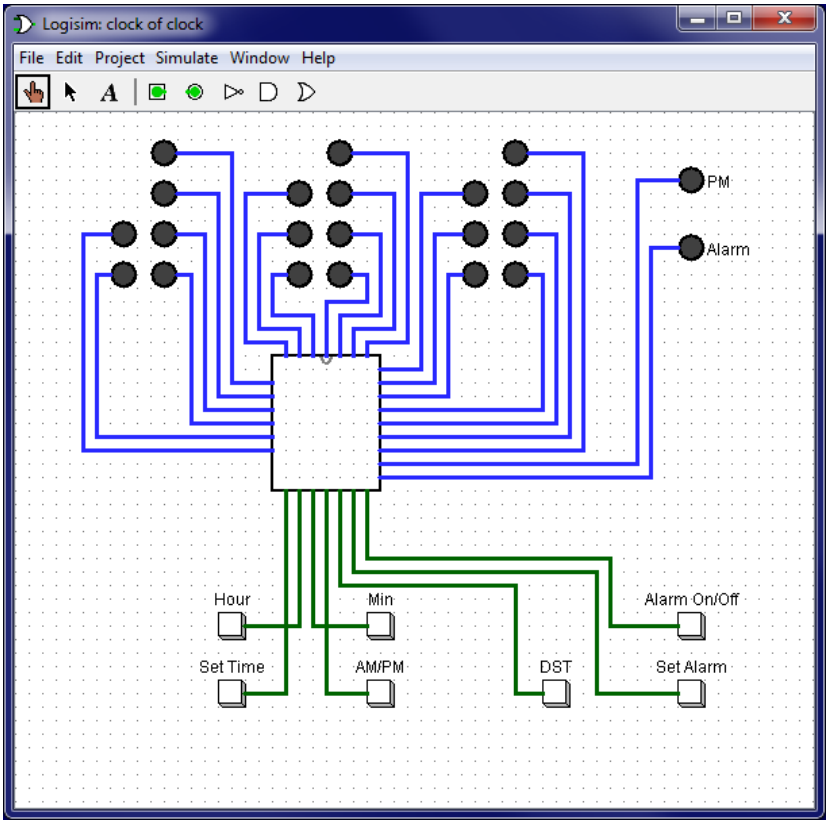
Intro to Computer Science
Logisim Binary Clock Project

The purpose of this project is to build a simulated "Binary Clock", similar to the one on top of the whiteboard in the class as a Logisim circuit. User input must be from "Buttons", and output via LEDs. The top level circuit must include ONLY your Buttons, LEDs, and a single sub-circuit that drives them all. It must include labels similar to what you would find on a clock to allow the user to operate it. Here's one possible way of doing this, but you may invent your own too.



This circuit shows Buttons and LEDs required for all Milestones below; you only need to provide those required at each Milestone. Note that Logisim only allows you to press one button at a time.

The Milestones for this project are listed below. All circuits should use a Clock input for normal operation. They are cumulative; for example, Milestone 5 must include all the functionality of Milestones 1-4.

1	4-bit Counter	Use a Clock input to cause four output LEDs to count repeated from 0 to 15 (in binary). You may NOT use the Logisim "counter" chip, but you may use any of the other memory components you wish. *Hint: T(oggle) flip-flops are one approach.*
2	BCD Counter	Use a Clock input to cause four output LEDs to count from 0-9 in binary. This is commonly called "Binary Coded Decimal" or BCD.

		Make sure each transition, including 9 to 0, takes the same number of clock ticks. No "interim states" should be visible to the user.
3	Second Counter	Use 7 LEDs (3 for 0-5 and 4 for 0-9) to count seconds from 00-59. All transitions should take one second, as defined by the Logisim "Tick Frequency". No "interim states" should be visible to the user, including 59-00, 09-10, 19-20, etc.
4	Basic Clock	Use the 20 LED arrangement shown above to count seconds from 00:00:00 to 23:59:59. Test your transitions by manually setting the state values in your flip-flops to cause important hour and minute transitions, and by increasing the Tick Frequency.
5	Set Time	Set the hours and minutes using Buttons. You do not have to set the seconds. You may use any approach you wish for this, but the user should be able to set any time in a minute or less. *Note that Logisim only allows you to press one Button at a time.* The clock design above suggests pressing the "Set Time" button, and then advancing through the hours quickly by holding down the "Hour" button, and doing the same for minutes.
6	Power Up	The clock initially goes into a "power fail state" displaying some special, attention getting pattern and requires that the user set the time before it starts ticking.
7	DST	Use a Button to adjust for Daylight Savings Time quickly without having to actually set the time. The design above suggests pressing the DST button once to advance the hour, and a second time to count it back, or holding it down and having it alternate between the two slowly.
8	Set Alarm	Use Buttons to set an alarm time in hours and minutes and display it on request. The clock should continue to keep accurate time while the user is setting the alarm. Separately, the user can turn the alarm on, or off, and the clock should indicate with an LED that the alarm is on. If the user turns it off, it should still remember the set time so that turning it on again does not require entering in the alarm time.
9	Alarm	If an alarm is set, the clock does something attention-getting when the alarm time is reached until the user turns off the alarm. Note that this should NOT clear the set alarm time; it just disables it. The design above suggests the user must hold down the Alarm On/Off button to turn it off.
10	Snooze	Instead of turning off the alarm, the user can cause it to "snooze" for 10 minutes, and then go off again. They can keep doing this for as many alarm/snooze cycles as they want. This should NOT clear or change the set alarm time. The design above suggests the user can whack any button momentarily to accomplish this.
11	AM/PM	Give the user the option to change to AM/PM display instead of 24-hour time. This means the clock cycles from 12:00 to 12:59 to 1:00 to

		11:59AM to 12:00PM, etc. The Set Time and Set Alarm features must work in AM/PM version if selected by the user as well.

You should submit only one .circ file for a Milestone, but it may include as many subcircuits as you wish.

For full credit for a milestone, the submitted .circ file must include:

a) A "MilestoneX" circuit as the top level circuit
b) Any subcircuits required; no additional files should be needed (this means you should probably start each new milestone with a copy of your previous one)
c) All inputs and outputs with component labels indicating their purpose
d) A text block in the MilestoneX circuit that provides a standard copyright notice to claim the work as your own.
e) Additional text blocks providing user operation instructions (for example, "click the DST button to switch between daylight savings time and standard time")
f) No "vestigial" circuits that are not actually used
g) No duplicate circuitry that could be eliminated by an appropriate subcircuit

In addition to your circuit, you must also deliver a Google Doc in your shared CS folder named **Theory of Operation**. This document must contain a section for each Milestone with the following:

1. Explain how your circuit works to someone knowledgeable about Logisim, but not your specific circuit (for example, MrH!).
2. List any sources (web of otherwise) you used, and briefly state what ideas you incorporated from it.
3. Explain and list sources for any new subcircuits you built to support the MilestoneX circuit.