1. Explain the difference between a register and the ALU.

A register is an electronic storage component. In the MIPS architecture each register is a component with a capacity to hold a 32-bit number. The ALU (Arithmetic Logic Unit) is a digital logic circuit that is can perform binary arithmetic operations such as AND, OR, and XOR. The operation that the ALU performs is determined by the operation code (opcode) in the Instruction Register (IR).

2. Explain the difference between assembly language and machine language.

Assembly language is a mnemoic representation of a program that is converted into machine code by a utility program called an assembler. Machine code is stored in a file on disk. When someone wants to execute the program, another utility program, called a linking loader, loads and links together all of the necessary machine language modules into main memory.

3. Explain the difference between Cache Memory and the Register File.

Cache memory is memory located on the CPU chip. Cache memory provides fast memory access to instructions and data that were recently accessed from memory. The register file is an array of processor registers in the CPU. In the case of the MIPS architecture, the register file contains 32 registers.

4. Explain the difference between the Instruction Register and the Program Counter.

The instruction register holds a copy of the most recently fetched instruction whereas the Program Counter points to the position in memory where the next instruction can be loaded.

5. Explain the difference between a bus and a control line.

A bus is simply a set of electrical conducting paths over which different sets of binary values may be transmitted whereas a control line is a single line or bit that is used as input in a multiplexer to determine what action should be taken.

6. Identify a kitchen appliance that contains a finite state machine.

A microwave oven contains a finite state machine. A microwave oven can either be in the ready state, open door state, or active state (the number of states can vary depending on the complexity of the FSM representing it).

7. If a 500 MHz machine takes one clock cycle to fetch and execute an instruction, then what is the instruction execution rate of the machine?

A 500 MHz machine has 500*10^6 cycles per second. If it only takes one clock cycle to fetch and execute an instruction, then the effective instruction execute rate is 500 million instructions per second (500 MIPS).

8. How many instructions could the above machine execute in one minute?

The above machine could execute 500*10^6 instructions in one second. Thus the machine can execute 500*10^6*60 or 30 billion instructions in one minute.

9. Let’s suppose we have a 40-year-old computer that has an instruction execution rate of one thousand instructions per second. How long would it take in days, hours, and minutes to execute the same number of instructions you derived for the 500 MHz machine?

To calculate the total time it would take for the old computer to calculate 30 billion instructions, we take the total number of instructions (30 billion) over the instructions per second (1000). This gives us a time of 30*10^6 seconds to calculate 30 billion instructions. If we convert 30*10^6 seconds into days, hours, and minutes, we get that the total time it would take is 347 days, 5 hours, and 17 minutes.

10. What is an algorithm?

A list of instructions that we want the processor to perform to accomplish some task.