

1、提前分支判断使 T_{usert} 与 T_{users} 等于 0, 会产生数据冒险并产生阻塞暂停, 影响效率, eg. $lw \$t0, 0(\$t1)$ 对于不提前分支, 在 M 级的 add

$add \$s0, \$s1, \$s2.$ 等待 M, W 级的转发
 $beq \$t0, \$s0, loop1.$

和在 W 级 lw 向 E 级的 beq 转发数据; 而对于提前分支, beq 在 D 级时需暂停一个周期, 和 W 级 lw 向 E 级的 beq 转发数据; 而对于提前分支, beq 在 D 级时需暂停一个周期,

2、因为延迟槽的存在, B/J 指令的后一条指令不会被清空, 会继续执行, 因此 JAL 绑定“下”一条指令时, 会绑定“下下”条指令, 即 $PC+8$.

3、转发来自功能部件的数据会浪费各级运转的时间, 极大的降低了效率, 而流水线的意义就在于提高效率, 减少每个时钟周期的时间, 与之相悖, 故不转发。

4、可以不进行内部转发, 而进行外部显式转发, 即 $W \rightarrow D$

5、供给者: M, W 级流水线寄存器, 准确说为应写入 GRF 的数据

需求者: $E: ALUA, ALUB, RD1, RD2.$ $D: M: WD$

九条通路 $M: (ALUR/PC8)$ $\left\{ \begin{array}{l} RD1 \\ RD2 \\ ALUA \\ ALUB \end{array} \right.$ $W(ALUR/PC8/LoadData)$ $\left\{ \begin{array}{l} RD1 \\ RD2 \\ ALUA \\ ALUB \\ WD \end{array} \right.$

7、集中译码 + 指令驱动型

优: 直观, 易懂, 不易遗漏 劣: 各级间传输信号的量, 加指令时易于添加 整体代码量会随指令数量增多而显著增多

6、R: 对于位运算, $ALUSrc$ 需添加一个 Rt 输入, 相应多一位控制信号

其余改变 $ALUControl$ 即可, $ExtOp$

存储: lb, lh 等, 添加 $LoadControl$ 信号, 将 $Addr[11:2]$ 位读取到的数据

拆成 4/2 块, 根据 $Addr[1:0]$ 位选择相应的块数, 此刻是一字节或二字节, 后进行

符号拓展存入 GRF

sb, sh 等, 依然使用组合逻辑, 先将原读取到的数据提出,

在对应位置进行数据替换后存入即可, 可共用一个 $DMControl$.

J/B: 增加判断信号, $nPCSel$ 信号

使用组合逻辑判断是否分支

