

北京大学 - 智能硬件体系结构

2024年秋季 作业3

1、寄存器重命名 (40分)

考虑以下指令汇编代码：

```
start:  R1=MEM[R2+0]
        R2=MEM[R2+4]
        branch done if (R2==0)
        R3=R1+R3
        branch start
done: halt
```

如果重命名表 (Map Table或Register Alias Table) 中的所有条目架构寄存器 (Architectural Registers) 当前都指向了适当的物理寄存器 (Physical Registers) , 并且下一个可用的物理寄存器是 P1 (然后是 P2、P3, 依此类推) , 请展示上述程序每个汇编指令的执行将如何翻译成物理寄存器。假设 R2 被第 3 次更改后将分支到 “done: halt”, 且所需要的分支预测均是正确的。必要时, 需要多次“翻译”其中的一些指令。同时, 可以假设物理寄存器的数量是无限的, 答案应类似于如下示例。此外, 请用 1-2 句话简单描述上述程序的可能起到的作用 (可以考虑 Class 和 Pointer 功能等) 。

R1=MEM[P7+4]	// A	P1=MEM[R3+4]
R2=MEM[R3+8]	// B	P2=MEM[R3+8]
R1=R1*R2	// C	P3=P1*P2
MEM[R3+4]=R1	// D	MEM[R3+4]=P3
MEM[R3+8]=R1	// E	MEM[R3+8]=P3
R1=MEM[R3+12]	// F	P4=MEM[R3+12]
R2=MEM[R3+16]	// G	P5=MEM[R3+16]
R1=R1*R2	// H	P6=P4*P5
MEM[R3+12]=R1	// I	MEM[R3+12]=P6
MEM[R3+16]=R1	// J	MEM[R3+16]=P6

答案示例图 (非本题答案, 仅做示例)

2、乱序执行数据流 (40分)

考虑以下指令汇编代码：

```
A: LD 0(R1),F2      // F2 = Mem[R1+0]
B: MULTD F0,F2,F4    // F4 = F0*F2
C: ADDD F2,F4,F6     // F6 = F2+F4
D: LD 8(R1),F4      // F4 = Mem[R1+8]
E: MULTD F0,F4,F10   // F10 = F0*F4
F: ADDD F2,F2,F2     // F2 = F2+F2
G: ADDD F0,F2,F2     // F2 = F0+F2
H: SD 8(R1),F2      // Mem[R1+8] = F2
I: ADDDI F0, #16, F0 // F0 = F0+16
J: DIVDI F10, #3, F12 // F12 = F10+3
K: BNEQ F6,F12,A     // 如果F6不等于F12, jump to A
```

Functional units & latencies:	
1 Load	2
1 Store	1
1 Branch	1
1 FP Adder	2
1 FP Multiplier	5
1 FP Divider	10

假设该代码段运行在一个具有无限发射宽度、无限数量计算功能单元的乱序执行处理器上，该处理具有广播功能的通用数据总线 CDB，且实现了理想的寄存器重命名（即消除了所有 False Dependencies，假依赖性 WAW 和 WAR）。指令在其依赖的结果完成写回之前不能开始执行。请完成下表并填写 Cycle 序号，指令旁边的括号内数字是该指令在 EX 阶段的时长。表中顺序按指令依赖的所有数据Ready的时间顺序；若有并列情况，则按程序中的顺序排列，较早的指令排在较后的指令之上。

指令	所有数据Ready	EX开始	EX结束	CDB写回
A: LD 0(R1), F2 (2)	1	2	3	4
D: LD 8(R1), F4 (2)	1	2	3	4
I: ADDDI F0, #16, F0 (2)	1	2	3	4
。。。请完成该表。。。				

3、缓存基础知识 (20分)

- a) 假设缓存 Cache 容量是 4KB，缓存组织形式为 2-way associative write-back 模式，cache 的每个 block 长度为 32-byte，地址空间总长度为 40-bit. 请问该 cache 需要多少 bit 来作为 set index?
- b) 下列哪种缓存 cache 类型不需要 dirty bits (标记 cache 数据已被更改)：1) write-back; 2) write-allocate; 3) 2-way set associative; 4) write-through; 5) victim cache。请用一句话简要说明原因。