Jose Franco Baquera

February 11, 2018

CS – 273

Homework #3

1. Define the concept of a memory address.

In essence, a memory address is a number that signifies a specific memory location in a computer's memory. It is important to note that a memory address is assigned to each location in a computer's memory, regardless if it is being used to store data. As an analogy, if we think of memory as a huge 1-dimensional array, a memory address would be the equivalent of an index in the array, thus accessing memory with a memory address is like using indices in the array to find a specific element. The CPU uses memory addresses to track where in memory data and instructions are stored. In addition, the CPU uses the address bus to both read and write into a computer's memory. In the case of reading a value from memory, the CPU presents the memory address to the memory controller, which then looks up that particular address and returns the bits stored in that location. When the CPU writes a value into memory, both the memory address and the data are presented to the memory controller.

2. How many registers do an AVR CPU have? What are their names?

The AVR CPU has 32 registers with each register being 1-byte in size. They are named from r0 to r31 (in other words, they are named r0, r1, r2, r3, ……., r30, r31).

3. What is the function of a program counter in a CPU?

A program counter (PC) is a special register located inside the CPU whose solely function is to always contain the address (location) of the next instruction to be executed. In addition, whenever the CPU goes to fetch the next instruction, it uses the value in the program counter as

the address to read the program memory at. As each instruction in a program executes, the program counter is automatically changed to point to the next instruction in the sequence that needs to get fetched. Whenever the computer restarts or resets, the program counter is normally set back to 0.

4. **How many binary bits are used to code the "add" instruction in AVR?**

A takes a total of 16 binary bits to code the "add" instruction in AVR.

5. **How many bytes will the binary code of the compute.s program on page 12 have? Each byte is 8 bits.**

- lds r18, val2

  This instruction is 32 bits = 4 bytes.

- ldi r19, 23

  This instruction is 16 bits = 2 bytes.

- add r18, r19

  This instruction is 16 bits = 2 bytes.

- sts val1, r18

  This instruction is 32 bits = 4 bytes.

- ret

  This instruction is 16 bits = 2 bytes.

Now we add them.
Total = 4 Bytes + 2 Bytes + 2 Bytes + 4 Bytes + 2 Bytes = 14 Bytes.
Therefore, there is a total of 14 Bytes in the binary code of the compute.s program.