Tu Lam

ECE 375 / Dr. Justin Goins

Oct 12th, 2020

# Homework #1

(Due Date: Oct 13th, 2020)

**3**. Explain the meaning of each term in the list below. Be sure that you describe how each item is used within the pseudo-computer discussed in class.

a) *Operation code (opcode)*: opcode can be described as the field that specifies an operation to be performed by an instruction. To demonstrate this, an example would be $ADD\ 23, 43$. In this case the opcode would the 'ADD' operation as it works behind the scene to add these two numbers up together.

b) *ALU*: This is also call Arithmetic logic unit (ALU) which is a digital circuit is used to do arithmetic such as ADD, SUB, etc… in the machine. Also, it can do logic unit in the circuit and this can be seen as AND, OR, etc…

c) *Effective Address*: *Effective Address (EA)* is the encoded in the instruction that would directly referring it to an operand in the memory. An example of this would "J 0x1E" which would jump to the address pointing to the operand in the memory.

d) *Program counter (PC)*: This is holding the address of the next instruction to be fetched from the memory. An example of this when the program is running and being executed, the PC would get the next address of the instruction to run next and increment it counter by adding 1 and reset to 0 when computer restart or reset.

e) *Internal Data Bus*: It is an internal line in the computer system that carry operation to be performed. An example of this is when $AC \leftarrow AC + MDR$ when MDR hold the information and needed to be added with AC, the internal data bus help it transfer into the ALU including with AC to be added.

**4**. Imagine that you want to introduce a new assembly code instruction: “Decrement and Skip if Zero (DSZ)” of the form

Fetch Cycle:

```
Step 1: $MAR \leftarrow Y$                                  //Move the Y into MAR
Step 2: $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$       //Increment PC & read instruction
Step 3: $MDR \leftarrow MDR - 1$                            //Take content in Y and decrement it
Step 4: $If\ (MDR = 0)\ then\ PC \leftarrow\ PC + 1$        //Check if Y = 0, if so increment the PC
```

**5**. Consider the following hypothetical 1-address assembly instruction called “Store Accumulator Indirect with Post-increment” of the form.

Fetch Cycle:

```
Step 1: $MAR \leftarrow x$                                  //Move the x into MAR
Step 2: $MDR \leftarrow M(MAR), PC \leftarrow PC + 1$       //Increment PC & read instruction
Step 3: $MDR \leftarrow M(MDR), PC \leftarrow PC + 1$       //Read the instruction & increment PC
Step 4: $MDR \leftarrow AC$                                 //Assign AC into the address of MDR
Step 5: $MDR \leftarrow MDR + 1$                            //Post increment the MDR
```