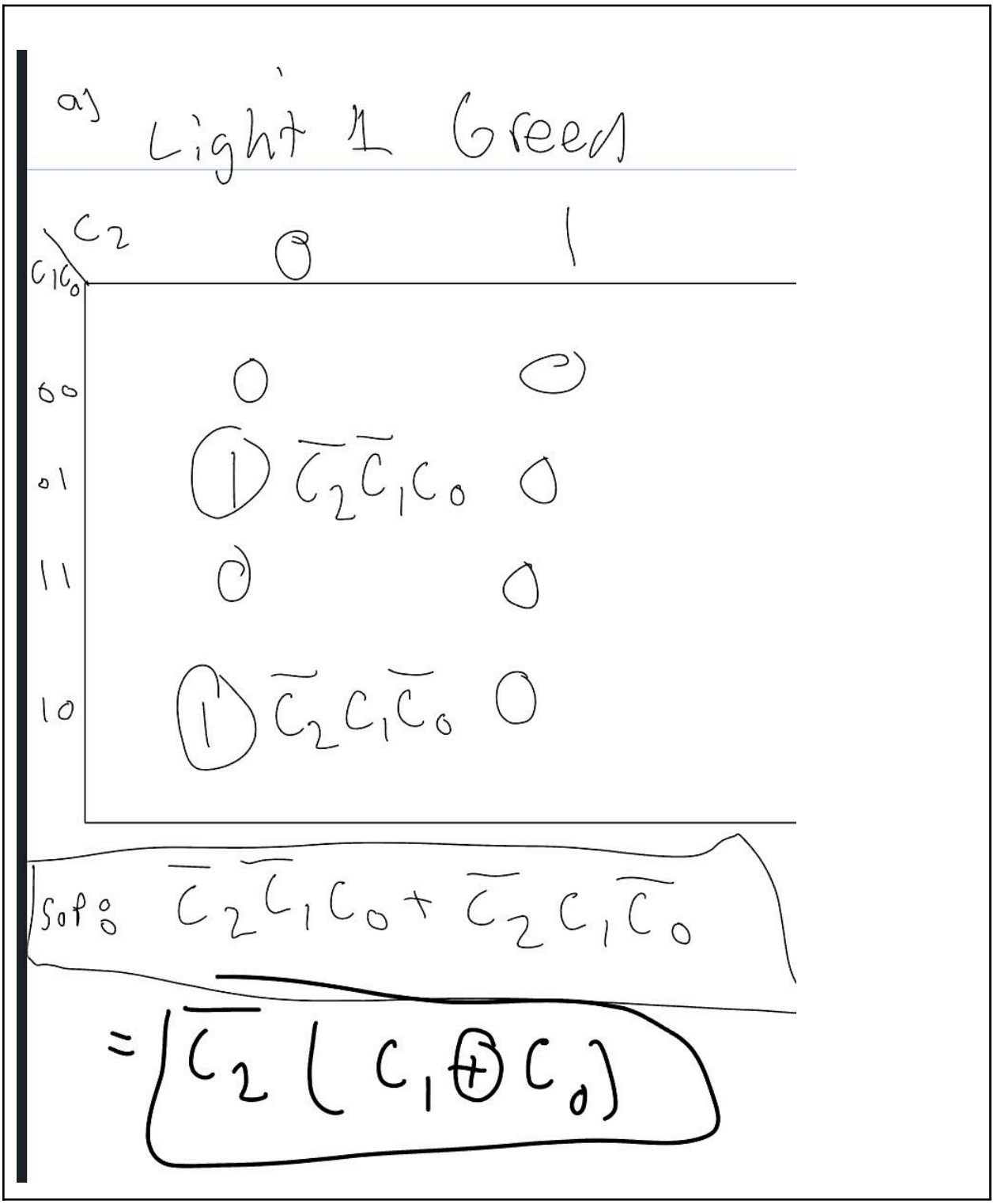
Name: Danyal Falsafi
Section: 002
X500: falsa003

Lab 2:

LIGHTS		CLOCK INPUT BITS			LIGHT 1			LIGHT 2		
1	2	C2	C1	C0	G	Y	R	G	Y	R
R	R	0	0	0	0	0	1	0	0	1
G	R	0	0	1	1	0	0	0	0	1
G	R	0	1	0	1	0	0	0	0	1
Y	R	0	1	1	0	1	0	0	0	1
R	R	1	0	0	0	0	1	0	0	1
R	G	1	0	1	0	0	1	1	0	0
R	G	1	1	0	0	0	1	1	0	0
R	Y	1	1	1	0	0	1	0	1	0

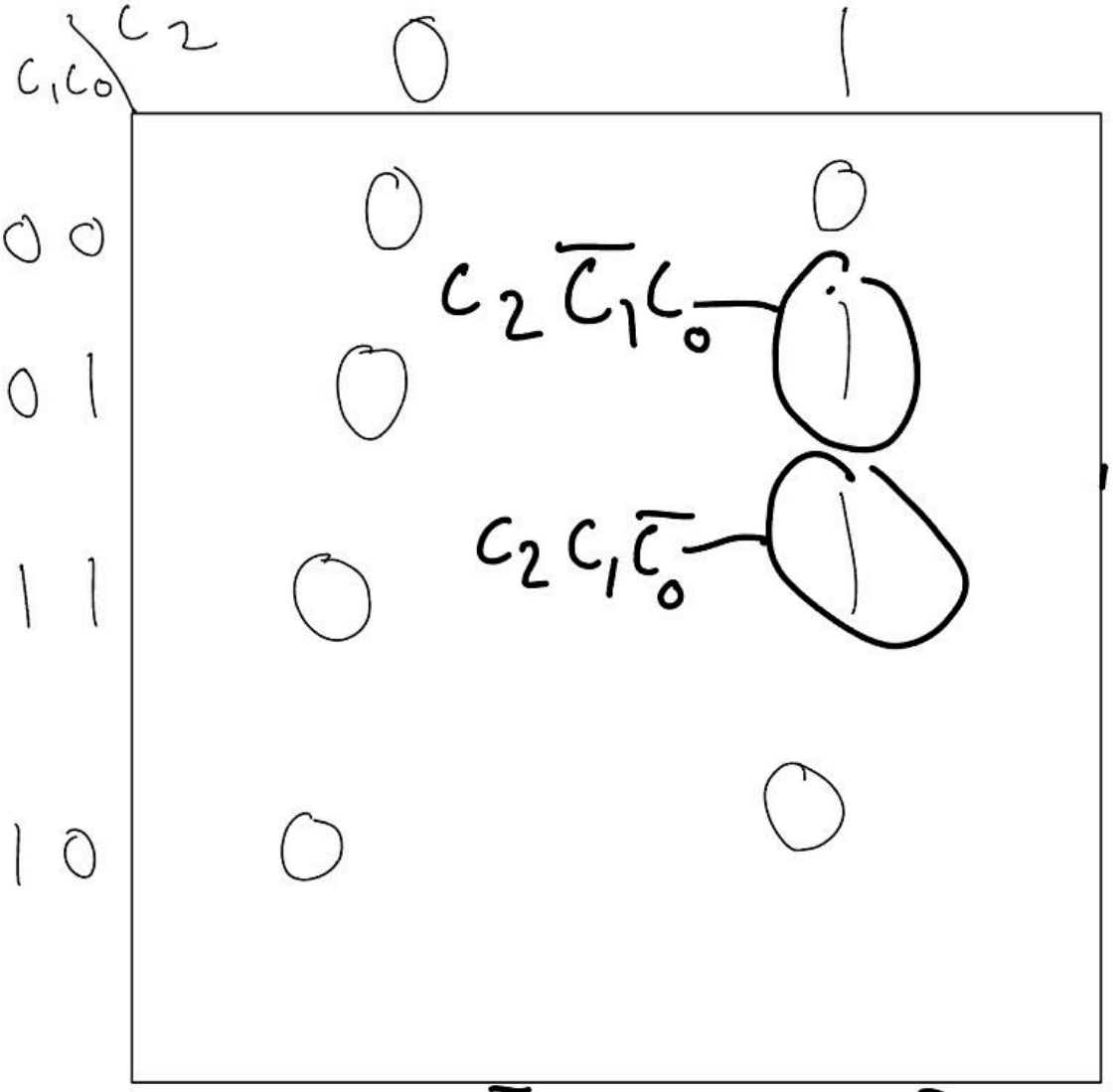
Truth Table for 4-way stop light controller (5 points)

GREEN


a) Light 1 Green
C_2
C_1C_0
0
1
00
01
11
10
$\bar{C}_2\bar{C}_1C_0$
$\bar{C}_2C_1\bar{C}_0$
SoP: $\bar{C}_2\bar{C}_1C_0 + \bar{C}_2C_1\bar{C}_0$
$= \bar{C}_2(C_1 \oplus C_0)$

Karnaugh Map Reduction and Simplified Equations for **Light 1** - G (2.5 pts)

d) Light 2 Green



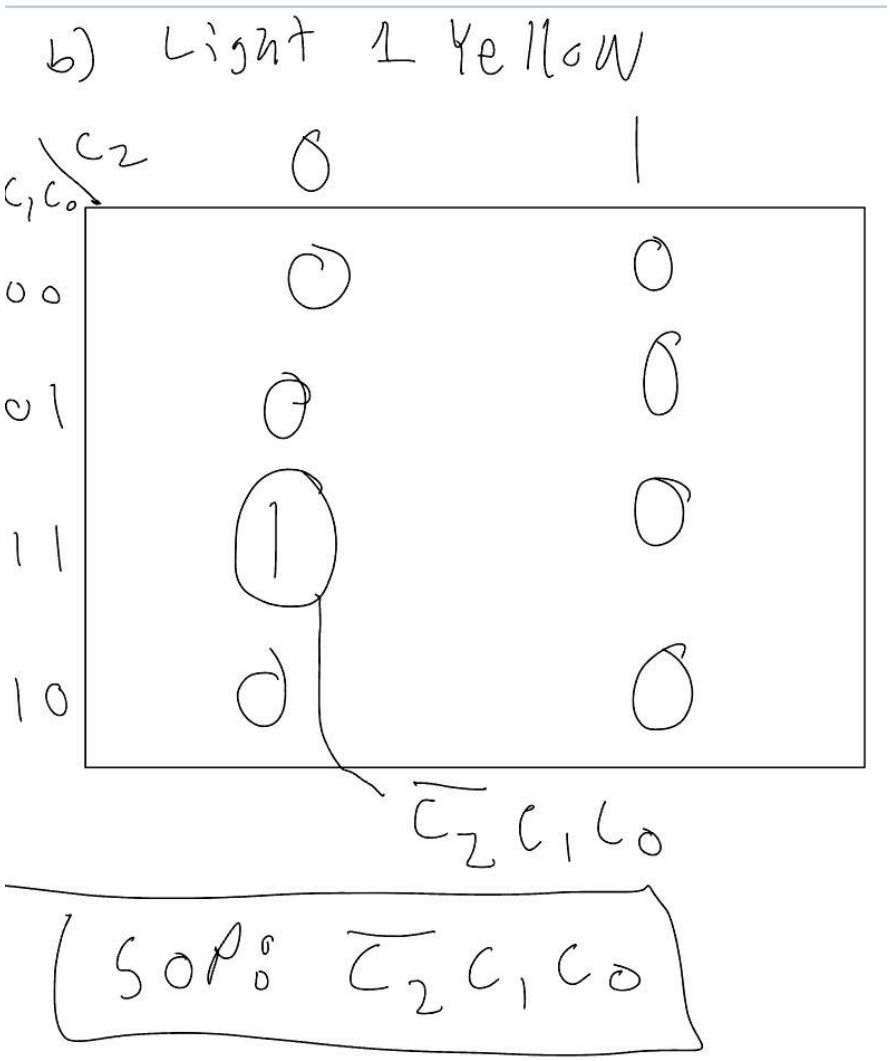
SOP: $C_2\bar{C}_1C_0 + C_2C_1\bar{C}_0$

$C_2(\bar{C}_1C_0 + C_1\bar{C}_0)$

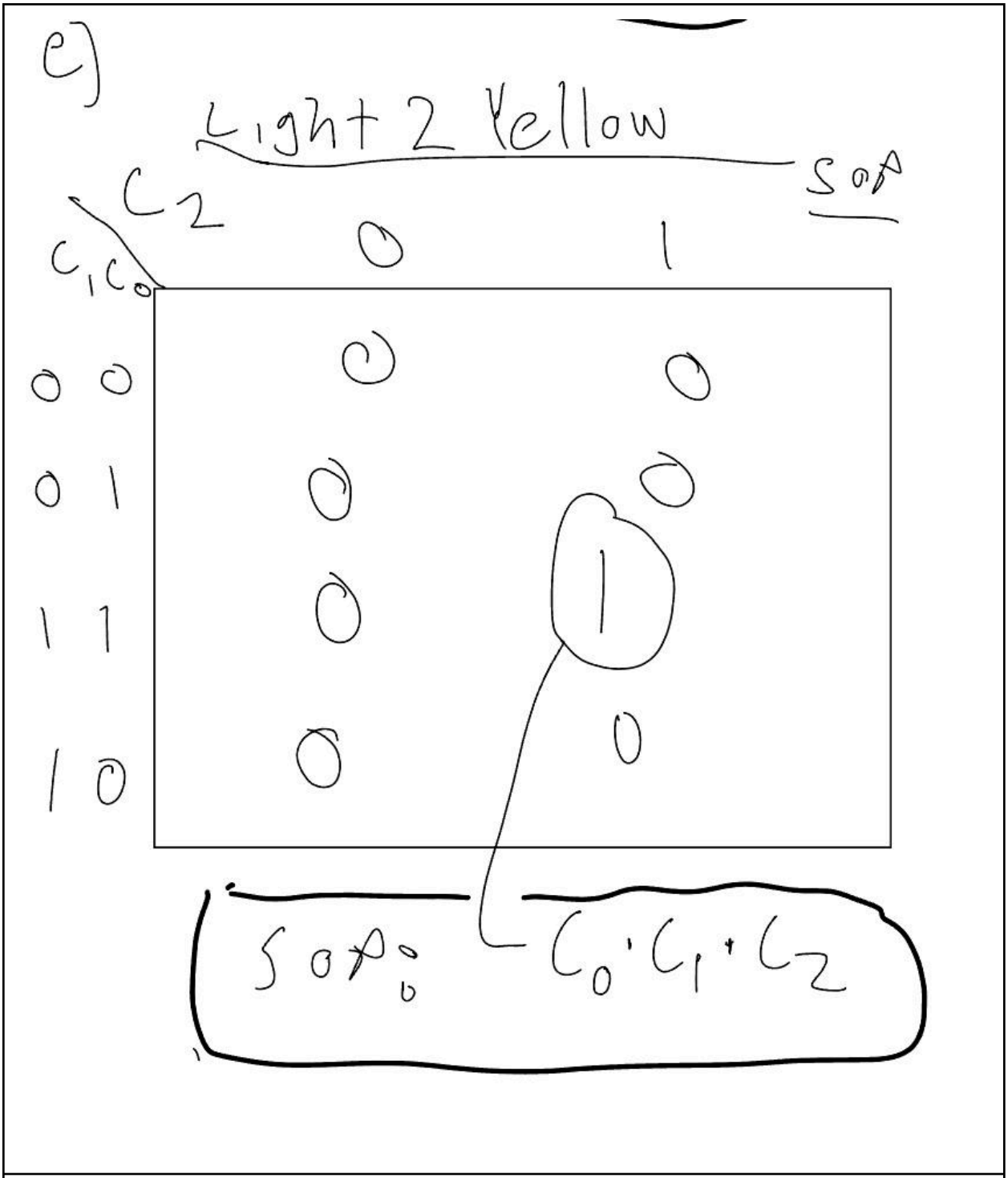
$C_2(C_0 \oplus C_1)$

Karnaugh Map Reduction and Simplified Equations for **Light 2** - G (2.5 pts)

YELLOW



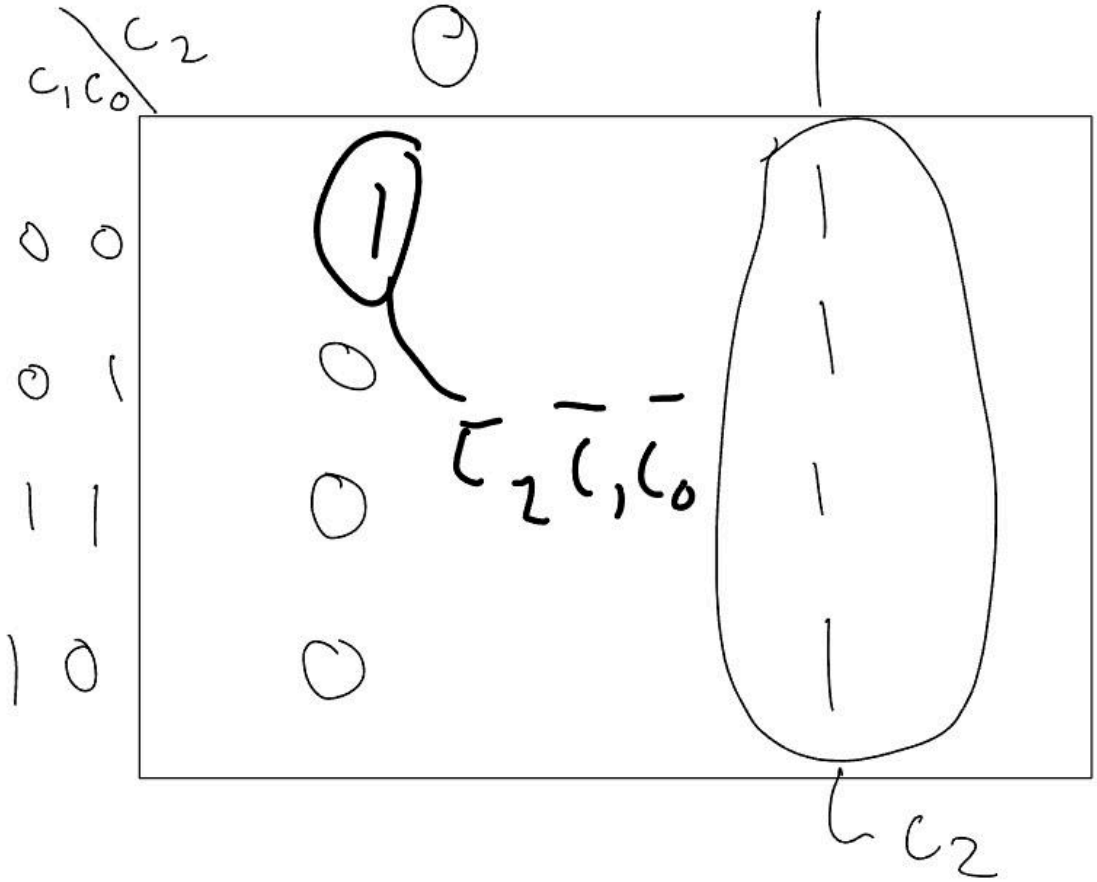
Karnaugh Map Reduction and Simplified Equations for **Light 1** - Y (2.5 pts)



Karnaugh Map Reduction and Simplified Equations for **Light 2** - Y (2.5 pts)

RED

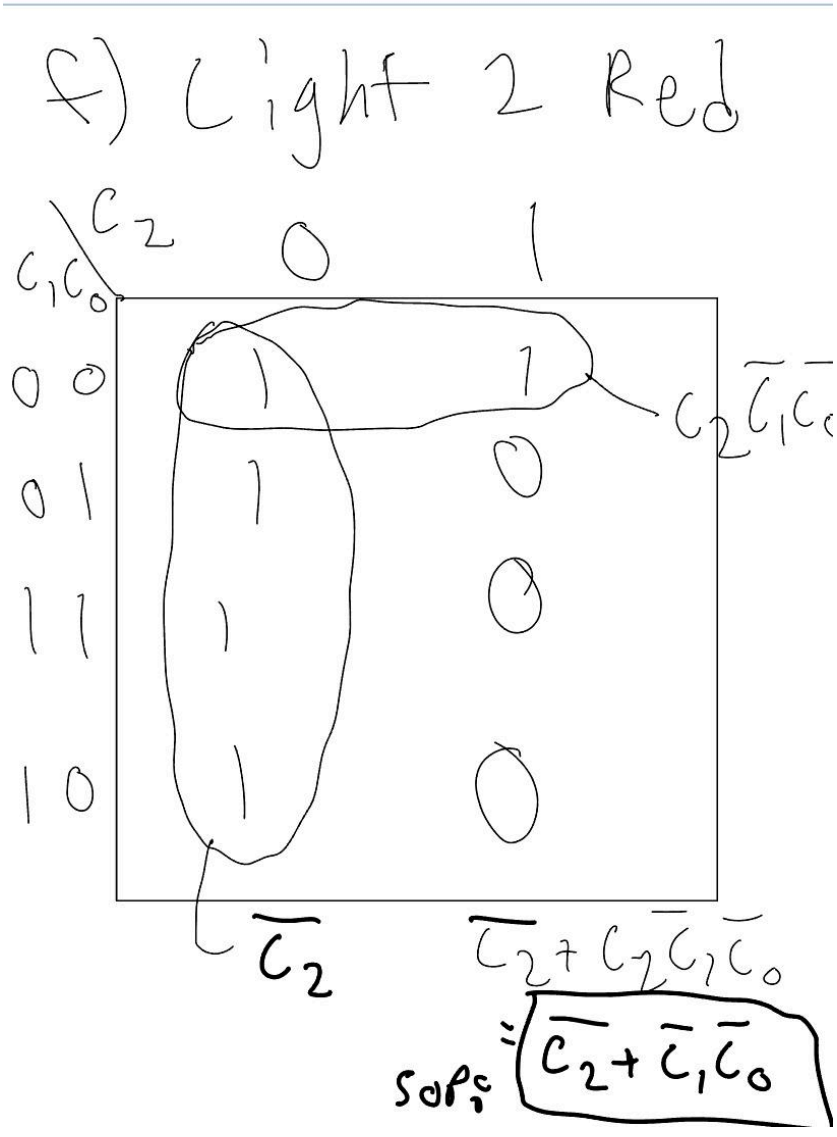
c) Light 1 Red 100 101 110 111



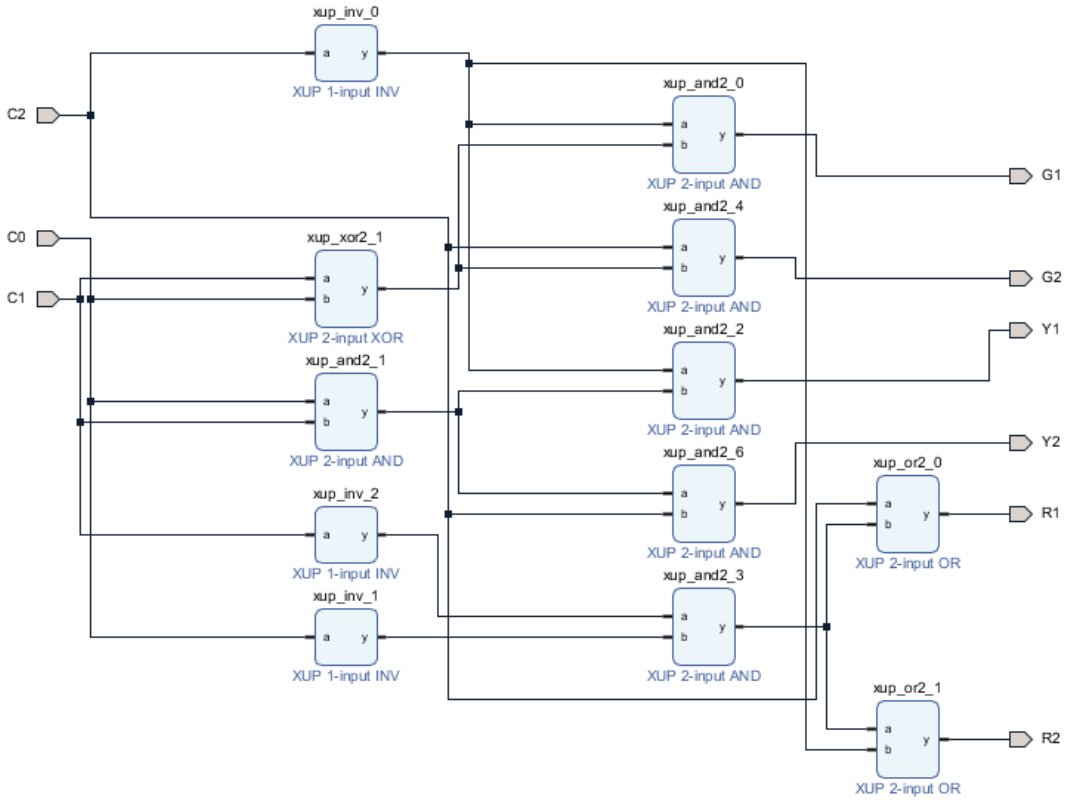
SOP: $C_2 + \bar{C}_2\bar{C}_1\bar{C}_0$

SOP: $C_2 + \bar{C}_1\bar{C}_0$

Karnaugh Map Reduction and Simplified Equations for **Light 1** - R (2.5 pts)



Karnaugh Map Reduction and Simplified Equations for **Light 2** - R (2.5 pts)



Include a copy of Block Diagram. (From Vivado) (10 pts)

Include a copy of your Simulation Waveform showing all the input combinations. (10 pts)

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////
// Module Name: lab2_tb
//////////////////////////////////////////////////////////////////

module lab2_tb(
);
reg inputC0, inputC1, inputC2;
wire G1;
wire Y1;
wire R1;
wire G2;
wire Y2;
wire R2;

design_1_wrapper instance_1(
    .C0(inputC0),
    .C1(inputC1),
    .C2(inputC2),
    .G1(G1),
    .Y1(Y1),
    .R1(R1),
    .G2(G2),
    .Y2(Y2),
    .R2(R2));
initial
begin
    inputC0 = 0; inputC1 = 0; inputC2 = 0;
    #50
    inputC0 = 0; inputC1 = 0; inputC2 = 1;
    #50
    inputC0 = 0; inputC1 = 1; inputC2 = 0;
    #50
    inputC0 = 0; inputC1 = 1; inputC2 = 1;
    #50
    inputC0 = 1; inputC1 = 0; inputC2 = 0;
    #50
    inputC0 = 1; inputC1 = 0; inputC2 = 1;
```

```
    #50
    inputC0 = 1; inputC1 = 1; inputC2 = 0;
    #50
    inputC0 = 1; inputC1 = 1; inputC2 = 1;
    end
endmodule
```

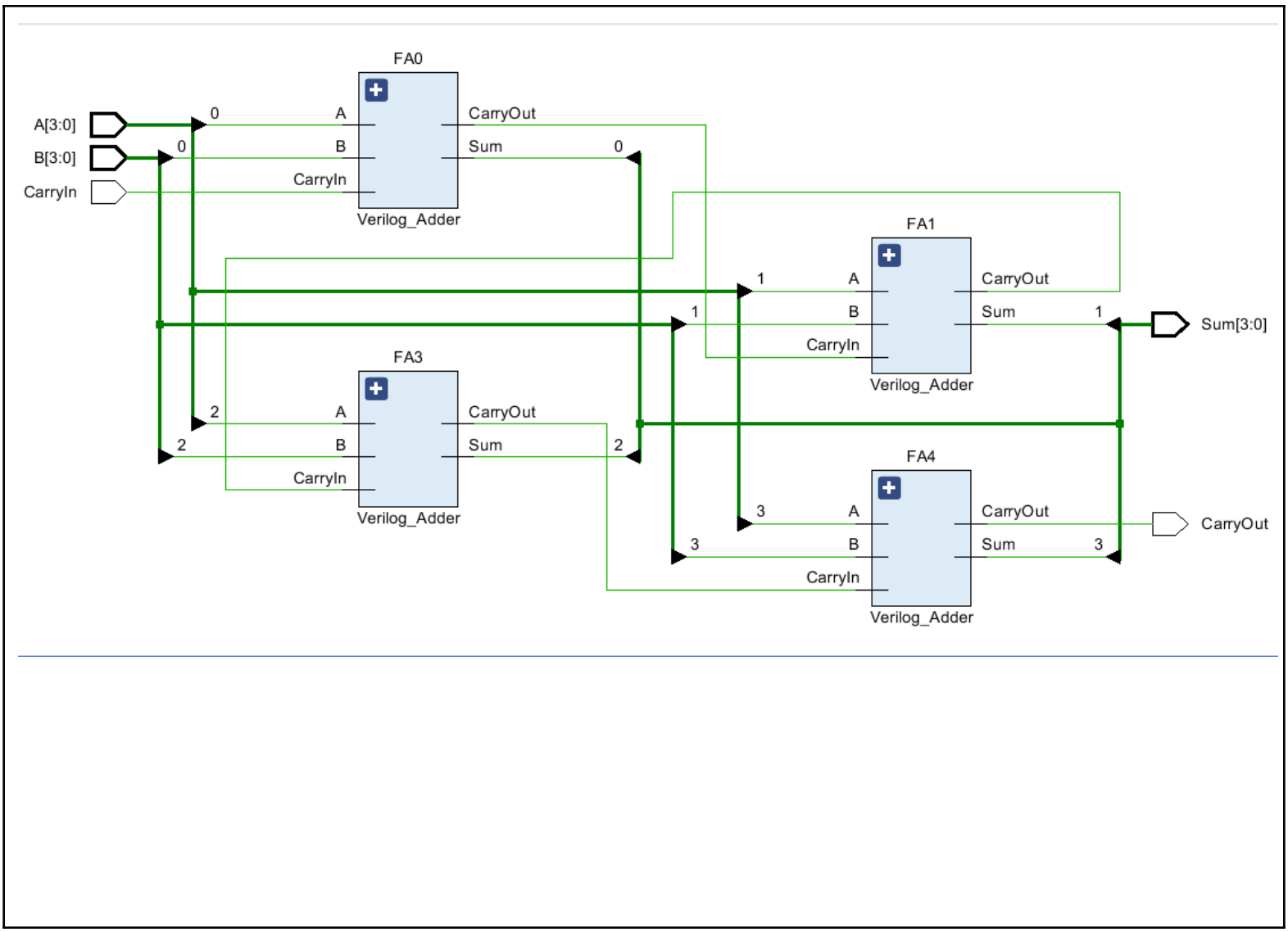
Include a copy of the **Testbench File** (Delete unnecessary Lines) (10 pts)

Lab 3:

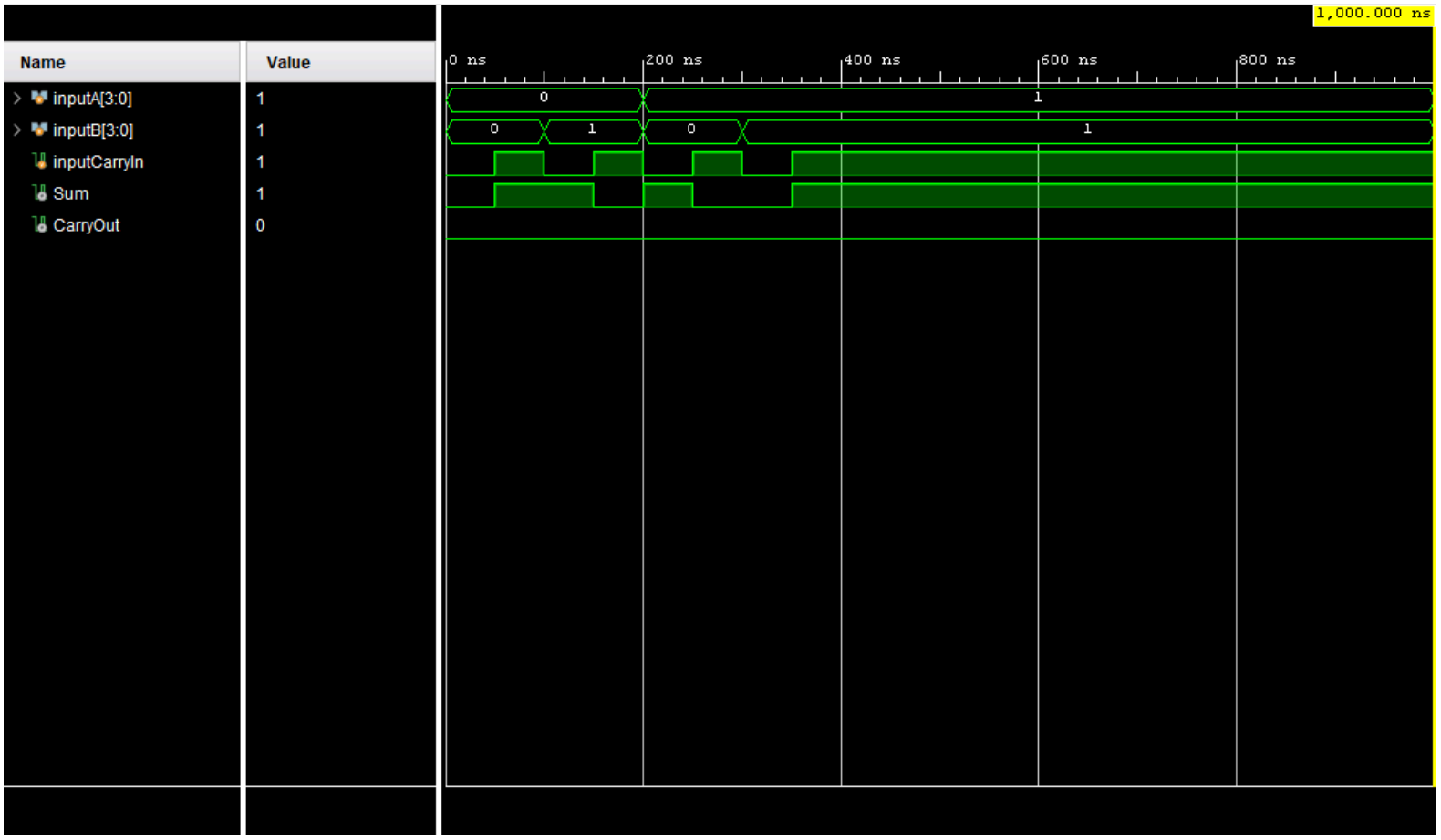
Prelab for 4-bit Ripple Carry Adder :

```
`timescale 1ns / 1ps


module prelab3_TB (
    // list your ports here if any
);

reg [3:0] inputA, inputB;
reg inputCarryIn;
wire Sum, CarryOut;

Ripple_Carry_Adder_4 Ripple_Carry_Adder_4_inst (
    .A(inputA),
    .B(inputB),
    .CarryIn(inputCarryIn),
    .Sum(Sum),
    .CarryOut(CarryOut)
);

initial begin
    inputA = 0; inputB = 0; inputCarryIn = 0;
    #50;
    inputA = 0; inputB = 0; inputCarryIn = 1;
    #50;
    inputA = 0; inputB = 1; inputCarryIn = 0;
    #50;
    inputA = 0; inputB = 1; inputCarryIn = 1;
    #50;
    inputA = 1; inputB = 0; inputCarryIn = 0;
    #50;
    inputA = 1; inputB = 0; inputCarryIn = 1;
    #50;
    inputA = 1; inputB = 1; inputCarryIn = 0;
    #50;
    inputA = 1; inputB = 1; inputCarryIn = 1;
end
```

endmodule
Include a copy of the **Verilog Code** used for the testbench (Delete unnecessary Lines) (8 pts)



Include a screenshot of the Schematic obtained after Elaboration (All gates should be visible). To find this schematic go to: RTL Analysis → Open Elaborated Design → Schematic (8 pts)

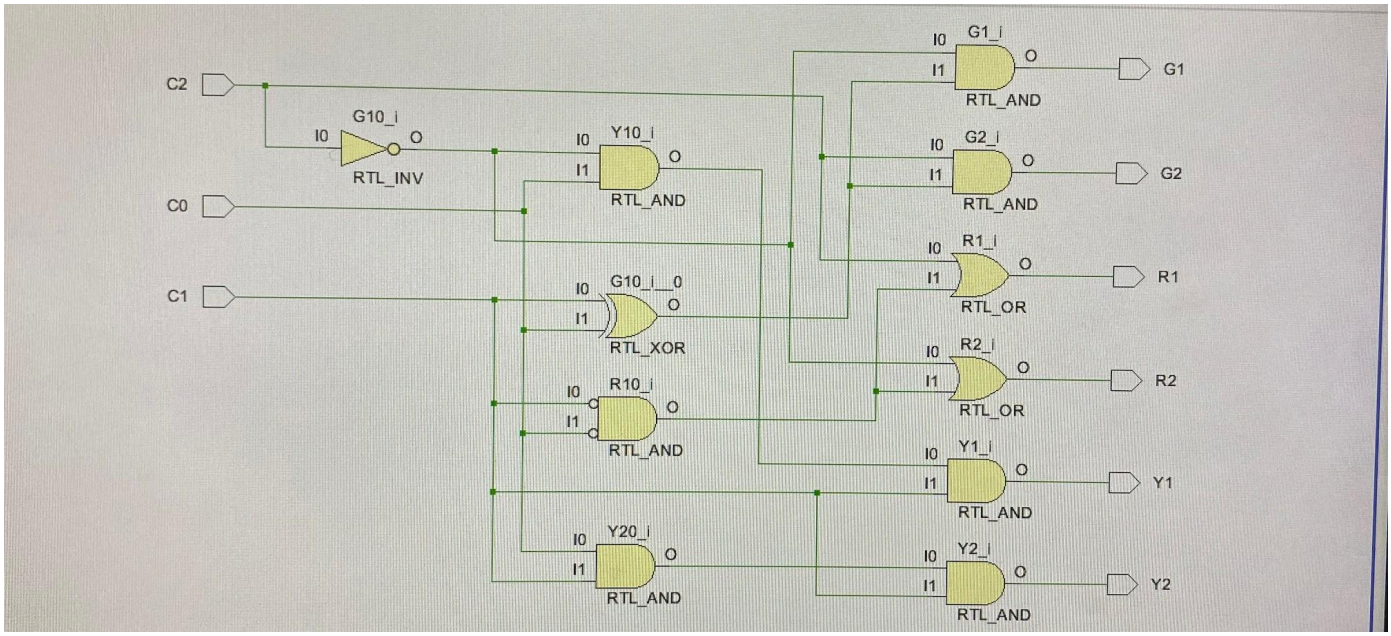


Include a screenshot of the simulation, a part of A(changing from 0 to 15) or B(changing from 0 to 15) is enough. (8 pts)

Stop Light Controller

```
`timescale 1ns / 1ps

module TrafficLightController(
output G1, G2, Y1, Y2, R1, R2,
output [2:0] led,
input C1,
input C2,
input C0
);
assign led[0] = C0;
assign led[1] = C1;
assign led[2] = C2;

assign G1 = ~(C2)&(C0^C1); // Green light 1
assign G2 = (C2)&(C0^C1); // Green light 2
assign Y1 = ~(C2)& C0 & C1; // Yellow light 1
assign Y2 = (C2)& C0 & C1; // Yellow light 2
assign R1 = C2 | ((~C0) & (~C1)); // Red light 1
assign R2 = ~C2 | ((~C0) & (~C1)); // Red light 2
endmodule
```

Include a copy of the **Verilog Code** used for Design Entry (Delete unnecessary Lines) (6 pts)



Include a screenshot of the Schematic obtained after Elaboration (All gates should be visible). (5 pts)

Briefly explain the trade-offs you have implemented in your “optimum” design.

The trade off would be that the “optimum” design(lab 2) is that there is a time delay greater, almost by a factor of 2, in lab 2 than in the lab 3’s simulation. However, the design of lab 2 was more simple and efficient the trade off was that lab 3’s design unlike the design in lab 2 was more adaptable to changes, whereas the elaborated design from lab 3 had an easier time to make changes with the operations that went on based on the design acting in a shorter execution time.

Compare your block diagram design from lab 2 to the elaborated design from lab 3, and describe the advantages and disadvantages of each. (5 pts)

Lab 3 Advantages Lab 2 Disadvantages:

- Lab 3 has less inverters (means only one input had to get inverted) than the design displayed for lab 2 causing lab 3's design to have less of a time delay to occur between the time it takes to change the input versus the time it takes for the new output to be produced.
- Lab 3 appears to have one less level than in lab 2 which is an advantage for performing a much more efficient series of steps to reach the exact same output you would get in lab 2 in a shorter period of time.

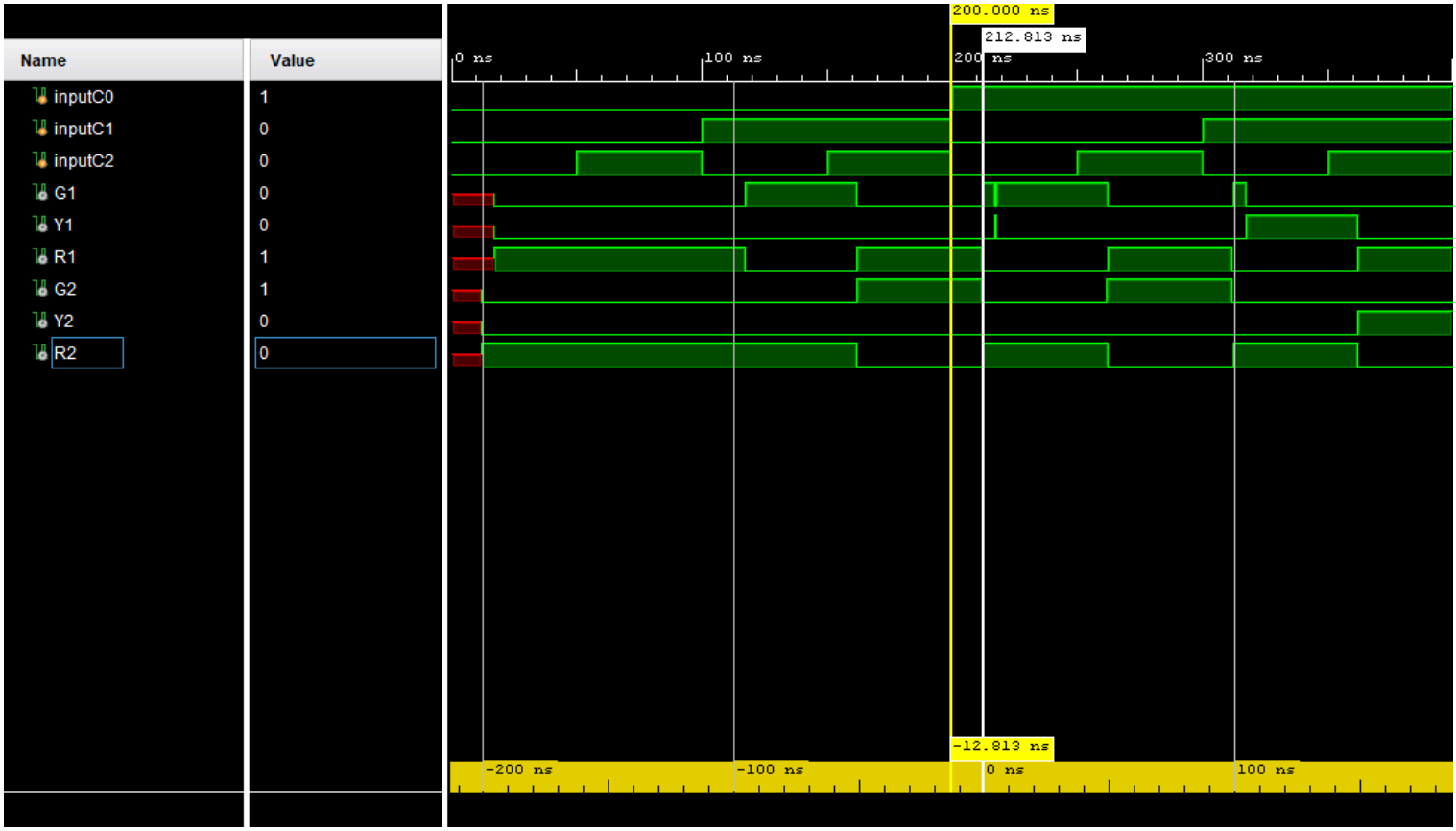
Lab 3 Disadvantages Lab 2 Advantages:

- Lab 3 seems to have the exact same logic gates (excluding inverters) in comparison to lab 2 as well as one additional AND gate that is for each output(G1,Y1,..., R2) is given an AND gate unlike the block design shown for lab 2.
- Lab 2 seems to have less wire connections between the logic gates(excluding inverters)

According to the post behavior time ___synthesis there is a time delay when the inputs are changing from C1 and C2 being inputted to a transition of only input C0 being inputted there is a 6.371ns time delay between sending the input and receiving the output (R1 & G2).

Lab 2:

Similarly, according to the post behavior time___synthesis there is a time delay when the inputs are changing from C1 and C2 being inputted to a transition of only input C0 being inputted there is a 12.813ns time delay between sending the input and receiving the output (R1 & G2).



Briefly explain the trade-offs you have implemented in your “optimum” design. Compare your block diagram design from lab 2 to the elaborated design from lab 3, and describe the advantages and disadvantages of each. (5 pts)

```
## Switches
set_property PACKAGE_PIN V17 [get_ports C0]
set_property IOSTANDARD LVCMOS33 [get_ports C0]
set_property PACKAGE_PIN V16 [get_ports C1]
set_property IOSTANDARD LVCMOS33 [get_ports C1]
set_property PACKAGE_PIN W16 [get_ports C2]
set_property IOSTANDARD LVCMOS33 [get_ports C2]




#Pmod Header JXADC

set_property PACKAGE_PIN J3 [get_ports G1]
set_property IOSTANDARD LVCMOS33 [get_ports G1]

set_property PACKAGE_PIN L3 [get_ports Y1]
set_property IOSTANDARD LVCMOS33 [get_ports Y1]

set_property PACKAGE_PIN M2 [get_ports R1]
set_property IOSTANDARD LVCMOS33 [get_ports R1]

set_property PACKAGE_PIN N2 [get_ports G2]
set_property IOSTANDARD LVCMOS33 [get_ports G2]

set_property PACKAGE_PIN K3 [get_ports Y2]
set_property IOSTANDARD LVCMOS33 [get_ports Y2]

set_property PACKAGE_PIN M3 [get_ports R2]
set_property IOSTANDARD LVCMOS33 [get_ports R2]
```

Include a copy of the **Constraint File (.xdc)** used in the implementation (Delete unnecessary Lines) (5 pts)

Include a discussion of the circuit you used to display the results on LED's.

The circuit I designed had 6 LED lights placed on a breadboard, each traffic light consisting of 3 LEDs, one for each color, was placed on the breadboard as each light's LEDs were grouped together individually. The inputs assigned to the JXADC pins sent out signals from the FPGA to the specific LEDs that the constraint file indicated the color of each light to be connected on the breadboard. Furthermore, the other pin of each LED (not receiving input signal) had a resistor attached to ground as a safe escape of the left over voltage.

What is the current through each LED assuming the Basys3 outputs 3.3 volts?

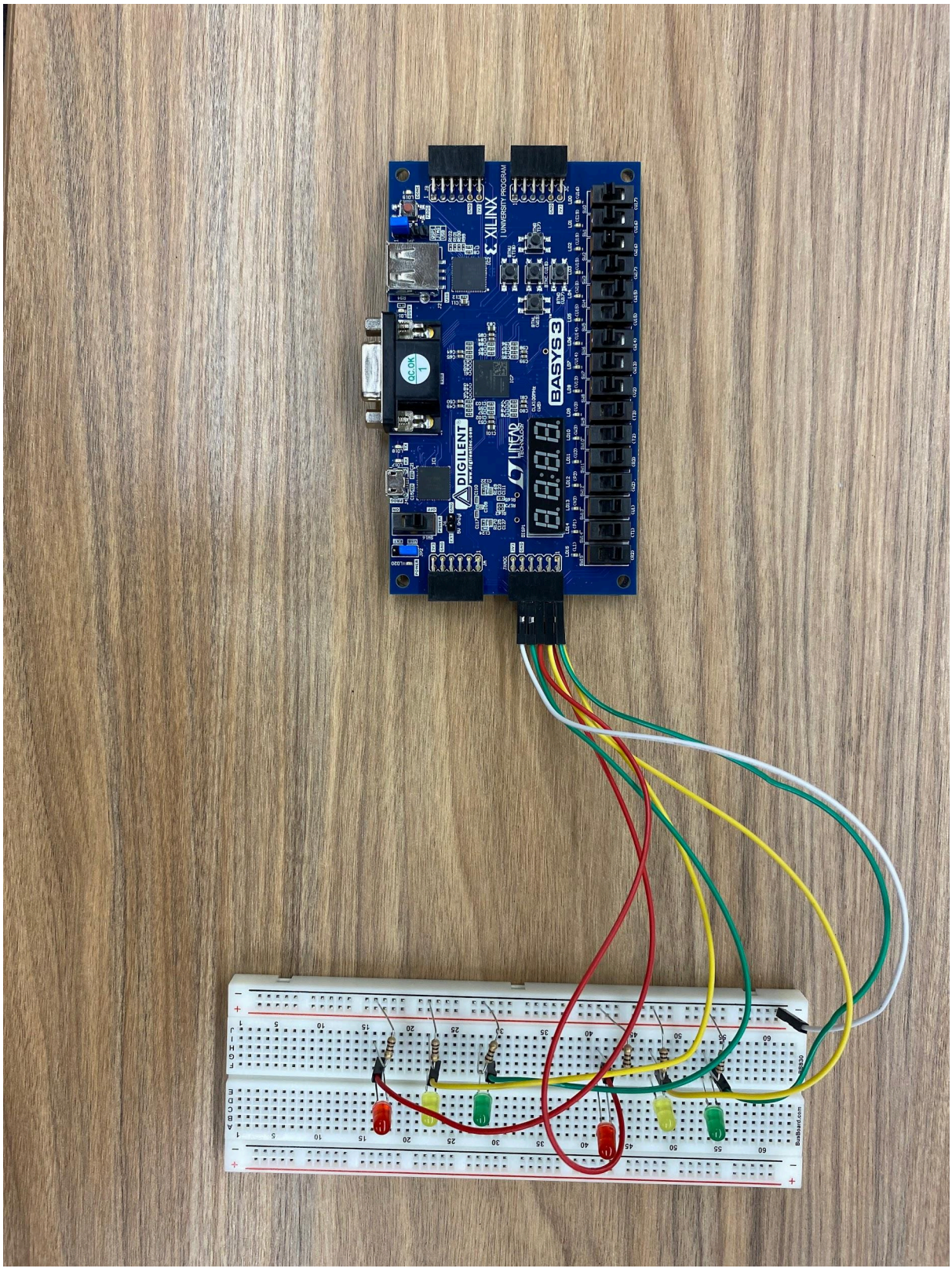
The circuit for the implementation of the traffic controller consisted of 6 LEDS, each LED connected in parallel with a 570Ω resistor. The electric potential drop for the green, yellow, and red LEDs were 2 Volts.Hence, the current flowing through the green, yellow, red and LEDs would be 21.05 mA.

$I = \frac{2.0}{570}$ = 3.51 mA

What is the total current draw on the Basys3 at any given time? (ignore the internal resistance of the LED) (5 pts)

The total current draw that was sent back to the Basys 3 from the output of the circuit from all of the LED lights would be 21.05 mA.

$3.51 \times 6 = 21.05mA$



voltage remaining after turning on the light.

Include a discussion of the circuit you used to display the results on LED's. What is the current through each LED assuming the Basys3 outputs 3.3 volts? What is the total current draw on the Basys3 at any given time? (ignore the internal resistance of the LED) (5 pts)