Department of Computer Science & Engineering
Microprocessor & Computer Architecture

UNIT 1 Q & A

1. Differentiate between Microprocessor and Microcontroller.

Answer:

Microprocessor	Microcontroller
Microprocessor consists of only a Central Processing Unit. It doesn't consist of RAM, ROM, I/O ports.	Micro Controller contains a CPU, Memory, I/O all integrated into one chip. It consists of CPU, RAM, ROM, I/O ports.
Microprocessor is used in Personal Computers	Micro Controller is used in an embedded system.
Microprocessor uses an external bus to interface to RAM, ROM, and other peripherals,	Microcontroller uses an internal controlling bus.
Microprocessors are based on Von Neumann model	Micro controllers are based on Harvard architecture
Microprocessor is complicated and expensive, with a large number of instructions to process	Microcontroller is inexpensive and straightforward with fewer instructions to process.
Its power consumption is high because it has to control the entire system.	It is built with CMOS technology, which requires less power to operate.
Its designing and hardware cost is high.	Its designing and hardware cost is low.

2. Differentiate between Computer Organization and Computer Architecture.

Answer:

Computer Architecture	Computer Organization
Computer Architecture is concerned with the way hardware components are connected together to form a computer system.	Computer Organization is concerned with the structure and behaviour of a computer system as seen by the user.
It acts as the interface between hardware and software.	It deals with the components of a connection in a system.
Computer Architecture helps us to understand the functionalities of a system.	Computer Organization tells us how exactly all the units in the system are arranged and interconnected.
A programmer can view architecture in terms of instructions, addressing modes and registers.	Whereas Organization expresses the realization of architecture.
While designing a computer system architecture is considered first.	An organization is done on the basis of architecture.
Computer Architecture deals with high-level design issues.	Computer Organization deals with low-level design issues.
Architecture involves Logic (Instruction sets, Addressing modes, Data types, Cache optimization)	Organization involves Physical Components (Circuit design, Adders, Signals, Peripherals)

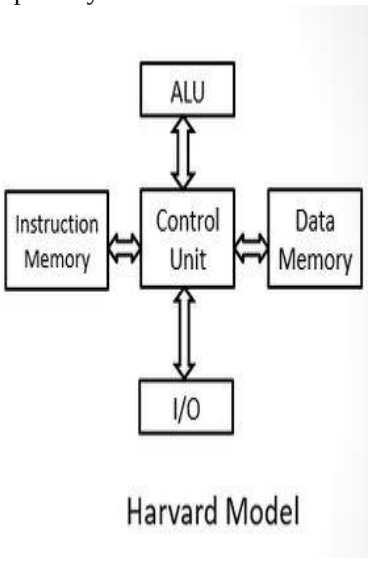
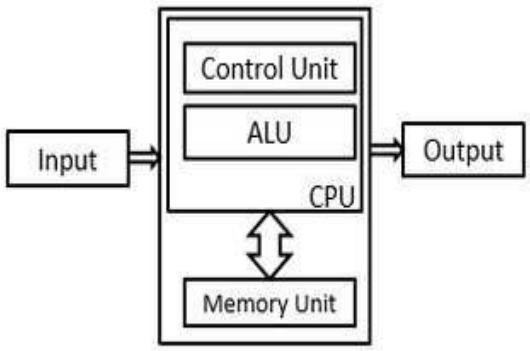
3. Differentiate between RISC and CISC.

Answer:

CISC	RISC
It has a microprogramming unit.	It has a hard-wired unit of programming.
The instruction set has various different instructions that can be used for complex operations.	The instruction set is reduced, and most of these instructions are very primitive.
Performance is optimized with emphasis on hardware.	Performance is optimized which emphasis on software
Only single register set	Multiple register sets are present
They are mostly less or not pipelined	This type of processors are highly pipelined
Execution time is very high	Execution time is very less
Code expansion is not a problem.	Code expansion may create a problem.
Decoding of instructions is complex.	The decoding of instructions is simple.
It requires external memory for calculations	It doesn't require external memory for calculations
Examples of CISC processors are the System/360, VAX, AMD, and Intel x86 CPUs.	Common RISC microprocessors are ARC, Alpha, ARC, ARM, AVR, PA-RISC, and SPARC.
Single-cycle for each instruction	Instructions can take several clock cycles
Heavy use of RAM (can cause bottlenecks if RAM is limited)	More efficient use of RAM than RISC
Simple, standardized instructions	Complex and variable-length instructions
A small number of fixed-length instructions	A large number of instructions
Limited addressing modes	Compound addressing modes
Important applications are Security systems, Home automation.	Important applications are : Smartphones, PDAs.
Varying formats (16-64 bits for each instruction).	fixed (32-bit) format
Unified cache for instructions and data.	Separate data and instruction cache.

4. How is Harvard and Von Neumann Architecture different from each other?

Answer:

Point of Comparison	**Harvard Architecture**	**Von Neumann Architecture**
Arrangement	In Harvard architecture, the CPU is connected with both the data memory (RAM) and program memory (ROM), separately.  Harvard Model	In Von-Neumann architecture, there is no separate data and program memory. Instead, a single memory connection is given to the CPU.  Von Neumann Model
Hardware requirements	It requires more hardware since it will be requiring separate data and address bus for each memory.	In contrast to the Harvard architecture, this requires less hardware since only a common memory needs to be reached.
Space requirements	This requires more space.	Von-Neumann Architecture requires less space.
Speed of execution	Speed of execution is faster because the processor fetches data and instructions simultaneously .	Speed of execution is slower since it cannot fetch the data and instructions at the same time.
Space usage	It results in wastage of space since if the space is left in the data memory then the instructions memory cannot use the space of the data memory and vice-versa.	Space is not wasted because the space of the data memory can be utilized by the instructions memory and vice-versa.
Controlling	Controlling becomes complex since data and instructions are to be fetched simultaneously.	Controlling becomes simpler since either data or instructions are to be fetched at a time.

5. What are the most notable features of the ARM instruction set?

Answer:

a. Load-store architecture
b. 3-address instructions
c. Conditional execution of every instruction
d. Possible to load/store multiple registers at once
e. Possible to combine shift and ALU operations in a single instruction

6. Explain the various modes of operation in ARM processor

Answer:

The ARM has several operating modes that are used for various purposes:

a. User mode is the basic mode in which application programs run. User mode is the only unprivileged mode, and it has restricted access to system resources. Typically, a processor spends more than 99% of its time in user mode.
b. System mode provides unrestricted access to all system resources. Supervisor mode can only be entered in certain controlled ways (discussed below), and it is typically only entered when required to manage a particular resource.
c. Supervisor mode also provides unrestricted access to all system resources. Supervisor mode is entered on reset or power-up, or when software executes a Supervisor Call instruction (SVC). Supervisor mode is similar to system mode, but offers access to a few more registers.
d. Abort mode is entered if a program attempts to access a non-existing memory location. Abort mode also offers access to a few private registers that other modes can't access.
e. Undefined mode is entered for any instruction-related exceptions, including any attempt to execute an unimplemented instruction.
f. IRQ mode is entered in response to a normal interrupt request from an external device.
g. FIQ mode is entered in response to a fast interrupt request from an external device. It is used to provide faster service for more urgent requests.

7. Explain the following instructions of the ARM processor
   a) TSTEQ R1,R2, R3
   b) BIC R5, R6, R7

Answer: Example: PRE: r6 = 0b1111, r7 = 0b0101

```
BIC r5, r6, r7      ; r0 = r6 AND (NOT (r7))
POST: r5=0b1010
```

8. Explain briefly program status register instructions.

Answer: The ARM instruction set provides two instructions to directly control a program status register (PSR).

The MRS instruction transfers the contents of either the CPSR or SPSR into a register; in the reverse direction, the MSR instruction transfers the contents of a register into the CPSR or SPSR. Together these instructions are used to read and write the CPSR and SPSR.

Syntax: MRS {} Rd MSR {}, Rm MSR {} , #immediate

The table shows the program status register instructions

MRS	Copy program status register to a general-purpose register	Rd=psr
MSR	Move a general-purpose register to a program status register	psr[field]=Rm
MSR	Move an immediate value to a program status register	psr[field]=immediate