



人工智能的硬件基石：从物理器件到计算架构

2025年春季 - 作业3

1、寄存器重命名 (25分)

```
1. R1 = R2 / R3
2. R0 = MEM[R1]
3. R1 = R2 + R3
4. R0 = R0 + R1
5. R1 = MEM[R1]
6. R1 = R0 * R3
7. R4 = R3 + R1
8. MEM[R3] = R2 // a store instruction
```

将上述汇编代码进行寄存器重命名，以消除代码段中的所有伪依赖（false dependencies），同时保持相同的功能。要求如下：第一、不能重新排序、删除或添加指令（只能更改寄存器编号）；第二、确保代码段结束时，寄存器 R0-R4 中的值与上述代码段执行后完全相同。例如，如果上述代码段执行后 R0=5，那么在你重写的代码段结束后，R0 仍然必须是 5。任何内存MEM的写操作结果也必须保持一致；第三、可使用的寄存器为 R0-R11。

2、乱序执行微架构 (25分)

假设我们有一个乱序执行处理器，RS可容纳3条指令，ROB可容纳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，同时给出简要解释。（指令会一直停留在RS，直到它被发射到计算单元。）

Program A
R1=MEM[R2+0]
R2=R4+4
R3=R2+R7
R4=R1+6
R1=R2+R3

Program B
R1=MEM[R2+0]
R2=R1+4
R3=R4+R4
R4=R2+6
R1=R2+R3

$R6=R2+R6$	$R6=R9+R10$
$R7=R6+19$	$R7=R3+19$
$R8=R3+R6$	$R8=R9+R3$

3、缓存一致性 (50分)

假设有3个处理器核及其附属缓存采用MESI缓存一致性协议，所有3个处理器核都有一个2行直接映射的缓存（2-line direct-mapped cache），每行由16字节组成（16-byte cache line）。缓存开始时所有行都标记为无效。请完成以下几个表中内容，其中访存状态表1包括：

- Hit/Miss：处理器缓存是命中还是未命中（如果数据不在缓存中或数据状态无法避免总线操作，则为未命中Miss）
- 总线操作：处理器核执行的总线操作（如果存有请写明；若不存在请写不存在）。
- 操作后的状态：指明本地处理器缓存行在读/写后的状态。
- Miss类型：对于仅未命中的情况，指明未命中是由于强制（Compulsory）、容量（Capacity）、冲突（Conflict）或是缓存一致性（Coherence）。其中，缓存一致性未命中是指：如果不是由于另一个处理器核的操作，本该会命中的情况。

注意，总线操作可能包含多个子操作，子操作按顺序写明。所有缓存开始为空。

表1、访存状态表

处理器核序号	访存地址	读/写	Hit/Miss	总线操作	操作后状态	Miss类型
1	0x120	Read				
1	0x120	Write				
1	0x100	Read				
1	0x120	Write				
2	0x120	Read				
2	0x110	Write				
1	0x110	Read				
3	0x100	Write				
1	0x100	Read				
1	0x120	Read				
3	0x130	Read				
3	0x100	Read				
1	0x100	Write				

请填写处理器核缓存最终状态表：

	处理器核 1	
	Tag	State
Set 0		
Set 1		

	处理器核 2	
	Tag	State
Set 0		
Set 1		

	处理器核 3	
	Tag	State
Set 0		
Set 1		