Homework 2: Answers

Profs. John McLeod & Arash Reyhani

ECE3375, Winter 2022

1. *Problem*: Which microprocessor is used in the labs for this course?

   (a) ARM®Cortex-A9

   (b) ARM®Cortex-M3

   (c) Intel Core-i7

   (d) Microsoft Word

   Please note that the order of these options is randomized.

   Solution: The ARM®Cortex-A9 is used in the labs in this course. This information is found in the course notes and in various lab manuals and documentation.

   The ARM®Cortex-M3 (or ARM®Cortex-M4) is used in the textbook, but is not used in the labs for this course.

2. *Problem*: A line of code in ARM Assembly used in the labs in this course that contains an instruction must have the following format:

```
1 label: mnemonic operand1, operand2, operand3 /* comment */
```

Please choose the best description of this format from the options given below.

(a) A mnemonic is always necessary.
The number of operands depends on the mnemonic; some require zero.
Comments are always optional.
A label is optional but often needed for branch or if-then statements. Each label must be unique within the entire program code.

(b) At least one operand is necessary.
The mnemonic is optional.
A label is only required for new distinct sections of code.
Comments are always optional.

(c) A mnemonic and all three operands are necessary.
Comments are always optional.
A label is only required after a directive.

(d) Every instruction must have a label, a mnemonic, and a comment.
The number of operands required depends on the mnemonic.

Please note that the order of these options is randomized.

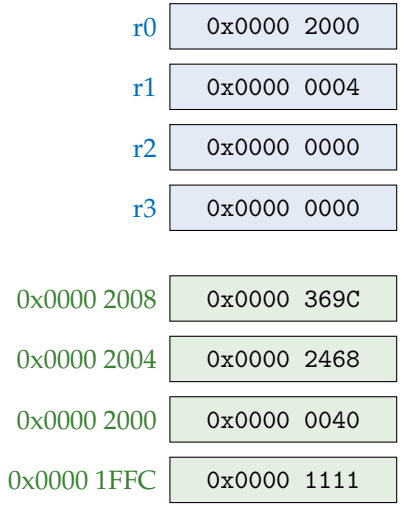
Solution: A mnemonic is always necessary: a *line of code* does not require a mnemonic — it may be a label or a comment — but an *instruction* always requires a mnemonic. The number of operands depends on the mnemonic; some require zero.

- The `nop` mnemonic is an example of a mnemonic that requires no operands. It does nothing (it is used to idle or to pad code so other instructions align at particular memory addresses).
- This wasn't discussed in class, but you can deduce this was the correct answer by eliminating the other incorrect options.

Comments are always optional. A label is optional but often needed for branch or if-then statements. Each label must be unique within the entire program code.

3. *Problem*: Some registers and memory cells in a microcontroller are initialized as shown. The following code is executed:

```
1 mov r2, #4
2 ldr r3, [r0]
3 sub r0, r0, r1
4 add r1, r3
5 ldr r3, [r0]
6 add r2, r3
7 str r2, [r0]
```



What are the contents of the registers and memory after this code is executed?

Solution: Trace the code line-by-line.

- Register `r2` now holds the value 4.
- Register `r3` holds the contents of memory address `r0` = `0x2000`, so `r3` contains `0x0040`.
- The value in register `r1` is subtracted from the value in register `r0` and stored in `r0`. `0x2000` − `0x0004` =`0x1FFC`.
- The values in registers `r1` and `r3` are added and stored in `r1`. `0x0004` + `0x0040` = `0x0044`.
- Register `r3` holds the contents of memory address `r0` = `0x1FFC`, so `r3` contains `0x1111`.
- The values in registers `r2` and `r3` are added and stored in `r2`. `0x0004` + `0x1111` = `0x1115`.

- The value stored in `r2` is written to the memory address in `r0`. The memory starting at address `0x1FFC` now holds the value `0x1115`.