Assignment 1

Due date: 26-09-2023

Spondon Sayeed

Section 1 (30 marks)

1. What is the decimal representation of each of the following unsigned binary integers? (4 marks)

a. 1111 1000   128+64+32+16+8 = 248

b. 1100 1010   128+64+8+2 = 202

2. What is the binary representation of the following hexadecimal numbers? (4 marks)

a. 0 1 2 6 F 9 D 4 = 0000 0001 0010 0110 1111 1001 1101 0100

b. 6 A C D F A 9 5 = 0110 1010 1100 1101 1111 1010 1001 0101

3. What are the number of inputs for a truth table with 3 variables x, y, z? (1 mark)

8 inputs

4. What is the 8-bit binary (two's-complement) representation of each of the following signed decimal integers? (1 mark)

a. -72

signed: 1100 1000

1s: 0011 0111

2s: 0011 1000

b. -98

signed: 1110 0010

1s: 0001 1101

2s: 0001 1110

5. Is assembly language portable? Explain. (4 marks)

Assembly language is not portable, but it remains relevant in applications such as gaming and embedded systems. Moreover, while high-level programming languages like C++, Java, or Python are standardized because of its compatibility with every computer, assembly lacks standardization. There is no general assembly language that works with every computer architecture. However, assembly can be used for optimizing sections of code for a specific architectures like gaming and embedded systems. This level of control allows programmers to achieve maximum performance but sacrifice portability in the process.

6. Compare between the following: (16 marks)

a. Carry and sign flags

Carry Flag: is used to indicate whether a math operation creates a carry out of the most significant bit (MSB) during addition or to borrow into the MSB during subtraction.

Sign Flag: is used indicate the +/- sign of the result in a math operation. If the MSB is 1, the Sign Flag is set to 1 to indicate a negative number, and if it's 0 the flag is set to 0 to indicate its a positive number.

b. Flat and multi-segment models

Flat Memory Model: A memory model that acts like a seamless straightforward road for simple memory management because it is so straightforward the management is consistent and allows for a uniform address space for active processes. The consistency and simplicity are the main reasons why this model is popular in modern systems.

In the Multi-Segment Memory Model, memory is divided into segments with unique base addresses, sizes, and designated functions, offering precise control and security. However, dealing with these segments involves detailed memory operations, potentially making the system more complex. This model was more widespread in older computer systems and early operating systems.

c. Data, control, address buses

A Data Bus is responsible for moving information between different components in the computer like external devices, internal memory, and the CPU. This bus allows data to move in and out of these components allowing the bus to be able to read and write memory to the devices.

The Control Bus acts like organizer, directing the various operations within the computer. The bus sends signals that instruct actions which manage input and output tasks and coordinating the CPU's activities. The Control Bus uses specific lines, like a baton, such as "Read," "Write," "Clock," and "Reset," to ensure that all components work together.

The Address Bus is like the postal service of the computer. It tells the CPU exactly where to find the data in the memory. This bus carries binary address information, which allows the CPU to navigate to specific memory locations or peripheral devices. Unlike the Data Bus, which can go both ways, the Address Bus mainly sends addresses unidirectionally from the CPU to memory and devices.

d. Logical and physical address

Logical addresses are crucial locations in the computer memory management unit, allowing programs to communicate with the computer's memory during the execution phase without dealing with the hassles of physical memory. Additionally, logical addresses provide natural security by keeping programs and their respective data isolated from each other.

Physical addresses are like permanent destinations in the computer's memory, providing data consistency. They are used by hardware components, and when logical addresses need to access memory, they are translated into these fixed physical addresses. This process maintains program separation, efficient memory management.