

Assignment 2

Due: 11:59PM, Friday May 26th

Digital Stopwatch (in ARMLite)

In this assignment you will implement the Digital Stopwatch as specified in Assignment 1, in the ARMLite ARM Assembler. Your aim is to replicate the functionality of the stopwatch as described in Assignment 1 of this unit, as closely as you can, in the ARMLite simulator. You will also demonstrate your working solution in a video you create of no more than 5 minutes, showing all the features you have implemented (and explaining how it works). See further details below.

Assessment

This assignment is worth 15% of your total grade for this unit.

Your solution will be graded on the basis of three key criteria:

1. Level of Completion (50%)
2. Quality of solution (style, commenting, layout) (35%)
3. Demonstration (clarity and depth of understanding of solution) (15%)

Each of these is outlined in more detail below.

Levels of Completion (50% of marks)

Below aims to provide you some guidance on how the 60% of marks for this criterion will be awarded. Note that solutions may implement partial elements of any of these and will be assessed on the merit of what is ultimately demonstrated in the video.

1. *Stopwatch Beginnings*: a program which displays Seconds incrementing in the text output display area in a sensible and perceivable way (i.e, so we can see each tick!) from when the program launches and allows the timer to be paused and un-paused. (20/50)
2. *Stopwatch Basics*: a program which achieves all the above, and shows time ticking in real-time, and also offers the ability to start/pause and reset the timer. (30/50)
3. *Stopwatch with Benefits*: a program which achieves all the above, but also allows a split time to be displayed when requested (e.g., when “s” is entered), while also continuing to show the timer (As per Stage 5 in Assignment 1). (50/50)
4. ***Stopwatch with Buffering***: a program which achieves all the above, and also keeps track of the last 5 “Split” times during a single timed activity. The solution should show only one split time at a time, and thus allow the user to view each split time in turn (as per Stage 6 of Assignment 1). **(Bonus +5%)**
5. ***Stopwatch of Beauty***: a program that implements a stopwatch proper! – that is, all of the above, but displays all the digits on the ARMLite display screen (rather than the text output field), and replicating as closely as possible the functional requirements of Stage 7 of Assignment 1. **(Bonus +5%)**

Quality of Solution (35 % of marks)

This criterion will consider the layout and readability of your code, but also the quality of the solution. That is, the extent to which your solution exhibits sound use of relevant data structures such as variables and arrays, the stack, as well as the functional decomposition (i.e., use of functions) to implement the solution, and your solution’s adherence to ABI conventions.

Video Demonstration (15% of marks)

You will create a video of no more than 5 minutes demonstrating your solution. In this, you will be expected to show all the features of your working solution. It should also include some highlighted selected examples (no more than 2) from your code demonstrating how you have applied key concepts from the unit in your solution. Note that your program will also be tested independently of this video, and so must work as demonstrated in the video.

Submission

You will submit a single zip/z7 file with the following items on Canvas by **11:59PM, Math 26th**:

1. The project source file that implements the solution for execution in the ARMLite simulator.
2. A video file or document with URL to an online video (and any relevant video access details if required)
3. A written report of no more than 5 pages providing an overview of your solution and design, clear instructions on how to run and use your program, and any issues/bugs that you are aware of (voluntary disclosure of errors may reduce penalties associated with them).

Academic Integrity

This is not a team project, and it is expected that the solution provided is your own design and derived from your own understanding of the concepts. Anything submitted is thus expected to be your own work, and explainable by you, noting that you may be asked to attend an interview with a panel of the teaching team if we feel further information about your solution is required. Any cases of plagiarism, code sharing or obtaining of code from on any online platform, or contract cheating will receive a grade of 0 for this assignment, and possible further disciplinary action. An inability to properly explain your solution should you be asked to attend an interview will also result in a 0 grade.

Getting Help

You are reminded that this unit has online consultation times scheduled throughout the week. The schedule is linked from the Canvas unit home page. Outside of your lab classes, this is your next port of call for help, along with a dedicated Discussion Board for this assignment. You are advised to start early and allow plenty of time for debugging, and seeking any help you might need.