Advanced Computer Architecture

Lecture 21

Reading Material

Vincent P. Heuring&Harry F. Jordan Chapter 5
Computer Systems Design and Architecture 5.2

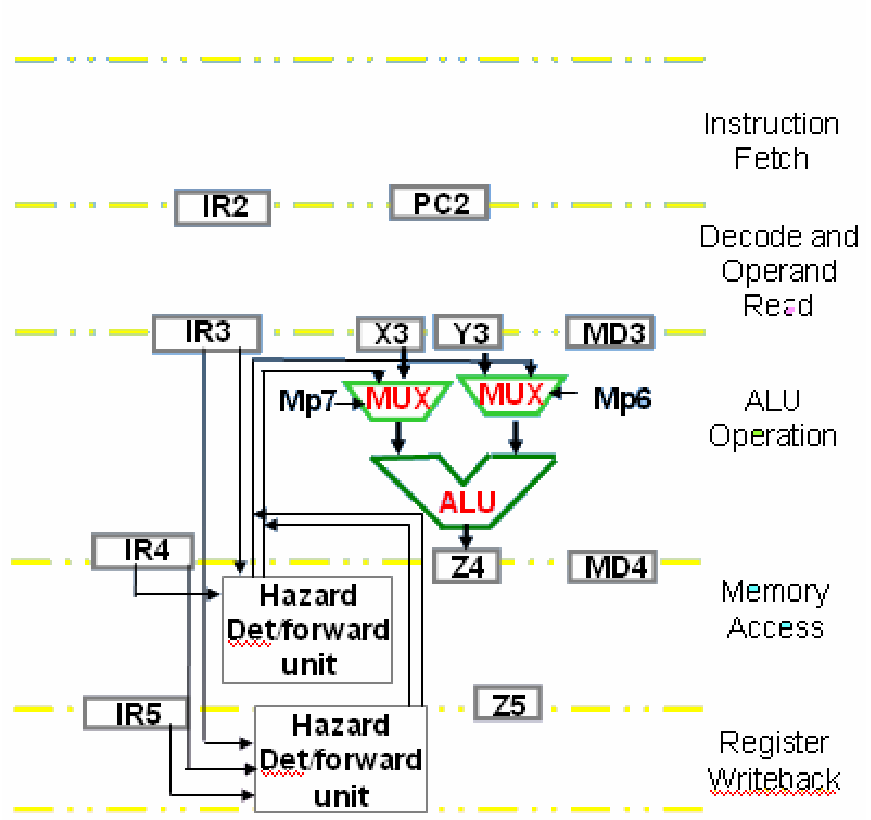
Summary

- Data Forwarding Hardware
- Instruction Level Parallelism
- Difference between Pipelining and Instruction-Level Parallelism
- Superscalar Architecture
- Superscalar Design
- VLIW Architecture

Maximum Distance between two instructions
Example
Read page no. 219 of Computer System Design and Architecture (Vincent P.Heuring, Harry F. Jordan)

Data forwarding Hardware

The concept of data forwarding was introduced in the previous lecture.



RTL for data

forwarding in case of ALU instructions

Dependence	**RTL**
Stage 3-5	alu5&alu3:((ra5=rb3):X←Z5, (ra5=rc3)&!imm3: Y ← Z5);
Stage 3-4	alu4&alu3:((ra4=rb3):X←Z4, (ra4=rc3)&!imm3: Y ← Z4);

Instruction-Level Parallelism

Increasing a processor's throughput

There are two ways to increase the number of instructions executed in a given time by a processor

- By increasing the clock speed
- By increasing the number of instructions that can execute in parallel

Increasing the clock speed

- Increasing the clock speed is an IC design issue and depends on the advancements in chip technology.
- The computer architect or logic designer can not thus manipulate clock speeds to increase the throughput of the processor.

Increasing parallel execution of instructions

The computer architect cannot increase the clock speed of a microprocessor however he/she can increase the number of instructions processed per unit time. In pipelining we discussed that a number of instructions are executed in a staggered fashion, i.e. various instructions are simultaneously executing in different segments of the pipeline. Taking this concept a step further we have multiple data paths hence multiple pipelines can execute simultaneously. There are two main categories of these kinds of parallel instruction processors VLIW (very long instruction word) and superscalar.

The two approaches to achieve instruction-level parallelism are

– **Superscalar Architecture**
A scalar processor that can issue multiple instructions simultaneously is said to be superscalar

– **VLIW Architecture**
A VLIW processor is based on a very long instruction word. VLIW relies on instruction scheduling by the compiler. The compiler forms instruction packets which can run in parallel without dependencies.

Difference between Pipelining and Instruction-Level Parallelism

Pipelining	**Instruction-Level Parallelism**
Single functional unit	Multiple functional units
Instructions are issued sequentially	Instructions are issued in parallel
Throughput increased by overlapping the instruction execution	Instructions are not overlapped but executed in parallel in multiple functional units
Very little extra hardware required to implement pipelining	Multiple functional units within the CPU are required

Superscalar Architecture

A superscalar machine has following typical features

- It has one or more IUs (integer units) , FPUs (floating point units), and BPUs (branch prediction units)
- It divides instructions into three classes
    - Integer
    - Floating point
    - Branch prediction

The general operation of a superscalar processor is as follows

- Fetch multiple instructions
- Decode some of their portion to determine the class
- Dispatch them to the corresponding functional unit

As stated earlier the superscalar design uses multiple pipelines to implement instruction level parallelism.

Operation of Branch Prediction Unit

- BPU calculates the branch target address ahead of time to save CPU cycles
- Branch instructions are routed from the queue to the BPU where target address is calculated and supplied when required without any stalls
- BPU also starts executing branch instructions by speculating and discards the results if the prediction turns out to be wrong

Superscalar Design

The philosophy behind a superscalar design is

- to prefetch and decode as many instructions as possible before execution

- and to start several branch instruction streams speculatively on the basis of this decoding
- and finally, discarding all but the correct stream of execution

The superscalar architecture uses multiple instruction issues and uses techniques such as branch prediction and speculative instruction execution, i.e. it speculates on whether a particular branch will be taken or not and then continues to execute it and the following instructions. The results are not written back to the registers until the branch decision is confirmed. Most superscalar architectures contain a reorder buffer. The reorder buffer acts like an intermediary between the processor and the register file. All results are written onto the reorder buffer and when the speculated course of action is confirmed, the reorder buffer is committed to the register file.

Superscalar Processors

Examples of superscalar processors

- PowerPC 601
- Intel P6
- DEC Alpha 21164

VLIW Architecture

VLIW stands for "Very Long Instruction Word" typically 64 or 128 bits wide. The longer instruction word carries information to route data to register files and execution units. The execution-order decisions are made at the compile time unlike the superscalar design where decisions are made at run time. Branch instructions are not handled very efficiently in this architecture. VLIW compiler makes use of techniques such as loop unrolling and code reordering to minimize dependencies and the occurrence of branch instructions.