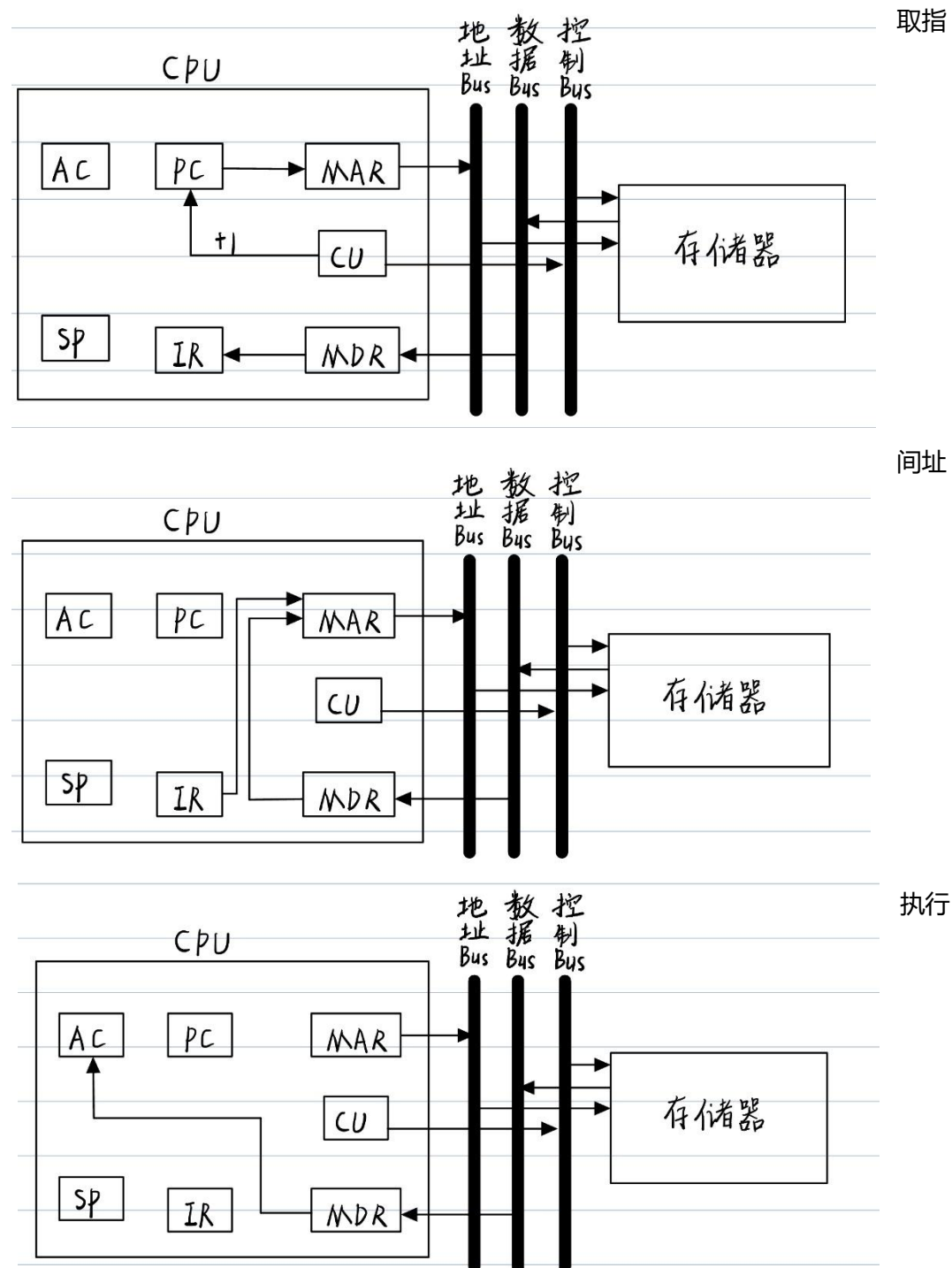


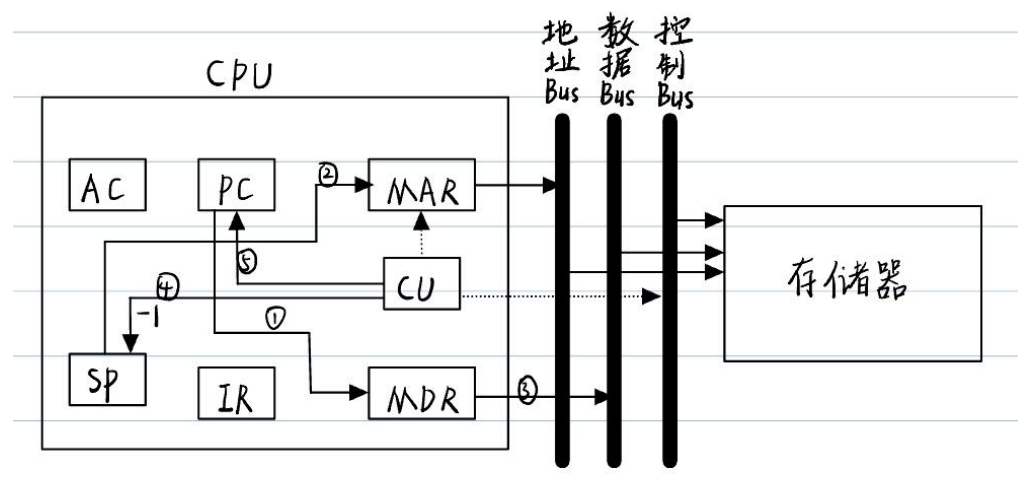
## 8.4

CPU 内有这些部件：PC IR SP AC MAR MDR CU

(1) 画出完成间接寻址的取数指令 “LDA@X” (将主存某地址单元的内容取至 AC 中) 的数据流 (从取指令开始)。



(2) 画出中断周期的数据流。



① PC → MDR

② SP → MAR

③ MDR → Mem[SP]

④ SP-1 → SP

⑤ 中断程序地址 → PC

## 8.8

什么是指令流水？画出指令二级流水和四级流水的示意图，它们中哪一个更能提高处理器速度，为什么？

(1) 指令流水：利用了完成一条指令的不同阶段其操作过程 and 使用的部件不同这一特点，在同一周期内同时处理多条指令的不同阶段，从而大大提高执行效率和部件利用率。

(2) 二级流水：



四级流水:

1	IF	ID	EX	WB				
2		IF	ID	EX	WB			
3			IF	ID	EX	WB		
4				IF	ID	EX	WB	

四级流水更能提高处理器速度, 原因是:

1. 各个阶段的用时更加接近, 有效减少了部件空闲和等待的时间, 提高了效率。
2. 把指令周期划分得更细, 提高了流水级, 使更多的指令在同一时间内执行, 单位时间内完成的指令数量更多, 提高了吞吐率。

## 8.11

今有四级流水线, 分别完成取指(IF)、译码并取数(ID)、执行(EX)、写结果(WR)4个步骤。假设完成各步操作的时间依次为 90ns、90ns、60ns、45ns

- (1) 流水线的时钟周期应取何值?
- (2) 若相邻的指令发生数据相关, 那么第 2 条指令安排推迟多少时间才能不发生错误?
- (3) 若相邻两指令发生数据相关, 为了不推迟第 2 条指令的执行, 可采取什么措施?

(1) (若不考虑其他因素) 流水线的时钟周期应取各步操作的最大时间, 即 90ns

(2) 要等待前一条指令的 EX 和 WR 阶段完成, 即至少等待两个时钟周期, 即 180ns

(3) 可采取旁路技术, 设置直接传送数据的通路。不必等待前面指令的执行结果送回到寄存器后, 再从寄存器中取出该结果, 作为下一条指令的操作数, 而是直接将执行结果送到其他指令所需要的地方。

## 8.12

在 5 个功能段的指令流水线中, 假设每段的执行时间分别是 10ns、8ns、10ns、10ns、7ns。对于完 12 条指令的流水线而言, 其加速比为多少? 该流水线的实际吞吐率为多少?

流水线指令周期应为 10ns。流水级  $m=5$ 。指令数  $n=12$

非流水线用时  $t_1=12 \times (10+8+10+10+7)=540\text{ns}$

流水线用时  $t_2=(12-1+5) \times 10=160\text{ns}$

加速比  $S_p = t_1/t_2 = 540/160 = 3.375$

实际吞吐率  $T_p = 12/t_2 = 0.075 \text{ 条/ns}$