Name __ ***Answers*** ____

CE-1921-11 – Dr. Durant – Quiz 1
Spring 2017, Week 1

1. **Define** architecture
2. **List** the two locations that hold operands.
3. **Define** instruction.
4. **State** the bit width of every ARM instruction.
5. **Explain** the purpose of an assembler.
6. **Define** mnemonic.
7. **Define** source operands.
8. What is a **disadvantage** of a RISC architecture relative to a CISC architecture?
9. **Describe** the impact the number of instructions in an instruction set has on code density thus instruction storage size.
10. **Explain** what the following instruction does: ADD R0, R6, R7.

Answers

1. It is the programmer's view of a computer. It specifies the instruction set (language) and where operands may be stored.
2. Registers and memory
3. A single operation that a microprocessor's hardware can execute. These are often very basic, such as adding 2 numbers, or moving a number from a register to memory. They can be more complex, though.
4. 32 bits
5. Translate human-readable assembly language into machine language, which is the binary values that tell the microprocessor hardware what to do.
6. An abbreviation or initialism that represents an instruction or operation. It is mean to be human readable and suggestive of its function, in contrast to the binary machine encoding.
7. The part of the instruction that specifies what values are the input to the calculation or operation being done.
8. Since its instructions are constrained to be simpler, it takes more of them to do most tasks.
9. More instruction in the set means, at some point, more bits will be needed to encode them into machine language. However, this is often more than offset by the fact that fewer of the more complex instruction suffice to complete a given task, so the code will tend to become more dense and thus smaller.
10. It reads the contents of registers 6 and 7, adds them, and stores the result in register 0.

Name __***Answers***__

CE-1921-21 – Dr. Durant – Quiz 1
Spring 2017, Week 1

1. **List** the two primary items that make up an architecture.
2. **Contrast** system architecture and micro-architecture.
3. **Define** instruction set.
4. **Describe** the difference between assembly language and machine language.
5. **Explain** the purpose of a compiler.
6. **List** two of the four modern architectural design rules.
7. **Define** destination operand.
8. **Describe** the impact the number of instructions has on mnemonic encoding and thus circuit design.
9. **State** the number of ARM registers.
10. **Translate** the following Java/C statement into ARM assembly: a = b + c; Use R0, R1, and R2 for the variables, respectively.

Answers

1. Instruction set (language) and operand location
2. System architecture is a higher level view, focusing on what a developer using the system needs to know. Micro-architecture concerns the specific arrangement of ALUs, registers, etc., to realize a particular system architecture. There are infinite possible micro-architectures for any given system architecture. This is analogous to how a VHDL architecture is used to implement a VHDL entity.
3. The list of operations that an architecture is capable of directly executing.
4. While both represent particular instructions, assembly language is easily human readable and uses mnemonics (abbreviations or initialisms), while machine language is the binary version of the instruction used by the system architecture.
5. It translates a high level language, like C, into an assembly language (not machine language).
6. Regularity leads to simplicity, smaller is faster, (make the common case fast), (design is compromise)
7. The part of the instruction that specifies where the instruction result should be stored. (We will see there is some variation to how this term is used, but this is the primary meaning.)
8. More available instructions means that, at some point, more bits are needed to differentiate among them. Thus the hardware to differentiate among and implement all the options becomes more complex.
9. 16
10. add r0,r1,r2