Name: \_Schuyler Davis\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CS 2700 – Assembly Language and Computer Architecture (Worksheet – 60 pts)

1. The most common Instruction Set Architectures (ISAs) in use today are x86 and ARM. Which devices tend to use the x86 architecture, and which tend to use ARM?

Desktops and laptop computers use the x86 architecture. Mobile phones, tablets and other more mobile devices use the ARM architecture.

1. x86 is a type of CISC architecture, whereas the ARM is a type of RISC architecture. What are some of the fundamental differences between the two?

CISC architecture is more powerful than RISC architecture but it creates so much heat that it requires a fan or external cooling system.

RISC architecture is smaller than CISC architecture so it creates less heat, which is why we use it for phones but is less powerful than CISC architecture.

1. The bus is the means by which a computer is able to transfer data to and from various components of a given system. What are the names of the three major elements of the bus and what purpose do they serve?

The address bus transports memory addresses which the processor wants to access in order to read or write data. The requests go only one way.

The data bus transfers instructions coming and going from the processor.

This was a bi-directional bus.

The control bus transports orders and synchronization signals coming from the control unit and traveling to all other hardware components. It is a bi-directional bus, as it also transmits response signals from the hardware.

1. A CPU is comprised of both the Control Unit (CU) and the Arithmetic Logic Unit (ALU) among other elements. Explain what the CU and ALU are and the function they serve.

The control unit directs the computer what to do next. The ALU carries out the arithmetic and logic operations that is passed from the control unit.

1. What is the difference between a compiler and an assembler?

A compiler takes the source code and then converts it into the target code which could be machine language or virtual code. A compiler also performs optimizations to the code.

An assembler is just a straight translation from assembly language to machine code.

1. What purpose do alternate number systems such as octal and hexadecimal serve?

Octal and hexadecimal are better ways of representing binary for humans. It is more readable and therefore it is better for us to read rather than long strings of bits.

1. Convert the decimal number 212 into both binary and hexadecimal

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

212-128 =84-64=20-16=4-4=0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 4 | 2 | 1 | 8 | 4 | 2 | 1 |
|  |  |  |  |  |  |  |  |
| Hex | C4 | | | | | | |

1. Both the x86 and ARM architectures contain flags registers. What is the purpose of the zero flag and the overflow flag?

A zero flag is used to check the result of the arithmetic operations and if that is zero then the flag turns on.

The overflow flag is flipped on when addition or subtraction is performed and the register cannot hold the last bit of the data of the over flow and signals to the programmer that they need to do something with the new data.

1. The x86 architecture has four segment registers; the code segment, the data segment, the extra segment, and the stack segment. What is the purpose of each segment?

The code segment is the section where your code goes under.

The data section is where you define all of your variables and constants.

The stack section is where you can set the size of the stack. You can manually set aside the amount of data that the stack of your program can take up.

The extra segment is a 16 bit register containing address of 64KB segment with program data.

1. What is a base address, and how does it differ from an offset address?

A base address is a unique location in primary storage that serves as a reference point for other memory locations called absolute addresses.

An offset address is the address of an item that you are creating in assembly language.

A base address is the starting point of the program and in this class we are using the 100h in our programs.