UCSD CSE140L Spring 2014

**LAB#2 Report**

Demonstration Date : 4 / 29 / 14 Student CID\_\_\_\_\_\_\_\_\_\_268\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name: \_\_\_\_\_\_\_ Kieth\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Vo\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

first M.I. Last

**Submission Date & Time :**

(FILLED BY Student BEFORE DEMO) (\*\*\* FILLED BY TUTOR/INSTRUCTOR \*\*\*)

**Self-test Report** Demo Reviewer

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Working Not working **Demo** score **Report** score

**Part1**: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_/1 a) \_\_\_\_\_\_/1

**Part2**: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_/1 b)\_\_\_\_\_\_\_/1

**Part3**: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_/1 c) \_\_\_\_\_\_\_/1

**Part4**: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_/1 d) \_\_\_\_\_\_\_/1

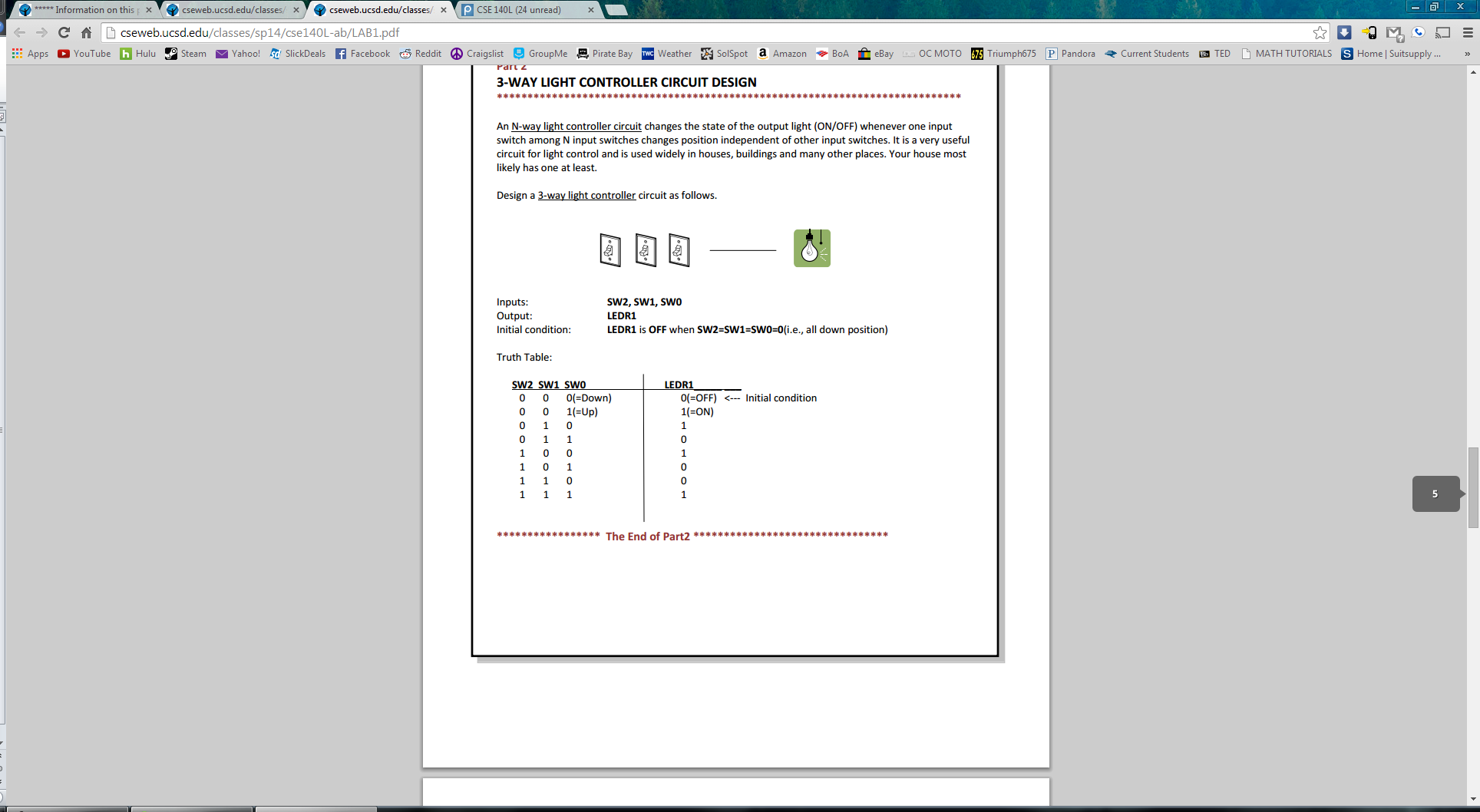
**Part5**: \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ /1 e) \_\_\_\_\_\_\_\_/1

**Subtotal**  **Subtotal**

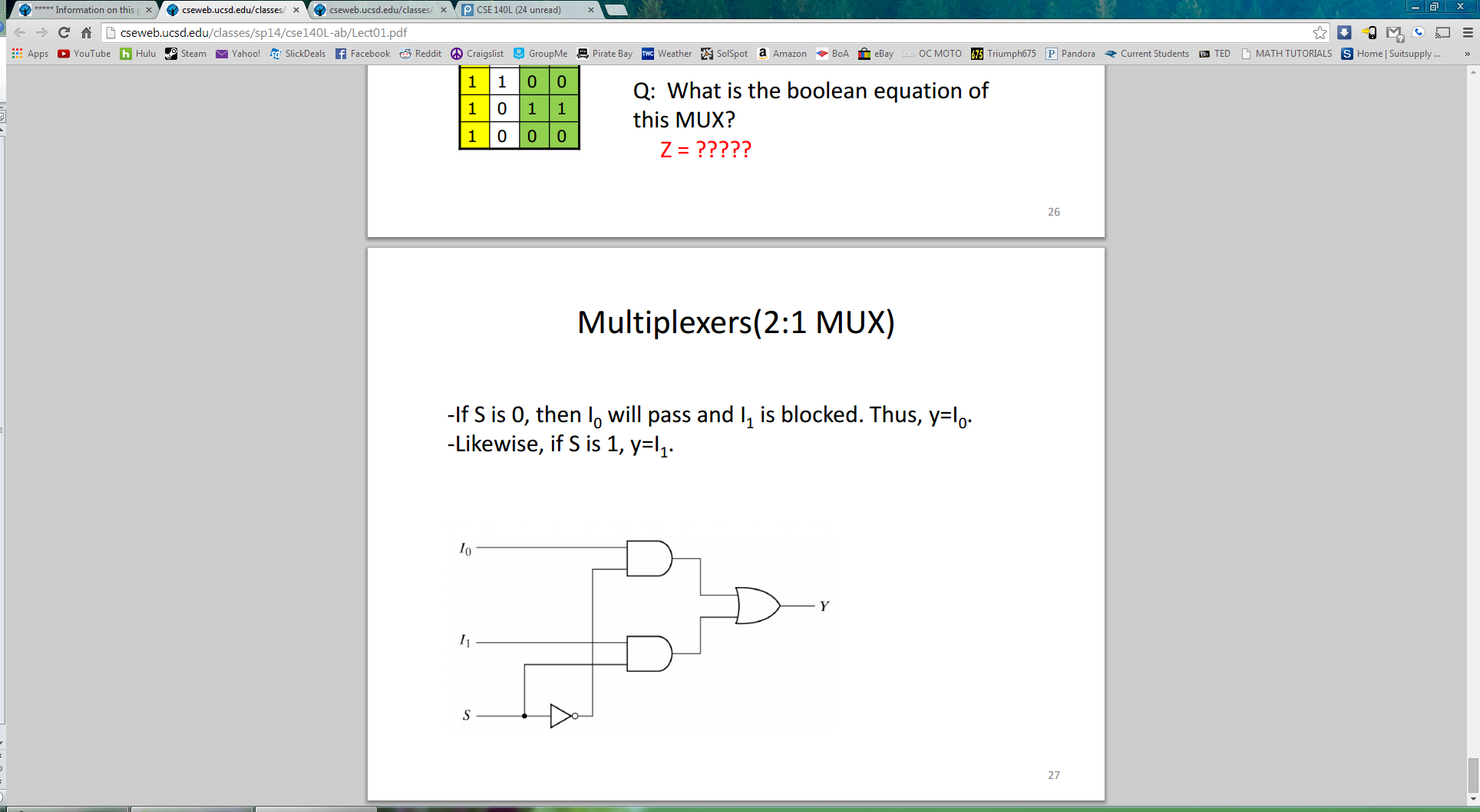
\_\_\_\_\_\_\_\_\_\_\_/**5** \_\_\_\_\_\_\_\_\_\_\_\_\_/**5**

**TOTAL Score:** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/10**

1. For part 1 I used the input from SW0 and split the path into two. One if the switch is positive and one if the switch is negative. When it is negative the switch will have an output that takes off the middle segment in the 7 segment LED display in HEX 1 2 and 3 which will give us zeros. When the switch is positive, it will do the same things but this time the result would be my CID number which is 268 by blocking out the respective segments in the 7 segment LED display.
2. For part 2 I used the truth table given to create a sum of products equation. The equation I got was (SW2`SW1`SW0) + (SW2`SW1SW0`) + (SW2SW1`SW0) + (SW2SW1SW0). Then I created the appropriate logic gates that are equivalent to the equation.



1. For part 3 I used the equivalent logic gates that mimic the MUX design. This was discussed in class and the professor showed us the circuit design for a MUX design using AND and OR gates and I just recreated the design.



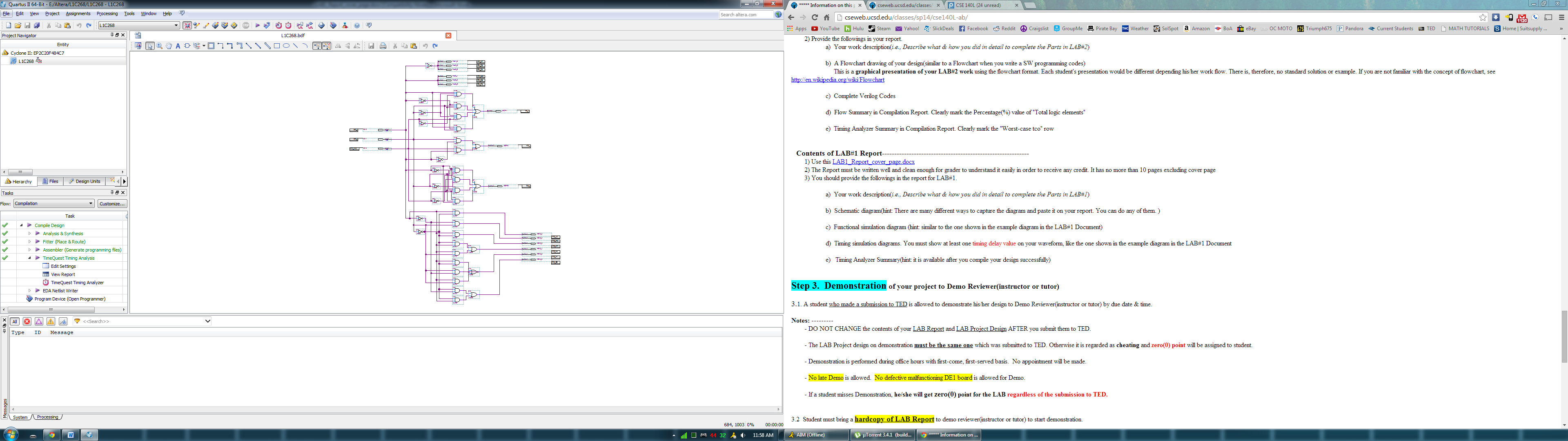
1. For part 4 I used a similar truth table like in part 2 but this time the equation was different. Since the output only equaled 1 when there is a majority of 1’s. I figured out the sums of products equation which was (SW2`SW1SW0) + (SW2SW1`SW0) + (SW2SW1SW0`) + (SW2SW1SW0). Again, I just used logic gates to create replicate the equation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SW2 | SW1 | SW0 |  | LEDR8 |
| 0 | 0 | 0 |  | 0 |
| 0 | 0 | 1 |  | 0 |
| 0 | 1 | 0 |  | 0 |
| 0 | 1 | 1 |  | 1 |
| 1 | 0 | 0 |  | 0 |
| 1 | 0 | 1 |  | 1 |
| 1 | 1 | 0 |  | 1 |
| 1 | 1 | 1 |  | 1 |

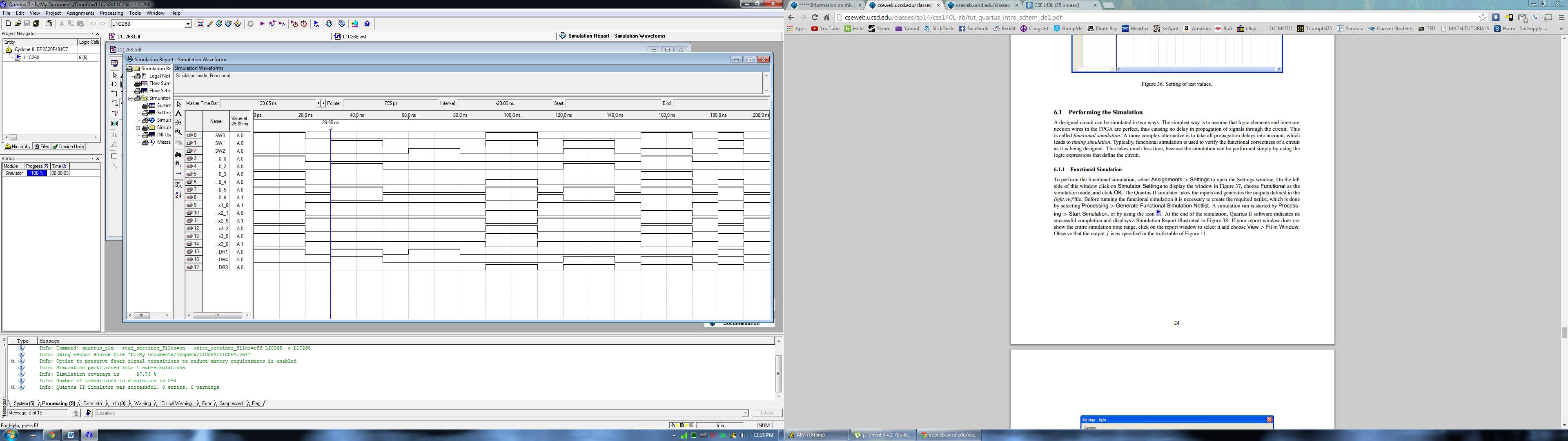
1. For part 5 I used a truth table that had SW1 and SW0 with the different segments in the 7 segment LED display of HEX0 as outputs so I knew which ones needed to be turned off or on. This gave me six equations so I just implemented them as logic gates. The equations are: 0 = SW1`SW0 2 = SW1SW0` 3 = SW1`SW0 4 = (SW1`SW0) + (SW1SW0) 5 = (SW1`SW0) + (SW1SW0`) + (SW1SW0) 6 = (SW1`SW0`) + (SW1`SW0)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SW1 | SW0 |  | HEX0\_1 | HEX0\_2 | HEX0\_3 | HEX0\_4 | HEX0\_5 | HEX0\_6 |
| 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 |  | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 |  | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 |  | 0 | 0 | 0 | 1 | 1 | 0 |

B)

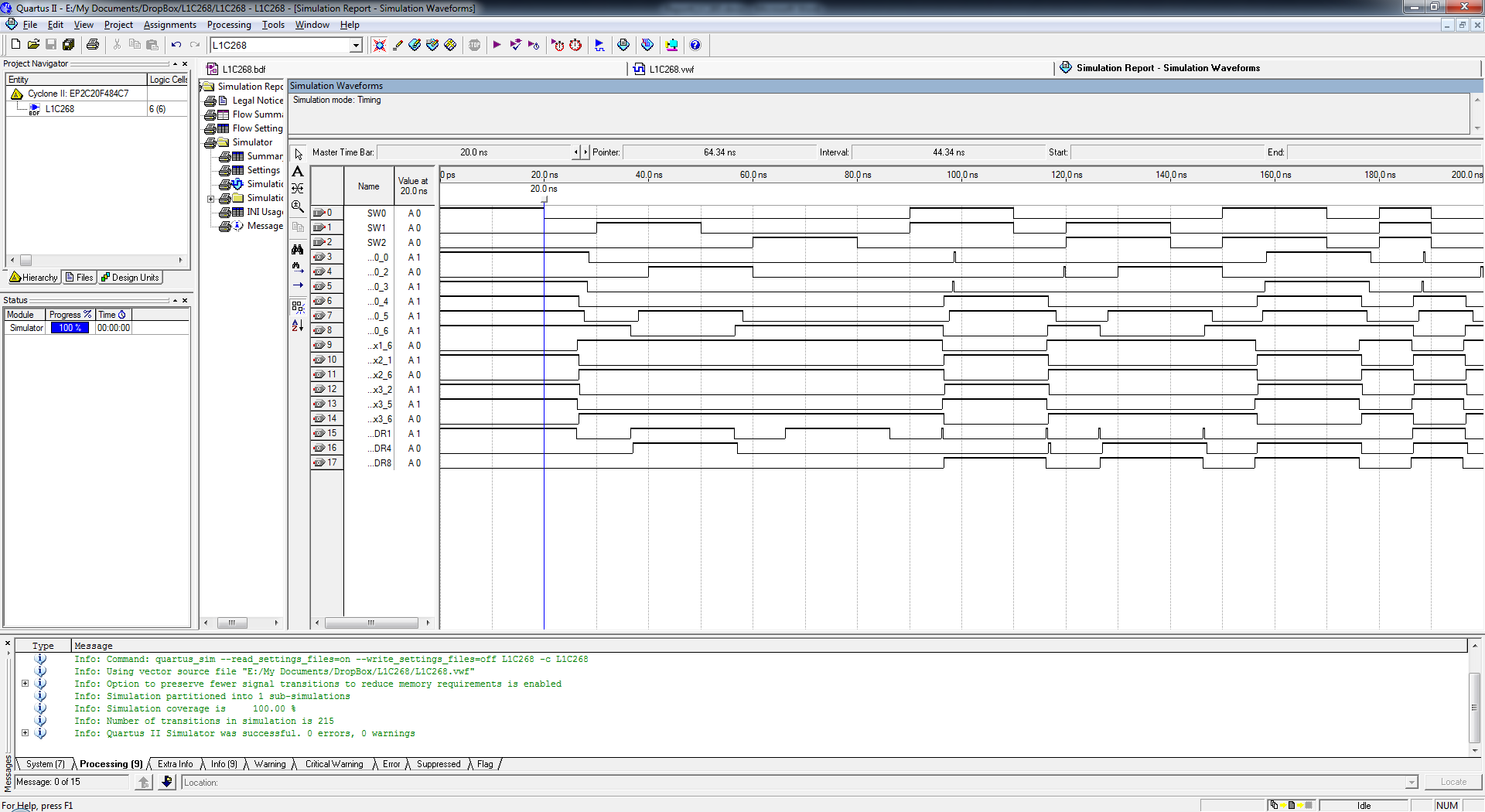


C)



D)

|\_\_\_| 🡨 8 NS delay



E)

