

CPSC 359 – Winter 2017
Assignment 1
Navigation Sign Light Controller
Due: January 27th @ 11:59PM
Weight: 8% of final mark

Objective:

In this assignment, you will design and implement a sequential circuit that simulates a flashing traffic navigation sign for drivers.

Background:

The main job of a navigation sign is to inform drivers which direction they should navigate to due to construction or unexpected events. In this assignment you will create a circuit that will manage how led lights will be turned on and off in such a signal. This will be done in such a way to grab the attention of the drivers to the direction of the arrow it's pointing to. **Figure 1** shows how your overall circuit should look like.

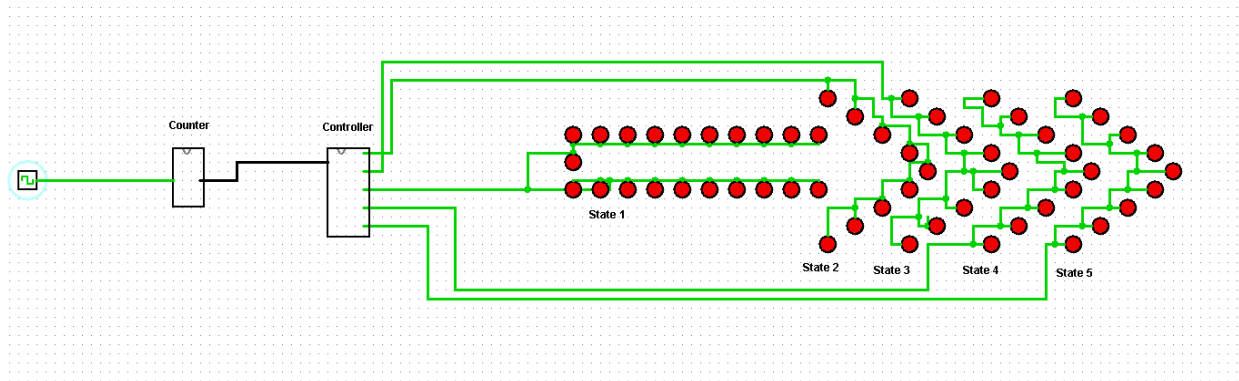


Figure 1: Circuit Components

The LEDs in the arrow of Figure 1 will be lit in the direction of the arrow in a sequential order, and then they will all flash. The order in which the LEDs are lit is done in five stages (indicated as states in Figure 1). The order in which the arrow is animated depends on a “tick” value and is performed as follows:

1. When tick 1, All the LEDs of State 1 are lit
2. At tick 3, State 2 LEDs are lit; they stay lit with State 1 LEDs until tick 4
3. At tick 5, State 2 LEDs are turned off and State 3 LEDs are turned on (State 1 LEDs remain lit) until tick 7
4. AT tick 7, State 3 LEDs are turned off and State 4 LEDs are turned on (State 1 LEDs remain lit) until tick 9
5. AT tick 9, State 4 LEDs are turned off and State 5 LEDs are turned on (State 1 LEDs remain lit) until tick 11
6. AT tick 11, all LEDs are turned on.
7. At tick 12, all LEDs are turned off
8. At tick 13, all LEDs are turned on
9. The next tick after 13 is tick 0 (that is, these steps are repeated indefinitely)

Figure 2 shows in details when states 2-5 should be activated. When the tick value is between 3 and 10, inclusive, one state (from 2-5) at a time should be active. When the tick value is equal to 11 all states from 1-5 should be activated. In order to make the LEDs flash, you should turn off all LEDs when tick is 12 and turn them on again when it is equal to 13.

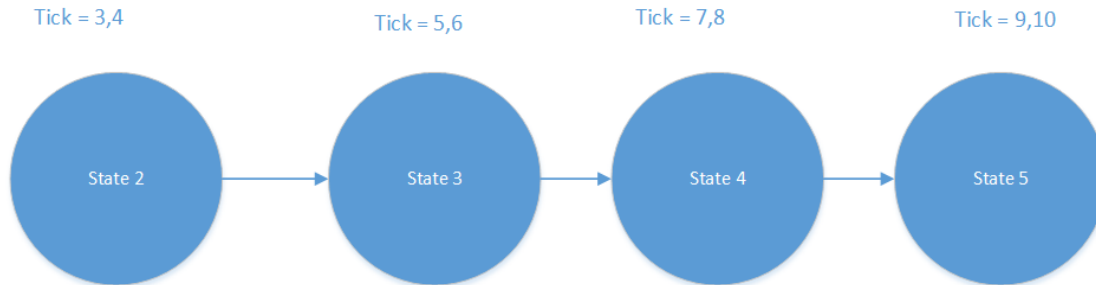


Figure 2: Transition between States 2 to 5

Rules:

- State 1 should always be turned on unless tick is 0 or 12.
- States 2, 3, 4 shouldn't be turned on together unless tick is greater 10.
- All LEDs should be off when tick is equal to 12 or 0
- All LEDs should be on when tick is equal to 13 or 11
- Tick is incremented as $\text{tick} = (\text{tick} + 1) \% 14$

Components:

- **LED:** You can use the built-in LED component available in Logisim to simulate the navigation arrow.
- **Clock:** You can use the built-in clock component available in Logisim.
- **D-Flipflop:** You can also use the built-in D-flipflop in Logisim
- ☐ **Navigation sign controller:** Build a light controller managing the states of the navigation arrow of Figure 1.
- **Navigation sign counter (tick):** Counts from 0 to 13, adding one at a rising edge of a clock, the output is in the form of 4-bit binary number as follows. Reset to 0000 when the counter reaches 1101. All lights should be turned off when counter is 0000.

Notes:

- You will build the **states of the lights** using a sequential circuit (Finite State Machine)
- All steps for creating the **Navigation sign controller** must be reported in a pdf report.
- Using the available components in Logisim is not permitted except for wiring, gates, clock, LED, and D-flipflops. You can use any color for the LEDs.

Report: Write a report that shows your steps for building the 4-bit counter and the controller, including the truth tables, the Boolean functions resulting from the truth tables, and the simplification steps of the functions.

Deliverables:

1. Design and implement the corresponding circuit in Logisim.
2. Use building blocks to create the corresponding components (Your design should be modular). See Figure 1.
3. A description of all steps of building the **Navigation light controller** must be included in a PDF report.

Marking Guide:

Navigation Sign Controller	7
4-bit Counter	5
Using Blocks	2
Labeling	1
Building Circuit	2
Satisfying the Rules	3
Written Report	4
Total	24

Teams: You are advised to work with another student in class in order to complete the assignment, but you are not required to do so. You and your partner must be in tutorials taught by the same TA. Peer evaluation in teams may be conducted.

Submission: Submit your .circuit file and the PDF file to the drop box on D2L. Only one submission per team is required.

Late Submission Policy: Late submissions will be penalized as follows:

-12.5% for each late day or portion of a day for the first two days

-25% for each additional day or portion of a day after the first two days

Hence, no submissions will be accepted after 5 days (including weekend days) of the announced deadline

Academic Misconduct: Any similarities between assignment submissions will be further investigated for potential academic misconduct.

Code sharing with a different group is prohibited. Code sharing includes looking at others' code on paper and on the computer screen. Discussions with other groups can only be carried out at the concept level. While you are encouraged to discuss the assignment with your colleagues, your final submission must be your team's original work. Any re-used code in excess of 10 lines must be acknowledged and cited. Violation of this policy may be considered academic misconduct. If unsure, always check with your instructor or TA.

For the purpose of the assignment, the term **code** should be interpreted as "**circuit**". A line of code should be interpreted as a "gate".

D2L Marks: Any marks posted on D2L or made available using any other mean are tentative and are subject to change (after posting). They can go UP or DOWN due to necessary corrections.