

4.1

(1) \because 没有使用内存. $\therefore \text{MemRead} = 0, \text{MemWrite} = 0.$

又 \because 无 Branch 指令. $\therefore \text{Branch} = 0.$

$\because \text{Zero} = x. \therefore \text{RegWrite} = 1$

ALU operation = AND.

Regd mutex = 0. ALU mutex = 0.

(2) ~~1~~ Add 单元 (PC+4).

寄存器. (获取数据). ALU 单元. 数据存储器. 指令存储器.

(3). Memory 单元和第 2 个 Add 单元产生输出. 但不会被指令用到.

所有功能单元都会产生输出.

4.4.

(1). $25\% + 10\% = 35\%$. Load / Store.

(2). 所有指令.

(3). $28\% + 25\% + 10\% + 11\% = 74\%.$ I / Load / Store / Branch.

(4). 空闲状态. 不执行操作.

4.5.

(1). 0110.

(2). PC+4.

$0 \times 00C6BA23 + 4 \Rightarrow 0 \times 00C6BA27.$

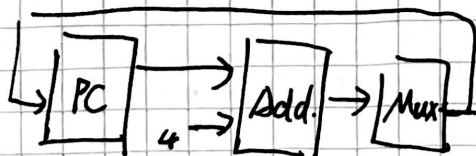
(4). ALU 的输入: rs, rt 寄存器值.

两个加法单元: 一个计算 PC 值.

另一个未使用 (R 型) / 计算跳转 (分支)

(5) ALU 的结果: 写回寄存器 rd 的数据. 00110000.

路径:



(3). 目的寄存器. 输入: 00110. 10110 输出: 10111.

ALU 操作数: 输入为寄存器堆的输出. 及 16 位的符号扩展.

输出为堆的输出

写回: 输入为 ALU 的执行结果和保存阶段的数据.

输出为 ALU 的执行结果.

PC: 输入输出均为 PC+4



4.7.

(1). R型延迟:

$$250 + 30 + 200 + 20 = 500 \text{ ps}$$

(2) lw 延迟.

$$250 + 30 + 50 + 200 + 250 + 20 = 800 \text{ ps}.$$

(3). Sw 延迟

$$250 + 30 + 50 + 200 + 250 = 780 \text{ ps}.$$

(4) beq 延迟

$$250 + 30 + 200 + 150 = 630 \text{ ps}.$$

(5). 非load 加载的I型指令. 不需访问数据存储器. 但需要符号扩展.

$$250 + 30 + 150 + 200 + 20 = 550 \text{ ps}.$$

(6) 最小时钟周期内执行. 最长的指令决定. 800 ps (lw).

4.10.

由4-7. 4-8.

延迟: R: 1120 ps lw: 1420 ps Sw: 1250 ps beq: 1105 ps .

解/逻辑/I型: 1170 ps . 最小时钟周期: 1420 ps

$$(1). \frac{52\% \times \frac{1}{2} (1120 + 1170) + 25\% \times 1420 + 11\% \times 1250 + 12\% \times 1105}{52\% \times \frac{1}{2} (1140 + 1190) + 25\% \times 1440 + 11\% \times 1260 + 12\% \times 1115} = 1.08.$$

(2) 性能提高.

开销 =	R.	lw	Sw	beq.	I型.
原开销.	2075.	4175	3975	1986	2175
后来.	2475.	4575	4175	2186	2575.

$$\frac{\text{原}}{\text{后}} = \frac{52\% \times \frac{1}{2} (2075 + 2175) + 25\% \times 4175 + 11\% \times 3975 + 12\% \times 1986}{52\% \times \frac{1}{2} (2475 + 2575) + 25\% \times 4575 + 11\% \times 4175 + 12\% \times 2186} = 0.99.$$

\therefore 开销增加.

(3) 当lw和Sw指令较多. 对寄存器堆访问较少时. 增加更多寄存器. 相反. 对寄存器堆访问较多时. 无用.

4.15. (1). 600 ps

(2) 更慢. 增加了 $200 + 150 + 20 = 370 \text{ ps}$.

(3) 指令数目. CPU时钟频率

(4) 性能新值.

