ICS 332 Fall 2021

Homework Assignment #1: Pencil-and-Paper Assignment – Computer Architecture: The Fetch-Decode-Execute cycle [30 pts]

You are expected to do your own work on all homework assignments. (See the statement of Academic Dishonesty on the Syllabus.)

Check the Syllabus for the late assignment policy for this course.

How to turn in?

Assignments need to be turned in via Laulima. Check the Syllabus for the late assignment policy for the course.

What to turn in?

You should turn in single **plain text** file named `README.txt` with your answers to the assignment's questions. Your file must be readable "as is" and points will be removed if the report is not readable.

Exercise #1: The Fetch-Decode-Execute Cycle

The purpose of this exercise is to test the understanding of the Fetch-Decode-Execute cycle.

Consider a fictitious and very simplified Von-Neumann architecture with:

- 5-bit addresses/9-bit values memory
- 2 9-bit registers for computation named A and B;
- 1 5-bit Program Counter register;
- 1 9-bit Current Instruction register;
- 1 1-bit error register (set at the end of the cycle) when the execution of an instruction creates an "error" condition.
- 6 instructions: LOAD, STORE, SUBTRACT, JE (Jump on Error), and STOP (detailed in the table below)

Each instruction is encoded using 9 bits:

- the first 3 bits give the mnemonics (or operation code) identifying the instruction (see table below);

ICS 332 Fall 2021

- the last 5 bits are an operand which can be anything (e.g. an integer number, an address, or even ignored) depending on the instruction.

CPU Instruction	Opcode	Description
LOAD	010	Load register value with value from address (operand) Example: 010110101 = 010110101 is executed as 010 = LOAD, 1 = B, with the value at address (10101, i.e. LOAD B, [10101]. For instance, if the memory content at address 10101 is 11, then, at the end of the fetch-decode-execute cycle, the register B will contain the value 11. The value of the error register is not used. It is set to 0 when this instruction has completed its execution.
STORE	011	Store register value to address (operand) Example: 011010101010 = 011010101 is executed as 011 = STORE, 0 = A, with the value at address (10101, i.e. STORE A, [10101]. For instance, if the content of the register A is 25d, then, at the end of the fetch-decode-execute cycle, the memory location at address 10101 will contain the value 25. The value of the error register is not used. It is set to 0 when this instruction has completed its execution.
SUB	101	Subtract register B contents from register A contents. The result is written in A. The error register is set if an underflow happens. 101000101: SUB A, B. Note that 6 bits are ignored If initially A contains the value 7 and B 5, then, at the end of the fetch-decode-execute cycle, A will contain 2 (=7-5), i.e. b00010. The error register is set to 0 (no error). If initially A contains the value 3 and B 5, then, at the end of the fetch-decode-execute cycle, A will contain 0 (while it should be "-2" (=3-5)), and the error register will be set to 1 (signaling the underflow error) The value of the error register is not used for the execution of this instruction.

ICS 332 Fall 2021

JE	110	If the error register value is set to 1, set the Program Counter (PC) value to (operand) Otherwise, nothing happens (i.e. increment the PC by 1). In any case the error register value is set to 0 at the end of the fetch-decode-execute cycle For instance: 110000101 translates to JE 00101 Case 1: If initially the error register is 0 and the PC value is 11110, then at the end of the fetch-decode-execute cycle, the PC value will be set to 11111 Case 2: If initially the error register is 1 and the PC value is 11110, then at the end of the fetch-decode-execute cycle, the PC value will be set to 00101 The value of the error register is used for the execution of this instruction.
STOP	111	Terminates program For instance: 11100101 translates to STOP. Note that all bits after the first three ones (00101 in this example) are ignored. The value of the error register is not used for the execution of this instruction.

Part 1: Warm-up

Question 1.1 [2 points]

What does the instruction encoded by 110010001 do? Show your work.

Question 1.2 [2 points]

Translate “LOAD A, [10110]” to binary. Show your work.

Question 1.3 [2 point]

Among the instruction encodings below, which one is invalid? Give an explanation.

1. 11010110
2. 10011011
3. 01111010

Part 2: Case study

Assume that initially:

- The Program Counter value is set to 12d=01100b; The values of the other registers are undefined.
- The memory has the following contents:

Address	Contents	Address	Contents	Address	Contents	Address	Contents

ICS 332 Fall 2021

00001	110101001	01001	110010000	10001	011010110	11001	101101001
00010	001111011	01010	010101010	10010	111000000	11010	000001011
00011	100110110	01011	010000001	10011	001011010	11011	000000000
00100	101000110	01100	010010110	10100	010001100	11100	100010111
00101	010011110	01101	010110111	10101	010100000	11101	000000100
00110	100011001	01110	101100000	10110	001000001	11110	111101101
00111	000111011	01111	110010001	10111	010111100	11111	100100000

Question 2.1 [2 points]

What is the address and the first instruction executed by the program? Explain.

Question 2.2 [18 points]

Detail the program execution instruction by instruction (the program stops after executing the first STOP).

Detail what each stage of the Fetch-Decode-Execute cycle does for each instruction. Make sure to describe thoroughly how the system (CPU and memory) is or is not modified by each stage.

Detail the program execution in the following way: for each instruction, give the assembly-like code of the instructions (e.g., “LOAD A, [0101]”), and state what register and or memory values change. For instance, here is a sample answer (which has nothing to do with the real answer):

```
PRINT [0100]        ; print 'R'
LOAD A, [0101]      ; set the value of A to 01010101 (85d)
STORE B, [0000]     ; set the value in RAM at address 0000 to 77d
JE [1111]           ; do not jump to instruction at address 1111
...
STOP                ; terminate
```

Hint: The program executes 6 instructions, counting the last STOP instruction.

Question 2.3 [2 points]

What is the content at address 10010? What is the content at address 10110?

Question 2.4 [2 points]

The last STORE instruction writes to memory the output of the program. What does this program compute? Also consider other cases with different input values. Explain.

Hints: Rewrite the CPU mnemonics into a higher level pseudo-language (e.g. pseudo-C, or English). Swap the two initial input values in memory. Play around with other values. Execute

ICS 332 Fall 2021

Powered by the Morea Framework (Theme: cerulean-green)
Last update on: 2021-09-01 21:59:10 -1000
2 modules | 10 outcomes | 39 readings | 15 experiences