

21 Lecture - CS501

Important Subjective

1. **What is Instruction Level Parallelism (ILP)?**

Answer: Instruction Level Parallelism (ILP) refers to the ability of a computer processor to execute multiple instructions in parallel, thereby improving the overall performance of the system.

How is ILP different from Thread Level Parallelism (TLP)?

Answer: ILP and TLP are two different forms of parallelism. ILP focuses on executing multiple instructions in parallel within a single thread of execution, while TLP involves executing multiple threads in parallel on a multi-core processor.

What are the benefits of ILP?

Answer: The main benefit of ILP is improved performance. By executing multiple instructions in parallel, ILP can reduce the overall execution time of a program and increase the throughput of the processor.

What are the challenges of ILP?

Answer: One of the main challenges of ILP is the issue of dependencies between instructions. If an instruction depends on the results of a previous instruction, it cannot be executed until the previous instruction has completed, which can limit the level of parallelism that can be achieved.

What techniques are used to overcome the challenges of ILP?

Answer: Techniques such as instruction scheduling, register renaming, and speculative execution can be used to overcome the challenges of ILP by allowing instructions to be executed out of order and predicting the outcome of branches.

How does superscalar processing relate to ILP?

Answer: Superscalar processing is a type of processor architecture that is designed to exploit ILP by allowing multiple instructions to be issued and executed in parallel.

What is dynamic scheduling in the context of ILP?

Answer: Dynamic scheduling is a technique used in ILP to allow instructions to be issued and executed out of order based on their availability and the availability of resources such as registers and functional units.

What is speculation in the context of ILP?

Answer: Speculation is a technique used in ILP to predict the outcome of conditional branches and execute instructions based on the predicted outcome before the actual outcome is known.

How does ILP relate to pipelining?

Answer: Pipelining is a technique used to increase the throughput of a processor by breaking down the execution of instructions into a series of stages. ILP can be used in conjunction with pipelining to allow multiple instructions to be executed in parallel within each stage.

What is the role of the compiler in ILP?

Answer: The compiler plays an important role in ILP by optimizing the code to reduce

dependencies between instructions and exploit available parallelism, such as by reordering instructions or breaking them down into smaller units that can be executed in parallel.