**1. What do you understand by Instruction Set Architecture?**  
**Not given in lecture slide.**

**2. Discuss types/classification of processors based on Instruction set architecture.**  
**Not given in lecture slide.**

**3. What is addressing mode? Discuss different types of addressing modes.**  
**Not given in lecture slide.**

**4. What do you understand by instruction format?**  
The MIPS instruction format has 6 fields:

* Opcode field (1st)
* Source registers (middle)
* Destination register (4th field)
* Shift amount (5th field)
* Function code (last field)  
  (Referenced from MIPS instruction format slide.)

**5. What do you understand by memory addressing mode?**  
**Not given in lecture slide.**

**6. Show instruction**  
Example given:  
add $r12, $r7, $r8  
(Add contents of $r7 and $r8, store in $r12)

**7. What is Accumulator-based CPU?**  
**Not given in lecture slide.**

**8. Functions of PC, IR, MAR, Accumulator?**  
**Not given in lecture slide.**

**9. What do you understand by Instruction fetch?**  
Instruction Fetch is the first stage in the MIPS instruction cycle. (IF stage in 5-stage pipeline)

**10. How does the CPU calculate memory address of next instruction?**  
**Not given in lecture slide.**

**11. How a CPU is designed to run program?**  
CPU follows 5 stages in instruction execution:  
IF → ID → EXE → MEM → WB

**12. How the PC is updated while the CPU runs a program?**  
**Not given in lecture slide.**

**13. What is Control unit? Approaches to design?**  
**Not given in lecture slide.**

**14. (i) microprogram (ii) microinstruction (iii) microoperation (iv) control memory (v) micro-programmed control unit?**  
**Not given in lecture slide.**

**15. Characteristics of CISC and RISC processors?**  
✅ RISC:

* Small optimized instruction set
* One clock cycle per instruction (CPI = 1)
* Pipelining
* Large number of registers

✅ CISC:

* More instructions per instruction set
* More cycles per instruction
* Often uses microcoded instructions
* Instructions can vary in size

**16. Addressing modes of CISC and RISC processors?**  
**Not detailed in lecture slide.**

**17. What is pipelining? Benefits and challenges?**  
✅ Benefits:

* Simultaneous execution of instruction stages
* Higher instruction throughput
* Ideally 1 CPI

✅ Challenges:

* Data dependency can stall pipeline
* Branch instructions may cause pipeline flush if prediction is wrong

**18. What is data dependency?**  
Data dependency occurs when one instruction depends on the result of another. It causes pipeline stalls if the needed data isn’t yet written to register.

**19. What is Branch Instruction? How does it affect pipeline?**  
Branch instructions can stall the pipeline since the decision depends on values not yet computed.  
Solution: branch prediction or instruction fetch from both paths.

**20. What is CPI? How does pipeline reduce average CPI?**  
CPI = Cycles Per Instruction  
Pipelining allows overlapping execution of instructions, targeting CPI ≈ 1.

**21. Write CPU performance equation. Explain terms.**  
Time/Program = (Instructions / Program) × (Clock Cycles / Instruction) × (Time / Clock Cycle)

**22. Compare run time of RISC vs CISC.**  
CISC: Fewer instructions, but more clock cycles per instruction (5–6x more)  
RISC: More instructions, but each takes only 1 clock cycle → ~4× faster overall

**23. Show the instruction formats of MIPS R2000 processor.**  
MIPS instructions are 32 bits. Fields include:

* Opcode
* Source Register(s)
* Destination Register
* Shift Amount
* Function Code  
  Visual format shown in slide.

**24. John Cocke’s work and IBM RISC motivation?**  
In 1974, IBM’s John Cocke aimed to automate telephone exchanges.  
Target: 300 calls/sec = 12 million instructions/sec  
→ Required simple, fast instruction set → Led to RISC.

**25. Emer and Clark’s 1984 VAX findings?**

* 20% of instructions used 60% of microcode
* Represented only 0.2% of execution time  
  → Justified simpler RISC instructions without microcode  
  → Enabled direct execution, use of cache, register optimizations.

**26. Show how PC can be incremented (diagram)?**  
**Not given in lecture slide.**

**27. How addresses of instructions are generated in CPU?**  
**Not given in lecture slide.**

**28. How does CPU generate address of DATA?**  
**Not given in lecture slide.**

**29. Address generation if operand field has relative address?**  
**Not given in lecture slide.**