|  |  |
| --- | --- |
|  | **Computer Organization & Assembly Language**  **BSCS 3rd**  **Department of Computer Science**  **Bahria University, Lahore Campus** |

**Quiz: 1**

Date: Week 4, 13th March 2023

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation of CLO** | **Question Number** | **Marks** | **Obtained Marks** |
| **CLO1: CLO statement**  *Identify the major components of computer architecture, and explain their purposes and interactions* | 1 | 2 |  |
| 2 | 3 |  |
| **Total Marks** | | **5** |  |

**Question 1.** What is the binary representation of D5B7h? Also calculate the unsigned and signed decimal interpretation of this number.

Note: You can check the answer from assignment.

**Question 2.**  What are Assembly Registers?

The processor operation mostly works on processing the data. Registers are the internal memory storage locations that help to speed up processor operations.  In IA-32, there are six 16-bit processor registers and ten 32-bit registers. The registers can be divided into three categories −

* ****General registers****

****Data register**** - 32-bit data registers: EAX, EBX, ECX, EDX. AX is the primary accumulator, BX is the base register, CX is the count register and DX is the data register.

****Pointer register**** – These are 32-bit EIP, ESP, and EBP registers. There are 3 types of pointers namely Instruction Pointer (IP), Stack Pointer (SP) and Base Pointer (BP)

****Index register****– These registers are further divided into Source Index (SI)  and Destination Index (DI)

* ****Control registers**** – Control registers can be defined as the combination of the 32-bit instruction pointer register and the 32-bit flags register. Some of the popular flag bits are Overflow Flag (OF), Trap Flag (TF), Interrupt Flag (IF), Sign Flag (SF), Direction Flag (DF), Zero Flag (ZF), etc.
* ****Segment registers**** – These are further divided into Code Segment, Data Segment, and Stack Segment