# Comp 3350: Computer Organization & Assembly Language

# HW # 3: Theme: Data declarations, Small program

*All main questions carry equal weight.*

*(Credit awarded to only those answers with work shown)*

1. Explain the two ways of generating a clock for a CPU, as discussed in the class and explain which is preferable and state reasons.

A clock generator is an electronic oscillator that produces a clock signal for synchronizing circuit operations, such as between the CPU and system bus. This is a built-in circuit for every computer, but other options are available. The two options for generating a clock signal for a CPU are a quartz crystal and an RLC circuit. The quartz crystal oscillator has the benefit of a very precise resonant frequency. This allows the CPU to reach speeds above 10 MHz, which is very beneficial in terms of overclocking the PC. The RLC circuit consists of a resistor (the R), an inductor (the L) and a capacitor (the C) connected. The RLC circuit is used to oscillate (and therefore act as a clock) with minimal damping, so the resistance is made as low as possible. The quartz crystal oscillator is the preferable way of generating a clock for a CPU. We prefer these because they are much more accurate than the RLC circuit, they are smaller, have lower temperature coefficients and low drift at a low cost. We can therefore use this structure to emulate an RLC circuit at a fraction of the cost and space, which achieves the desired effect of generating a clock for a CPU.

1. Discuss a synchronous memory read cycle.

First, place the address of the value you want to read on the address bus. Second, assert (change the value of) the processor’s RD (Read Line) pin (set it to low). Third, wait one clock cycle for the memory chips to respond. Fourth, the RD goes to high, indicating that the data is on the data bus and is ready to be read. Finally, we drop the memory read control signal to terminate the read cycle. This is synchronous because the CPU and data bus work together (synchronized) to achieve the goal of reading something from memory. As a further note, each step generally requires a single clock pulse, causing this process to be somewhat slower than desired.

1. Declare the following:
2. An un-initialized data declaration for an 8-bit signed and unsigned integer

value1 SBYTE ? ; signed 8-bit integer

value2 BYTE ? ; unsigned 8-bit integer

1. An initialized data declaration for a 32-bit signed integer with the value “BC12h” and a signed integer with the value of your choice.

value1 SDWORD BC12h ; signed 32-bit integer

value2 SDWORD -2147483648 ; signed 32-bit integer

1. A null terminated string variable containing the name of your favorite car.

str BYTE “Honda Skyline”,0 ; unsigned 8-bit integer

1. A symbolic constant named “*SecondsinWeek*” using the equal-sign directive and assign it an arithmetic expression that calculates the total number of hours in a Week.

SECONDSINWEEK = 7 \* 24

1. Show the order of individual bytes in memory *(lowest to highest****)*** for the following double word variable *(use little endian order****):*** GoTigers DWORD 1A2B3C4Dh

4Dh would be stored in the first byte, 3Ch would be stored in the second byte, 2Bh would be stored in the third byte, and 1Ah would be stored in the fourth byte. The offsets in order with respect to the bytes are: 0000, 0001, 0002, & 0003.

1. Show the following using assembler directives:
2. How to declare a signed array of five elements and initialize the array with the following values: 1111h, 2222h, 3333h, 4444h, 5555h. (You have to choose the right data type which uses the minimum memory space.)

list SWORD 1111h, 2222h, 3333h, 4444h, 5555h

1. Using the array created in part A of this question, show how to calculate the number of elements contained and assign that value to a symbolic constant named “*ArraySize*”

ListSize = ($ - list) / 2

1. Why is a string variable declared using the reserved word BYTE as opposed to WORD, DWORD or QWORD?

Each character uses a byte of storage, and strings are exceptions to the rule that byte values must be separated by commas. This is included as convenience to the programmer, because without it, doing anything involving strings would be too tedious for anyone to attempt. With this exception, all the characters can be effectively concatenated, and the storage space would be calculated by combining each individual byte from every character into a final storage value. Strings are also usually null-terminated (with a 0) and enclosed in quotation marks to signal that it is a string to the compiler.

1. Using the *AddVariables* program from the textbook as a reference, write a program that adds three signed byte sized integers using only *32-bit registers*. Please embed program code into your homework submission.

