

## Homework 5

PB22051080 王珏

2024 年 4 月 8 日

## 目录

<b>1 流水线延迟问题</b>	<b>1</b>
<b>2 设计新的操作指令</b>	<b>2</b>
2.1 例题 . . . . .	2
<b>3 流水线改进 2</b>	<b>5</b>
3.1 影响时钟周期 . . . . .	5
3.2 是否可以提高 or 降低流水线性能 . . . . .	5
<b>4 Thinking</b>	<b>5</b>
4.1 影响流水线性能发挥的因素有哪些? . . . . .	5
4.2 为何 RV 只有 Load/store 指令访存 . . . . .	6
4.3 单周期、多周期的控制信号无须 buffer, 而流水线的控制信号需要 buffer 的原因 . . . . .	6
4.4 流水线控制器的实现方式 . . . . .	7

## 1 流水线延迟问题

**4.16** 在本题中将讨论流水线如何影响处理器的时钟周期。假设数据通路的各个流水级的延迟如下：

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

同时，假设处理器执行的指令分布如下：

ALU/Logic	Jump/Branch	Load	Store
45%	20%	20%	15%

**4.16.1** [ 5 ] < 4.5 > 在流水化和非流水的处理器中，时钟周期分别是多少？

**4.16.2** [ 10 ] < 4.5 > 在流水化和非流水的处理器中，对于 ld 指令的延迟分别是多少？

**4.16.3** [ 10 ] < 4.5 > 如果我们将数据通路中的一个流水级拆成两个新流水级，每一个新流水级的延迟是原来的一半，那么我们将拆分哪一级？新处理器的时钟周期是多少？

图 1: 基本延迟

在非流水线中，以单周期为例， $T = 250ps + 350ps + 150ps + 300ps + 200ps = 1250ps$ 。

而指令流水线以延迟最长的阶段为主，则  $T = 350ps$

对于 ld 指令的延迟来说，非流水处理器执行五个环节，延迟  $T = 250ps + 350ps + 150ps + 300ps + 200ps = 1250ps$

但是为了避免结构冒险，流水化处理器仍然有整个机器周期的延迟  $T = 350 \times 5 = 1750ps$

拆分 ID，300ps。

## 2 设计新的操作指令

### 2.1 例题

4.13 尝试添加 RISC-V 中的指令: `ss rs1, rs2, imm` (存储两数之和)。

250第 4 章

指令释义:  $\text{Mem}[\text{Reg}[\text{rs1}]] = \text{Reg}[\text{rs2}] + \text{immediate}$

4.13.1 [10] <4.4> 对于这条指令, 需要添加的新功能部件是什么?

4.13.2 [10] <4.4> 现有的哪些功能部件需要改造?

4.13.3 [5] <4.4> 对于这条指令, 需要新添加的数据通路是什么?

4.13.4 [5] <4.4> 为支持这条指令, 为控制单元新添加的控制信号有哪些?

4.13.5 [5] <4.4> 修改图 4-21 并实现该条指令。

图 2: 设计新指令

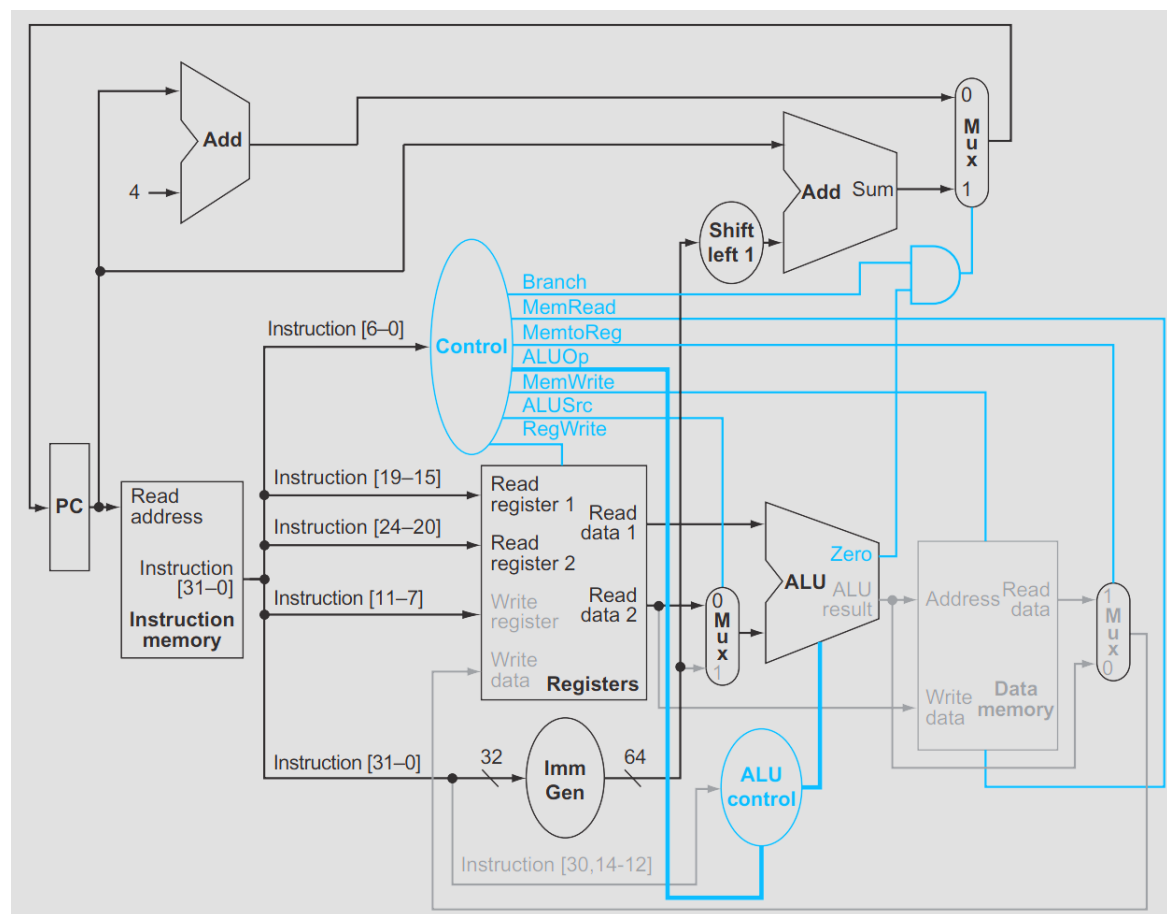


图 3: 控制信号通路

- 1 新增加的功能部件是几个多路选择器，将其连接在寄存器堆的第二个读出数据的端口处，通过控制信号决定是否要将该数据传入内存的读地址端口。
- 2 需要修改控制信号的生成单元（增加一些信号），其他无需改变。
- 3 除了第一点提到的以外，在 `ALUresult` 输出之后，也需要进行由控制信号控制的多路选择器，决定该运算结果是作为地址访存还是直接写回寄存器的值**以外的**——作为 `write data` 写入 MEM。
- 4 首先 `ALUop` 信号需要添加，其次需要在写一个能否将目的寄存器读出值作

为访存地址的信号，以及运算器运算结果是否作为写数据写入内存的控制信号。

5 如图。

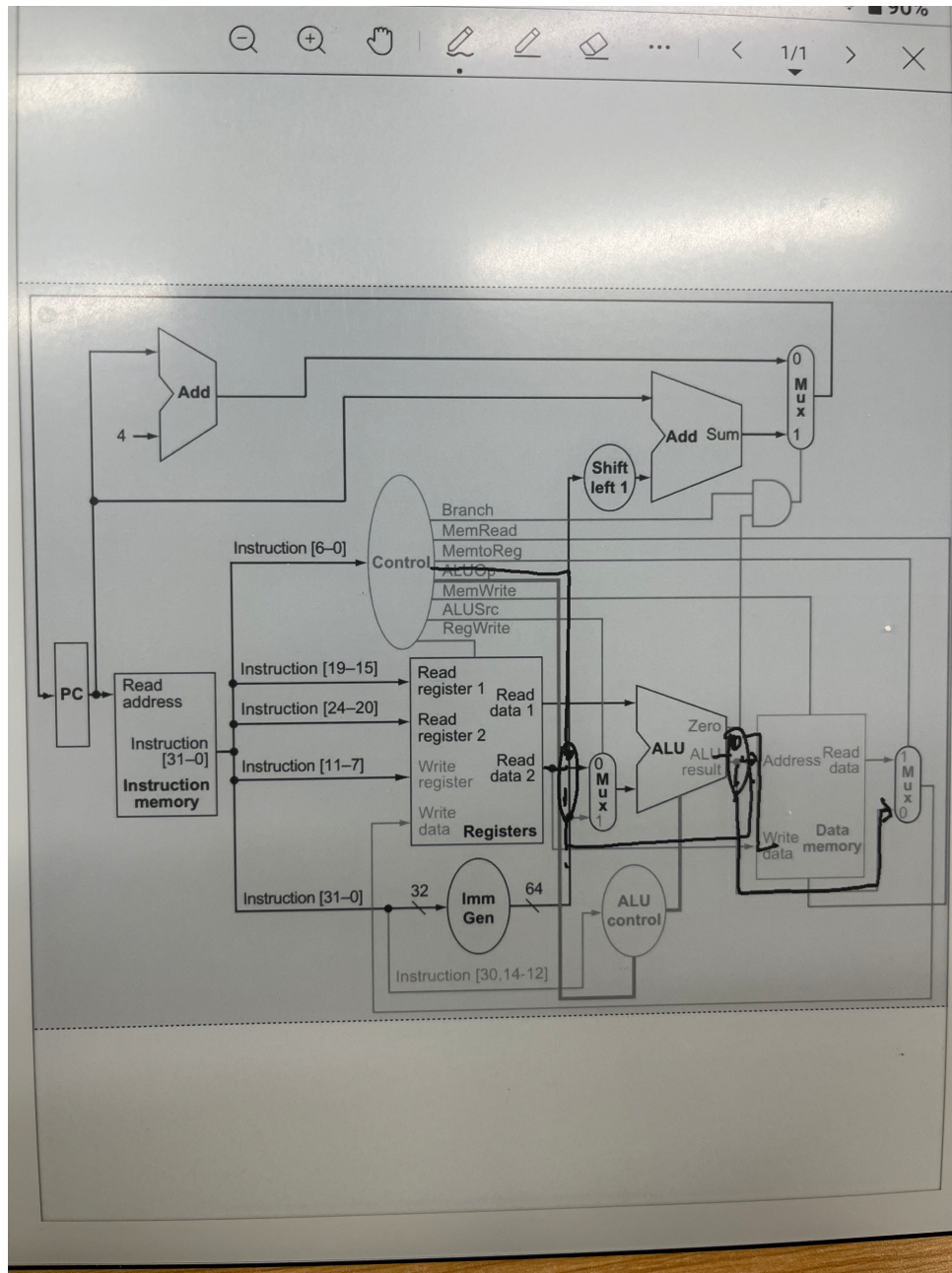


图 4: 控制信号通路添加的结果

## 3 流水线改进 2

**4.23** 如果我们改变 load/store 指令格式，使用寄存器（不需要立即数偏移）作为访存地址，这些指令就不再需要使用 ALU（具体见习题 4.15）。这样的话，MEM 阶段和 EX 阶段就可以重叠，流水级数变为四级。

**4.23.1** [10] < 4.5 > 流水线级数的减少会影响时钟周期吗？

**4.23.2** [10] < 4.5 > 这样的变化会提高流水线的性能吗？

**4.23.3** [10] < 4.5 > 这样的变化会降低流水线的性能吗？

图 5: 改变 ld 和 st 指令

### 3.1 影响时钟周期

其实是有可能影响的，当我们重叠的阶段是实际耗时（延迟）最长的阶段，则合并后可能会使最长的耗时降低从而影响时钟周期。

### 3.2 是否可以提高 or 降低流水线性能

流水级数减少会导致每个时钟周期内可以并行执行的指令数量减少，使流水线的吞吐率降低。从而使得其处理指令的整体效率降低。但是另一方面，较短的流水线可以减少硬件资源的需求，简化处理器的设计降低功耗。同时它可能可以减少流水线冲突的发生（如数据冲突、控制冲突等），可以在发生冒险的时候减少依赖，提高运行效率（在存在大量分支时也可以减少分支损失）。

## 4 Thinking

### 4.1 影响流水线性能发挥的因素有哪些？

- 首先就是最重要的一点——流水线的冲突（冒险）问题。包括数据冲突（在后续指令需要前面指令的执行结果作为输入时），控制冲突（在分支语句判断时，stall——流水线暂停等待），结构冲突（不同指令同时访问一个硬件资源）

- 分支预测的准确性会直接影响到流水线的效率。
- 程序中不同指令之间的独立性决定了流水线能够同时处理多少条指令。如果依赖关系较多也会限制流水线的效率。

## 4.2 为何 RV 只有 Load/store 指令访存

- 这种设计原则体现了 RISC 架构的核心原则之一——“加载/存储架构”
- 这样做可以简化处理器设计（限制能够直接访问内存的指令类型），让处理器的其他部分（如 EX 模块）可以设计得更简单。
- 大部分指令只在寄存器里操作，这些数据访问和修改过程的时间耗费远低于内存访问时间。有助于减少处理器在执行大多数指令时的延时。
- 可以增强指令的并行性，因为寄存器之间的操作通常比涉及内存的操作有更少的依赖和冲突。

## 4.3 单周期、多周期的控制信号无须 buffer，而流水线的控制信号需要 buffer 的原因

单周期和多周期的控制信号在整个指令执行过程中是同时生效的，因此无需使用缓冲器。每条指令在单周期和多周期处理器中执行所需的时间是固定的，因此可以简单地通过控制信号来控制整个处理过程。

而在流水线处理器中，各阶段的指令处理是并行的，每个阶段的操作可能需要不同的时间完成。因此，在流水线中，不同阶段之间的控制信号需要进行时序上的协调，以确保指令在流水线中正确地流动。由于不同阶段的处理速度可能不同，因此需要使用缓冲器来暂存某些阶段产生的结果，以便在后续阶段需要时使用。



简而言之，流水线处理器需要使用缓冲器来协调不同阶段之间的控制信号，以及处理不同阶段之间的速度不匹配问题，而单周期和多周期处理器由于整个指令执行过程是按照固定的时间完成的，因此不需要使用缓冲器。

#### 4.4 流水线控制器的实现方式

流水线控制器是处理器架构中的一个关键组件，以下是其两种主要的实现方式：

- **硬连线控制 (Hardwired Control)**

- 通过硬件逻辑电路直接实现控制信号的生成。
- 优点：速度快，适用于高性能处理器设计；简化设计，对于简单或中等复杂度的流水线。
- 缺点：灵活性较低，修改控制逻辑需要重新设计和制造电路；设计复杂性高，难以管理和调试。

- **微程序控制 (Microprogrammed Control)**

- 通过一个称为微指令序列的固件程序来实现控制逻辑。
- 优点：灵活性高，通过修改微程序可以较容易地更改或更新处理器的控制逻辑；设计简化，使设计更加模块化和可管理。
- 缺点：速度慢，由于控制信号的生成依赖于微指令的执行；占用额外资源，需要额外的存储空间来存放微程序。