In this module we will delve into superscalar execution, a method that improves processor performance by taking advantage of a concept called Instruction Level Parallelism (ILP). ILP is the ability of instructions in a program to be handled independently. The more independence there is, the higher amount of ILP we can extract from a program, which usually means we can execute faster. Real programs usually have dependences that limit their ILP. We will examine two techniques, one software based (instruction reordering by the compiler), and one hardware based (out-of-order processing), which try to mitigate the dependences to improve performance.

By the end of this module, you will be able to:

- Explain and evaluate the operation and performance benefits of a superscalar pipeline vs. a non-superscalar pipeline.
- Examine a piece of code and identify data dependences among instructions.
- Looking at a set of data dependent instructions, propose methods to avoid or mitigate the data dependences.
- Demonstrate the operation of an out-of-order processor by explaining how an instruction sequence can move through an out-of-order issue queue.

INTRODUCING INSTRUCTION LEVEL PARALLELISM

Help

3:08 / 3:08 1.0x

Download transcript .txt

Show Discussion New Post