An Introduction to VisUAL

1. Objectives

This lab will introduce you to VisUAL – a tool for editing, assembling and simulating a subset of ARM instructions. VisUAL is used in Lab1a, HW2, and HW3 to let you assimilate the ARM assembly instruction set. While this lab takes place in class hours, your submission to Canvas has 2 days, so that you'll have a plenty of time to play with VisUAL. (Please check the submission date and time, anyway.)

2. Download VisUAL

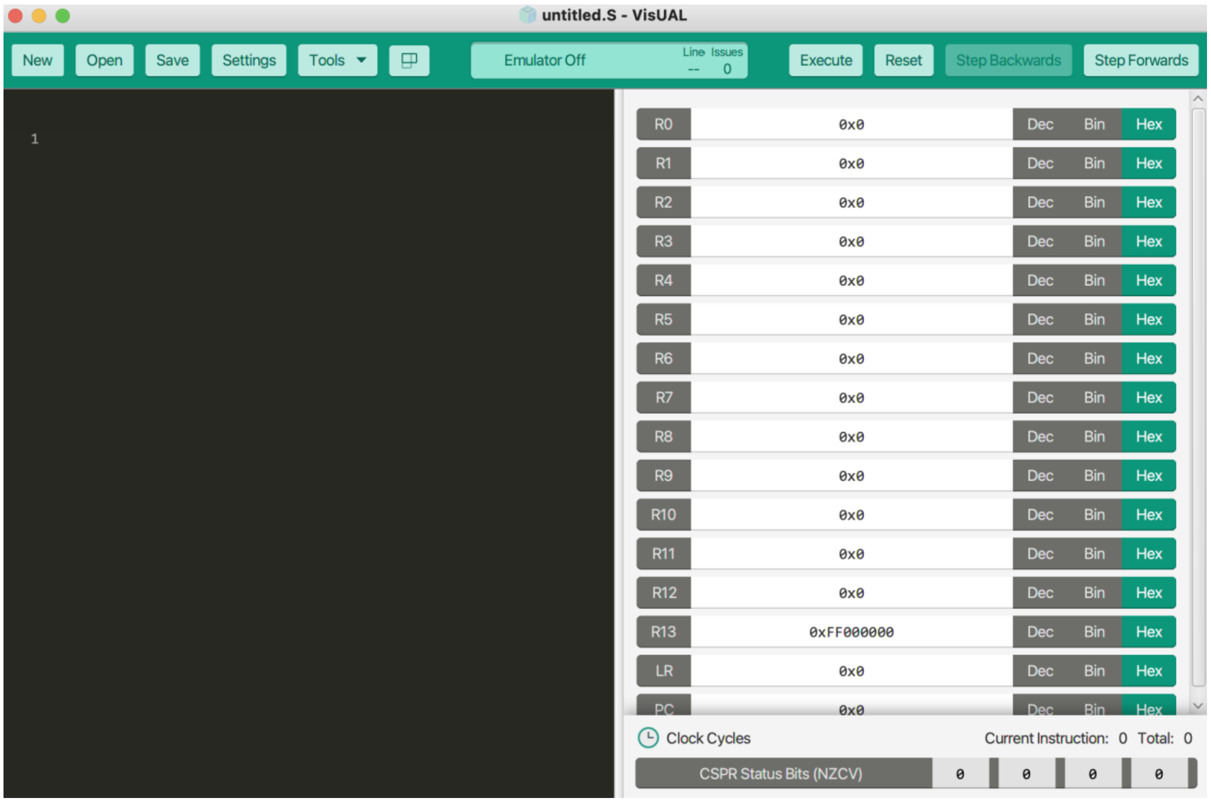
Visit https://salmanarif.bitbucket.io/visual/ and click "Downloads" to download VisUAL. Please choose the version corresponding to your operating systems, (Windows, MacOS, and Ubutu). Follow the installation instructions.

3. User Guide

From the main menu, please click "User Guide" to see VisUAL's functionality. You don't have to read all. All you need is to frequently check "List of supported instructions" when receiving simulation errors upon your code execution, as VisUAL's instruction set is quite limited.

4. Coding and Simulation

Upon starting VisUAL, you'll see the following window that allows you to code an ARM program on its left and to check the contents of CPU registers on the right.

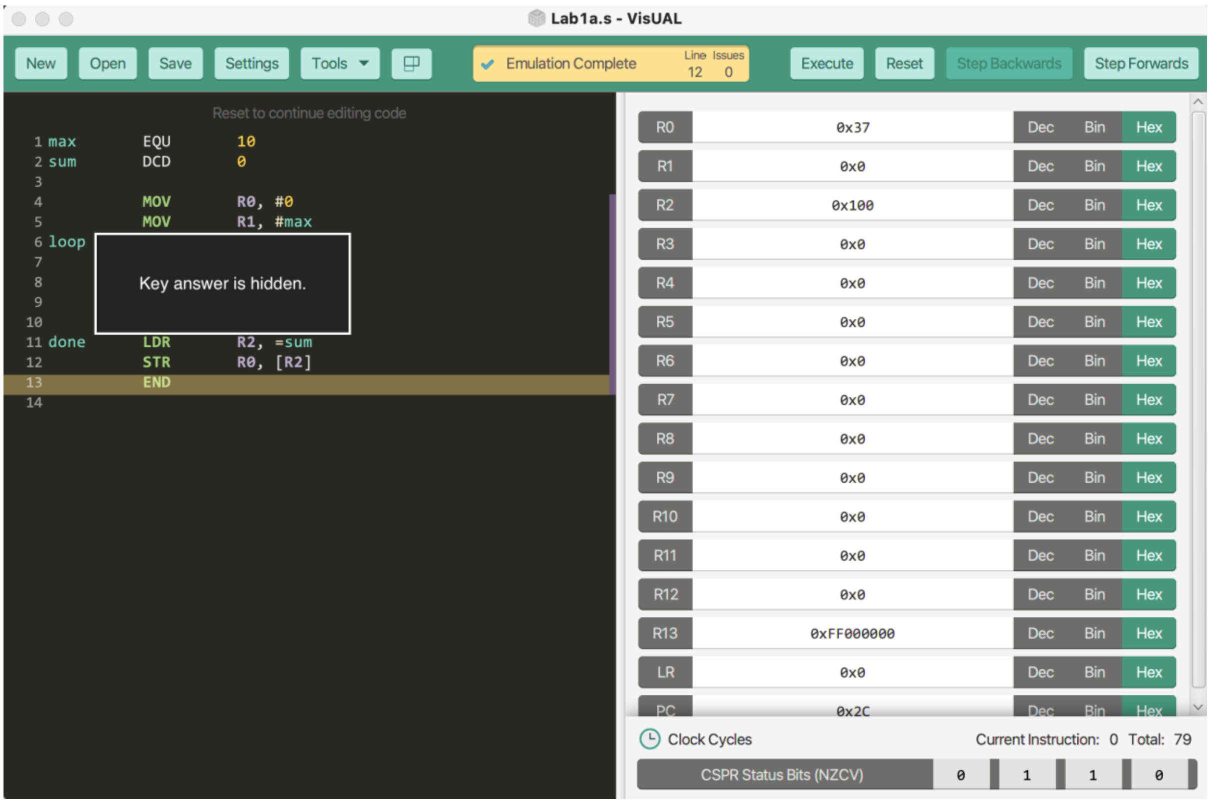


R0	0x0	Dec Bin Hex
R1	0x0	Dec Bin Hex
R2	0x0	Dec Bin Hex
R3	0x0	Dec Bin Hex
R4	0x0	Dec Bin Hex
R5	0x0	Dec Bin Hex
R6	0x0	Dec Bin Hex
R7	0x0	Dec Bin Hex
R8	0x0	Dec Bin Hex
R9	0x0	Dec Bin Hex
R10	0x0	Dec Bin Hex
R11	0x0	Dec Bin Hex
R12	0x0	Dec Bin Hex
R13	0xFF000000	Dec Bin Hex
LR	0x0	Dec Bin Hex
PC	0x0	Dec Bin Hex

Current Instruction: 0 Total: 0

CSPR Status Bits (NZCV) 0 0 0 0

Complete the following program that sums up 10 integers that are 10, 9, 8, 7, 6, 5, 4, 3, 2, and 1. R0 is used to maintain an intermediate and the final sum. The result of R0 should be stored into the sum variable at address 0x0100. Please note that VisUAL uses memory address 0x0000 – 0x00FF for code (which is actually not visible from the simulator) and thus allocates any readable/writable data to the memory space starting at 0x0100. When you define “sum” as shown below (in line 2), sum occupies memory 0x0100 – 0x0103 as an integer needs 4 bytes.



```
1  max      EQU      10
2  sum      DCD      0
3
4           MOV      R0, #0
5           MOV      R1, #max
6  loop
7
8             Key answer is hidden.
9
10
11 done     LDR      R2, =sum
12          STR      R0, [R2]
13          END
14
```

R0	0x37	Dec Bin Hex
R1	0x0	Dec Bin Hex
R2	0x100	Dec Bin Hex
R3	0x0	Dec Bin Hex
R4	0x0	Dec Bin Hex
R5	0x0	Dec Bin Hex
R6	0x0	Dec Bin Hex
R7	0x0	Dec Bin Hex
R8	0x0	Dec Bin Hex
R9	0x0	Dec Bin Hex
R10	0x0	Dec Bin Hex
R11	0x0	Dec Bin Hex
R12	0x0	Dec Bin Hex
R13	0xFF000000	Dec Bin Hex
LR	0x0	Dec Bin Hex
PC	0x2C	Dec Bin Hex

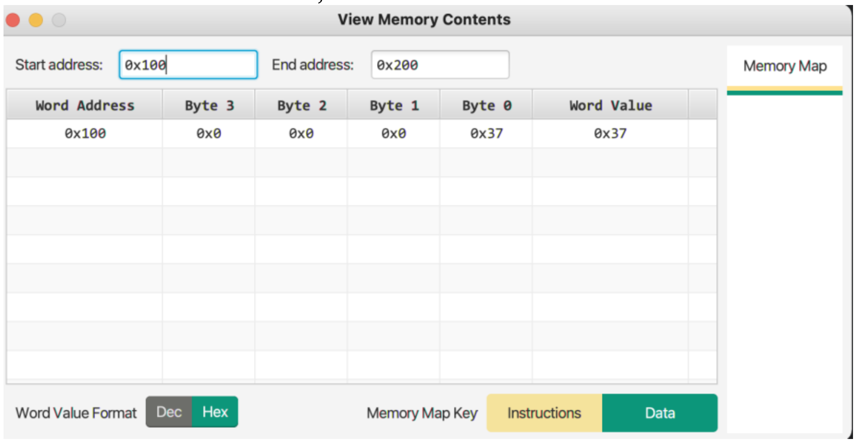
Clock Cycles   Current Instruction: 0 Total: 79

CSPR Status Bits (NZCV) 0 1 1 0

VisUAL’s menu buttons on the right side are quite descriptive:

1. To execute your program, click “Execute”
2. To stop your program, click “Reset”, which goes back to your code-editing mode.
3. To run your program in a single step, click “Step Forwards”.
4. To reverse back your program execution, click “Step Backwards”.
5. To view the memory contents, click “Tools” on VisUAL’s left top menu and thereafter choose “View memory contents”, which opens a “View Memory Contents” window. Set the memory region:

   Start address: 0x100, End address: 0x200



Start address: 0x100   End address: 0x200

Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x100	0x0	0x0	0x0	0x37	0x37

Word Value Format: Dec Hex

Please make sure to save your code by clicking “Save As”. Your file name should be **Lab1a.s**. Please don’t forget to add “.s” as the assembler file attribute.

Hints for your lab1a:

1. R1 maintains each of integers starting with 10 and decrementing all the way to 0.
2. Use “CMP R1, #0” and then “BEQ label” to check if R1 reaches 0 and to jump to label if so”.
3. Use ADD to add R1 to R0.
4. Use SUB to decrement R1.
5. Use B to go back to the top of loop.

===

What to submit

1. Your source code named Lab1a.s
2. A snapshot of VisUAL’s register contents and memory contents at 0x100-0x103

===