A black and red sign with white text

Description automatically generated with low confidence

Swinburne University of Technology

Stopwatch Program – Solution Overview and User Guide

COS10004

Computer Systems

Name : Nguyễn Ngọc Anh

Student ID : SWS00627

Video Link : <https://youtu.be/iBZan4eR62s>

Table of Contents

[Overview 3](#_Toc136390457)

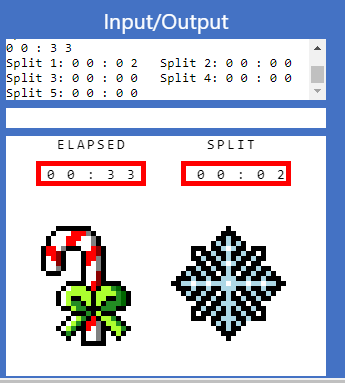
[Solution Design 3](#_Toc136390458)

[Instructions how to Run and Use the Program 4](#_Toc136390459)

[Issues/ Bugs 5](#_Toc136390460)

[Conclusion 5](#_Toc136390461)

# Overview



The Stopwatch Program is designed to function as a digital stopwatch with minute and second counters. It runs on an ARM assembly language platform which is called *ARMlite Simulator V1.2.4 © Peter Higginson 2020-23.* It utilizes memory addresses to store and manipulate the values. The program allows users to start and stop the stopwatch, as well as reset and split the time. The stopwatch time is displayed using Text Output Display and Pixel Display Area. It is displayed in the text output screen and it uses the ‘NewLine’ function that makes it look like it refreshes every second.

# Solution Design

The program is structured using subroutines for different functionalities. Here is a breakdown of the main subroutines and their purposes:

* ‘checkInput’
* Check the user’s input
* If “S” is pressed, call the ‘split’ subroutine
* If “P” is pressed, call the ‘pause’ subroutine
* If “R” is pressed, call the ‘reset’ subroutine
* ‘pause’
* Check the user’s input
* If “A” is pressed, call the ‘start’ subroutine
* If “R” is pressed, call the ‘reset’ subroutine
* Branch pause infinitely
* ‘reset’
* Call the ‘resetDisplay’ subroutine
* Update the reset value by calling ‘splitDisplay’ and ‘updateDisplay’
* ‘start’
* Call ‘cont’ to continue the program
* ‘addSecondsOneloop’
* Checks if it reaches 9, moves to the next digit (addSecondsTensLoop).
* Increments the secondOnes digit by 1.
* Calls the ‘display’ subroutine to update the increment.
* ‘addSecondsTensLoop’
* Checks if it reaches 5, moves to the next digit (addMinutesOnesLoop).
* Resets the secondOnes digit to 0.
* Increments the SecondsTens digit by 1.
* Calls the ‘display’ subroutine to update the increment.
* ‘addMinutesOnesLoop’
* Checks if it reaches 9, moves to the next digit (addMinutesTensLoop).
* Resets the secondOnes digit to 0.
* Resets the secondTens digit to 0.
* Increments the MinutesOnes digit by 1.
* Calls the ‘display’ subroutine to update the increment.
* ‘addMinutesTensLoop’
* Checks if it reaches 9, reset all the digits ( call resetDigits )
* Resets the secondOnes digit to 0.
* Resets the secondTens digit to 0.
* Resets the minutesOnes digit to 0.
* Increments the MinutesTens digit by 1.
* Calls the ‘display’ subroutine to update the increment.
* delay
* Implements a delay by calculating the elapsed time between iterations.
* Uses a loop to wait until the required delay time is reached.
* display
* Update the arguments to the main registers
* updateDisplay
* Prints the ‘minTens’, ‘minOnes’, ‘secTens’, ‘secOnes’ on the Text Output Display and Pixel Area Display and 5 split displays on the Text Output Display
* Pause and Reset
* Handle user input to pause and reset the stopwatch.
* Check for specific key codes (R for reset, P for pause) and perform the corresponding actions.
* splitDisplay
* Handle user input to split the time.
* Check for specific key code (S) and perform the action by taking the split time values that are stored in the memory locations ‘split\_minutes\_tens\_(1-5)’, ‘split\_minutes\_ones\_(1-5)’, ‘split\_seconds\_tens\_(1-5)’, ‘split\_seconds\_ones\_(1-5)’.
* splitdecoration & draw
* Used for decoration

# 

# Instructions how to Run and Use the Program

To run and use the Program, follow these steps

* Set up the ARM assembly language platform (ARMlite simulator).
* Load the program into the ARMlite simulator environment.
* Start the program execution.
* The stopwatch will start automatically and display the time on the text output screen.
* To pause the stopwatch, press the 'P' key. Pressing the 'P' key again will resume the stopwatch.
* To reset the stopwatch, press the 'R' key.
* The stopwatch will stop automatically when it reaches 99:99.
* The current stopwatch time will be displayed on the screen, with a colon separator between minutes and seconds.
* After stopping or resetting, you can start the stopwatch again by pressing ‘A’ key.

# Issues/ Bugs

* When press “R” to reset, only the Elapsed and Split1 time will be reset, the Split2 – Split 5 need multiple times of pressing “R” to reset all of it.
* Have not utilized the clock interrupt to optimize the program, thus making the program very long and using a lot of function call.

# Conclusion

In conclusion, the current implementation of the stopwatch program has shown some issues and bugs that need to be addressed. The reset function has not been optimized to reset all the split time displays at once.

Additionally, the program is very long which is approximately 500 lines of code excluding decoration due to having not utilized the clock interrupt in week 11 lessons to optimize the delay of the program and avoid many function call.

To enhance the program's usability and address these issues, modifications should be made to ensure that all the split time displays reset to zero when press “R” for only one time. Furthermore, the clock interrupt should be to fully optimize the program.

By addressing these issues, the stopwatch program can offer a more reliable and user-friendly experience, meeting the expectations of users who rely on accurate timing and split time measurements.