

前递和冒险部分提示

ICS (2021 秋)

11 月 2 日

1. 流水线中产生的信号可能未来得及更新到状态中，就需要使用，因此可能产生冒险。大部分解决方案为前递，即产生的信号立刻转发。要点：
 - 需要前递的都是要“写”更新的数据，其中都是写到寄存器中，都会在 E 过程中或者之后才产生；产生的信号如果需要前递，则在产生后立刻前递。
 - 由此理解并记忆 5 个前递数据来源 (`e_valE`, `m_valM`, `M_valE`, `W_valM`, `W_valE`, 思考分别可能对应什么情况) 和两个前递目的。
 - 前递时，判断要读的值是不是有阶段打算写入它，决定是否递入；各递送数据信号有优先顺序，流水线中**较早**阶段产生的信号优先 (为什么?)，由此理解并记忆 `val_A`, `val_B` 的前递选择逻辑。
2. 部分情形中，需要用到的信号还没产生，无法前递。包括 `ret`, `mispredicted jXX`, `load-use` 几种。这种冒险的处理有一定难度，思考要点如下：
 - 首先要确定流水线需要特别处理该指令的阶段：什么时候产生的影响开始？什么时候产生的影响结束？例如，`ret` 指令在通过 D、E、M 三个阶段时，流水线都有问题 (都在取错指令)：其中 D 阶段开始取错指令，M 阶段行将结束时才能拿到返回地址，进行转发。又如，`load-use` 的访存指令通过 E 时，前一阶段的 D 才出现问题，而访存通过 M 之后才能转发信号。`mispredicted jXX` 影响最短，E 时可能发现预测错误，E 结束之后也就能确定下一条指令位置了。
 - 由此理解并记忆检查特殊冒险是否出现的逻辑 (糟糕的指令在哪些阶段时要特别处理)，例如返回指令对应于 `IRET in { D/E/M_icode }`；`load-use` 只有将内存加载到寄存器的时候才发生，而且是 E 阶段开始出问题，所以是 `E_icode in { IMRMOVQ, IPOPOP }` (以及需要 `E_dstM in { d_srcA, d_srcB }`)，由于它没有取错指令的问题，而且后来会 `stall` 一下，所以 M 阶段无需判断。

- 发现问题后，对流水线各阶段的处理方式包括：bubble、stall 和 normal。Bubble 的意思是通过（类似）无意义的指令填充流水线（气球车），目的是使得下一阶段有事可做或者使得这条指令“消失”。Stall 的意思是暂停这条指令向下一阶段进行（反复在这阶段洗车）。暂停规律：若某阶段是 stall，则下一阶段如无暂停必要就是 bubble，最后一个暂停后一定是 bubble；stall 的前一阶段一定是 stall。
- 然后确定产生影响的每个阶段，各流水线寄存器的下个瞬间处理方式（FDEM 的表）。这里强调每个阶段是因为只有 ret 比较特殊（尽管处理方式一样），剩下两种情况都是只需要处理一个阶段。分析：
 - mispredicted jXX 在 E 时，可能发现前两条指令都取错了。这时，我们需要让这两条指令消失，这两条指令即将（“下个瞬间”）分别进入 D、E 阶段，因此 D、E 设为 bubble，其他均为 normal。接下来正确指令的地址就知道了，冒险结束。
 - 类似地，load-use 在 E 时，前面的指令有问题。这时它需要等一下，所以下一瞬间 D 设为 stall，根据前面的规律，E 设为 bubble，F 设为 stall，其他均为 normal。由于它没有取错指令的问题，而且 F 会 stall 一下，所以 M 阶段无需其他判断或处理，正常前递，冒险结束。
 - ret 最为“麻烦”。它在 D 时，前面处于 F 阶段的指令已经出现了问题。那 F 阶段是 stall 还是 bubble 呢？我们指出 F 阶段是不可以 bubble 的（Y86-64 实现中的 bubble 是将寄存器设为 reset 状态；F 仅仅是预测的 PC，无法模拟出 nop 的状态）。这样我们令 F 阶段 stall（反复地尝试取指），根据规律，后一阶段应该是 bubble，其余 normal。这样一来 ret 到 E、M 两个阶段时，都是让 F stall，D bubble，实际上连续产生了三个气泡。等 M 结束后，下一次取指才能正常，冒险结束。
- 根据插入气泡的总个数，三种冒险中，load-use 浪费一个周期，mispredicted jXX 浪费两个周期，ret 浪费三个周期。这在今后计算执行消耗周期数的时候可能有用。

3. 2 中的组合情况。注意，返回有三种位置，DEM 皆有！

- 首先，不是每种组合情况都能发生。这需要结合影响开始的阶段来分析。例如，ret 在 E、M 两个阶段都不可能发生组合，因为前面都是气泡。load-use 和 mispredicted jXX 也不可能发生组合，因为二者开始影响的时期都是 E，结束都是 M，重叠了。

- 组合时，一般采用的策略是 stall > bubble > normal，当然也要求符合前述暂停规律。

要求：在理解原理的基础上，完整阅读 pipe-full.hcl 文件，熟记异常处理、其他组合逻辑等代码。遇到不寻常的行为结合图 4-52 中的数据通路找原因（有一些信号连过去了就应当要用，没有连过去的就不能用！）。可以通过做课本上的 practice problem 练习回忆 HCL 代码。然后做往年题练习，思考方式可类比上面。