



## 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2

#### 1、五级流水线，stall 与 forwarding 前馈优化 (40 分)

- A: LD 0(X1),R1            R1 = Mem[X1+0]  
 B: LD 8(X1),R2            R2 = Mem[X1+8]  
 C: ADDD R1,R2,R3         R3 = R1+R2  
 D: ADDDI R3,#12,R1       R1 = R3+12  
 E: SD 0(X1),R1            Mem[X1+0]=R1

- (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（表格可以根据实际调整宽度）。假设没有引入前馈（Forwarding）机制；若 reg 读写地址相同，可以在前半个周期写入 reg，后半周期读出正确的值

Inst	1	2	...	...	...
LD	IF	ID	...	...	...
LD		IF	...	...	...
ADDD			...	...	...
ADDDI				...	...
SD					...

- (2) 在 5 级流水线中引入 Detect and Forwarding，对于 LD 指令可以从 Mem 阶段将待写入 reg 结果 forward 回 ID，其他指令可以从 EX 阶段 forward 回 ID，重新画出时序表

#### 2、数据依赖图与寄存器重命名 (40 分)

- A: LD 0(X1), R1            R1 = Mem[X1+0]  
 B: LD 8(X1),R2            R2 = Mem[X1+8]  
 C: ADDD R1,R2,R3         R3 = R1+R2  
 D: MULTD R1,R2,R4        R4 = R1\*R2  
 E: ADDD R3,R4,R1         R1 = R3+R4  
 F: MULTD R3,R4,R2        R2 = R3\*R4

**G: SUBD R1,R2,R1      R1 = R1-R2**

**H: SD 0(X2),R1      Mem[X2+0]=R1**

- (1) 给出代码运行结束后，Mem[X2]中存储的值
- (2) 画出上述代码的数据依赖关系图（要包括 RAW，WAW，WAR 并区别标注）
- (3) 请根据课上所学的寄存器重命名的方法，对上述代码中的 WAW 和 WAR 进行消除，写出改后代码，并重画数据依赖关系图

### 3、Winograd 卷积（20 分）

- (1) 基于二维 Winograd 卷积方法，计算下面卷积，要求列出具体计算过程

input				kernal		
1	2	3	4	3	2	1
5	6	7	8	2	3	2
9	10	11	12	1	2	3
13	14	15	16			

- (2) 计算用传统卷积计算上述输入与卷积核所需要的乘法与加法次数，同时计算使用 Winograd 卷积方法计算所需要的乘法与加法次数（计算 Winograd 加法时，input-input 之间、kernal-kernal 之间的加法可以忽略，因为这些操作可以通过预计算得到，只需考虑 input 值-kernal 值之间的加法即可）作对比