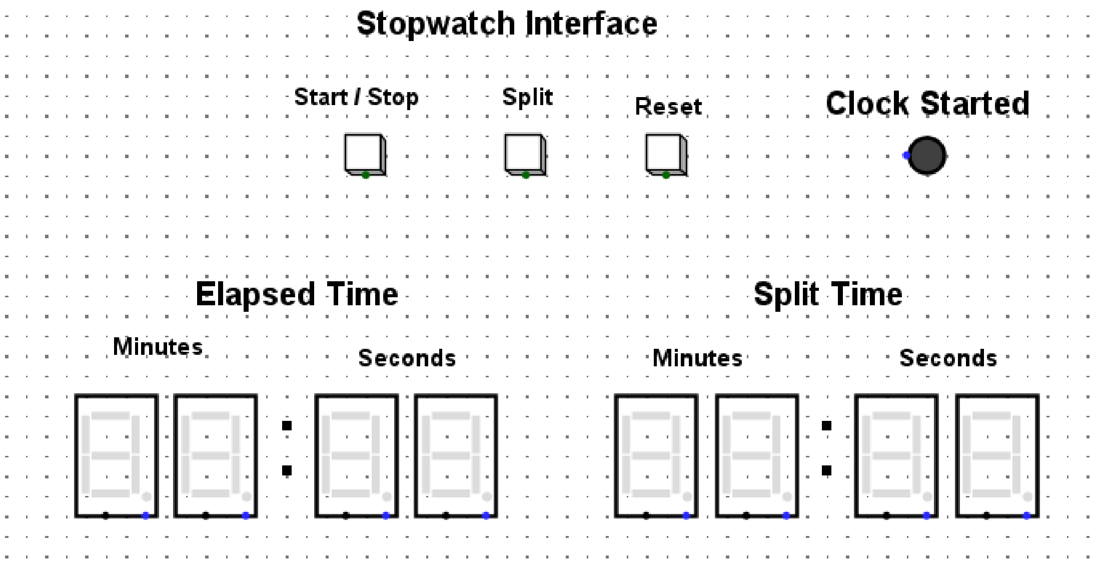
Assignment 1 - Digital Stopwatch



This assessment is an individual assignment. For this assignment, you are going to implement the functionality for a simple stopwatch interface as shown above. The interface itself is already provided as a Logisim file named main.circ . *Your assignment must be built using this file as the interface*. You are encouraged to use multiple circuits to implement specific components however your whole circuit must be operable and displayed using the buttons and display components in this file. That is, your solution should not require any additional interface components (e.g., buttons, displays, pins etc) in order to test the functionality of your circuit.

Logisim Version

Your assignment must be implemented using Logisim Evolution 3.9.0, which can be downloaded from: https://github.com/logisim-evolution/logisim-evolution/releases

This is the version we will test with, and we will not be using any other version, or making special accommodations. If your solution is incompatible with ours, it will not be able to be tested and thus will be ineligible for most marks on offer.

You can verify that your version is correct by loading the provided main.circ file with the interface as shown above.

Allowable Logisim Components

Only the following components may be used to develop your solution:

- Logic Gates: any
- Flip Flops: JK, D, S-R, T
- LEDs
- Clock (only one)
- Hex Digit Display (already provided in interface)
- Buttons (already provided in interface)
- Pins (for connecting circuits)
- Constants (for setting inputs that will not change)
- Splitter (for using HEX Digit Display)

The use of any other components *will be penalised* – in particular, you must not use any pre-built circuits such as registers, shift registers, etc).

Assessment

The assignment will be assessed according to the following criteria:

- Successful completion of each stage according to the specified requirements (30% of total mark)
- Quality of the solution (10% of total mark)
  - Clarity and readability of circuit layout
  - Modularity and use of appropriate circuit components
- In-lab interview (60% of total mark)
  - Quality and clarity of demonstration/presentation
  - Depth of understanding of implementation (including question answering) and critical reflection on the design and implementation
  - Explanation of solution's origins including use of Gen-AI, online resources

The weighting of marks reflects the focus of this assessment, which is on your ability to develop, explain and justify design choices for a simple digital logical circuit.

With respect to your solution, to get full credit for each stage of completion, it must work as described and will be tested using Logisim 3.9.0 by one of the teaching team.

Interview

An interview of approximately 10 minutes will be conducted after the submission date to see your solution in action provide you an opportunity to answer questions, explain design decisions made, and demonstrate your understanding of the concepts you have applied. The exact date, time and location of these interviews will be advertised on Canvas.

The high weighting of this assessment item reflects our focus on your ability to explain and justify the solution you have provided. This means a strong mark can still be achieved even if not all stages are implemented. For students who do not complete all stages, you should come well prepared to answer questions regarding what you would have done for later stages, and an ability to explain key concepts relevant to these stages if you are wanting to be considered for higher marks for the interview.

Assignment Stages

To break the problem down, you will implement the functionality of the stopwatch in stages. Each stage has a percentage weighting of marks contributing to the overall total of 100% You should implement each stage in order, and upon completion of each stage, save your file using the naming convention: stageX.circ.

This assignment will take a long time to complete in full. You should start early and work methodically through each stage. Note however that a good mark for this assignment does not require all stages being complete. In particular, Stages 6 and 7 are considered more advanced,

Stage 1: Implement the Start/Stop button (15% of Completeness marks)

Using the Start/Stop button provided in main.circ, wire up a simple circuit that toggles between the *Start* and *Stop* states every time the button is clicked. Your solution should:

- Use the "Clock Started" LED and ensure it is turned on when in the Stopwatch is in the Start state, and off when in the Stop state,
- Make use of a Flip Flop to keep track of the current state.

Stage 2: Implement a single digit "Seconds" display (30% of Completeness marks)

This stopwatch will provide 1 second precision, and so needs to display the number of seconds that have elapsed since the Start button was pressed. As such, this will require the implementation of a counter. You will start by implementing a single digit "Seconds" counter for the units column of the "Seconds" display. That is, a counter that increments the "Seconds" display by 1s every clock tick, between "0" and "9". Specifically:

- Replace your flashing LED in Stage 1 with a counter that keeps track of the number of "seconds" (in increments of 1s, between 0s and 9s).
- The "Seconds" display should start from "00" when the Start/Stop button is first pressed
- The "Seconds" display should Stop when the Start/Stop button is pressed in the Start state.
- The "Seconds" display should resume counting when the Start/Stop button is pressed in the Stop state

Your counter must be *based on counters described in this unit's lectures/videos.* For this stage you can assume a single clock pulse equals 1s, and the display will only show the units column in seconds. As such, your "Seconds" display should only show values: "00, 01, 02, 03, 04, 05 …09", and then wrap back to "00".

Stage 3: Implement the full two-digit "Seconds" display (50% of Completeness marks)

You're now going to implement full "Seconds" display for your stopwatch. Modify your circuit so that:

- the seconds display now shows "Seconds" in 1s increments using both the units and tens column. That is, the display will now show values: "00, 01, 02 …. 57, 58, 59", and then wrap back to "00".
- the display resets to all zeros whenever the Reset button is clicked, and enters the Stop state (i.e., the Elapsed Time remains 00:00).
- Your circuit should explicitly ensure no illegal values are displayed (e.g., no hex values displayed or digits above "5" in the tens column, etc).

Stage 4: Implement the "Minutes" display (75% of Completeness marks)

In this stage you will implement the remaining time display components (i.e., "Minutes"). These are described in two sub-stages below. Implement a "Minutes" display using the *two hex digit displays* labelled "Minutes". Specifically, your "Minutes" display should:

- display decimal values only from "00" to "99", and then wrap back to "00"
- only increment when the "Seconds" display is wrapping back to "00" (and this should be at the same time).
- ensure the Start/Stop and Reset buttons work for the "Minutes" display as they do for the S

Stage 5: Implement the "Split" button (100% of Completeness marks)

Most stop watches provide a "Split" button that allows intermediate times (i.e., lap times) to be recorded and displayed. In this stage you will implement the "Split" display. For this you will use the second "Split Time" display as shown on the Interface above to show the stopwatch time at the time the "Split" button is pressed.

Specifically, your circuit should:

- display the elapsed time on the "Split Time" display at the moment when the "Split" button is pressed, *and only when the stopwatch is not in the Stop state*
- Ensure the displayed "Split" time remains displayed and unchanged until the next time "Split" is pressed, or the "Reset" button is pressed.
- If the Reset button is pressed, the "Split" Time display should read "00:00"

Note that the "Split" button should not impact the "Elapsed Time" display. It should continue to count as normal.

Submission

Your completed submission must be made through Canvas - (Go to Assignment 1 under "Assignments" before the due date/time).

Everyday day late will incur a *10% deduction of the original marks available*.

Each *submission must be a zip file containing*:

- the actual Logisim file (.circ source file) for testing. This MUST be labelled main.circ and implement the most complete version of your assignment.
- In addition to the above, you must also include a Logisim file for each stage of your solution's development, each be labelled stageX.circ (i.e, ensure you save your solution as a separate file before moving to the next stage).
- a Solution Development Log (see below)

Solution Development Log

Your submission must include a log of all resources used to help develop your submitted solution, with specific explanations as to how each resource was utilised. Resources that must be logged include:

- This unit's learning material, including specific reference to modules and video lectures/labs.
- Online resources
- Generative AI tools/models, including which specific AI tool/model was used, and a log of all queries used, including date and time.

A template for this document is provided on Canvas.

Use of Generative AI and online resources

You are permitted to make use of Generative AI and online resources to assist you in developing your solution, however, it is imperative that all such resources and/or Gen-AI queries are logged and explained in your solution's development log (see above), a template for which is provided on Canvas. You must use this template and must include it as part of your submission. Submissions without this document will receive no marks for this assignment.

In your interview you will be assessed on the basis of your understanding and justification of the solution you provide – that is – the solution you have developed. You will also be assessed on how well you are able to explain how the resources you refer to in your solution development log were utilised to inform your solution.

Academic Integrity

This is an individual assessment task and it is required that you work alone on your solution for assignment. This means you *must not share your solution with any other student*, *or make any part of your solution available online*. Markers will be cross-checking work and will expect to see progress being made on assignments during the dedicated lab classes.