Question 01:

Construct a Table to compare computer architecture, computer design and computer organization.

Answer:

Computer architecture, computer design and computer organization are the three important steps to design a computer system. Here is the comparison between these three:

Topic	Computer Architecture	Computer Organization	Computer Design
Idea	Computer architecture refers to those attributes of a system visible to a programmer. It deals with high level design of computers.	Computer Organization refers to the level of abstraction above the digital logic level, but below the operating system level	Computer design refers to the low-level hardware design of a computer.
Level of Concern	Computer Architecture is concerned with the structure and behavior of the computer.	Computer Organization is concerned with the way the components are connected together	Computer Design is concerned with the development of hardware to achieve a set of specifications.
Purpose	Computer Architecture describes what the computer does	Computer organization explains how a computer works	Computer design describes in which way the hardware should be designed to get desired computer organization
Order	Computer architecture is fixed first.	Computer organization is decided after having its architecture	Computer design comes after architecture and organization.
Field of Work	Computer architecture deals with the number of registers of a computer, the instructions should be followed, the memory addressing scheme etc.	Computer organization deals with the size of cache memory, RAM, peripherals, implementation technology etc to achieve that architecture.	Computer design deals with hardware materials and connections.

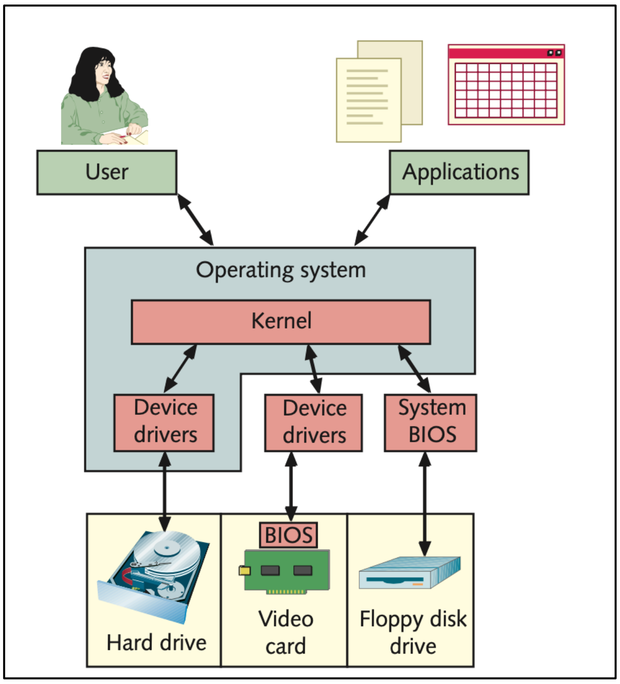
(Answer)

Question 02:

Explain how application software, operating system and computer hardware interact each other.

Answer :

Application software, operating systems, and hardware are three different layers in computing. Application software is on the top of the layer and the hardware is under these hierarchy. Here, an operating system acts as the middleman between application software and hardware. See the picture below to get an overview of these layers and the way they interact:



An operating system (OS) is a software that controls a computer. It manages hardware, runs applications, provides an interface for application software, and stores, retrieves, and manipulates files. Every OS has its own set of APIs that allow an application to interact with its kernel. On the other hand, the OS does not relate directly to the hardware. Rather, **the OS uses device drivers and the BIOS to interface with hardware.**

An application software or a software is just a list of instructions of things for the hardware to do. To run a software, we must need an operating system to interact with hardware. **The application software communicates with the OS through API (Application Program Interface).** These system calls allow access to software processor control, interruption, file management, network access etc. Then, **the OS communicate with kernel using the device drivers to send commands to the hardware to do specific tasks.**

Device drivers teach the OS to interact with each bit of hardware. All the peripherals connected with a computer relies on drivers. Without drivers, OS will not know how to use peripherals to do specific tasks.

Example of how a document writing software (ex : MS Word) will print a document using the hardware 'Printer' :

The hardware 'Printer' has its onboard firmware as well as a device driver in the systems memory to connect these both. Now the System Software is directly in contact with the device driver and knows that if a print command is requested by some application (ex : MS word) then it will use this device driver code as an entry point to communicate with the printer.

So this is the flow of that operation:

Step 1:
First, user clicks on 'Print' option on MS Word application.

Step 2:
MS-Word makes a request to the OS to inform of a print command to be executed.

Step 3:
Now The OS forward that request to the printer using device driver.

Step 4:
The printer's on-board firmware understands the incoming request and data (the file user wants to print) and does the required Job.

This is how application software, operating system and computer hardware interact with each other.

(Answer)

Question 03:

What is meant by 64-bit processor? Consider a processor with cycle time of 0.5ns. It executes 5000 floating points instructing by 9000 clocks and 8000 integer type instructions by using 12000 clocks. Calculate its MIPS.

Answer :

A 64-bit processor is a processor with a word size of 64 bits where the 'word size' refers to the number of bits processed by the computer's CPU in one go. In short, A 64 bit processor is a processor which can process 64 bits at a time.

Given that,

The cycle time of the processor, $T_c = 0.5$ ns $= 0.5 \times 10^{-9}\, s$
No. of floating point instructions, $I_{float} = 5000$
No. of integer type instructions, $I_{int} = 8000$

So, the total number of instructions,

$I_c = I_{float} + I_{int}$
Or, $I_c = 5000 + 8000$
Or, $I_c = 13000$

And, The frequency of the processor (in MHz),

$$f = \frac{1}{T_c} = \frac{1}{0.5 \times 10^{-9}\, s} = 2 \times 10^{9}\, Hz = 2000\, Mhz$$

Let CPI_i be the number of cycles required for instruction type i, and I_i be the number of executed instructions of type i for a given program. Then we can calculate an average *CPI* as follows:

$$\text{CPI} = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$$

Now,

The processor executes 5000 floating points instructing by 9000 clocks
So, the clock it takes per each floating points instruction, $CPI_{float} = \frac{9000}{5000} = 1.8$

Similarly,

The processor executes 8000 integer type instructions by 12000 clocks
So, the clock it takes per each integer type instruction, $CPI_{int} = \frac{12000}{8000} = 1.5$

So, The average clock per instruction (CPI) = $\frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$

$= \frac{(1.8 \times 5000)+(1.5 \times 8000)}{13000}$

$= 1.6154$

Finally,

$$\boxed{\text{MIPS} = \frac{Frequency\ in\ Mhz}{CPI} = \frac{2000\ MHz}{1.6154} = \mathbf{1238.08}}$$

So, the processor can process 1238.08 million instructions in 1 second (MIPS).

(Answer)