SCombinational Logic Blocks

Calvin Reese
cjreese@fortlewis.edu

3/1/22

1 Introduction

This Lab, we designed a timed trafic light with a parity bit.

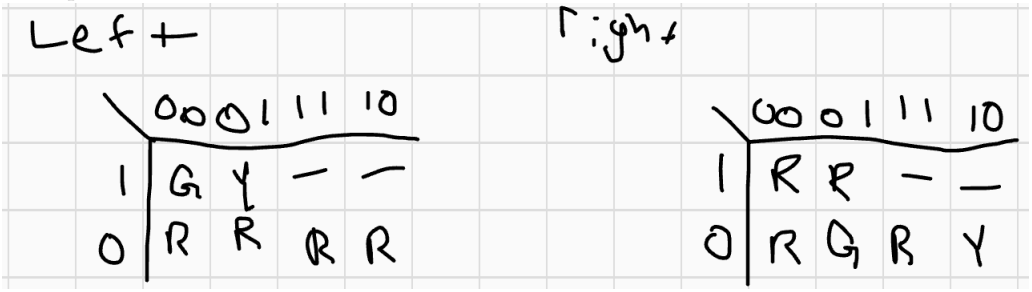
2 Materials and Methods

The tutorial for making these examples are in `http://www.yilectronics.com/Courses/CE433_Labs/s2022/Lab4_combinationalBlocks/Lab4_combinationalBlocks.html`

3 Results

3.1 Task 1

K-map from the counter



Traffic Lights Code

```
22 module TraficLight(clk,sw,led);
23 input clk;
24 input[1:0] sw;
25 output reg [5:0] led = 6'b001001;
26 parameter cntmax = 4'b1110;
27 reg [3:0] cnt;
28 parameter secmax = 32'd100000000;
29 reg [31:0] sec;
30 wire error;
31 parity UUT(.in(cnt[2:0]),.parity(cnt[3]),.errorout(error));
32 always @(posedge clk or negedge sw[0])
33 if(~sw[0])
34 begin
35     cnt <=3'd0;
36     sec <=32'd0;
37 end
38 else if (sec==secmax)
39 begin
40     if (cnt==cntmax)
41     begin
42         cnt<=4'b0000;
43     end
44     if(error)
45     begin
46         sec <= 32'b0;
47         led[0] <= (~cnt[1]&~cnt[0])|(cnt[2]&~cnt[1])|(~cnt[2]&cnt[1]&cnt[0]);
48         led[1] <= ~cnt[2]&cnt[1]&~cnt[0];
49         led[2] <= ~cnt[2]&~cnt[1]&cnt[0];
50         led[3] <= ~cnt[2];
51         led[4] <= cnt[2]&~cnt[1]&cnt[0];
52         led[5] <= cnt[2]&~cnt[1]&~cnt[0];
53         cnt<=cnt+4'b1001;
54     end
55     else
56     begin
57         led=led;
58         cnt[3] <= cnt[0]^cnt[1]^cnt[2];
59     end
60 end
61 else
62 begin
63 sec<=sec+32'd1;
64 end
65 endmodule
```

Parity Check Code

```
23 module parity(parity,in,errorout);
24 input[2:0] in;
25 input parity;
26 output errorout;
27 assign errorout = parity == in[0]^in[1]^in[2];
28 endmodule
```

Output: `https://youtu.be/yAmZ5CytY_Q`

4 Discussion

Parity was fairly simple, I only had a difficult time translating the running lights code to this one since I didn't quite understand how the second gap was made the first time. When the parity is enacted it will hold onto the previous light then move on when it receives valid info.