

Toby Chappell (tchappell@chapman.edu)
Dana Davidson (ddavidson@chapman.edu)
Donner Hanson (hanso127@mail.chapman.edu)

Introduction

For this assignment, we simulated a decoder using Vivado.

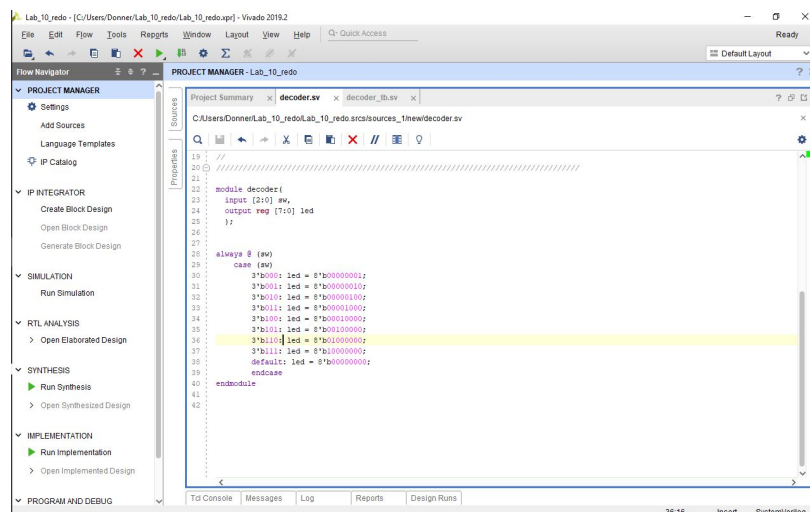
Expected Truth Table

sw	led
000	00000001
001	00000010
010	00000100
011	00001000
100	00010000
101	00100000
110	01000000
111	10000000

Procedure

Decoder

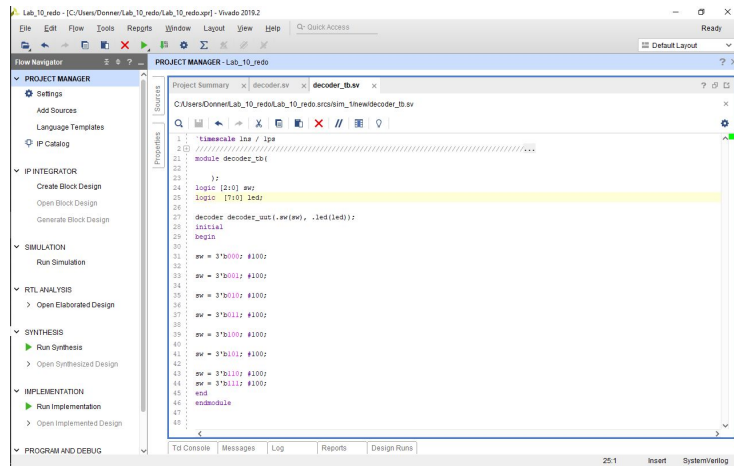
To simulate the decoder we specified 1 3-bit input (sw) and one 8-bit output (led). We then assigned led according to the truth table given above



```
13 //
14 //
15 //
16 //
17 //
18 //
19 //
20 //
21 //
22 module decoder(
23     input [2:0] sw,
24     output reg [7:0] led
25 );
26
27
28 always @ (sw)
29     case (sw)
30         3'b000: led = 8'b00000001;
31         3'b001: led = 8'b00000010;
32         3'b010: led = 8'b00000100;
33         3'b011: led = 8'b00001000;
34         3'b100: led = 8'b00010000;
35         3'b101: led = 8'b00100000;
36         3'b110: led = 8'b01000000;
37         3'b111: led = 8'b10000000;
38         default: led = 8'b00000000;
39     endcase
40 endmodule
41
42
```

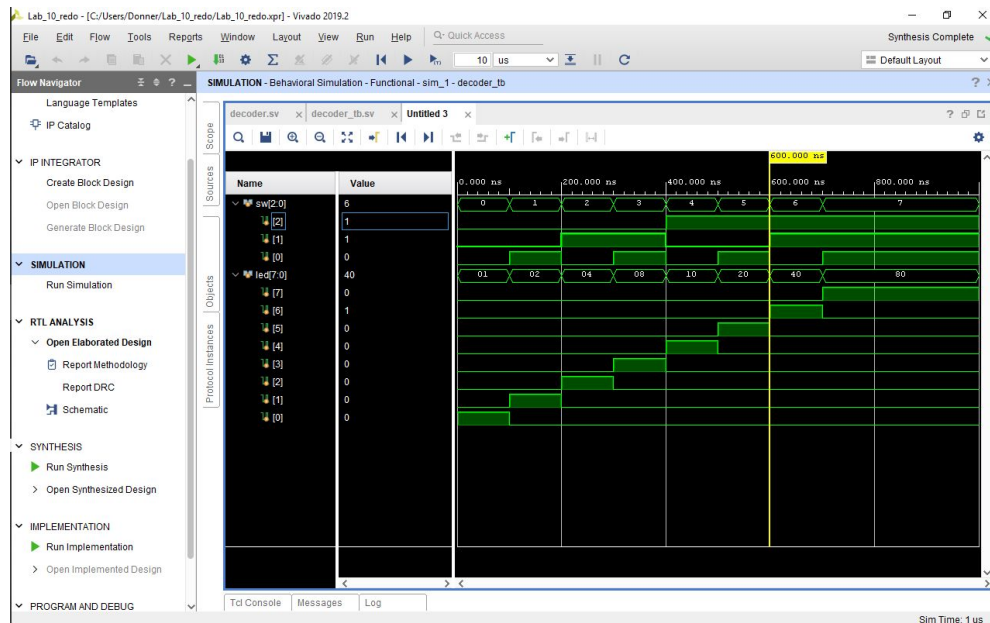
Decoder Test Bench

We then created a test bench for our decoder to simulate our design.



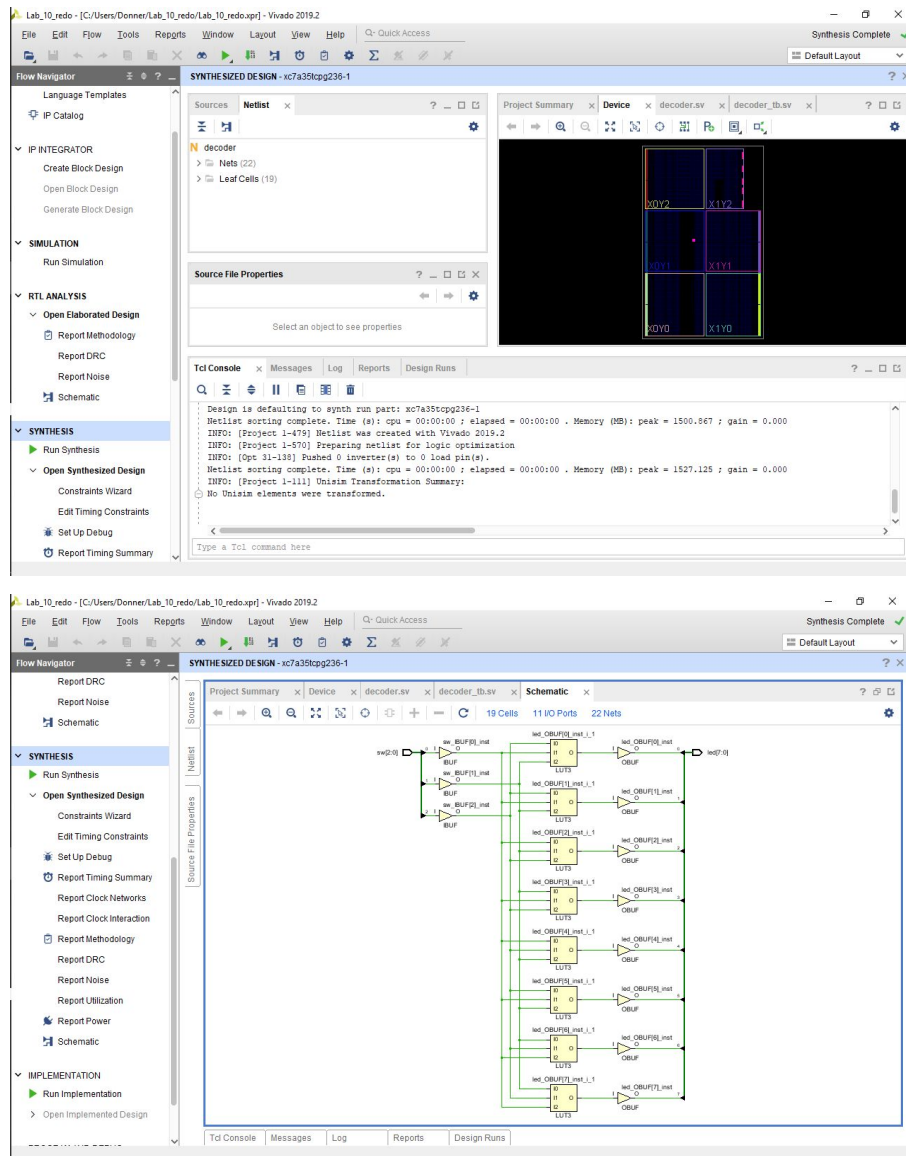
Decoder Waveform

We then ran the simulation of our decoder and came up with the following waveform.



Synthesis

Lastly, we synthesized the design and came up with the following output.



Conclusion

Based on our results for the decoder, our design works correctly. Each value of led corresponds to the correct sw value.