

# MINI INTERN

## 중간보고서

---

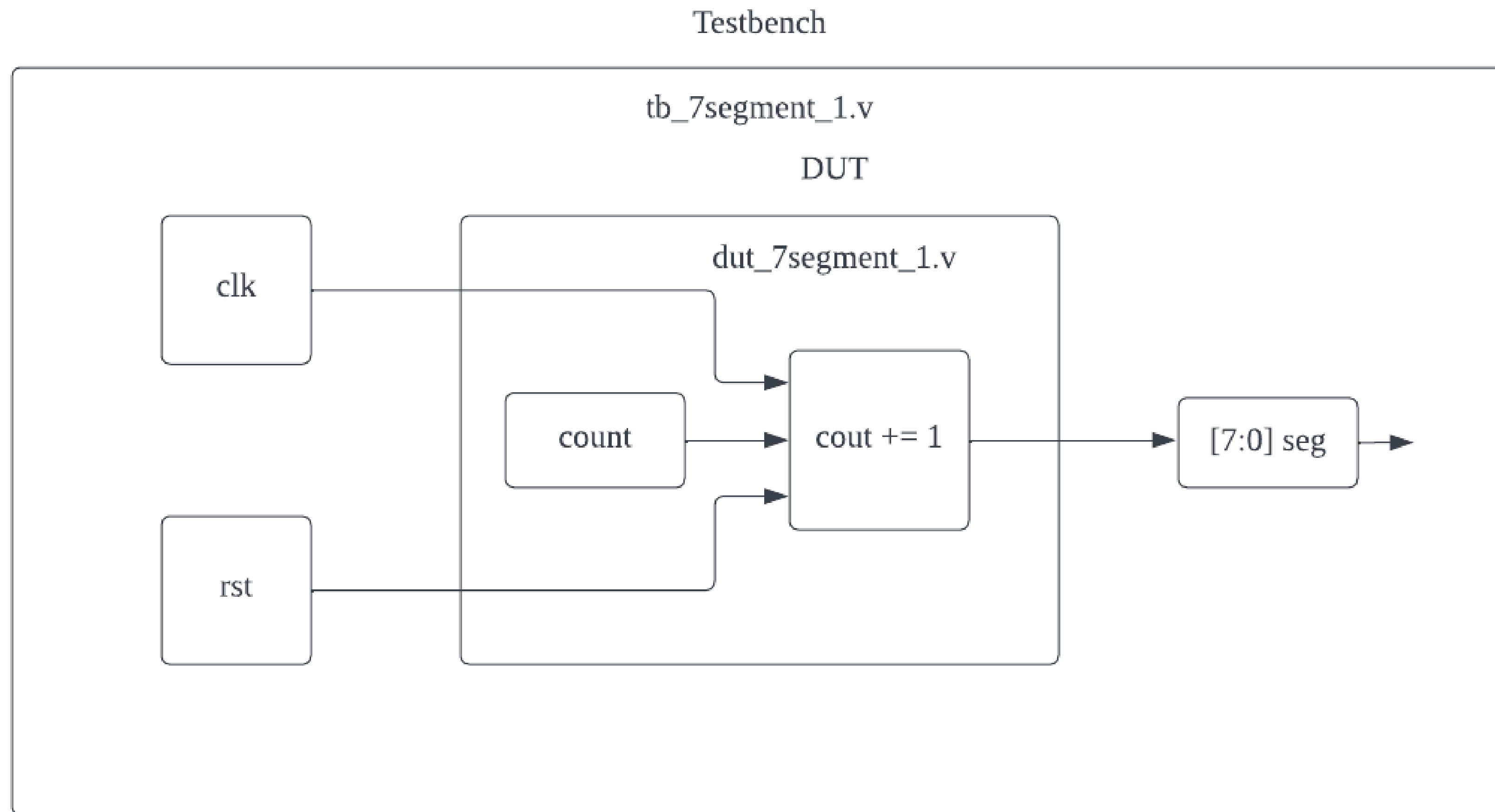
기업명: TSN LAB

작성자: 노지훈

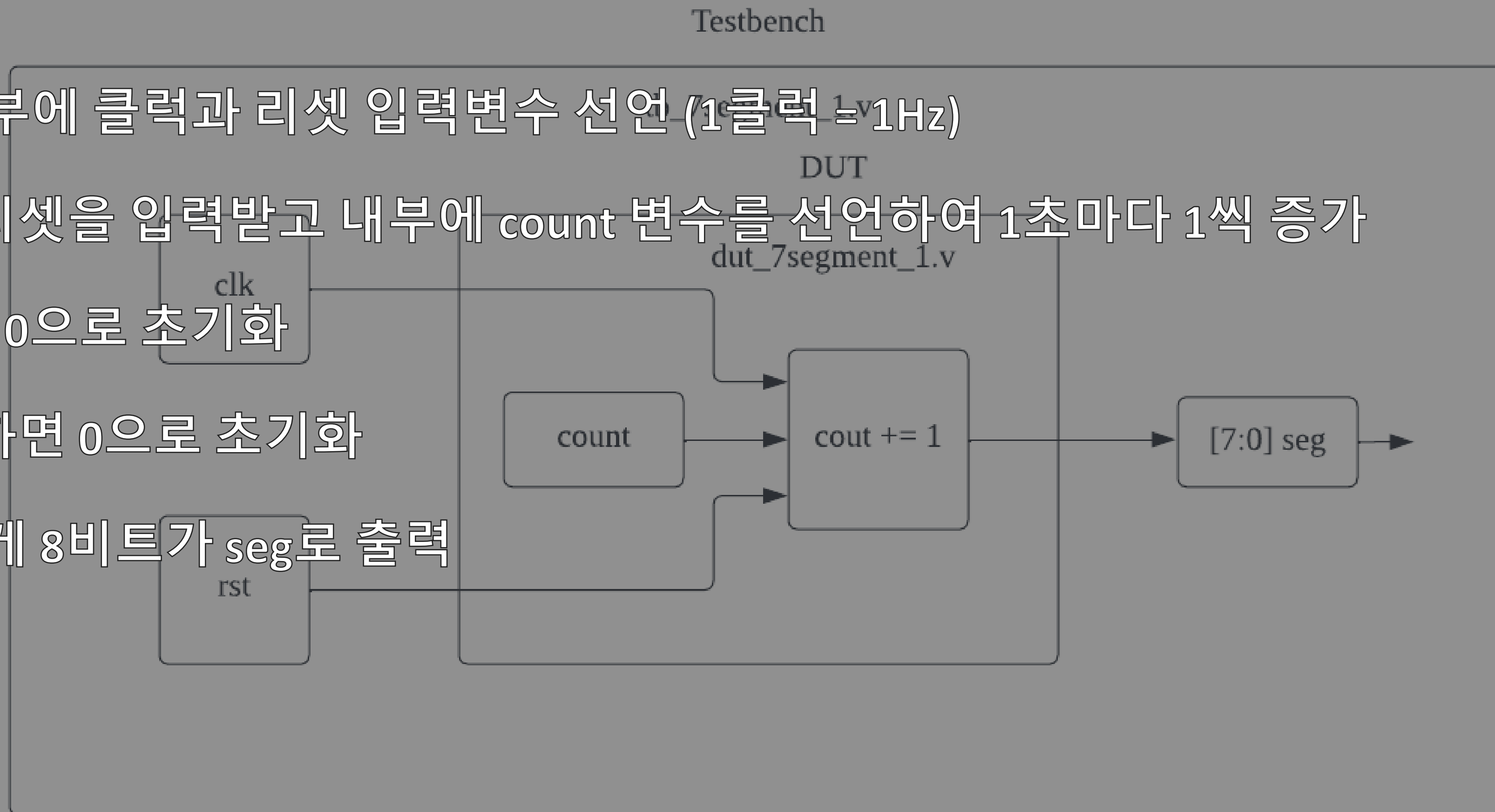


## 본과제: 7 SEGMENT 구현하기

- ▶ 7 segment 한 자리 표기하기 (0초부터 9초까지)
- ▶ 9초를 넘으면 0초부터 시작
- ▶ 7 segment가 없으면? -> GPIO로 구현
- ▶ 보드가 없으면? -> Simulation 으로 구현 (해당 0)



- ▶ Testbench 내부에 클럭과 리셋 입력변수 선언 (1클럭 = 1Hz)
- ▶ DUT 클럭과 리셋을 입력받고 내부에 count 변수를 선언하여 1초마다 1씩 증가
- ▶ 9초가 지나면 0으로 초기화
- ▶ 리셋이 동작하면 0으로 초기화
- ▶ 각 숫자에 맞게 8비트가 seg로 출력



# TESTBENCH

- ▶ segment의 dot부분도 포함하여 8비트의 출력 설정
- ▶ 처음 20초는 타이머 동작
- ▶ 그 후 20초는 리셋에 의한 0으로 초기화
- ▶ 그 후 20초는 다시 타이머 동작 후 종료

```
8 // 현실시간과 매칭을 위해 1초 단위로 설정
9 `timescale 1s/1ms
10
11 module tb_7segment;
12 reg clk;
13 reg rst;
14 wire [7:0] seg;
15
16 // 1Hz 클럭 생성
17 always
18 | #0.5 clk = ~clk;
19 initial begin
20 | // 초기값 생성
21 | clk = 0;
22 | rst = 0;
23 #20
24 | // 리셋 동작 확인
25 | rst = 1;
26 #20
27 | // 재동작 확인
28 | rst = 0;
29 #20
30 $finish;
31 end
32
33 // dut 파일 연결
34 dut_7segment DUT(
35 | .clk (clk),
36 | .rst (rst),
37 | .seg (seg)
38 );
39 endmodule
```

## DUT -1

- ▶ Testbench와 마찬가지로 timescale 및 변수 선언
- ▶ 내부에서 사용할 count와 s 변수를 int와 reg로 선언
- ▶ 리셋이 1이면  
: count = 0으로 선언하여 초기화
- ▶ 리셋이 0이면  
: count는 1클럭(상승 엣지)마다 1씩 증가
- ▶ 내부 count: 상승 에지, segment: 하강 에지에서 출력  
-> count가 10이 아닌 9일때 0으로 초기화

```
8 // 현실시간과 매칭을 위해 1초 단위로 설정
9 `timescale 1s/1ms
10
11 module dut_7segment(
12     input clk,
13     input rst,
14     output [7:0] seg
15 );
16     integer count = 0;
17     reg [7:0] s;
18
19 // 리셋 여부에 대한 내부 count 동작 선언
20 always @ (posedge clk) begin
21     if (rst == 1) begin
22         count = 0;
23     end else begin
24         if (count == 9) begin
25             count = 0;
26         end else begin
27             count = count + 1;
28         end
29     end
30 end
```

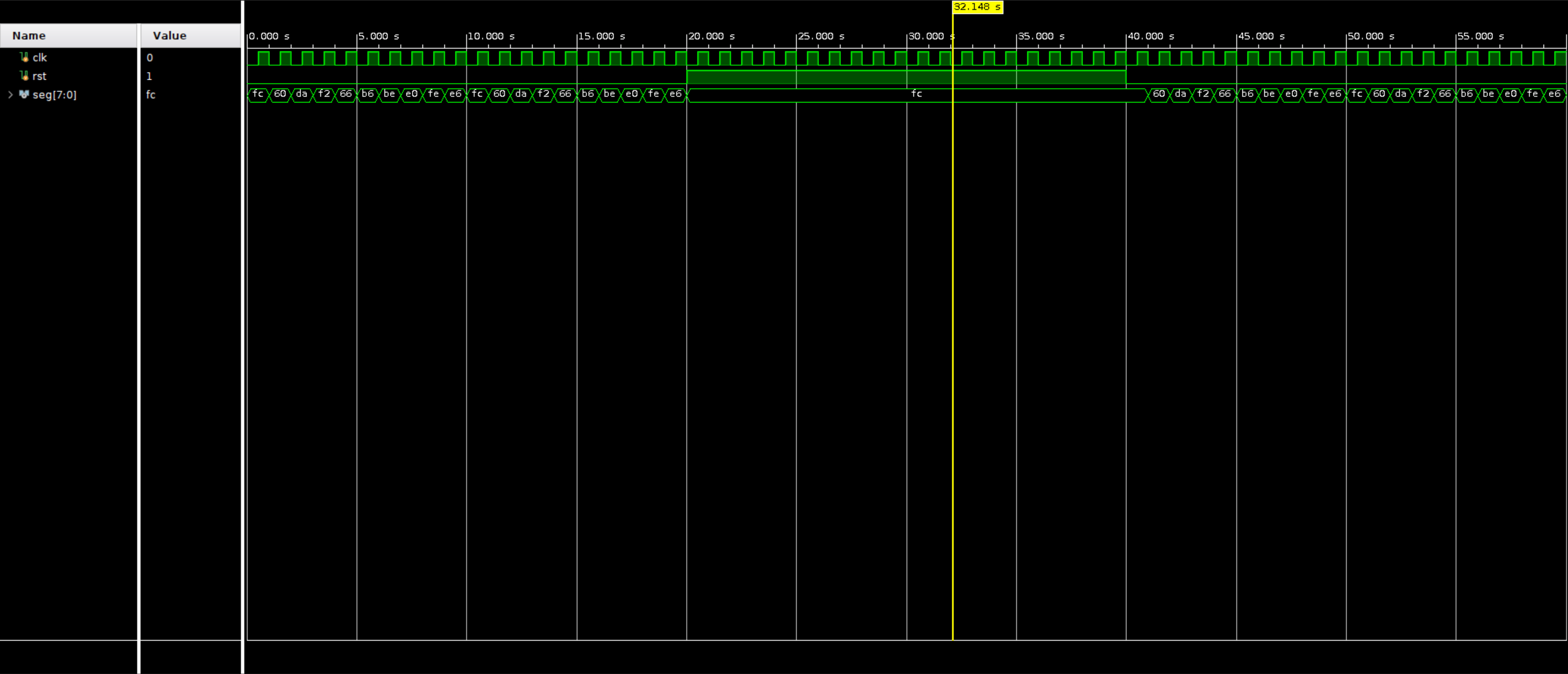
## DUT -2

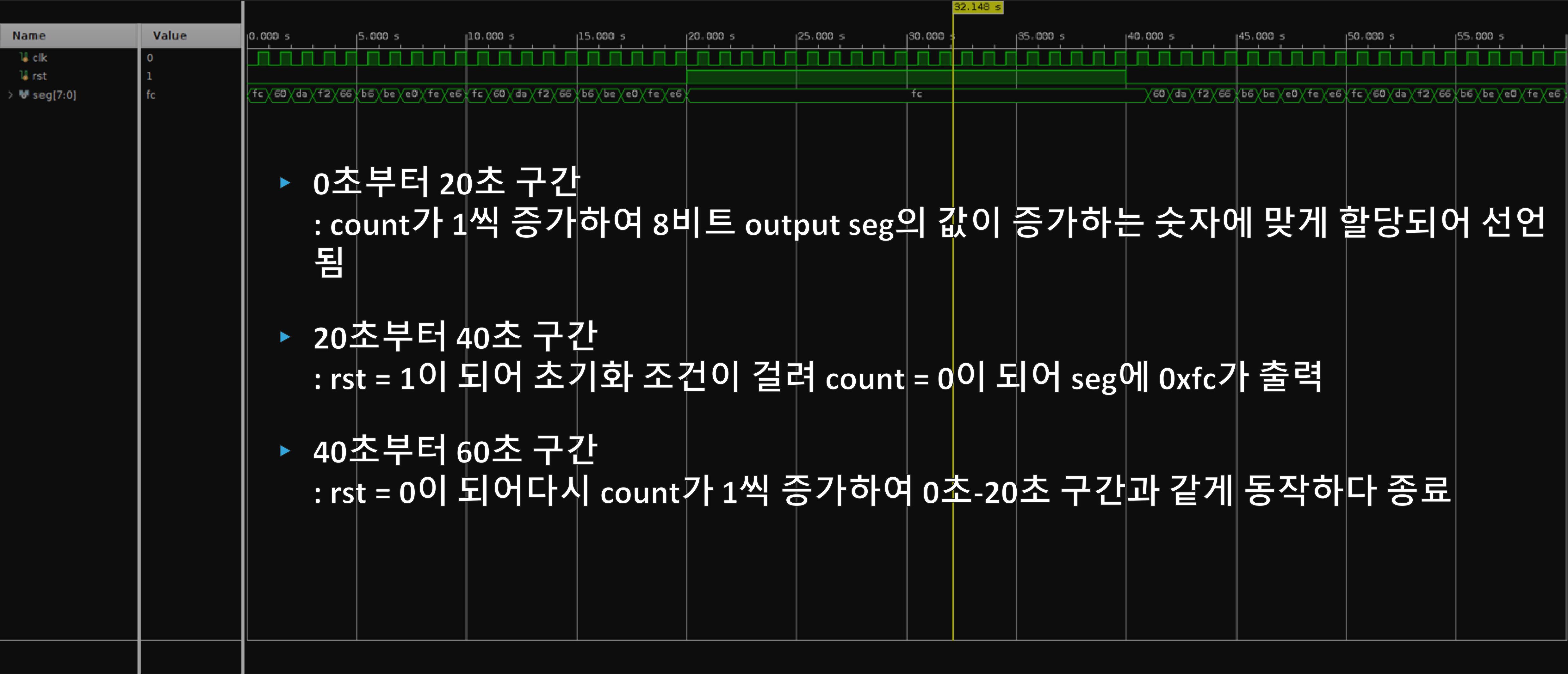
- ▶ 하강 에지에서 현재의 count값에 따라 segment 값 선언
- ▶ 최종 output인 seg에 현재 s값 할당하여 출력

```
32 // count 값에 따른 7-segment 선언
33 always @ (negedge clk) begin
34     s <= (count == 0) ? 8'b11111100: //0
35           (count == 1) ? 8'b01100000: //1
36           (count == 2) ? 8'b11011010: //2
37           (count == 3) ? 8'b11110010: //3
38           (count == 4) ? 8'b01100110: //4
39           (count == 5) ? 8'b10110110: //5
40           (count == 6) ? 8'b10111110: //6
41           (count == 7) ? 8'b11100000: //7
42           (count == 8) ? 8'b11111110: //8
43           (count == 9) ? 8'b11100110: 8'b00000000; // 9, error: 0
44 end
45
46 // 최종 output인 seg에 값 할당
47 assign seg = s;
48
49 endmodule
```



결과





## 추가 과제 계획

- ▶ 기존 1자리에서 3자리로 늘어남에따라 segment도 3개로 증가  
-> output인 seg를 seg0, seg1, seg2로 증가하여 선언
- ▶ 000초 부터 999초까지 타이머 동작 후 000으로 초기화  
-> 본과제와 동일한 원리로 설계
- ▶ 000초를 0분00초로 표현, 실제 타이머처럼 9분59초 후 0분00초로 초기화  
-> 10의자리 초 변수(seg1)만 5를 넘어갈때 초기화  
-> display 함수를 사용해 seg2분 seg1seg0초로 출력

이상입니다

---

THANK YOU