2022/5/13

方框图问题

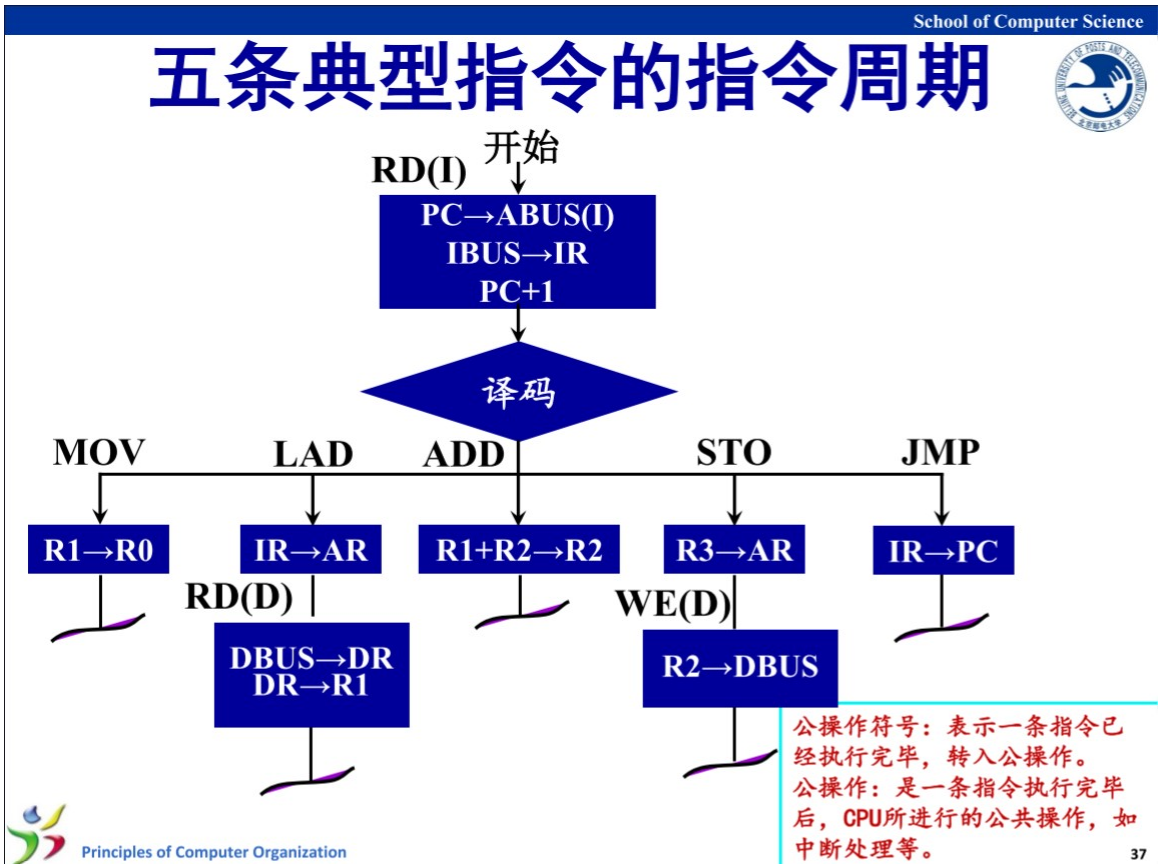
1 基本概念

> “在进行中央处理器设计时，可用方框图语言来描述一条指令的指令周期，其中，方框代表一个CPU周期，方框中的内容表示数据通路的操作或某种控制操作，菱形通常用来表示某种判别或测试，在时间上依附于紧接它的前面一个方框的CPU周期，不单独占用一个CPU周期。”
>
> ——易秋萍PPT《第5章 中央处理器1》

分析：

- “可用方框图语言来描述一条指令的指令周期”：并不是说一条指令必须使用一个方框描述，而是说：对于一条指令的整个执行过程而言，可以用若干个方框构成的方框语言来描述
- “其中，方框代表一个CPU周期”：一个方框代表一个CPU周期，一个CPU周期就是一个机器周期，由此可知，一个指令周期由若干个机器周期组成，也就是一条指令应当拆分成多个方框。其中，一个CPU周期常定义为“从存储器中读取一个数据的时间”
- 关于机器周期：是完成一个基本操作的时间。另，指令周期和机器周期可见于[CPU中的指令周期、CPU周期和时钟周期 - 知乎]

2 分析



按照之前的分析，应当将RD(I)方框拆开（从而写成多个方框，分别对应单个CPU周期），但是这里没有拆开，原因是此处将指令分成了四个基本阶段（见补充材料3.1和3.2）：

- 取指周期：取指令并PC+1
- 间指周期：如果指令应当间接寻址，则需要此操作，否则不需要，当然如果是多重间接取值则有多个该周期

- 执行周期：执行操作
- 中断周期：检测是否有I/O请求（在上图中就是进入了“公操作”阶段）

每个方框对应一个周期，对于菱形而言，在**基本概念**一节中已经说明了“菱形通常用来表示某种判别或测试，在时间上依附于紧接它的前面一个方框的CPU周期，不单独占用一个CPU周期”，因此属于取值周期的任务。

似乎，上图的画法并不是所谓的“指令周期流程图”，原因有二：

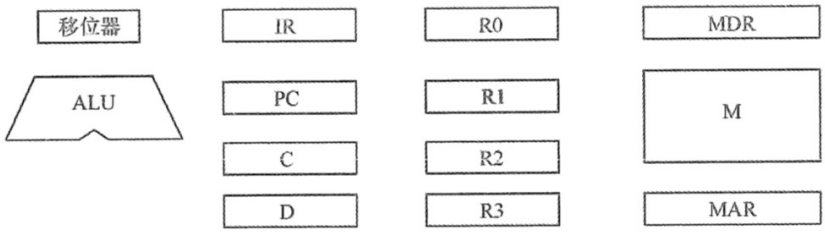
2.1 原因之一

易秋萍PPT《第5章 中央处理器1》P38问“画出其指令周期流程图”，并在P40中给出了“单语句单方框的画法”。

2.2 原因之二

我找到了如下题目及其答案：

6. 某机主要功能部件如下图所示，其中 M 为主存，MDR 为主存数据寄存器，MAR 为主存地址寄存器，IR 为指令寄存器，PC 为程序计数器（并假设当前指令地址在 PC 中），R0 ~ R3 为通用寄存器，C、D 为暂存器。

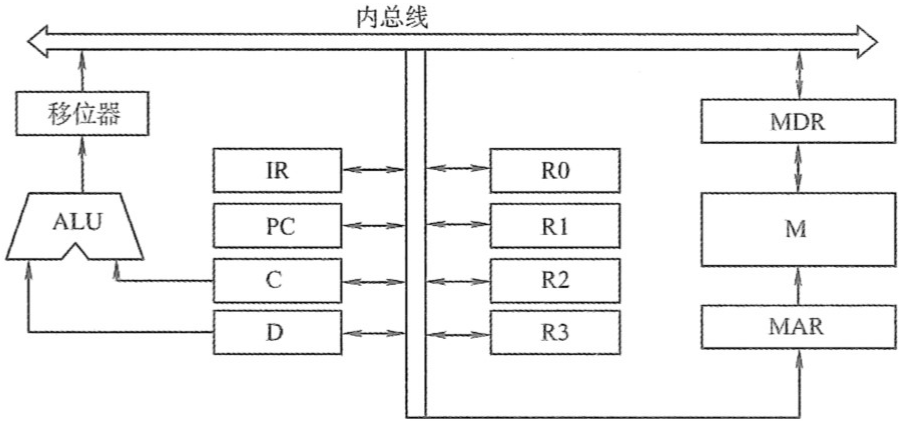


1）请补充各部件之间的主要连接线（总线自己画），并注明数据流动方向。

2）画出“ADD (R1), (R2) +”指令周期流程图。该指令的含义是进行求和运算，源操作数地址在 R1 中，目标操作数寻址方式为自增型寄存器间接寻址方式（先取地址后加 1），并将相加结果写回 R2 寄存器。

6．解答：

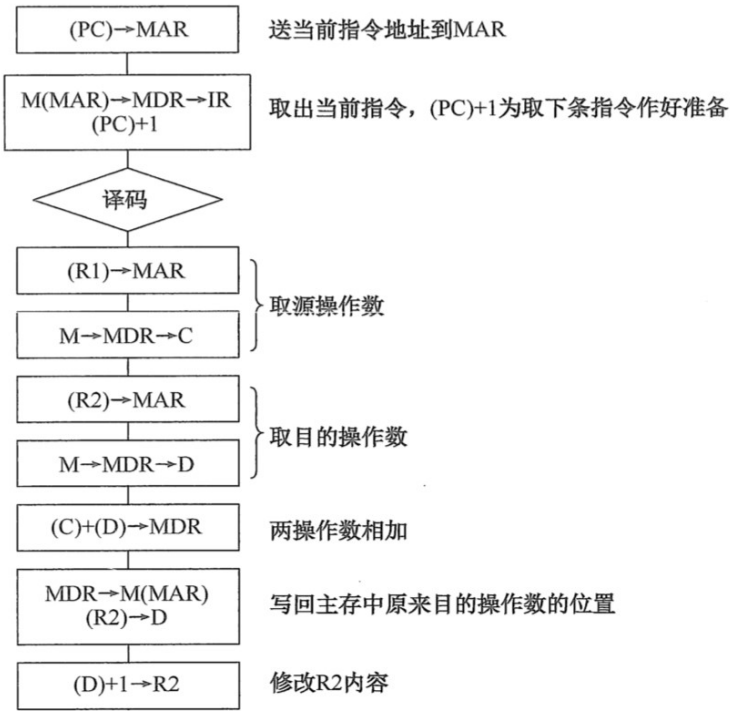
1）各功能部件的连接关系，以及数据通路如下图所示。



（2）分析过程如下：

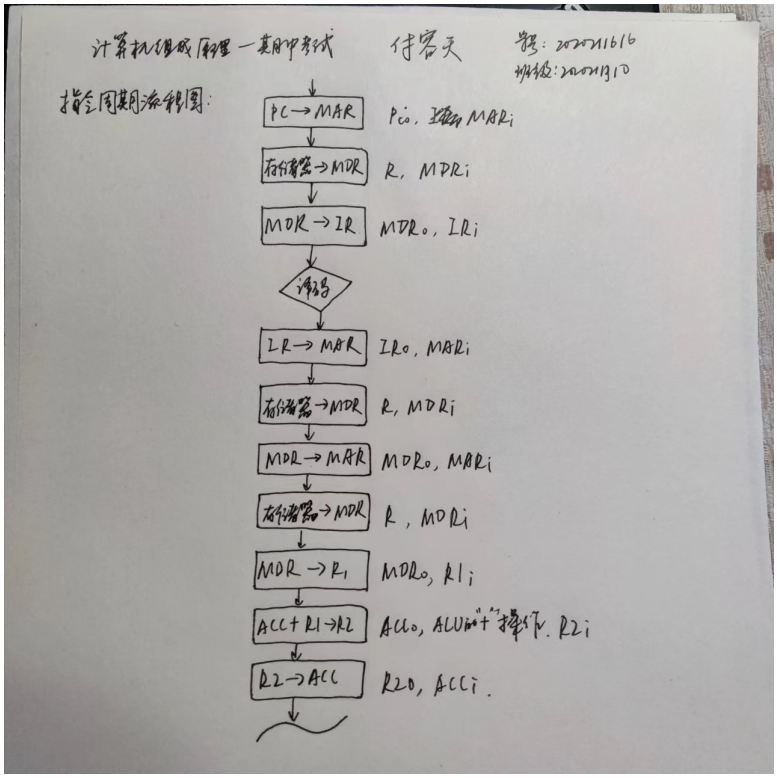
- 取指令地址送到 IR 并译码。
- 取源操作数和目的操作数。
- 将源操作数和目的操作数相加送到 MAR，随之送到以前目的操作数所在内存的地址。
- 将寄存器 R2 的内容加 1。

取指周期流程如下图所示。



（网络资料显示，上图中PC两侧有括号，表示值的流动，但是画法中可以没有括号，比如易秋萍PPT上的画法，这种没有括号的画法表示静态描述的数据路线而不是动态进行的数据流动）可以看出，似乎所谓的“指令周期流程图”总是画成“单方框单指令”的形式。

2.3 所以我把答案改为了：



3 补充材料

3.1 唐朔飞，《计算机组成原理（第二版）》，P375-378

9.1 微操作命令的分析

控制单元具有发出各种微操作命令(即控制信号)序列的功能。

概括地说,计算机的功能就是执行程序。在执行程序的过程中,控制单元要发出各种微操作命令,而且不同的指令对应不同的命令。进一步分析发现,完成不同指令的过程中,有些操作是相同或相似的,如取指令、取操作数地址(当间接寻址时)以及进入中断周期由中断隐指令完成的一系列操作。为更清晰起见,下面按指令周期的 4 个阶段进一步分析其对应的微操作命令。

9.1.1 取指周期

为了便于讨论,假设 CPU 内有 4 个寄存器,如图 8.10 所示。MAR 与地址总线相连,存放欲访问的存储单元地址;MDR 与数据总线相连,存放欲写入存储器的信息或最近从存储器中读出的信息;PC 存放现行指令的地址,有计数功能;IR 存放现行指令。取指令的过程可归纳为以下几个操作。

① 现行指令地址送至存储器地址寄存器,记作 PC→MAR。

② 向主存发送读命令,启动主存作读操作,记作 1→R。

③ 将 MAR(通过地址总线)所指的主存单元中的内容(指令)经数据总线读至 MDR 内,记作 M(MAR)→MDR。

④ 将 MDR 的内容送至 IR,记作 MDR→IR。

⑤ 指令的操作码送至 CU 译码,记作 OP(PC)→CU。

⑥ 形成下一条指令的地址,记作(PC) +1→PC。

9.1.2 间址周期

间址周期完成取操作数有效地址的任务,具体操作如下。

① 将指令的地址码部分(形式地址)送至存储器地址寄存器,记作 Ad(IR)→MAR。

② 向主存发送读命令,启动主存作读操作,记作 1→R。

③ 将 MAR (通过地址总线)所指的主存单元中的内容(有效地址)经数据总线读至 MDR 内,记作 M(MAR)→MDR。

④ 将有效地址送至指令寄存器的地址字段,记作 MDR→Ad(IR)。此操作在有些机器中可省略。

9.1.3 执行周期

不同指令执行周期的微操作是不同的,下面分别讨论非访存指令、访存指令和转移类指令的微操作。

1. 非访存指令

这类指令在执行周期不访问存储器。

(1) 清除累加器指令 CLA

该指令在执行阶段只完成清除累加器操作,记作 0→ACC。

(2) 累加器取反指令 COM

该指令在执行阶段只完成累加器内容取反,结果送累加器的操作,记作$\overline{ACC}$→ACC。

(3) 算术右移一位指令 SHR

该指令在执行阶段只完成累加器内容算术右移一位的操作,记作 L(ACC)→R(ACC),ACC_0→ACC_0(ACC 的符号位不变)。

(4) 循环左移一位指令 CSL

(4) 循环左移一位指令 CSL

该指令在执行阶段只完成累加器内容循环左移一位的操作，记作 R(ACO)→L(ACC)，$ACC_0 \rightarrow ACC_n$(或 $\rho^{-1}(ACC)$)。

(5) 停机指令 STP

计算机中有一个运行标志触发器 G，当 G=1 时，表示机器运行；当 G=0 时，表示停机。STP 指令在执行阶段只需将运行标志触发器置"0"，记作 0→G。

2. 访存指令

这类指令在执行阶段都需要访问存储器。为简单起见，这里只考虑直接寻址的情况，不考虑其他寻址方式。

(1) 加法指令 ADD X

该指令在执行阶段需要完成累加器内容与对应于主存 X 地址单元的内容相加，结果送累加器的操作，具体如下：

① 将指令的地址码部分送至存储器地址寄存器，记作 Ad(IR)→MAR。

② 向主存发读命令，启动主存作读操作，记作 1→R。

③ 将 MAR(通过地址总线)所指的主存单元中的内容(操作数)经数据总线读至 MDR 内，记作 M(MAR)→MDR。

④ 给 ALU 发送加命令，将 ACC 的内容和 MDR 的内容相加，结果存于 ACC，记作 (ACC) + (MDR) →ACC。

当然，也有的加法指令指定两个寄存器的内容相加，如"ADD AX,BX"，该指令在执行阶段无须访存，只需完成 (AX) + (BX) →AX 的操作。

(2) 存数指令 STA X

该指令在执行阶段需将累加器 ACC 的内容存于主存的 X 地址单元中，具体操作如下。

① 将指令的地址码部分送至存储器地址寄存器，记作 Ad(IR)→MAR。

② 向主存发写命令，启动主存作写操作，记作 1→W。

③ 将累加器内容送至 MDR，记作 ACC→MDR。

③ 将累加器内容送至 MDR，记作 ACC→MDR。

④ 将 MDR 的内容(通过数据总线)写入到 MAR(通过地址总线)所指的主存单元中，记作 MDR→M(MAR)。

(3) 取数指令 LDA X

该指令在执行阶段需将主存 X 地址单元的内容取至累加器 ACC 中，具体操作如下。

① 将指令的地址码部分送至存储器地址寄存器，记作 Ad(IR)→MAR。

② 向主存发读命令，启动主存作读操作，记作 1→R。

③ 将 MAR(通过地址总线)所指的主存单元中的内容(操作数)经数据总线读至 MDR 内，记作 M(MAR)→MDR。

④ 将 MDR 的内容送至 ACC，记作 MDR→ACC。

3. 转移类指令

这类指令在执行阶段也不访问存储器。

(1) 无条件转移指令 JMP X

(1) 无条件转移指令 JMP X

该指令在执行阶段完成将指令的地址码部分 X 送至 PC 的操作，记作 Ad(IR)→PC。

(2) 条件转移(负则转)指令 BAN X

该指令根据上一条指令运行的结果决定下一条指令的地址，若结果为负(累加器最高位为 1，即 $A_0=1$)，则指令的地址码送至 PC，否则程序按原顺序执行。由于在取指阶段已完成了 (PC)+1→PC，所以当累加器结果不为负(即 $A_0=0$)时，就按取指阶段形成的 PC 执行，记作 $A_0 \cdot Ad(IR) + \overline{A}_0 \cdot (PC) \rightarrow PC$。

由此可见，不同指令在执行阶段所完成的操作是不同的。如果将访存指令分为直接访存和间接访存两种，则上述三类指令的指令周期如图 9.1 所示。

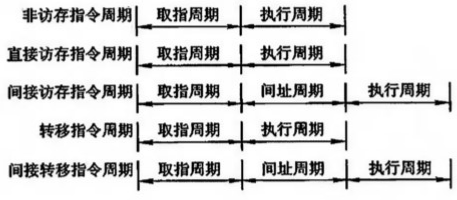


图 9.1 三类指令的指令周期

9.1.4 中断周期

9.1.4 中断周期

在执行周期结束时刻，CPU 要查询是否有请求中断的事件发生，如果有则进入中断周期。由 8.4.4 节可知，在中断周期，由中断隐指令自动完成保护断点、寻找中断服务程序入口地址以及硬件关中断的操作。假设程序断点存至主存的 0 地址单元，且采用硬件向量法寻找入口地址，则在中断周期需完成如下操作。

① 将特定地址“0”送至存储器地址寄存器，记作 0→MAR。

② 向主存发写命令，启动存储器作写操作，记作 1→W。

③ 将 PC 的内容（程序断点）送至 MDR，记作 PC→MDR。

④ 将 MDR 的内容（程序断点）通过数据总线写入到 MAR（通过地址总线）所指示的主存单元（0 地址单元）中，记作 MDR→M(MAR)。

元（0 地址单元）中，记作 MDR→M(MAR)。

⑤ 将向量地址形成部件的输出送至 PC，记作向量地址→PC，为下一条指令的取指周期作准备。

⑥ 关中断，将允许中断触发器清零，记作 0→EINT（该操作可直接由硬件线路完成，参见图 8.30）。

如果程序断点存入堆栈，而且进栈操作是先修改栈指针，后存入数据（参见图 7.18），只需将上述① 改为(SP) - 1→SP，且 SP→MAR。

上述所有操作都是在控制单元发出的控制信号（即微操作命令）控制下完成的。

3.2 网络资料