

COD

HW7

T1

h5 (1)

$$(3 + 3 + 2 + 1 + 1) \times 1000 = 10000ns$$

h5 (2)

最小时钟周期为 3ns

$$\text{执行时间为 } (3 \times 5) \times 1000 = 15000ns$$

h5 (3)

最小时钟周期为 3ns

$$\text{执行时间为 } 3 \times 1004 + 100 \times 3 + 60 \times 6 = 3672ns$$

h5 (4)

时钟周期：时钟周期越小，性能越好

指令结构：lw 后紧跟 R-type ALU 指令会造成停顿，停顿越少性能越好

分支预测：预测失败要冲刷流水线，所以预测成功率越高，性能越好

T2

h5 (1)

最多 64 位

h5 (2)

不能，无法在一个时钟周期内完成

h5 (3)

t2 的正确结果在 ALU 的输出线上（EX 段）；t1 的正确结果在 DM 的读地址端（MEM 段）

h5 (4)

t0 的正确结果在乘法模块的输出端（EX 段）

流水线应该停顿一个周期，等待 mul 指令计算完成；流水线应该将 t0 的数据前递给 ALU 的输入端

T3**h5 (1)**

改写如下：

```
1 addi x5, x0, 0x14
2 addi x6, x0, 0x18
3 lw x7, 0(x5)
4 lw x8, 0(x6)
5 add x6, x7, x8
```

h5 (2)

对于新加的三个指令，更改 EX 阶段的功能为读内存（内存需要多个读端口），更改 MEM 阶段的功能为运算（需要加一个 ALU 模块），其他级不变。

ALU 的第一个输入数据为 EX 阶段读出的 MEM(x[rs1])；

ALU 的第二个输入数据有三种选择：

- ① EX 阶段读出的 MEM[x[rs2]]
- ② ID 阶段读出的 x[rs2]
- ③ 立即数

以 addmr 为例，各阶段主要任务如下：

- IF：在 IM 中取指令
- ID：指令译码，获得 x[rs1], x[rs2], x[rd]
- EX：读内存，取得 MEM[x[rs1]]
- MEM：计算 MEM[x[rs1]]+x[rs2]
- WB：将计算结果写回 x[rd]

h5 (3)

需要将还未写入内存的 x5 的数据前递到 ALU 的输入端

h5 (4)

优点:

- 结构简单, 易于设计
- 指令精简, 使用率均衡
- 程序执行效率高

缺点:

- 指令数少, 功能不如 CISC 强大
- 寻址方式不灵活