第 9 章 控制单元的功能

9.1 微操作命令的分析

完成一条指令分 4 个工作周期：取指周期、间址周期、执行周期、中断周期。

一、取指周期

要想把指令从内存单元取出，就要执行下面的操作：PC（指令地址）->MAR->地址线，最后送到存储器。控制单元向存储器发出读命令，读出的数据通过数据总线送到 MDR 中，后送到 IR 当中。然后 IR 送到 CU 进行译码知道要做什么指令。然后对 PC 进行加 1。

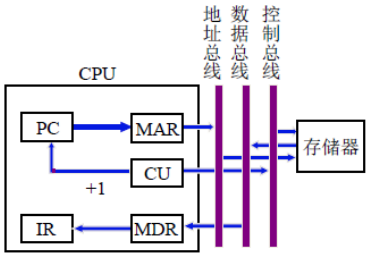

图 1.1 取指周期

二、间址周期

首先将指令的形式地址（IR 当中的地址码部分）送到 MAR，再通过 MAR 送到再送到存储器的地址总线上，要实现这个操作，控制器要发出 IR 的地址码部分送到 MAR 的控制信号，然后 CU 向存储器发出读操作命令，存储器接收到命令和地址以后，读出指令的真实地址，然后传输给 MDR。被取出的地址进而送到 IR 寄存器的地址码部分，这个时候地址码就是操作数的真实地址，就结束了。

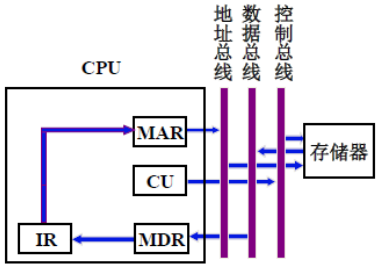

图 1.2 间址周期

三、执行周期

1.非访存指令

CLA	清 A	0->ACC
COM	取反	ACC'->ACC
SHR	算术右移	L(ACC)-> R(ACC), ACC0->ACC0
CSL	循环左移	R(ACC)->L(ACC), ACC0->ACCn
STP	停机指令	0->G

2.访存指令

（1）加法指令

操作命令	微操作步骤
ADD X	Ad(IR)->MAR

	1->R M(MAR)->MDR (ACC)+(MDR)->ACC

（2）存数指令

操作命令	微操作步骤
STA X	Ad(IR)->MAR 1->W ACC->MDR MDR->M(MAR)

（3）取数指令

操作命令	微操作步骤
LDA X	Ad (IR)->MAR l->R M(MAR)->MDR MDR->ACC

3.转移指令

（1）无条件转移指令

操作命令	微操作步骤
JMP X	Ad (IR)->PC

（2）条件转移指令（只举了一个例子）

操作命令	微操作步骤
BAN X（负则转）	$A_0 \cdot Ad(IR) + \overline{A}_0(PC) \rightarrow PC$

4.三类指令的指令周期

如图 1.3 所示。

图 1.3 三类指令的指令周期

四、中断周期

保存断点的方法（以此为例分析）：

程序断点存入**“0”地址**	程序断点**进栈**
“0”地址->MAR 1->W（控制器向存储单元发出写命令） PC->MDR（断点地址由 PC 保存到 MDR 中） MDR->M(MAR)（MDR 写 MAR 指向地址）	**(SP-1)->MAR** 1->W（控制器向存储单元发出写命令） PC->MDR（断点地址由 PC 保存到 MDR 中） MDR->M(MAR)（MDR 写 MAR 指向地址）
中断识别程序入口地址 M->PC（**硬件向量法**和**软件查找法**，这里以前者为例）	
0->EINT（关中断）	0->EINT（关中断）

9.2 控制单元的功能

一、控制单元的外特性

如图 2.1 所示。

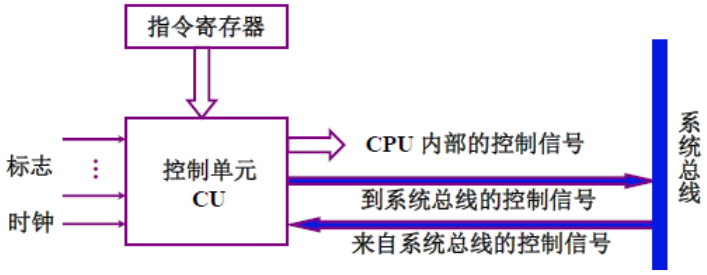

图 2.1 控制单元的外特性

1.输入信号

（1）时钟

完成每个操作都需要占用一定的时间，各个操作是由先后顺序的。为了能够使控制单元按一定的先后顺序、一定的节奏发出各个控制信号，**CU 必须受时钟控制**。一个时钟脉冲发一个操作命令或一组需同时执行的操作命令。

（2）指令寄存器

OP(IR)->CU：这个微指令代表一个译码操作。

保存了要执行的那条指令。现行指令的操作码决定了不同指令在执行周期所需完成的不同操作，故指令的**操作码**字段是控制单元的输入信号，它与时钟配合可产生不同的控制信号。

（3）标志

CU 受标志控制

（4）外来信号

如 INTR（中断请求）、HRQ（总线请求）等。

2.输出信号

（1）CPU 内部的各种控制信号

（2）送至控制总线的信号

例如访存控制信号、访 IO/存储器的控制信号、读命令、写命令、中断响应信号、总线响应信号等。

二、控制信号举例

1.不采用 CPU 内部总线的方式

（1）取指周期

C0~C4 均由 CU 产生。

①控制信号 C0 有效，打开 PC 送往 MAR 的控制门；

②控制信号 C1 有效，打开 MAR 送往地址总线的输出门；

③通过控制总线向主存发读命令；

④C2 有效，打开数据总线送至 MDR 的输入门；

⑤C3 有效，打开 MDR 和 IR 之间的控制们，至此指令送至 IR；

⑥C4 有效，打开指令操作码送至 CU 的输出门，CU 在操作码和时钟的控制下，可产生各种控制信号；

⑦使 PC 内容加 1。

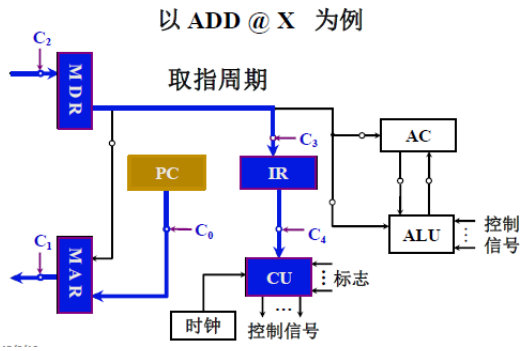

图 2.2 不采用 CPU 内部总线的方式（取指周期）

（2）间址周期

①C5 有效，打开 MDR 和 MAR 之间的控制门（假设间址周期的时候，直接地址是从 MDR 送入 MAR 的），将指令的形式地址送入 MAR；

②C1 有效，打开 MAR 送往地址总线的输出门；

③通过控制总线向主存发读命令；

④C2 有效，打开数据总线送至 MDR 的输入门，至此，有效地址存入 MDR；

物 C3 有效，打开 MDR 和 IR 之间的控制门，将有效地址送至 IR 的地址码字段。

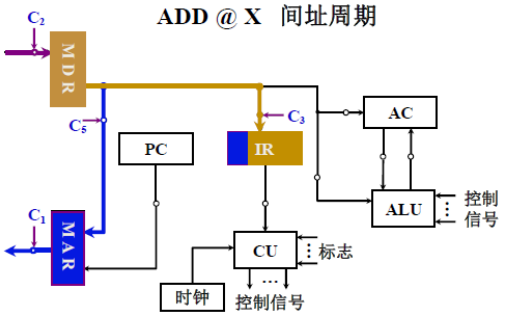

图 2.3 不采用 CPU 内部总线的方式（间址周期）

（3）执行周期

①C5 有效，打开 MDR 和 MAR 之间的控制门，将有效地址送至 MAR；

②C1 有效，打开 MAR 送往地址总线的输出门；

③通过控制总线向主存发读命令；

④C2 有效，打开数据总线送至 MDR 的输入门，至此操作数存入 MDR；

⑤C6/C7 同时有效，打开 AC 和 MDR 通往 ALU 的控制门；

⑥通过 CPU 内部控制总线对 ALU 发 ADD 加控制信号，完成 AC 的内容和 MDR 的内容相加；

⑦C8 有效，打开 ALU 通往 AC 的控制门，至此将求和结果存入 AC。

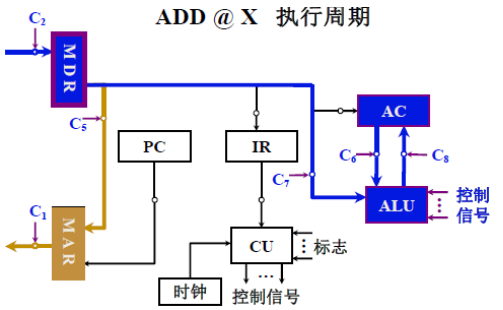

图 2.4 不采用 CPU 内部总线的方式（执行周期）

2.采用 CPU 内部总线方式

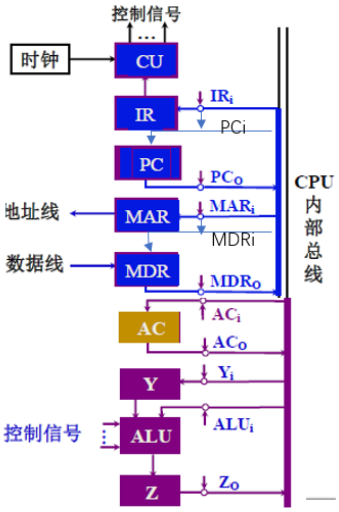

图 2.5 采用 CPU 内部总线方式

图中每一个小圈处都有一个控制信号，它控制寄存器到总线或总线到寄存器的传送。例如 IRi 表示控制从内部总线到指令寄存器的输入控制门；PCo 表示控制从程序计数器到内部总线的输出控制门。下标为 i 表示输入控制，下标为 o 表示输出控制。与图 2.2~2.4 相比，多了两个寄存器 Y 和 Z，这是由于 ALU 是一个组合逻辑电路，在其运算过程中必须保持两个输入端不变，其中一个输入可以从 Y 寄存器中获得，另一个输入可以从内部总线上获得。当 CPU 内有多个通用寄存器时，由于设置了寄存器 Y，可实现任意两个寄存器之间的算术/逻辑运算。此外，ALU 的输出不能直接与内部总线相连，因为其输出又会通过总线反馈到 ALU 的输入，影响运算的正确性，故用寄存器 Z 暂存运算结果，再根据需要送至指定的目标。

下面依然以上面的命令分析控制单元发出的控制信号。

（1）取指周期

①PCo 和 MARi 有效，完成 PC 经内部总线送至 MAR 的操作，即 PC->MAR；

②通过控制总线向主存发出读命令，即 1->R；

③存储器通过数据总线将 MAR 所指单元的内容（指令）送至 MDR；

④MDRo 和 IRi 有效，将 MDR 的内容送至 IRi，即 MDR->IR，至此，指令送至 IR，其操作码字段开始控制 CU；

⑤使 PC 内容加 1。

（2）间址周期

①MDRo 和 MARi 有效，将指令的形式地址经内部总线送至 MAR，即 MDR->MAR；

②通过控制总线向主存发出读命令，即 1->R；

③存储器通过数据总线将 MAR 所指单元的内容（有效地址）送至 MDR；

④MDRohe IRi 有效，将 MDR 中的有效地址送至 IR 的地址码字段，即 MDR->Ad(IR)。

（3）执行周期

①MDRo 和 MARi 有效，将有效地址经内部总线送至 MAR，即 MDR->MAR；

②通过控制总线向主存发出读命令，即 1->R；

③存储器通过数据总线将 MAR 所指单元的内容（操作数）送至 MDR；

④MDRo 和 Yi 有效，将操作数送至 Y，即 MDR->Y；

⑤ACo 和 ALUi 有效，同时 CU 向 ALU 发“ADD”加控制信号，使 AC 的内容和 Y 的内容相加（Y 的内容传送至 ALU 不必经过总线），结果送寄存器 Z，即(AC)+(Y)->Z；

⑥Zo 和 ACi 有效，将结果存入 AC，即 Z->AC。

三、多级时序系统

1.机器周期

机器周期指**所有指令执行过程中的一个基准时间**。机器周期取决于**指令的功能**和**器件速度**。确定机器周期时，通常要分析**机器指令的执行步骤**及**每一步骤所需的时间**。

基准时间的确定有两种方式：

①以完成**最复杂**指令功能的时间为准；

②以**访问一次存储器**（一般情况下最复杂的指令是这个指令）的时间为基准。

若指令字长=存储字长，则**取指周期=机器周期**。

2.时钟周期（节拍、状态）

一个机器周期内科完成多个微操作，每个微操作需一定的时间。将一个机器周期分成若干个时间相等的时间段（**节拍、状态、时钟周期**）。

时钟周期是控制计算机操作的最小单位时间。用时钟周期控制产生一个或几个微操作（一般是并行执行）命令。

如图 2.6 所示，它反映了机器周期、时钟周期和节拍的关系，一个机器周期由 4 个节拍 T0、T1、T2、T3。

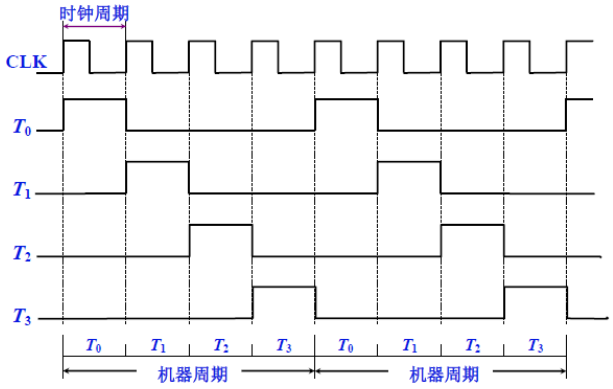

图 2.6 机器周期、时钟周期和节拍的关系

一个指令周期包含若干个机器周期，一个机器周期又包含若干个时钟周期（节拍），每个指令周期内的机器周期数**可以不相等**，每个机器周期里面的节拍数也**可以不相等**。

3.多级时序系统

机器周期、节拍（状态）组成多级时序系统。一般来说，CPU 的**主频越快**，机器的运行**速度也越快**。在机器周期所含**时钟周期数相同**的前提下，**两机平均指令执行速度之比=两机主频之比**。

机器速度不仅与**主频**有关，还与机器周期中所含**时钟周期**（主频的倒数）数以及**指令周期中所含的机器周期数**有关。

四、控制方式

产生不同微操作命令序列所用的时序控制方式

1.同步控制方式

任一微操作均由统一基准时标的时序信号控制。

（1）采用**定长**的机器周期：以**最长**的微操作**序列**和**最复杂**的微操作作为标准，机器周期内**节拍数相同**。显然这种方法对于较短指令来说会造成时间上的浪费。

（2）采用不定长的机器周期：机器周期内的**节拍数可以不等**。这种控制方式可以解决微操作**执行时间不统一**的问题。

（3）采用中央控制和局部控制相结合的方法：这种方案将机器的大部分指令安排在统

一的、较短的机器周期内完成，称为中央控制，而将少数操作复杂的指令中的某些操作采用局部控制方式来完成。

局部控制的节拍宽度与中央控制的节拍宽度一致；将局部控制节拍作为中央控制中及其街拍的延续，插入到中央控制的执行周期内，使机器以同样的节奏工作，保证了局部控制和中央控制的同步。如图 2.7 所示。

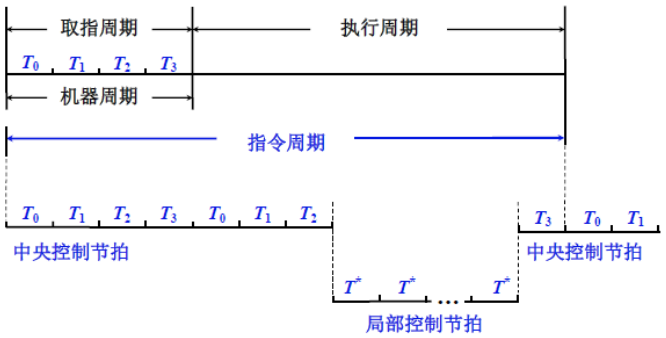

图 2.7 中央控制与局部控制结合的方法

2.异步控制方式

无基准时标信号，无固定的周期节拍和严格的时钟同步，采用**应答方式**。

3.联合控制方式

同步与异步相结合。对各种不同指令的微操作实行**大部分统一、小部分区别对待**的方法。

4.人工控制方式

（1）Reset 键；

（2）连续和单条指令执行转换开关；

（3）符合停机开关。