**CSE 3302: Programming Languages**

**Spring 2018**

**Homework 04**

**Due on October 4, 2018 [ before 11:59 pm]**

**Name: Jerry Olds**

**1000 number: 1001533643**

**Date: 10/02/2018**

**INSTRUCTIONS**

1. **Do NOT plagiarize.**
2. **No group-work. All work should be your own.**
3. **Do not discuss your work with other students in the class.**
4. **Cite sources where necessary.**
5. **Turn in your word document using Blackboard. Do not email your documents.**
6. **Name your document as netid.docx where *netid* is your UTA NetID. If you do not know your NetID, check what it is using NetID Self Service. Your 1000 number is NOT your NetID.**
7. **Try to answer each question within 5-7 lines.**

**Questions:**

**1.** What’s the regular speed of your computer running (in Hertz)? How many instructions will they run in one second, assuming one instruction per clock cycle?**[7 points] 4.2 X 10^9 Hertz. 4,200,000,000 instructions/s**

**2.** What does a compiler do**? [8 points] Transforms source code written in high-level programming language in to a low-level assembly or machine language.**

**3.** Explain branch prediction. **[7 points] When a jump instruction is queued, the CPU will guess which direction the jump will go and load the instructions of that branch. If the guess was correct the CPU can carry out it’s execution of instructions without any delay. If the guess was wrong it must preform a pipeline flush and go back to the instruction to choose the other path.**

**4.** What is the FLOPs? What’s the FLOPs of the CPU in your computer? **[8 points] FLOPs stands for Floating Point Operations per second. It is the amount of operations your CPU can perform on floating point numbers in one second. 25 x 10^9 FLOPS**

**5.** What is the machine language? The relation to English? Show an example to describe them.**[9 points] Machine language is a set of 1’s and 0’s that a computer interprets to preform instructions. The pattern of the 1’s and 0’s determines the type of instructions to be run, just as we use a pattern of English words to communicate ideas to others. The relationship between machine language and the English language is very similar to the relationship between Morse Code and English. Morse code is comprised entirely of dots and dashes just like machine language is comprised entirely of 1’s and 0’s.**

**6.** Give a brief description on ALTAIR 8800 computer.[**6 points] It was the first successful home computer that used panel programming to store and execute instructions.**

**7.** How many instructions does an Intel 4004 chip have?**[5 points] 46 instructions.**

**8.** What is a cache? Why do we need it? **[8 points] A little piece of RAM memory built in to the CPU. We need it so that we don’t have to go from CPU to RAM back to CPU for every instruction.**

**9.** Explain the cache hit and cache miss. **[5 points] When a CPU requests memory for an instruction, the RAM returns a block of memory starting at the location of the instruction requested. If the next instruction the CPU requests is inside the block of memory returned by the RAM, it is called a cache hit and can perform the next instruction without going to RAM. If not, it is called a cache miss and must request a new block of memory from the RAM.**

**10.** What were the advantages of Assembly languages? **[7 points]**

**11.** How is assembly language different from machine language? **[7 points] Assembly language is more closely related to human language. It has words and mnemonics that is sent to an assembler that translates the assembly language instruction in to a binary machine language instruction.**

**12.** Why did we need punch cards in the early programming era? **[7 points] Punch cards were used to store data for machines that did not have electronic memory. They were also used to read data in to memory in early electronic memory computers.**

**13.** Is the divide operation implemented via subtraction efficient when handled in CPU as described in the video? Why?**[10 points] No. Because one subtraction takes one clock cycle of the CPU. A large division that would take numerous subtractions would take numerous clock cycles which means it would take more time.**

**14.** What were the benefits of each of the two early languages below?**[6 points]**

a. COBOL: **Was the first programming language that was designed with the idea of “Write Once, Run Anywhere”. A COBOL program could run on any machine with a COBOL compiler despite varying hardware.**

b. Fortran: **A Fortran program was on average 20 times shorter than an assembly language program. The computations took longer than assembly languages at the time, but the amount of time to write programs was significantly decreased.**

**Extra credit (bonus question):**

**15.** List the tricks boosting the CPU performance. **[10 points]**

**1. Add cache memory to processor.**

**2. Pipelining.**