

Logic Design Lab Report: Week 5

2013-12815 Dongjoo Lee

1. Introduction

이번 실습 수업을 통해 Verilog 상에서 Case 문 사용에 대한 요령과 reg 자료형에 대해 이해할 수 있었다. always 문 안에 value 가 변할 때에 output 값에 반영을 희망하는 변수를 포함시킨 뒤, 구문 안에 포함된 case 문을 통해서 input 값에 따른 reg 값을 대응할 수 있었고, 이 reg 를 output 에 assign 함으로써 최종적으로 input 에 대해 원하는 output 을 매핑하는 결과를 얻어낼 수 있었다.

2. Implementation & Result

Homework 는 Practice 에서 BCD to SevenSegmentDisplay 를 구현했던 것 과는 달리, 새로운 Klingon 이라는 새로운 Numbering System 을 구현한다. 이미 작성한 코드에서 reg 에 대응되는 값을 넘버 시스템에 맞게 바꾸어주었다. 추가적인 구현 사항은 case 에 default 문을 추가한 것이다. BCD to SSD 에서는 4 비트의 인풋 범위 [0, 15]가 모두 사용되었기에 default 문이 필요하지 않으나, Klingon 에서는 [0, 9]의 범위를 제외한 나머지는 어떤 LED 도 ON 되지 않도록 작성해야하기 때문에 위 구문의 도입이 가장 효율적이라고 생각했다. 작성한 코드와 Testbench 는 다음과 같다.

< seg.v >

```
1 `timescale 1ns / 1ps
2
3 module seg(
4     input [3:0] in,
5     output [7:0] out
6 );
7
8     seg7_display Seg7_display(
9         .in(out)
10    );
11    // Implement your code from here.
12
13    reg [7:0] klingon;
14    assign out = klingon;
15
16    always @(*)
17        case (in)
18            // in          klingon          dgfedcba          input
19            // in3210
20            4'b0000:    klingon = 8'b00111111; // 0
21            4'b0001:    klingon = 8'b00000001; // 1
22            4'b0010:    klingon = 8'b01000001; // 2
23            4'b0011:    klingon = 8'b01001001; // 3
24            4'b0100:    klingon = 8'b01100010; // 4
25            4'b0101:    klingon = 8'b01011100; // 5
26            4'b0110:    klingon = 8'b01010010; // 6
27            4'b0111:    klingon = 8'b01100100; // 7
28            4'b1000:    klingon = 8'b00110110; // 8
29            4'b1001:    klingon = 8'b01110110; // 9
30            default:    klingon = 8'b00000000; // over 9
31        endcase
32
33 endmodule
```

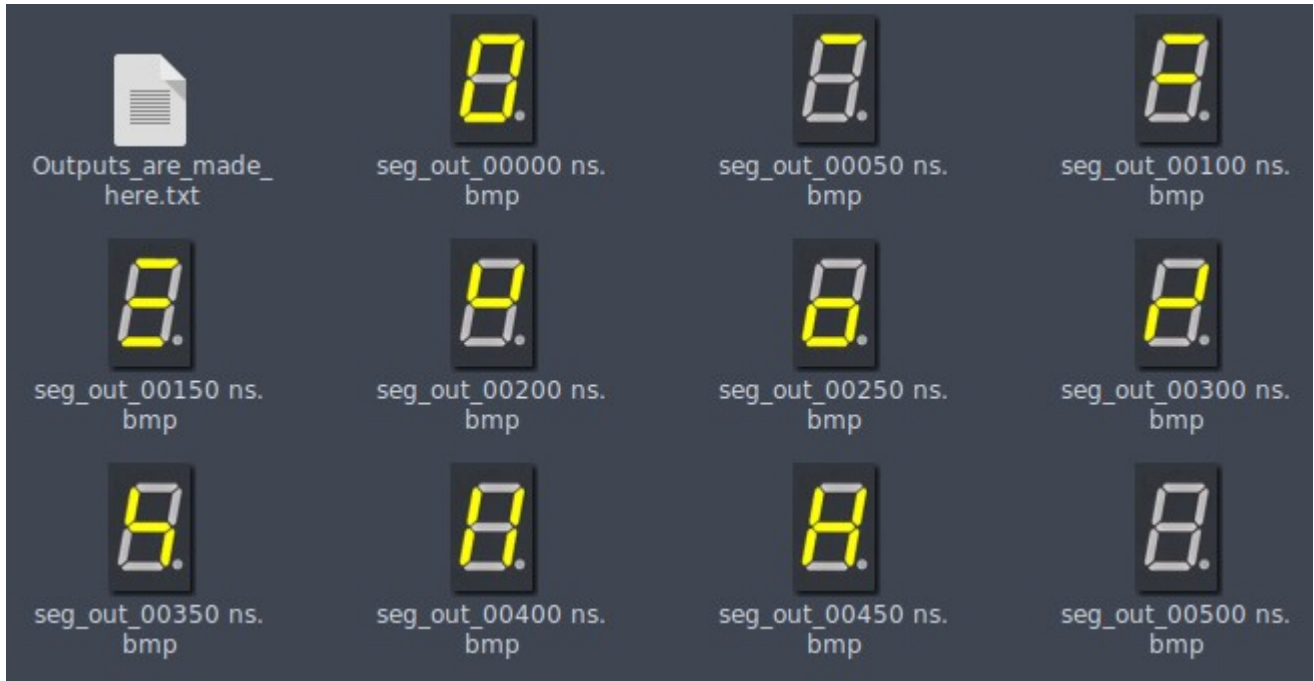
< seg_tb.v >

```
1 `timescale 1ns / 1ps
2 module seg_tb;
3     reg [3:0] in;
4     wire [7:0] out;
5     seg uut (
6         .in(in),
7         .out(out)
8     );
9
10    initial begin
11        // Initialize Inputs
12        in = 4'b0000;
13        #50;
14        in = 4'b0001;
15        #50;
16        in = 4'b0010;
17        #50;
18        in = 4'b0011;
19        #50;
20        in = 4'b0100;
21        #50;
22        in = 4'b0101;
23        #50;
24        in = 4'b0110;
25        #50;
26        in = 4'b0111;
27        #50;
28        in = 4'b1000;
29        #50;
30        in = 4'b1001;
31        #50;
32        in = 4'b1010;
33        #50;
34        in = 4'b1011;
35        #50;
36        in = 4'b1100;
37        #50;
38        in = 4'b1101;
39        #50;
40        in = 4'b1110;
41        #50;
42        in = 4'b1111;
43        #50;
44    end
```

3. Result

결과는 다음과 같다. 의도한대로 잘 동작하는 것을 확인하였으며 특이사항이라면, input 의 범위를 [0,9]뿐 아니라 [10,15]에 대해서도 넣었지만, 모두 default case 에 해당해서인지 output image 는 한 장만 생성되었다.

<./out>



4. Conclusion/Discussion

input 과 output 의 대응을 꾸준히 생각하다 보니 본 과제가 어려움 없이 해낼 수 있었고, Module Design 에 대해 어느 정도 이해해나가고 있다고 느껴졌다.