计算机组成 Lecture Note 4

本文探讨处理器应当如何设计。首先探讨一些基本原则，而后以构建一个支持主要 MIPS 指令的处理器为主线。

基本的设计思路

联系此前对于计算机的描述，容易发现，下面的实体是必要的：

- CPU 内部的，如控制单元、ALU、寄存器堆。
- CPU 外部的：指令存储器，数据存储器，I/O。

剩余的就是如何刻画数据通路了。这里先研究一下这些实体。

寄存器堆

寄存器堆（Register File）即寄存器的集合，其中的寄存器均可通过指定的寄存器号进行读写。

寄存器堆采用组合逻辑读数据，采用时序逻辑写数据。即其总是根据输入的寄存器号输出相应的寄存器内容，而写操作由时钟和写信号控制。

> 寄存器堆的写为什么要采用时序逻辑进行设计？考虑一种情况：如果某个寄存器同时作为读和写的目标（例如 `sw` 指令中 rs 和 rt 相同），那么采用组合逻辑会造成震荡。

ALU

ALU 接受一些运算数，然后产生一些结果。同时有时可能需要支持产生一些额外的信息，如结果是否为 0 等。

特殊情况

在设计 CPU 时需要处理一些可能出现的意外情况。

非法指令

在实际的 CPU 中，如果执行一条*不合法*的指令，会怎么样？

- 可能被直接改写为 NOP 指令。
- 可能会发生异常中断。
- 可能这是一条未公布但实际上有效的指令，这时无法知道其的动作。

异常处理

硬件可能还需要进行异常检测，在发生异常时要及时中断。中断机制见 Lecture Notes 3。

单周期数据通路设计

见“MIPS 基础”一章，此处不再赘述。

单周期在现代设计中很少见，因为它的效率太低：时钟周期由执行时间最长的指令决定，这违背了“加速大概率事件”的原则。

> 在 MIPS 指令集中，这条执行时间最长的指令一般是 `lw`。

流水线设计

概念

流水线技术用于提升指令执行效率。其基本思想是：将整个指令的执行划分成多个**不交的流水段（Pipe Segment）**，每一个流水段执行一个特定的功能。指令从流水段一段进入，从另一端流出。

指令流水线的**吞吐量（Throughput）**定义为单位时间里流出的完成指令数，其取决于指令流出流水线的数目。

指令沿流水线移动一次的时间间隔就是一个**机器周期（Processor Cycle）**，由执行时间最长的流水段决定。这个机器周期一般是一个时钟周期。因此，可以说，流水线减少了指令的 CPI，或者减少了时钟周期，或者二者兼有之。总的来说，流水线带来的性能提高，是**通过增加指令的吞吐量**实现的。

但流水线实现比较复杂，因此有可能相比单周期，某一个段上的运行时间变长了。且流水线本身并不能减少*单条指令*的执行时间。甚至由于段时间延长，还可能增加时间。

> 流水线技术是硬件级别实现的，程序员无法控制。

为了能让流水线能够流动，流水段之间需要保存必要的、下一段需要的信息。这些信息就保存在**流水线寄存器**中。

MIPS 流水线

在 MIPS 指令集下，流水线可以被划分成 5 段：

1. IF 段：取指令。
2. ID 段：译码。
3. EXE 段：运算。
4. MEM 段：读写存储器。
5. WB 段：结果写回寄存器堆。

流水线冒险

有的时候流水线上会出现一些问题，称为**冒险（Hazards）**。可以把问题分成下面几类：

1. **结构冒险（Structural Hazards）**
2. **数据冒险（Data Hazards）**
3. **控制冒险（Control Hazards）**

结构冒险

结构冒险往往由硬件资源的缺乏所导致。因此增加硬件往往就可以解决这一问题。

数据冒险

当后一条指令的参数要用到前一条指令的结果时，就会发生数据冒险。

通常情况下在 ID 阶段就可以检测冒险。冒险来源有：

- EXE 段指令的运算结果
- MEM 段指令的运算结果
- MEM 段指令的读存储器结果

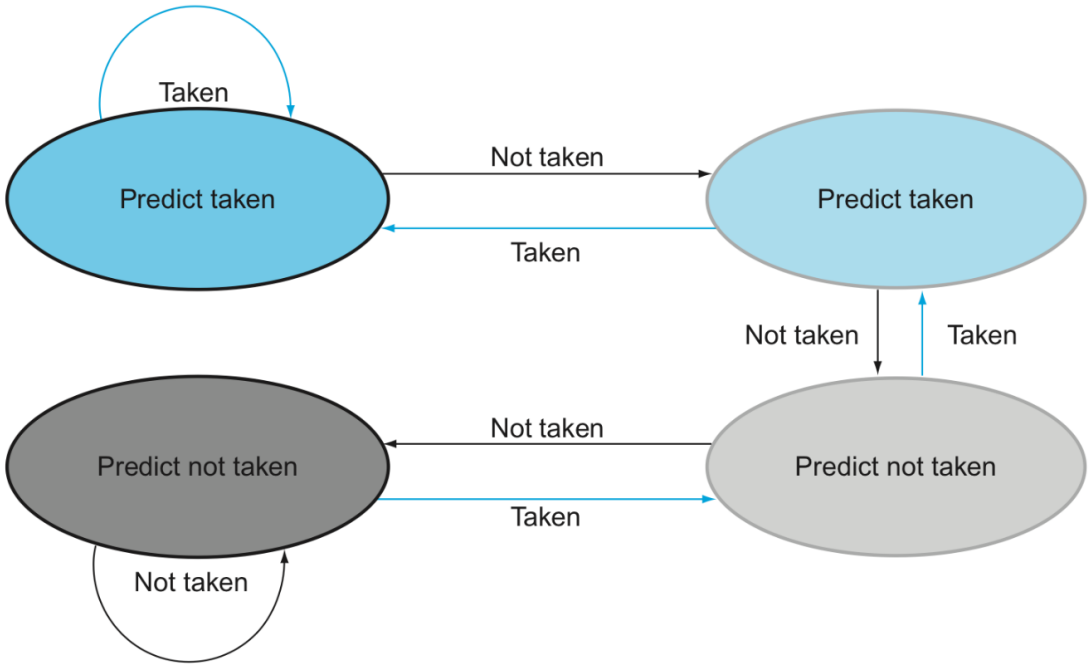
这时都可以直通到 ID 段。但如果出现像 `lw` 指令要读存储器才有、而下一条指令就要用到结果的情况时，由于没法一次跨两级，因此要停顿一个周期。

控制冒险

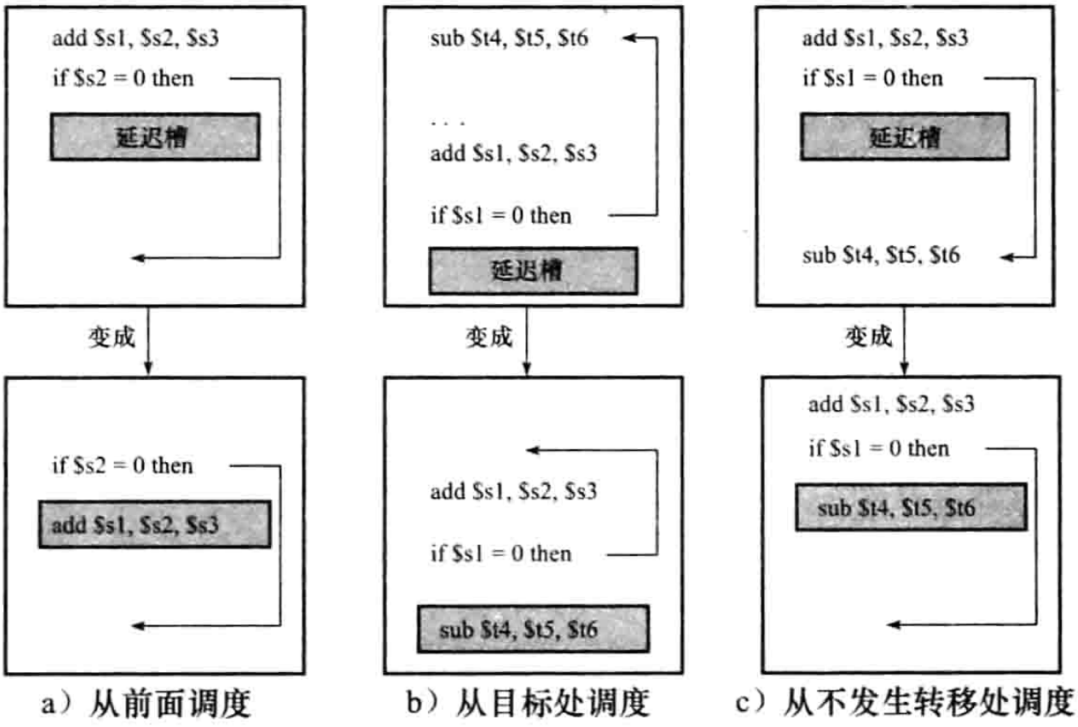
转移、分支指令可能导致此前已经进入流水线的指令作废。因此要尽可能避免。

几种简单的调度方法：

- 始终认为转移不发生。转移发生时，冻结或者冲刷流水线。
- 预测转移。使用某种状态机制，对分支指令建立状态，在 IF 阶段根据指令地址对应的*预测位*判断分支是否发生。例如 1 位预测位，按照预测位走错误时翻转预测位。又如 2 位预测位，预测位可以容忍一次错误。如图所示。



- 使用延迟槽。将总在分支后执行的那条指令放入延迟槽，从而不冲刷流水线，让这条指令执行完成（即让跳转的跳过延迟一个周期）。延迟指令的来源如图所示。当方案 a 无法实现时，用 b 或 c。b 一般对应*分支发生*时，c 一般对应*分支不发生*时。



> 实际的 MIPS 指令集中，分支和跳转指令是“延迟的”，即分支和跳转后的那条指令总是会在 PC 改变前执行。这样的话，如果把分支和跳转都前移到 ID 阶段，那么就没有这部分的损失了。

高速 CPU 的其他影响因素

- 芯片板上的布局和布线（Place & Route）方式对 CPU 的最大工作频率有很大的影响。
- 多发射 CPU。流水线发掘了**指令级并行（Instruction-level Parallelism）**。有两种方法可以增加指令级并行，一种是增多流水线的级数，另一种是复制内部元件数目，让一级流水线可以跑多个指令。这就是**多发射（Multiple Issue）**。

致谢

本文的主要内容参考自：

- 上海交通大学《计算机组成》（课程代号：EI332）一课的课程材料。
- 《Computer Organization and Design: Hardware/Software Interface (5th edition)》

本文的图片来自

- 《计算机组成与设计：硬件/软件接口（第 5 版）》
- 《Computer Organization and Design: Hardware/Software Interface (5th edition)》