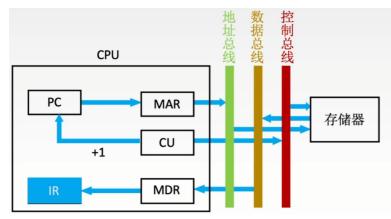
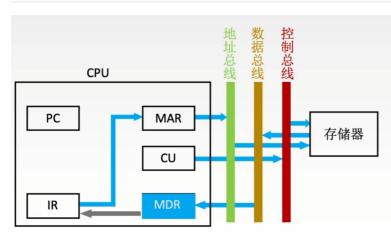
指令周期的数据流

取指周期



- 1. 当前指令地址送至存储器地址寄存器, 记做: (PC) → MAR
- 2. CU发出控制信号,经控制总线传到主存,这里是读信号,记做: $1 \rightarrow R$
- 3. 将MAR所指主存中的内容经数据总线 送入MDR,记做: M(MAR) → MDR
- **4.** 将MDR中的内容(此时是指令)送入IR, 记做: (MDR) → IR
- 5. CU发出控制信号,形成下一条指令地址,记做: (PC)+1 → PC

间址周期

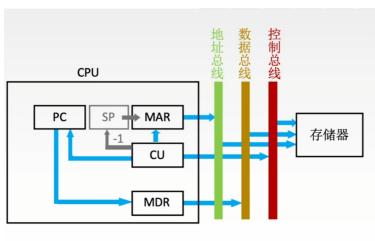


- 1. 将指令的地址码送入MAR,
- 记做: Ad(IR) → MAR 或Ad(MDR) → MAR
- 2. CU发出控制信号,启动主存做**读操作**,
- 记做: 1→R
- 3. 将MAR所指主存中的内容经数据总线 送入MDR,记做: M(MAR) → MDR
- 4. 将有效地址送至指令的地址码字段, 记做: (MDR)→ Ad(IR)

执行周期

执行周期的任务是根据IR中的指令字的操作码和操作数通过ALU操作产生执行结果。不同指令的执行周期操作不同,因此没有统一的数据流向。

中断周期



中断:暂停当前任务去完成其他任务。 为了能够恢复当前任务,需要保存断点。 一般使用堆栈来保存断点,这里用SP表示 栈顶地址,假设SP指向栈顶元素,进栈操 作是先修改指针,后存入数据。

- 1. CU控制将SP減1,修改后的地址送入MAR 记做: (SP)-1 \rightarrow SP,(SP) \rightarrow MAR 本质上是将断点存入某个存储单元,假设其地址为a,故可记做: a \rightarrow MAR
- 2. CU发出控制信号,启动主存做**写操作**,记做: 1 → W
- 3. 将断点(PC内容) 送入MDR, 记做: (PC) → MDR
- 4. CU控制将中断服务程序的入口地址 (由向量地址形成部件产生)送入PC, 记做:向量地址→ PC