1. What is the essential characteristic of the superscalar approach to processor design?

The essential characteristic of the superscalar approach is the ability to fetch, decode, and execute multiple instructions simultaneously during a single clock cycle. This parallelism increases instruction throughput and overall processor performance.

2. What is the difference between the superscalar and superpipelined approaches?

I. A superscalar processor has multiple execution units, allowing it to process multiple instructions at the same time — this is called horizontal parallelism.

II. A superpipelined processor, on the other hand, divides its pipeline into more, finer-grained stages, allowing for higher clock speeds, but it still processes one instruction at a time — this is vertical parallelism.

III. So, superscalar = multiple instructions in parallel and superpipelined = one instruction in smaller, faster steps

3. What is instruction-level parallelism?

Instruction-Level Parallelism (ILP) refers to the degree to which independent instructions in a program can be executed simultaneously. If there are no data or control dependencies between instructions, the processor can execute them in parallel, especially in superscalar architectures.

4. What are the key elements of a superscalar processor organization?

Key components of a superscalar processor include:

- Multiple instruction fetch units: Fetch several instructions per cycle
- Multiple decode units: Decode several instructions simultaneously
- Multiple execution units: ALUs, FPUs, etc., for parallel execution
- Dependency checking logic: Detects data/control hazards between instructions
- Instruction scheduling and reordering mechanisms: For out-of-order execution and optimal resource usage (for example; reorder buffer, reservation stations)