International University

School of Computer Science and Engineering

Digital Logic Design Laboratory

IT099IU

FINAL EXAMINATION

ANSWER PAPER

Nguyễn Trương Thanh Băng – ITITWE23034

Semester: 1

Academic Year: 2024 - 2025

Introduction

The topic is designing a traffic light system that satisfies these requirements:

1) Show how the traffic light colors change after each phase on both horizontal and vertical ways and the remaining time for each phase.
2) Have a button for resetting the system.
3) Have buttons for adjusting the time of both horizontal and vertical ways.

I used two ICs both of which are 74HC190, one JK flip-flop and some logic gates for building the logic behind with three LEDs (Red, Yellow, Green) and four BCD – 7Seg LEDs for displaying the lights and the remaining time.

74HC190 IC is an asynchronously presettable BCD decade counter. Counting occurs when LOAD is high, and the counter will be preset to the input data when LOAD goes low. Count enable (CTEN) is low as the counter will stop when it goes high, and the down/up (D/U) input is either high for down counting or low for up counting. The Max/Min or Terminal Count (called TC in SimulIDE) output is designed to indicate when the counter reaches its maximum (for count-up) or minimum (for count-down) value.

FUNCTION TABLE

INPUTS				FUNCTION
$\overline{LOAD}$	$\overline{CTEN}$	$D/\overline{U}$	CLK	
H	L	L	⌐	Count up
H	L	H	⌐	Count down
L	X	X	X	Asynchronous preset
H	H	X	X	No change

$D/\overline{U}$ or $\overline{CTEN}$ should be changed only when clock is high.
X = Don't care
⌐ Low-to-high clock transition

Approach

I solved this problem by breaking it down into three parts:

1) **Find an IC that can do the count-down, can be preset, adjusted and resettable while counting**.

The goal is here to design a system using as least number of ICs as possible but still finishing all the tasks effectively. I chose IC 74HC190 because I can set the D/U input as Low during the process so that my counter will only count down. Moreover, since it's a presettable counter, I can manipulate its data inputs and the LOAD input so that I can reset the counter to a preset number. This also benefits the time-adjusting requirement because I can set the remaining time by using the data inputs.

2) **Use the ICs and BCD 7-Seg LEDs to display the remaining time in two ways and design the resetting and adjusting part of the system.**

I approach this with the mindset of building one way first and based on that, I will adjust to build the other way.

I want to have the remaining time displayed with 2 digits so I use 2 ICs to build as each of them can only display one digit (from 0 to 9).

First, for the first IC (A), I set the D/U input as HIGH to always count down and connect a switch to the active low Counter Enable (CE) input in order to set HIGH to pause the counter when needed. A digital clock is connected to the counter's clock so that the outputs will be changed based on the signals.

Do the same with the second IC (B) but instead of connecting the same digital clock to its CLK, I connect it with the output Q3 of A so that after the first digit counting from 0 to 9, the second digit will start to count.

To display the remaining time, I connect the four data outputs of IC A to the first digit (the right BCD 7-Seg LED) and the same of IC B to the second digit (the right BCD 7-Seg LED). A switch bar consisting of 8 switches is connected to the 8 data input of them. Therefore, if I want my system to count down from a specific number, I can set it by toggling the switch bar and it will be displayed by the BCD 7-Seg LEDs.

To reset the system whenever I want, I use a push button (switch), connect it to the LOAD input which is active LOW.

However, this IC 74HC190 will also reset itself to count down from 9 after it counts to 0, so my system will reset automatically and count down from 99 after it counts from my chosen

number to 00 no matter what. To solve this problem, I will design a logic input for the LOAD so that when it resets to 99, it will reset again to count from my chosen number. I connect the outputs Q0 and Q3 of each counter to a 4-input AND gate so its output will be 1 whenever the system changes to 1001 1001 (number 99). Remember the input from the push button will be HIGH whenever I press it.
Here is a truth table to describe how these two inputs affect the LOAD input:

Output of AND gate	Push button	LOAD
0	0	1
0	1	0
1	0	0
1	1	0

Either the output of AND gate or the push button goes HIGH, the LOAD will be LOW so I use a NOR gate consisting of these two inputs and connect it to the LOAD.
Here is an example describing how it works:

I set the remaining time as 25 by toggling the data inputs. After finishing counting, instead of resetting automatically to 99, the system will reset to count down from 25 instead. Sometimes, after half of the time passes for a counting period, I want to change the remaining time to 15 immediately without waiting for it to go to 00 so I toggle the data inputs of counter B from 0011 to 0001 and push the LOAD button.

For the time adjusting requirement, toggling the 8 data inputs to change the specific time for the system to count down from as I described in the resetting part is also my solution.

So far I only work on one way of this system, to include another way, simply connect all the inputs and outputs of those two ICs to other two BCD 7-Seg LEDs. Therefore, when the traffic light on the horizontal way is counting from a specific time, the traffic light on the vertical way is also counting down from that.

3) **Design the logic to alternate the colors of traffic lights for three phases of each way and they are opposite to the other way.**

   I have the same approach to solve this problem which is designing one way first then finishing the other way based on this. I want the system to toggle from the RED phase to the GREEN + YELLOW phase whenever it finishes its one counting down period (goes to

00). Therefore, I use a JK flip flop to toggle the phases. Each phase will be turned on when all the data outputs go to 0 so I connect them to a 8-input NAND gate. When the state of the counter is 0000 0000, the output of the NAND gate will be 1. To enable the flip flop and also toggle the phase, I connect this output to all the inputs of its (J, K, CLK). The RED light is connected to the Q and the Q' will enable the GREEN and YELLOW phases.

To solve the problem how to change the GREEN light to the YELLOW light when there are 3 seconds left, I will use the Q' output, the TC output of the clock (which I already introduced before) and the Q3 and Q2 outputs of the counter A.
Here is a truth table to describe the cases in which the YELLOW light will be turned on:

Q'	TC	Q3 Q2 Q1 Q0	YELLOW
1	1	0011	1
1	1	0010	1
1	1	0001	1
1	1	0000	1

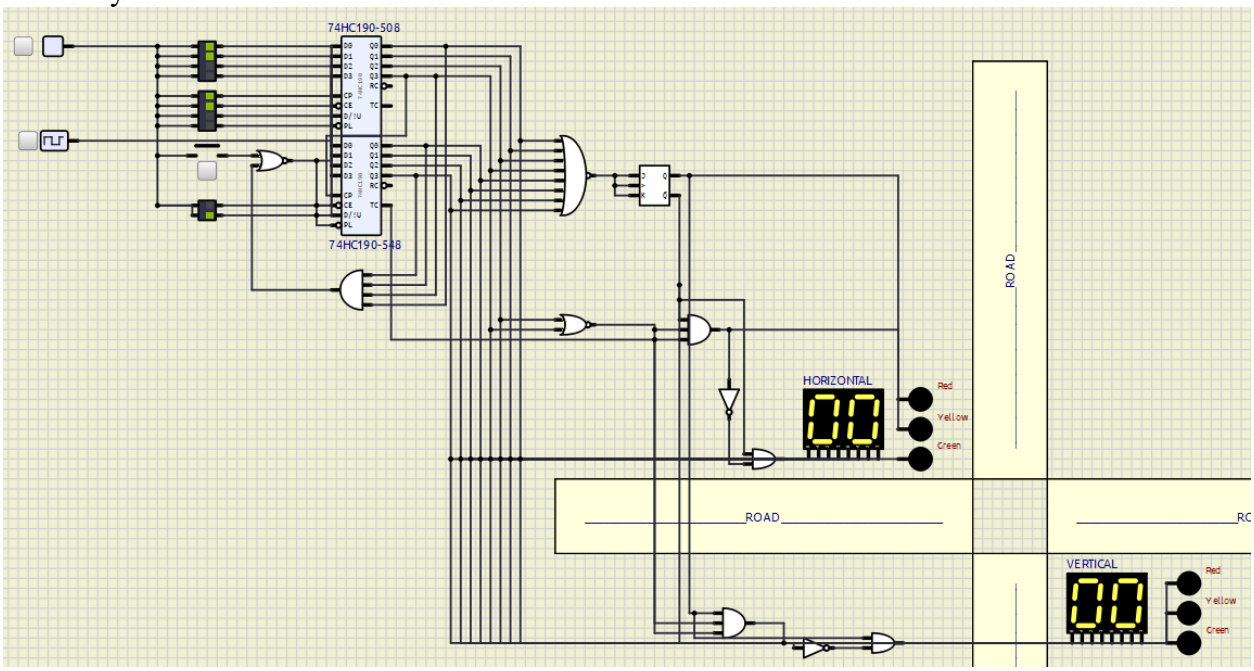
The Q' = 1 indicates when it's time for the GREEN and YELLOW phases.
The TC = 1 indicates when the second digit finishes counting.
I will connect the Q3 and Q2 inputs to a NOR gate so that the gate's output will be 1 whenever there are 3 seconds left. Finally, connecting all of them to a 3-input AND gate to enable the YELLOW phase when these three inputs are satisfied.

The GREEN light is only turned on when the YELLOW light is not enabled so I will connect the previous 3-input AND gate to a NOT gate. However, it is only turned on when the Q' = 1 so I connect the output of the NOT gate and Q' to a 2-input AND gate to avoid the GREEN light turned on when it is time for the RED light.

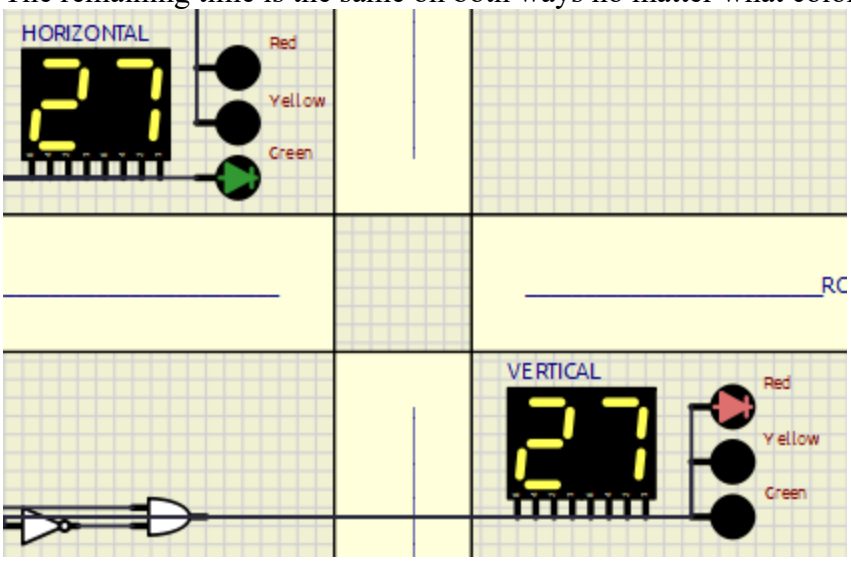
Do the same to the three traffic light colors on the other way but switch the logic applying between the RED phase and the GREEN + YELLOW phases.

Results and discussion

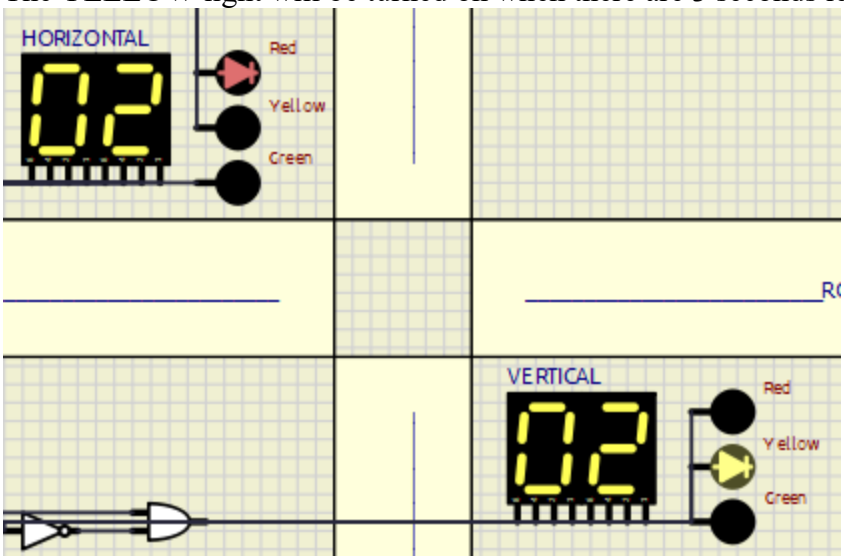
Overal system:



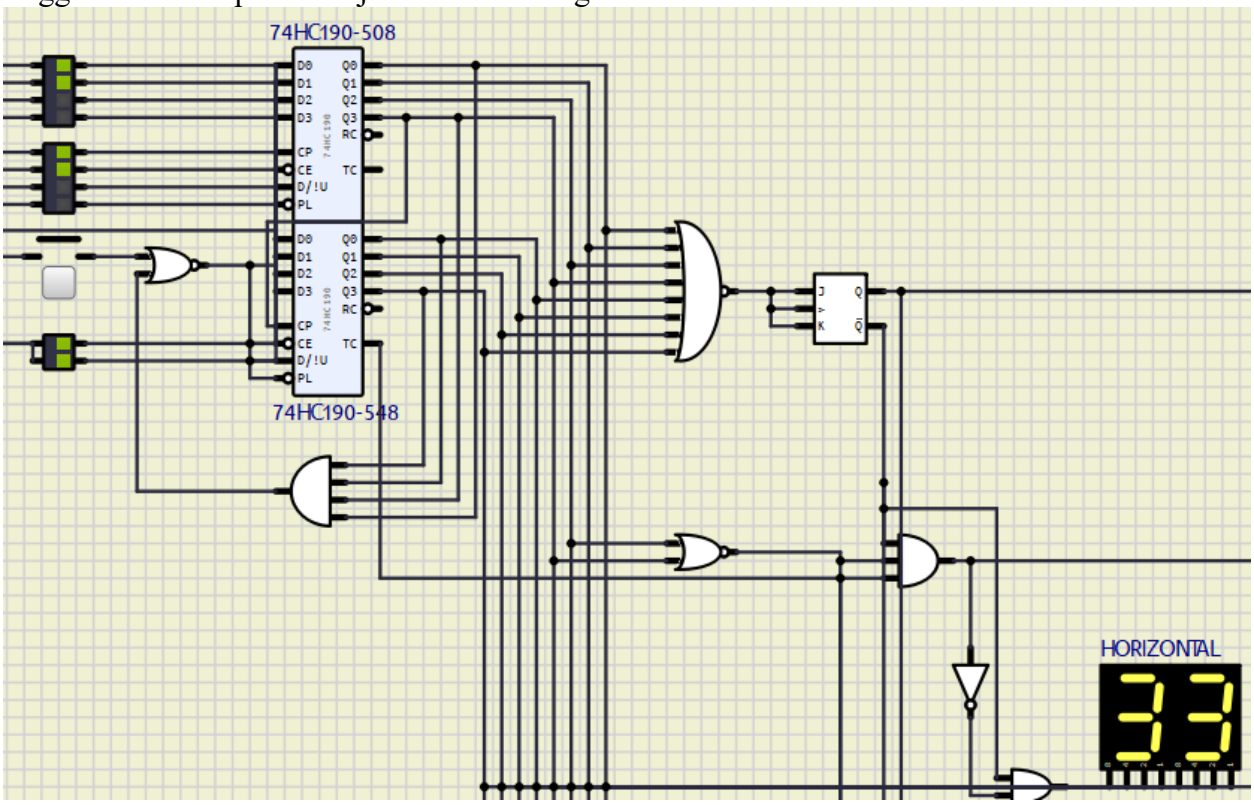
The remaining time is the same on both ways no matter what color phase they are:



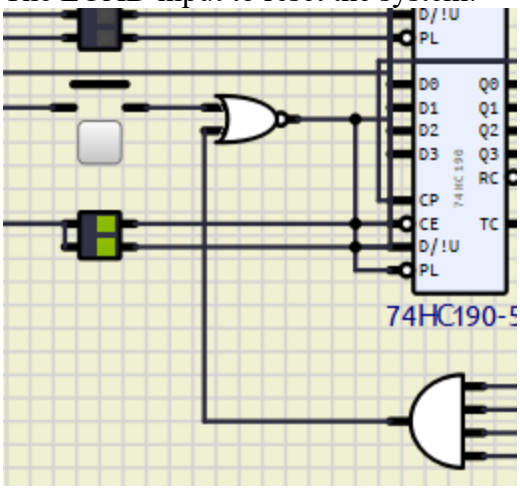
The YELLOW light will be turned on when there are 3 seconds left:



Toggle the data inputs to adjust the remaining time:



The LOAD input to reset the system:



Conclusion

For me, the hardest part is designing the function of the three colors of the traffic light and finding the right IC. To overcome that, I think we should do some research and have some ideas on the paper or drawing a draft design before starting to build the system so that we do not waste our time with the wrong ICs or struggle to come up with the logics on the simul IDE.

Reference

CD74HC190 data sheet, product information and support | TI.com. (n.d.). https://www.ti.com/product/CD74HC190

Texas Instruments Incorporated. (2003). *CD54/74HC190, CD54/74HC191, CD54/74HCT191 Presettable Counters*. https://www.ti.com/lit/ds/symlink/cd74hc190.pdf?ts=1733505060955&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD74HC190

THE END