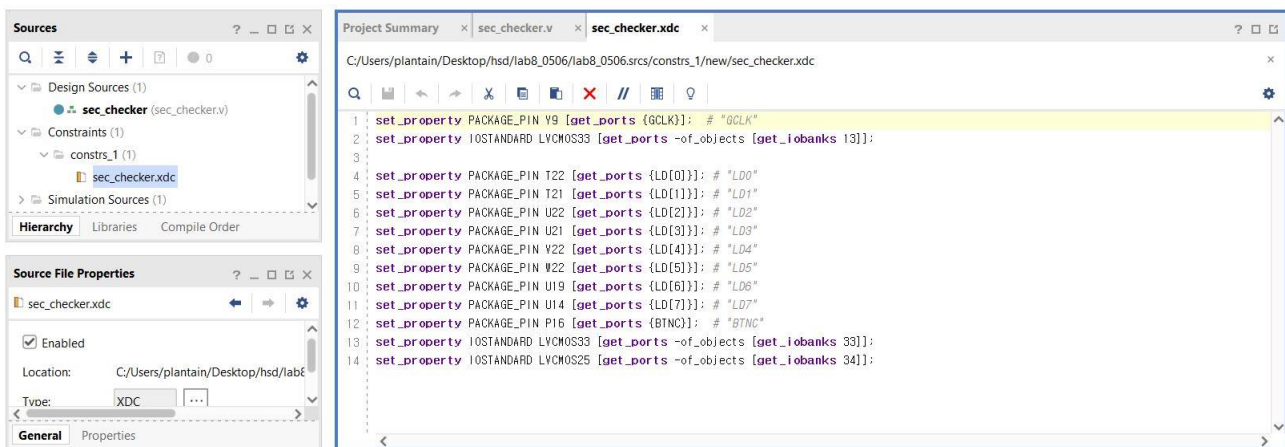
Lab 8의 목적은 주어진 zedboard에 직접 간단한 회로를 구현하면서 zedboard의 사용법을 익히는 것이다.

## 2. Implementation

### 2.1 lab8\_0506.srscs/constrs\_1/new/sec\_checker.xdc

먼저, down counter를 이용한 1sec\_checker를 구현하기 전에, zedboard의 어떤 부분을 사용할 것인지 지정해 주어야 한다.

이러한 부분을 지정하는것은 xdc파일을 통해, constraint를 지정하는것으로 해결할 수 있다.



<sec\_checker.xdc>

이는 위와 같이 구현하였다.

우리는 zedboard 하단의 LED 8개와 zedboard의 clk 생성기, 그리고 center button을 사용해야 하는데, 배포된 ZedBoard\_HW\_UG\_v2\_2.pdf의 4페이지를 보면, 각 부분의 이름과 어떤 bank에 물려 있는지가 나와있다. 같은 문서에서, 각 부분을 사용하려면 어떤 PIN을 지정해야 하는지도 나와있기 때문에 이를 보고 constraint를 위와 같이 지정해 주었다.

먼저, GCLK(클럭 생성기)는 PIN Y9에 물려있으므로 이를 설정해주고,

클럭 생성기가 있는 bank 13에는 3.3V의 전압을 주도록 되어있으므로 전압을 위와 같이 설정하였다.

그 외의 LED는 튜토리얼에서 진행했던 대로 설정해 주었고, BTNC(center button) 역시 문서를 참고하여 PIN위치를 알아낼 수 있었고, 이를 그대로 사용하였다.

이렇게 지정한 constraint를 통해, 이제 verilog module에서 이러한 포트들을 실제 input/output으로 쓸 수 있게 된다.

## 2.2 lab8\_0506.srscs/sources\_1/new/sec\_checker.v

constraint 설정을 끝냈으면, 이제 1sec-checker를 실제로 모듈로 구현할 차례다.

우리의 clk generator의 작동 주기가 100MHZ이므로, 1초당 100만번 \* 100 = 1억번의 클록을 발생시키게 된다.

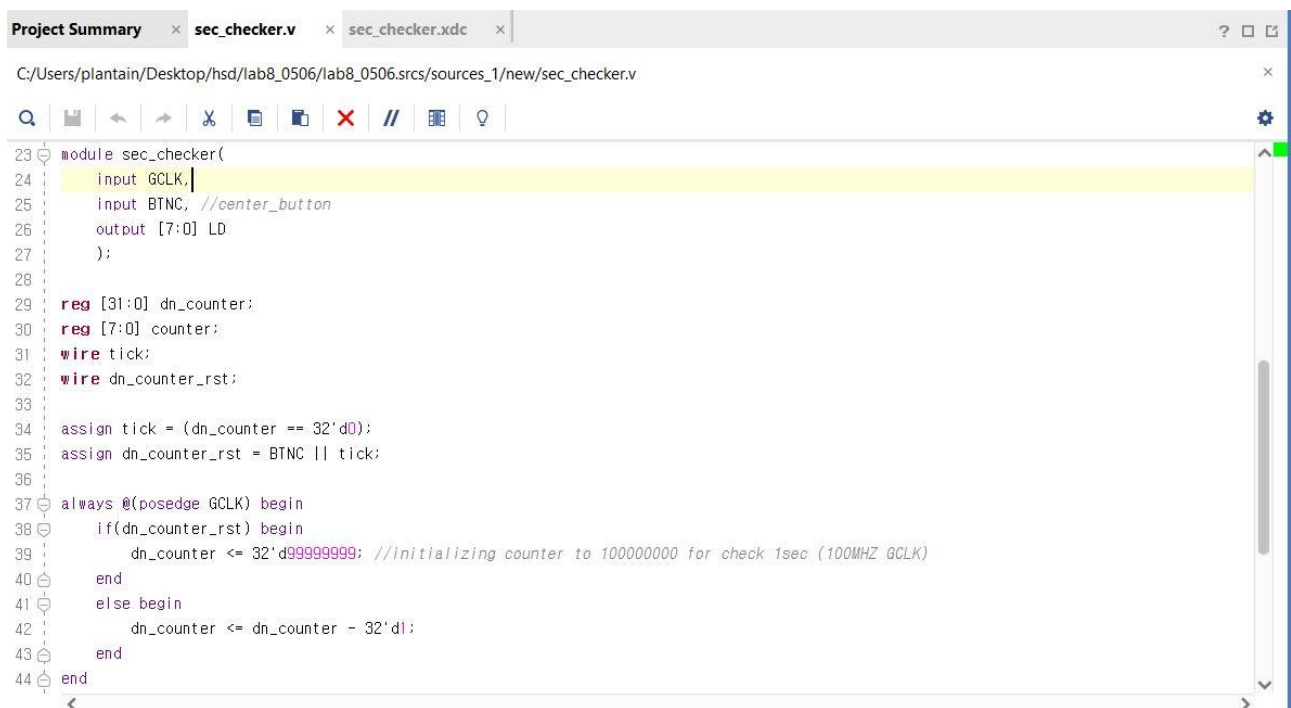
따라서, 1억번의 클록마다 1번씩의 tick을 발생시킬 수 있도록 down counter를 설계해야한다.

1억을 세어야 하므로,보편적인 int32를 저장할 수 있도록 32비트 down counter로 구현하였다.

또한, 이러한 tick이 들어왔을 때 LED의 입력으로 사용할 (출력할) up-counter가 하나 더 필요하므로,

내부적으로는 counter를 총 2개 가지게 된다. 출력이 8비트이므로 up-counter는 8비트로 구현하였다.

이는 다음과 같이 구현하였다.



```
23 module sec_checker(
24     input GCLK,
25     input BTNC, //center_button
26     output [7:0] LD
27 );
28
29 reg [31:0] dn_counter;
30 reg [7:0] counter;
31 wire tick;
32 wire dn_counter_rst;
33
34 assign tick = (dn_counter == 32'd0);
35 assign dn_counter_rst = BTNC || tick;
36
37 always @(posedge GCLK) begin
38     if(dn_counter_rst) begin
39         dn_counter <= 32'd99999999; //initializing counter to 100000000 for check 1sec (100MHZ GCLK)
40     end
41     else begin
42         dn_counter <= dn_counter - 32'd1;
43     end
44 end
```

먼저 down counter에 대한 부분이다. center button이 눌렸을 때와 tick이 발생할 때, counter를 1억 - 1로 초기화 하였고, down counter가 0일때 tick이 1이되도록 하여, 1초마다 tick이 발생하도록 하였다.

(1억-1 이 아닌, 1억으로 초기화하면 tick의 발생 주기가 1초가 아니라, 1초보다 약간 더 길어지게 된다.)

up counter에 대한 부분은 다음과 같다.

```

45 :
46 ⊖ always @(posedge GCLK) begin
47 ⊖     if(BTNC) begin
48 :         counter <= 8'd0; //initializing counter to 0.
49 ⊖     end
50 ⊖     else if(tick) begin
51 :         counter <= counter + 8'd1;
52 ⊖     end
53 ⊖ end
54 :
55 :     assign LD = counter;
56 ⊖ endmodule
57 :

```

tick이 1일때마다 작동하고, 그 외에는 center button이 눌렸을 때만 0으로 초기화를 하게 된다.

### 3. Result & Discussion

같은 폴더의 sec\_counter.avi (혹은 mp4)에 실행 결과가 들어있다.

mp4파일이 원본이고, 코덱 문제를 우려하여 avi로 변환까지 한 것을 같이 첨부하였다.

필요하지는 않지만 튜토리얼에 대한 동영상은 tutorial.avi로 첨부되어 있다.

### 4. Conclusion

실제 하드웨어를 사용하는 첫 실습이다. zedboard를 이용하여 우리가 지금까지 구현했던 verilog 코드가 보드에서 작동하는 모습을 보니 감회가 새로웠다.

비록 간단한 실습이었지만 실제 하드웨어가 돌아가는 모습을 볼 수 있었던 뜻 깊은 시간이었던 것 같다.