16.1 The essential characteristic is the ability to execute instructions independently and concurrently in different pipelines

16.2 Superscalar architecture is a method of parallel computing used in many processors. In a superscalar computer, the central processing unit (CPU) manages multiple instruction pipelines to execute several instructions concurrently during a clock cycle.

Super-pipelining is the breaking of stages of a given pipeline into smaller stages (thus making the pipeline deeper) in an attempt to shorten the clock period and thus enhancing the instruction throughput by keeping more and more instructions in flight at a time.

16.3 Instruction-level parallelism, or ILP, attempts to improve processor performance by having multiple processor components or functional units simultaneously executing instructions.

16.4

True data denpendency: is a situation in which a program statement (instruction) refers to the data of a preceding statement.

Procedural dependency: Can not execute instructions after a branch until the branch is executed

Resource conflicts: is a competition of two or more instructions for the same resource at the same time

Output dependency: An output dependency, also known as write-afterwrite (WAW), occurs when the ordering of instructions will affect the final output value of a variable.

Antidependency: ls known as write-after-read (WAR), occurs when an instruction requires a value that is later updated

16.5

Instruction-level parallelism (ILP) the average number of instructions in a program that a processor might be able to execute at the same time. Determined by the number of true dependencies and procedural control dependencies in relation to the number of other instructions.

Machine parallelism of a processor is the ability of the processor to take advantage of the ILP of the program. Its determined by the number of instructions that can be fetched and executed at the same time (the capacity of the hardware).

16.6

- In-order issue with in-order completion: Issue instructions in the exact order that would be achieved by sequential execution and to write results in that same order.

In order issue with out-of-order completion: Issue instructions in the exact order that would be achieved by sequential execution but allow instructions to run to completion out of order.

Out-of-order issue with out-of-order completion: The processor has alookahead capability, allowing it to identify independent instructions that can be brought into the execute stage. Instructions are issued with little regard for their original program order. Instructions may also run to completion out of order.

16.7

- is used to store instructions which are ready for executing. After a

processor has finished decoding an instruction, it is placed in it. As long as this buffer is not full, the processor can continue to fetch and decode new instructions.

16.8

Register renaming is a technique of duplication of resources (more registers are added)

Registers are allocated dynamically by the processor hardware, and they are associated with the values needed by instructions at various points in time. Thus, the same original register reference in several different instructions may refer to different actual registers.

16.9

- Instruction fetch strategies that simultaneously fetch multiple instruction.

Logic for determining true dependencies involving register values, and mechanisms for communicating these values to where they are needed during execution

Mechanisms for initiating, or issuing, multiple instructions in parallel

Resources for parallel execution of multiple instructions, including multiple pipelined functional units and memory hierarchies capable of simultaneously servicing multiple memory references

Mechanisms for committing the process state in correct order