CSC 120 Extra Credit Assignment

This assignment tests your understanding of chapters 0,1 and 2 as well as Python. If you score full points in this assignment, they will be counted towards your scores from Lab/Quiz 0, Lab/Quiz 1 and Lab/Quiz 2. For example, if you score 150/150, those 150 points will be awarded to make up for the points you lost in those labs/quizzes. However even if you score all the points in this assignment your Lab/Quiz scorres would not exceed 100 points which is the maximum.

* **Conceptual Questions (+90 points)**
* (+5) If an instruction is 32 bits long, how many hexadecimal digits would be needed to represent it?

You would need 8 hexidecimal digits to represent it.

* (+5) What is the difference between volatile and non volatile memory? Why do you have two different types of memory?

Volatile memory is memory that is able to be accessed at faster rates, is more easily transferred and altered, but is also generally smaller in capacity. Volatile memory is wiped automatically when the system is turned off and is not meant to be stored long term. Nonvolatile memory is larger in its capacity and is more permanent than volatile, but can not be changed or fetched as quickly.

* (+5) What is the goal of CPU registers VOLE architecture? (For example, Load, Store etc.)

CPU registers are memory cells that hold the instructions for data transfers that are being used in real time so that transfers can be executed faster and more efficiently.

* (+3) Is 102100 a valid binary number? Why?

No, binary is represented with only two digits, 0’s and 1’s. The 2 in this pattern makes it an invalid binary number.

* (+3) Convert 1F2 (hex number) to decimal.

498

* (+3) Convert FF04 (hex number) into binary.

1111 1111 0000 0100

* (+5) If you have a 6 bit binary number, what is the maximum value (in decimal) that can be represnted by this number. Describe with an example.

111111 is the largest possible 6 digit binary number that, when converted to decimal, would represent the number 63.

* (+5) Instead of a 6-bit binary number, if you have an n-bit binary number, what is the max value in decimal that can be represented by it? For example, start with a 1 bit binary number then a 2 bit binary number and so on and try to generalize

* (+3) Explain the concept of abstraction in software with your own example.

Abstraction in software engineering is the practice of simplifying different levels of engineering so that the intricacies of each individual part don’t overcomplicate the whole. Almost everyone owns a cell phone today, but only a small percentage of owners actually understand how it functions. You don’t have to understand the complexities of the tech in order to be able to use it. Depending on how the phone is manufactured, if you are responsible for building or maintaining one part of the phone, you don’t need to know the remainder if it doesn’t affect your specific part.

* (+5) If you want to represent a number like 1.02 in binary, what would you do?

* (+3) How many bits are used for opcode and operand in the Vole architecture?

The op-code is represented by the first two bits in the string, the operand are the four bits that follow. So 6 in total.

* (+5) How are ARM processors different from Intel processors? (You can read online for this and write your own understanding. Try and understand RISC vs CISC).

Intel processors use CISC (Complex Instruction Set Computing) while ARM uses RISC (Reduced Instruction Set Computing). In CISC, commands are executed using a more complicated set of instructions that is to go through several additional cycles before the task is completed. RISC only has to go through one cycle before completion. Intel is more commonly used in desktop computers and other larger devices that consume a larger amount of power and is also more dependent on hardware for these processes. ARM relies more heavily on software and is often used for processing in smaller tech like cell phones since its single cycle processing is more efficient to run on less power. Because ARM runs on less power, the processing speeds are often slower than Intels’.

* (+10) What are flip flops? Why are they necessary in digital circuit design?

A flip flop is a circuit, that’s output is always either a one or a zero, that can be altered to produce the opposite output value when affected by a pulse from a controlling circuit. Flip flops are a necessary component of computer memory, they will always hold a value of “1” or “0” that is able to be manipulated by outside stimuli (circuits), and other circuitry will use those outputs for other inputs, creating a chain of sorts. Therefore, they can be assembled in large numbers in a condensed package along with their circuitry to help form entire computer systems.

* (+10) What is the role of cache memory?

Cache memory is a smaller amount of high speed memory that is located within the CPU. This portion of memory is a more easily accessible copy of main memory that is of current interest; instead of transfers between main memory and the registers, transfers are made between cache memory and registers. If changes are made to this portion of cache memory while, they are transferred at a more opportune time later on.

* (+10) Classify the types of memories in terms of their prices and size: cache memory, main memory and hard drive.

In terms of size from smallest to largest, the order would be cache, main, hard drive. Cache is a small portion of memory, usually measured in KB’s. Most main memory today is no more than 5-10 GBs depending on the machine, modern hard drives however, are capable of holding as much as TBs of data and can be easily interchanged when filled. Cache is located in the CPU itself, so for larger, faster cache memory, it can get very expensive. Main memory is similar in this concept, a larger well of main memory will come standard with more expensive machines. Less expensive computers will likely contain less included main memory. Hard drives come in an array of sizes and can be very inexpensive depending on their storage size. Even the largest hard drives are a fraction of the cost of machines that contain large amounts of standard cache/main memory.

* (+10) What is the importance of studying chapter 1 and chapter 2? Why do you think it is necessary to learn these concepts?

For abstraction purposes, its important to have at least a general understanding of how all of these processes work. No matter which field of computer science someone chooses to specialize in, the topics covered in the first two chapters will be, at the very least, relevant and connected to any further specialization down the line. Having familiarity with these subjects will help relieve some of the complexities of the material later on, as you will have a general idea at least of how all of the functions work together and how they are connected to whatever your current work is.

**Programming Questions (+70 points)**

* (+5)Write a for loop that prints the following numbers up to 99. [0,3,6,9,12,15...99]

﻿

for i in range(0,100,3):

print(i)

* (+5) Write a while loop that prints the following numbers up to 99. [0,3,6,9,12,15...99].

﻿i = 0

while i < 100:

print(i)

i=i+3

* (+10) What happens when you run this code? Why?

i = 0

while i < 100:

print (i)

It displays an error due to a syntax issue. The “print” line should be indented.

* (+10) Identify the problem with the following piece of code. What happens when you run this code? How would you fix the issue? and Why?

i = 0

while i < 100:

print (i)

The condition of the “while’ loop is never broken, the loop will continue to print “0” unless you force quit.

* (+10) How would you fix the following piece of code? Why does this code not work?

for i in range (10, 50, 2):

print ("The value of i is :" + i)

In this code, the statement “the value of i is” is considered a string, while “i” is an integer. To concatenate the two objects, the must be the same type, so “i” must be converted to a string.

* (+10) Explain the concept of loop initialization. Provide an example. (Refer online. This is meant to help you how figure things by reading online)

The initialization is an expression or declaration that initializes the loop, commonly used to help control the limitations of *for* loops:

General form of for loop in C, loop initialization highlighted:

for(initialization, condition, iteration, expression):

for(i=0, i<=10, i++)

* (+10) Write a loop that keeps accepting digits from a user and terminates when the user enters a non digit as input. For example, here is how the program would run.

**Please enter a digit:** (Program prints this line)

< User enters 6>

"You entered 6" (Program prints this line)

**Please enter a digit**

<User enters 2>

"You entered 2" (Program prints this line)

**Please enter a digit:**

<User enters a>

"You did not enter a digit. Terminating program" (Program prints this line)

<Program terminates>

﻿while(True):

user\_input = input("please enter digit: ")

if user\_input.isdigit() == False:

break

**Instructions: Upload the file with the screenshot on Blackboard with your firstname\_lastname.docx**