# Data Dependences and Hazards(冒险)

## Data Dependences

**Data Dependence = Read After Write**.

An instruction J is data dependent on instruction I if either of the following holds:

* Instruction I produces a result that may be used by instruction J.
* Instruction J is data dependent on instruction K, and instruction K is data dependent on instruction I.

## Name Dependences

A name dependence occurs when two instructions use the same storage, called a **name**. However, there is **no flow of data** between the instructions associated with that name.

A name dependence can execute simultaneously or be reordered if the name used in the instruction is changed so the instructions do not conflict.

### Antidependence

**Antidependence = Write After Read.**

An instruction J writes a storage that instruction I reads. The original ordering must be preserved to ensure that I reads the correct value.

### Output Dependence

**Output Dependence = Write After Write.**

This occurs when instruction I and instruction J write the same storage. The ordering between the instructions should be preserved to ensure that the value finally written corresponds to instruction J.

## Data Hazards

A **hazard** exists whenever there is a name or data dependence between instructions.

## Overcoming Data Hazards with Dynamic Scheduling

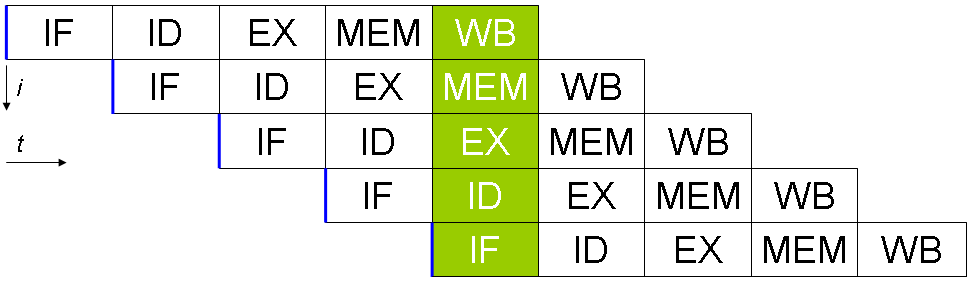
Stall: 暂停.

乱序执行比顺序执行好在哪? pp168

## Computer Architecture Cheat Sheet

### MIPS Pipelining

详见《计组：量化方法》附录C



IF = Instruction Fetch

ID = Instruction Decode

EX = Execution

MEM = Memory Access

WB = Register Write Back

### Structural Hazards (结构冒险)

多个指令争用同一个计算硬件.

对比: 数据冒险=>数据依赖.