**CIS 21JA Assignment 2 Name: Tom Ekshtein**

*Questions 1 - 7 are 1 pt each*

1. Name the 4 parts of a CPU, and next to each part name, write one sentence *in your own words* to describe its main purpose. (I'm familiar with the descriptions in the class notes, so a copy-paste won't get credit)

* Registers: These are temporary storage within the CPU that allow you to hold onto data that you might need quickly but are very expensive.
* Arithmetic and logic unit: The ALU can perform mathematical operations such as addition, subtraction, comparing, and more.
* Clock: The name gives it away here. The clock keeps the CPU in time and allows it to keep time for operations it might need to perform.
* Control Unit: The CU regulates the operations of the computer, it selects and gets instructions from the main memory and interprets them into what needs to be done.

2. With a 5-stage *pipelined* processor, where each stage takes 2 clock cycles, how many clock cycles does it take to execute 10 instructions?

* For k stages and n instructions, the number of required cycles is: k + (n-1). Therefore, 5 + (10-1) = 14 cycles.

With a 5-stage *non-pipelined* processor, with each stage also taking 2 clock cycles, how many clock cycles does it take to execute the same 10 instructions?

* A non-pipelined processor, the number of required cycles with n instructions and k stages is where each stage takes 2 cycles is: k\*2n. Therefore, 5\*10(2)=100 cycles

3. With respect to the instruction execution cycle, what is the advantage of storing data in registers instead of in memory variables? Make sure your answer refers to the instruction execution cycle to show the advantage.

* Looking at the instruction execution cycle, there are five steps when handling an instruction from a program. The biggest advantage of storing data in a register when compared to storing data in memory is being able to skip the operand fetching step. Skipping a step saves time when doing many instructions over a certain period of time.

4. Our assembly programs uses 32 bits to address memory and can access up to 4 GB of memory. If you write assembly code for a system that uses 16 bits to address memory, what size memory can your program access?

* Since 2^32bits = 4,294,967,296 unique addresses, or 4GB of memory. Therefore, 2^16bits = 65536 unique addresses or 64KB of memory.

5. If you convert an assembly program that is written for a CISC processor into a program that runs on a RISC processor, would the new program be longer or shorter? Why?

* Since CISC is complex and RISC is reduced which means that for the same instruction, CISC can be made in a shorter, more complicated statement. However, it might take a few, more simple instructions to do the same command in RISC. This means that a CISC program re-written in RISC would make it longer.

6. How can a program that accesses up to 4GB of memory run on a system that only has 1GB of physical memory?

* Thanks to virtual memory and paging this is possible. Paging lets the total memory used by a program to be in memory and the rest on the disk.

7. Since conventional memory is slower than the CPU, what does the computer have to help make memory access faster? Your answer should not include registers, they don't help memory access speed.

* Cache memory, it is a high speed but expensive form of RAM both in and outside the CPU.

8. (8pts) Download the file Assignment2.asm and bring it into the IDE Project. Then follow the steps in the file and fill in the data values you observe in the source file (the asm file). Then copy the data here to turn in your results.

mov ah, 101b ; AX = 05FE

sub ah, -2 ; AH = 07

inc al ; AL = FF

xor eax, 0FFFFH ; EAX = FFFFF800

bigData in memory = 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46