FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

CSC 1104: COMPUTER ORGANISATION AND ARCHITECTURE

TEST 1 2009 /2010

1. (a) Use examples to help you explain the difference between Computer Organization and Computer Architecture.

Computer Organisation is the way the hardware components are connected together to form a computer system i.e. the hardware details visible to the user e.g. the interfaces, memory technology etc.

Computer architecture is concerned with the structure and behaviour of the various functional modules of the computer and how they interact to provide the processing needs of the user. E.g. the instruction set, the number of bits used, I/O mechanisms, techniques for addressing etc.

(b) Explain briefly the main functions of a computer and for each function specify a computer component that executes the said function.

Main functions of a computer are:

- **Process data => done by the microprocessor**
- **Store data => by memory and the Mass storage devices**
- **Move the data between the different computer components and the external world; by the System Bus.**
- **To control all the above operations: Timing Circuitry and the Control lines.**

(c) Explain the following:

(i) Single Bus Architecture : Arrangement of computer components where there is only one processing element and all the other components are connected to a single link (the System Bus)

(ii) Multi Processing System:
Several processing elements are surrounded by different subsystems and a central link (the system bus) connecting the different subsystems together.

(iii) Contention
The competition for the shared resources by the different elements

2. (a) List the different **Numeric Formats** used to store data in a computer.

- **Integer or Fixed point Formats.**
- **Floating Point Formats**
- **Binary Coded decimal (BCD**

(b) Estimate using powers of 10 the biggest number that can be stored in 14 bits assuming:

(i) An ordinary Binary Number.

14 bits = $2^{14} – 1$ = 2^{14} = $2^{10} * 2^{4}$ = 16×10^{3}

(ii) A sign magnitude number

13 bits = $2^{13} – 1$ = 2^{13} = $2^{10} * 2^{3}$ = 8×10^{3}

(c) Given a binary number 100101101101 find its equivalent in base ten assuming it is:

(i) An ordinary binary number

100101101101 = **$(1 * 2^{11})+ (1 * 2^{8})+ (1 * 2^{6})+ (1 * 2^{5})+ (1 * 2^{3})+ (1 * 2^{2})+ (1 * 2^{0})$**

= 2048 + 256 + 64 + 32 + 8 + 4 + 1 = 2413_{10}

(ii) A sign magnitude number =

$(1 * 2^{8})+ (1 * 2^{6})+ (1 * 2^{5})+ (1 * 2^{3})+ (1 * 2^{2})+ (1 * 2^{0})$

= **256 + 64 + 32 + 8 + 4 + 1 = -365_{10}**

(iii) A two's complement number

Invert the bits and add 1 = >

100101101101 = 011010010010 + 1 = 11010010011

= $(1 * 2^{10})+ (1 * 2^{9})+ (1 * 2^{7})+ (1 * 2^{4})+ (1 * 2^{3})+ (1 * 2^{1})+ (1 * 2^{0})$

= 1024 + 512 + 128 + 16 + 8 + 2 + 1 = -1681

(iv) A packed BCD number.

1001 0110 1101 = -96

(d) Assume a base eight number 147.5_{8}. Find its equivalent in:

(i) Binary

$147.5_8 = 1100111.101_2$

(ii) Decimal

1100111.101 =

$(1 * 2^{6})+ (1 * 2^{5})+ (1 * 2^{2})+ (1 * 2^{1})+ (1 * 2^{0}) + (1 * 2^{-1})+ (1 * 2^{-3})$

64 32 + 4 + 2 + 1 + ½ + 1/8 = 103.625_{10}

(iii) Hexadecimal

0110 0111.1010 = $67.A_{16}$

3. (a) You are given two numbers **A = 11001110** and **B = 00011010**

(i) Assume that they are 2's *complement numbers*, find **A + B**

11001110
<u>00011010</u>
11101000

(ii) Assume that they are *Signed Magnitude numbers*, find **A + B**

11001110
<u>00011010</u>
10110100

(b) Perform the addition of the two IEEE/ INTEL Floating Point Numbers C3C40000 and 415A0000 and give the result as:

(i) A single precision IEEE Floating Point number.

C3C40000 = 1100 0011 1100 0100 000 00 $= -1.100010 \times 2^{8}$
415A0000 = 0100 0001 0101 1010 0000 $= 1.101101 \times 2^{3}$
$= -1.100010 \times 2^{8} = -110001 \times 2^{3} + 1.101101 \times 2^{3} =$
$= 101111.010011 \times 2^{3} = -1.01111010011 \times 2^{8}$
= Exponent = 127 + 8 = 135 = 10000111
1100 0011 1011 1101 0011 = ***C3BD3000***

(ii) A double precision IEEE Floating Point number

= Exponent = 1023 + 8 = 1031 = 10000000111
1100 0000 0111 0111 1010 0110 000 = ***C077A60000000000***

(iii) A typical (IBM) 32 bit Floating Point number

C3C40000 = 1100 0011 1100 0100 000 00 $= -1.100010 \times 2^{8}$
415A0000 = 0100 0001 0101 1010 0000 $= 1.101101 \times 2^{3}$
$= -1.100010 \times 2^{8} = -110001 \times 2^{3} + 1.101101 \times 2^{3} =$
$= 101111.010011 \times 2^{3} = -0.101111010011 \times 2^{9}$
= Exponent = 128 + 9 = 137 = 10001001
1000 1001 1101 1110 1001 1000 = ***89DE9800***

(c) Convert the following infix expression to postfix.

$$\frac{(x - y)(x + y/z)}{p - (q + r)}$$

= ***xy-xyz / + * pqr + - /***