

Week 13. 流水线.

Q1. $T = t_{\text{ok-to-q}} + t_{\text{memread}} + t_{\text{rfread}} + t_{\text{alu}} + t_{\text{memread}} + t_{\text{mux}} + t_{\text{rfsetup}}$
 $= 30 + 250 + 150 + 200 + 250 + 25 + 20 = 925 \text{ ps}$
 $f = \frac{1}{T} = 1.08 \text{ GHz}$

Q2. 取址: $t_1 = t_{\text{ok-to-q}} + t_{\text{memread}} + t_{\text{setup}} = 30 + 250 + 20 = 300 \text{ ps}$
译码: $t_2 = t_{\text{ok-to-q}} + t_{\text{rread}} + t_{\text{setup}} = 30 + 150 + 20 = 200 \text{ ps}$
ALU: $t_3 = t_{\text{ok-to-q}} + t_{\text{alu}} + t_{\text{setup}} = 30 + 200 + 20 = 250 \text{ ps}$
访存: $t_4 = t_{\text{ok-to-q}} + t_{\text{mread}} + t_{\text{setup}} = 30 + 250 + 20 = 300 \text{ ps}$
写回: $t_5 = t_{\text{ok-to-q}} + t_{\text{mux}} + t_{\text{rfsetup}} = 30 + 25 + 20 = 75 \text{ ps}$

$$T = \max(t_1, t_2, t_3, t_4, t_5) = 300 \text{ ps}$$

$$f = \frac{1}{T} = 3.33 \text{ GHz}$$

Q3. ① $\text{speed-up} = \frac{T_{\text{single}}}{T_{\text{pipeline}}} = \frac{925 \text{ ps}}{300 \text{ ps}} = 3.08$

②. 因为流水线执行时间不平衡, 有些步骤所需时间是比一个周期小的

Data Hazard. & Forwarding

Q1: 第2条指令中, rd 处的 指令 需要用 到 第1条指令中 rd 处的 指令 中的数据, 但 该值 在 第1条 执行 完 最后-阶段 才会在 相应寄存器中更新.

第3条中的 指令 也有着 同样的问题.

解决: 我们可以在 第1条指令 执行 完 EX 步骤, 就将 新的 指令 中 要存 的值 传递 给 接下来 的 指令.

Q2:

① if regWrite (1)

and. (rtw) == regDst(1) or rsw) == regDst(1),

AluOut(1)

② if regWrite (2)

and (rtw) == regDst(2) or rsw) == regDst(2),

AluOut(2)

③ if regWrite (3)

and (rtw) == regDst(3) or rsw) == regDst(3),

AluOut(3)

Data Hazard & stall

Q1: add 指令需要在 EX 之前 用到 $\$t0$ 中的值, 但 lw 在最早也是在 MEM 之后更新 $\$t0$ 中的值

Q2: 通过加入 nop 来延迟一个周期, 这样可以使 add 指令获得 lw MEM 之后所更新的 $\$t0$ 中的值
及时

Q3: Mem to Reg (→) && Reg to Mem (→) &&

if ($rs(w) == regDst(1) \parallel rs(w) == regDst(2)$)

Control Hazard.

Q1. Reordered set 1

```
addui $t0, $t1, 5.  
beg $t0, $s0, label.  
or $t2, $t3, 0xff  
lw $t4, 0($t0)
```

Reordered set 2.

```
addui $t0, $t1, 5.  
or $t2, $t3, 0xff  
beg $t0, $t2, label  
nop  
lw $t4, 0($t0)
```