

# 智能硬件体系结构

## 2025 秋-作业 2

2025 年 11 月 21 日

### 1、五级流水线数据冒险与优化分析（30 分）

考虑以下指令序列在五级流水线（IF, ID, EX, MEM, WB）中的执行：

1. A: LD 4(X2), R1 //  $R1 = \text{Mem}[X2+4]$
2. B: LD 8(X2), R3 //  $R3 = \text{Mem}[X2+8]$
3. C: ADD R1, R3, R4 //  $R4 = R1 + R3$
4. D: ADDI R4, #16, R2 //  $R2 = R4 + 16$
5. E: SD 12(X2), R2 //  $\text{Mem}[X2+12] = R2$

#### (1)

基于 5 级流水线架构，在以下表格中完成上述指令代码的时序表。假设：

- 没有引入前馈（Forwarding）机制
- 寄存器文件支持半周期读写：前半周期写入，后半周期可读出新值

- 检测到数据冒险时自动插入流水线气泡 (stall)

指令	1	2	3	4	5	6	7	8	9	10	11	12
A: LD	IF	ID	EX	MEM	WB							
B: LD		IF	ID	EX	MEM	WB						
C: ADD			IF									
D: ADDI												
E: SD												

(2)

在 5 级流水线中引入检测与前馈 (Detect and Forwarding) 机制:

- LD 指令可以从 MEM 阶段将待写入寄存器的结果前馈回 ID 阶段
- 其他指令可以从 EX 阶段将结果前馈回 ID 阶段
- 重新画出时序表, 展示前馈机制如何减少流水线停顿

## 2、乱序执行微架构分析 (30 分)

假设我们有一个乱序执行处理器, 其微架构参数如下:

- 保留站 (Reservation Station, RS) 容量: 3 条指令
- 重排序缓冲区 (Reorder Buffer, ROB) 容量: 6 条指令

由于第一个 Load 指令需要很长时间停顿, 以下两个程序 A 和 B 都会因为等待 Load 完成而停滞。对于每个程序, 请指出加载完成之前, 最后一条将被放入 ROB 的指令, 并给出简要解释。

(注意: 指令会一直停留在保留站, 直到它被发射到计算单元。)

Program A	Program B
1: $R1 = \text{MEM}[R2+0]$	1: $R1 = \text{MEM}[R2+0]$
2: $R3 = R4 + 8$	2: $R3 = R1 + 8$
3: $R5 = R3 + R1$	3: $R5 = R4 + R4$
4: $R6 = R1 + 12$	4: $R6 = R3 + 12$
5: $R7 = R3 + R5$	5: $R7 = R3 + R5$
6: $R8 = R9 + R10$	6: $R8 = R9 + R10$
7: $R11 = R8 + 20$	7: $R11 = R5 + 20$
8: $R12 = R5 + R8$	8: $R12 = R9 + R5$

### 3、缓存微架构分析（40 分）

考虑以下访存序列：A, B, A, C, B, A, C。假设 A,B,C 是分别位于不同缓存块（different cache block）的地址。此外，假设这些地址是以均匀随机的方式生成的，并且使用 LRU（最近最少使用）替换机制。若两个缓存块同时为无效（invalid），则任何给定的访存块都有相等的机会放置在任意的 Cache Way 中。

请提供简要说明，并计算以下情况下最后一次访问（即访问 C）命中（Hit）的概率：

1. 缓存有 2 行，并且是 fully-associative 的。[10 分]
2. 缓存有 4 行，并且是 fully-associative 的。[10 分]
3. 缓存有 8 行，并且是 direct-mapped 的。[10 分]
4. 缓存有 4 行，并且是 2-way set associative 的。[10 分]