

Swinburne University of Technology

Computer System - COS 10004 Assignment 2

Course name	Computer System
Course title	COS 10004
Tutor name	Mr MD Kafil Uddin
Assignment name	Digital Stopwatch (ARMLite)
Student ID	104209393
Student name	Ai Vi Tran

Stage 1: Stopwatch Beginning

The objective of this stage is to allow the program which displays seconds incrementing in the text output from where user can aware of each tick. In this part, the program allows users to pause and un-pause the clock through keyboard.

Instruction:

P	Pausing the clock
U	Un-pause the clock

Functions:

1. .WriteChar: I used for writing special characters like space without using “\n” and colon in ASCII conversion chart .
2. .Time: to announce the program of the start time once the button is clicked, and the update time.
3. .WriteUnsignedNum: to display exactly the number from the register to the text output display area.
4. .LastKey: to load into the register the key code for the most recent key pressed (receiving from the keyboard).

Operation:

- ✓ Seconds: I made a comparison between 9 and R1, if it less than then loop back to nine_sec function and continue to operate. Then when it reach 9, it jumps to the next unit of second simultaneously moving 0 to R1 and continue the loop.
- ✓ Minutes: when both of the digits meet the limits that I have set for counting the second (5 and 9). The operation of this section nearly the same as how the seconds operate but with the different limits (9 for both of the digits).
- ✓ When both of seconds and minutes reach their limits, then the clock will loop back (restart) and continue to count as the beginning.
- ✓ I made the subtraction of the update time and the start time so that the clock will increase exactly 1 second once the gap between them is enough.

Here is the interface of the program:

The screenshot displays the ARMLite Simulator V1.2.4 interface, which is divided into three main sections: Program, Processor, and Memory.

- Program Panel:** Shows the assembly code for the stopwatch program. The code starts with a 'begin:' label, initializes registers R1, R2, R3, R4, R5, and R6. It then enters a loop where it updates the time, writes the start time, and calculates the difference between the current time and the start time. The code uses instructions like MOV, LDR, STR, and CMP to manage the stopwatch logic.
- Processor Panel:** Displays the current state of the processor. It shows the PC (Program Counter) at 0x00000000, the Count register at 403315923, and the Current Instruction. The Status bits are shown as NZCV, and the Input/Output section displays '0 0 : 1 8'.
- Memory Panel:** Shows the memory dump, displaying hexadecimal values for various memory locations. The dump is organized into columns for different memory addresses, ranging from 0x00000000 to 0x0000000F.

The bottom of the interface includes a 'Hex' dropdown menu and a 'Clear' button, along with the version information: ARMLite Simulator V1.2.4 © Peter Higginson 2020-23.

Stage2 : Stopwatch Basic

The objective of this stage is to offer the ability to start/ pause and reset the timer.

P	Pausing the timer
U	Un-pause the timer
R	Reset the timer

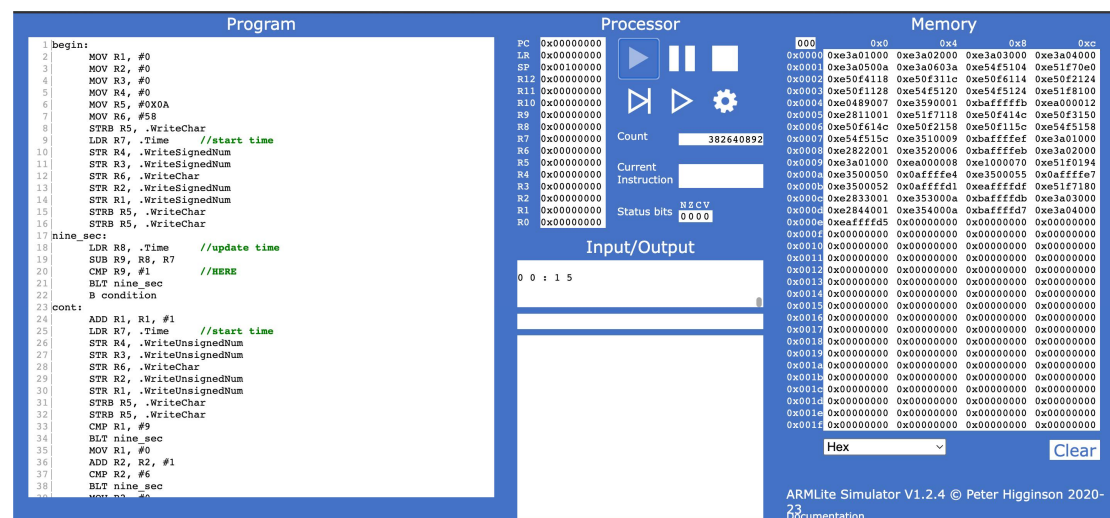
Function: In this stage, I used almost the same function as stage 1 but for different use and different operation.

1. .WriteChar: Used for writing strings and special characters which are converted to ASCII.
2. .Time: Start/ update the time to make the calculation.
3. .WriteUnsignedNum: Display exactly the number from register to the text output display area.
4. .LastKey: Loading the most recent key pressed (receiving from keyboard).

Operation:

- ✓ Following what have achieved in stage 1, in this stage I add two more .LastKey techniques to the condition of the timer, through that whenever user presses button R, the timer will loop back to the beginning and start counting again (which is the requirement of this stage)
- ✓ And also, when launching the program, the timer has not started yet, so, to start, people have to press button U (which is for unpausing as mentioned in stage 1).
- ✓ The main idea of this stage is make use of .LastKey function to reach the requirements.

Here is the interface of the program:



Stage 3 : Stopwatch with benefits

The objective of this stage is to split the timer whenever S is pressed.

P	Pausing the timer
U	Un-pause the timer
R	Reset the timer
S	Split the time

Function: Make use all of the functions have been used as the above stage

1. .WriteChar
2. .Time
3. .WriteUnsignedNum
4. .LastKey / LastKeyAndReset (up to what attempt)

Operation:

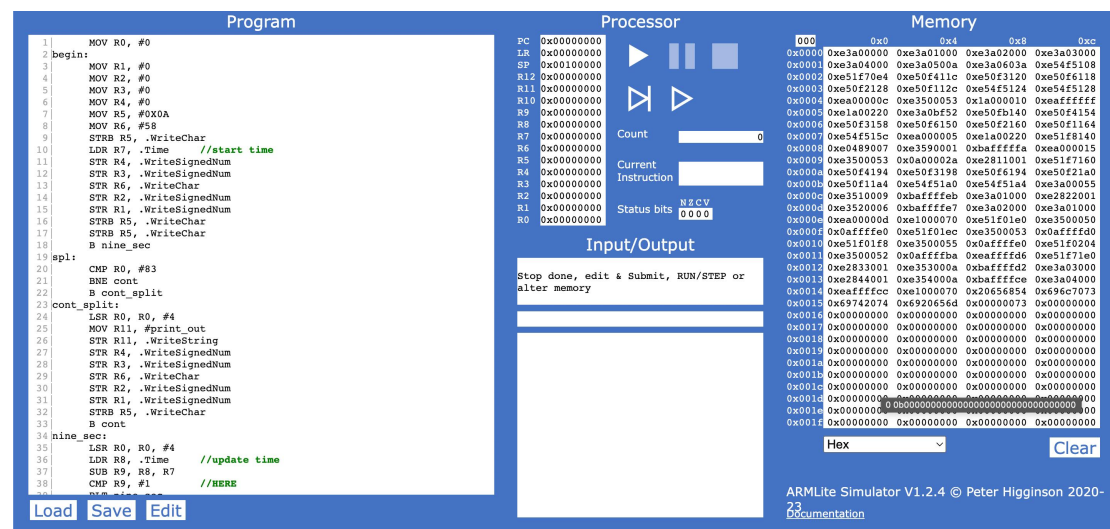
- ✓ In this stage, I added one more comparison in condition label, which will branch the program to split label whenever S is pressed when then turn back to the loop for continuing counting time.
- ✓ In this stage, I have tried many ways to have the output as requirement but it still can not. The following are some of my attempts in this stage.

Attempt:

1. First attempt:

For the first attempt, I still used .LastKey function for that, along with my logic, I think it should continue to count the time when the S button is pressed and the split time is displayed but the problem is although it keeps displaying the counting time but the split time still showing the increments in a second simultaneously with the clock counting. But if the user presses U/P then the program is running as usual.

Here is the interface of the program:



2. Second attempt:

For the second attempt, I tend to use .LastKeyAndReset to meet the requirement, along with my logical thinking, as expected, R0 should reset the information in R0 after receiving input from the user through keyboard and continuing counting the timer without keeping printing out the split time. But the problem is, the timer is paused after clicking the S button.

Here is the interface of the program:

