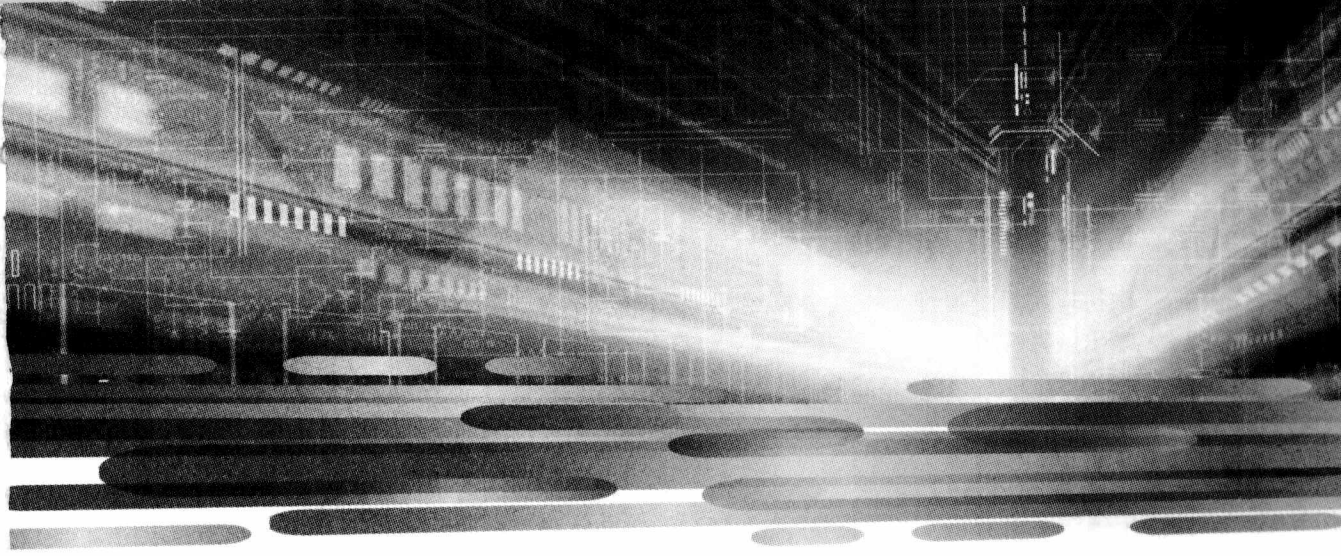
微机应用系统设计

——基于MCS-51单片机

冯介一　编著

湖南师范大学出版社



本书为湖南师范大学数学与计算机科学学院计算机科学技术课程教学团队
（项目经费号：830128-021）研究成果

微机应用系统设计
——基于MCS-51单片机

冯介一　编著

湖南师范大学出版社

图书在版编目(CIP)数据

微机应用系统设计——基于 MCS-51 单片机 / 冯介一编著. —长沙:湖南师范大学出版社, 2015.9

ISBN 978-7-5648-2265-1

Ⅰ.①微… Ⅱ.①冯… Ⅲ.①单片微型计算机—系统设计—高等学校—教材 Ⅳ.①TP368.1

中国版本图书馆 CIP 数据核字(2015)第 219902 号

微机应用系统设计——基于 MCS-51 单片机

冯介一 编著

责任编辑|廖小刚

责任校对|施 游

出版发行|湖南师范大学出版社

地址:长沙市岳麓山 邮编:410081

电话:0731.88873070 88873071

传真:0731.88872636

网址:www.hunnu.edu.cn/press

经 销|湖南省新华书店

印 刷|湖南雅嘉彩色印刷有限公司

开 本|787 mm×1092 mm 1/16

印 张|13

字 数|360 千字

版 次|2015 年 9 月第 1 版第 1 次印刷

书 号|ISBN 978-7-5648-2265-1

定 价|39.80 元

前　言

本书主要内容包含接口技术和单片机两大内容，第1章到第5章为接口技术部分，依次讲解微型计算机基本结构、总线、存储器、I/O接口和裸机的软件组织等内容。第6、7、8章讲解MCS－51单片机。第9章完整地讲解了一个微机应用系统开发实例。编写此书的目的是期待能给那些立志从事自动化系统设计和嵌入式系统设计的读者提供一个入门的阶梯。因此本书非常适合作为自动化、电气、电子信息类专业的相关课程的教材。

本书的特点：

1. 由抽象到具体来组织本书内容，全书主线突出，条理清晰。

开发技术很“具体”，但离“教室”较远。“教室”里的东西较抽象，但抽象的概念较为容易被读者接受。本书正是先让读者在整体上把握微机系统设计的方法之后，再具体落实单片机上来实施微机系统开发技术。CPU工作原理抽象成“CPU工作原理图”，计算机系统工作原理抽象成1.3.3节。全书以1.3.3节为核心展开，以“CPU工作原理图”贯穿全书主要内容。

2. 依工程开发规律组织本书内容，符合工程实际，也便于实践课与本课程的同步。

开发微机应用系统一般是先硬件设计，后软件设计调试等。本书内容正是依据这样的规律安排。这样安排的另一个好处是便于实践课与理论课的同步，不必等到理论学习结束或基本结束再开实践课。这种同步显然更利于读者学习和掌握本书内容。

3. 按微机系统开发规律详细讲解了一个开发实例，即第9章。该实例全面讲解了微机应用系统的开发过程与开发方法，为读者以后的开发实践提供了参照。

书中不足之处在所难免，恳请读者批评指正。

编　者

目　录

上篇　基础篇

下篇 实践篇

上篇　基础篇

第 1 章　微型计算机基本结构

目前的各种微型计算机(Microcomputer)系统,从体系结构来看,采用的基本上是计算机系统的经典结构——冯·诺依曼结构。这种结构的特点是:

(1)计算机由运算器、控制器、存储器、输入设备和输出设备 5 大部分组成。

(2)数据和程序以二进制代码形式不加区别地存放在存储器中,存放位置由地址决定,地址码也为二进制形式。

(3)控制器根据存放在存储器中的指令序列(即程序)工作,并由一个程序计数器(即指令地址计数器)控制指令的执行。控制器具有判断能力,能根据计算结果选择不同的动作流程。

由此可见,任何一个微机系统都是由硬件和软件(程序)两大部分组成的,其中硬件又由运算器、控制器、存储器、输入设备和输出设备 5 部分组成;软件即指令的有序集合。

1.1　微型计算机系统硬件结构

微型计算机系统由中央处理器 CPU(Center - Processing Unit)、存储器、输入/输出电路(I/O 接口)和形形色色的外部设备构成。芯片之间通过总线(Bus)连接。整个系统如图1 - 1所示。

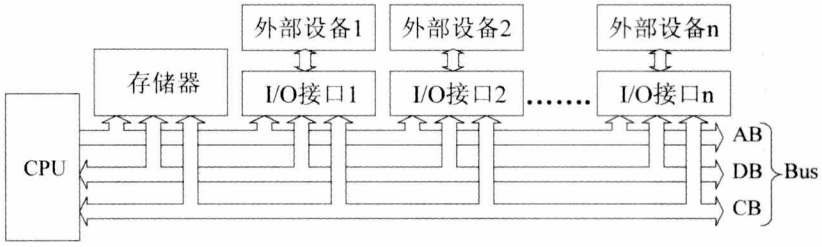


图 1 - 1　微型计算机系统图

1. CPU

CPU 是微型计算机系统的核心。不同型号的微机,其性能差别首先在于 CPU 性能的不同,而 CPU 性能又与它的内部结构、硬件配置有关。每种 CPU 有自己的指令系统,它的性能决定了整个微型计算机的各项关键指标。

无论哪种 CPU,其内部基本组成总是大同小异,包括三部分:运算器、控制器、内部寄存器阵列(工作寄存器组)。典型的 8 位中央处理器结构如图 1 - 2 所示。

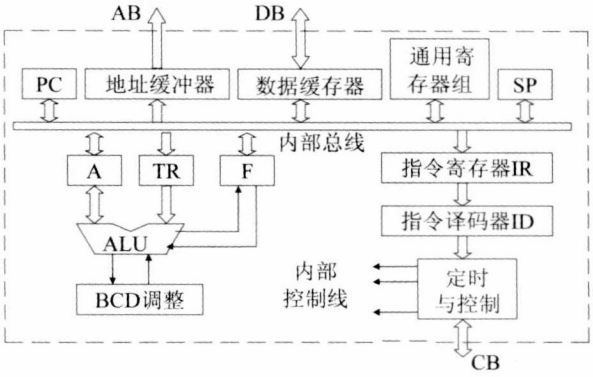


图 1-2　CPU 内部结构示意图

(1)运算器

运算器是对信息进行加工、处理及运算的逻辑部件。它由算术逻辑运算单元(Arithmetic Logic Unit,ALU)、累加器 A、暂存寄存器 TR、标志寄存器 F、BCD 调整电路等组成。新型处理器的运算器还可以完成各种高精度的浮点运算。

1)算术逻辑运算单元和累加器

算术逻辑运算单元是微型计算机执行算术运算和逻辑运算的主要部件。它有两个输入端:一个输入端与累加器 A 相连,另一个输入端与暂存寄存器 TR 相连。ALU 的输出端经累加器 A 与内部总线相连。累加器 A 是一个使用十分频繁的寄存器,它存放操作数,也存放运算结果。另一个操作数总是通过内部总线经 TR 送来的。由于总线只能分时传输数据,故用暂存寄存器 TR 在内部总线与 ALU 之间起缓冲作用。

例如在执行下述指令时

```
ADD     A,#24 H
SUB     A,R0
ANL     A,R1
```

内部总线先传输一个操作数至 TR,然后由控制器控制 ALU 对 A 和 TR 中的内容进行运算,运算结果再传输到累加器 A。

2)标志寄存器 F

标志寄存器 F 又称为状态寄存器,用来存放 ALU 运算结果的一些特征,如是否溢出、是否为零、是否为负、是否有进位、是否有偶数个“1”等。由于 ALU 的操作结果存放在累加器(A)中,因而 F 也反映了累加器(A)中所存放数据的特征。

F 中的特征标识常为 CPU 执行后续指令时所用,例如根据某种特征标志来决定程序是顺序执行还是跳转执行。

3)二—十进制调整电路

计算机在进行二—十进制(Binary - Coded Decimal,BCD)数运算时,要对运算结果进行调整,这由二—十进制调整电路来实现。

(2)控制器

控制器是计算机控制和调度的中心,计算机的各种操作都是在控制器的控制下进行的。

控制器包括指令寄存器 IR、指令译码器 ID 和定时与控制电路三部分。

计算机工作时，由定时与控制电路按照一定的时间顺序发出一系列控制信号，使计算机各部件能按一定的时间节拍协调一致地工作。

一条指令的执行分成取指令和执行指令两个阶段。具体步骤如下：

1）取指

从存储器中取回该指令的机器码，送指令寄存器 IR 暂存，直至该指令执行完毕。

2）执行指令

①由指令译码器 ID 将 IR 中指令译码，以识别该指令需要实施何种操作；

②由定时与控制电路产生一系列控制信号，送到计算机各部件以执行这一指令。

定时与控制电路除了接收译码器送来的信号外，还接收 CPU 外部送来的信号，如中断请求信号、复位信号等，这些信号由控制总线送入。定时与控制电路产生的控制信号一部分用于 CPU 内部，控制 CPU 各部件的工作；另一部分通过控制总线输出，用于控制存储器和 I/O 接口电路的工作。

（3）通用寄存器组

寄存器组可分为专用寄存器和通用寄存器。专用寄存器的作用是固定的，图 1－2 中的堆栈指针（SP）、程序计数器（PC）、标志寄存器（F）等即为专用寄存器。通用寄存器可由程序员规定其用途。通用寄存器的数目及位数因微处理器而异。由于有了这些寄存器，在需要重复使用某些操作数或中间结果时，就可将它们暂时存放在寄存器中，以避免对存储器的频繁访问，从而缩短指令长度和指令执行时间，加快 CPU 的运算处理速度，同时也给编程带来方便。

除了上述两类程序员可用的寄存器外，微处理器中还有一些不能直接为程序员所用的寄存器，如前述暂存器 TR 和后面将讲到的指令寄存器 IR 等，它们仅受内部定时与控制逻辑的控制。

从用户的角度讲，他们更关注所选择 CPU 的使用，比如在了解 CPU 功能的基础上对 CPU 内部寄存器的使用。因此，要正确使用 CPU 就必须掌握其内部寄存器的名字（符号）、大小（存放的二进制位数）及特殊功能。

（4）堆栈和堆栈指针 SP（Stack Pointer）

在计算机中广泛使用堆栈作为数据的一种暂存结构。堆栈由栈区和堆栈指针构成。

栈区是一组按先进后出或后进先出方式工作的寄存器或存储单元，用于存放数据。当它是由微处理器内部的寄存器组构成时，叫硬件堆栈；当它是由软件在内存中开辟的一个特定 RAM 区构成时，叫软件堆栈。目前绝大多数微处理器都支持软件堆栈。

堆栈指针 SP 是用来指示栈顶地址的寄存器，用于自动管理栈区，指示当前数据存入或取出的位置。在堆栈操作中，将数据存入栈区称为“压入”（PUSH），从栈区中取出数据称为“弹出”（POP）。无论是压入还是弹出，只能在栈顶进行。每当压入或是弹出一个堆栈元素，堆栈指针均会自动修改，以便自动跟踪栈顶位置。

SP 的初值是由程序员设定的。一旦设定初值后，便意味着栈底在内存储器中的位置已经确定，此后 SP 的内容即栈顶位置便由 CPU 自动管理。随着堆栈操作的进行，SP 值会自动变化，其变化方向因栈区的编址方式而异。栈区的编址方式有向下增长型和向上增长型两种。向下增长型堆栈是将新数据压入堆栈时，SP 自动减量，向上浮动而指向新的栈顶；当数据从栈中弹出时，SP 自动增量，向下浮动而指向新的栈顶。对于向上增长型堆栈 SP 的变化则相反。

堆栈主要用于中断处理与过程（子程序）调用。以后将会看到，堆栈的“先进后出”操作方

式给中断处理和子程序调用/返回(特别是多重中断与多重调用)带来很大方便。

(5)程序计数器

程序计数器 PC(Program Counter)是管理程序执行次序的特殊功能寄存器,它总是存放着下一条要执行的指令在存储器中的地址。程序中的各条指令一般是按执行顺序存放在存储器中。开始时,PC 中的值为程序第一条指令所在的地址编号。在顺序执行指令的情况下,每取得一条指令,PC 的内容自动递增,于是当从存储器取完一条指令后,PC 中存放的是下一条指令的首地址。若要改变程序的正常执行顺序,就必须把新的目标地址装入 PC,这称为程序发生了转移。指令系统中有一些指令用来控制程序的转移,称为转移指令。

程序计数器中的地址的改变有三种方式:

①复位:复位时计算机进入初始状态,PC 的内容将自动更新到初值,保证程序从头开始运行。这个初值很多情况下是 0。

②计数递增:CPU 读取一条指令时,总是将 PC 的内容作为指令地址,依此从存储器取回指令代码。同时,每取回指令代码的一个字节,PC 的内容自动加 1(加法计数)。因此,在取回一条指令后,PC 的内容已是按顺序排列的下一条指令的地址。

③直接置入新地址:PC 也能被 CPU 直接修改,从而改变程序的运行顺序。

2. *存储器*

存储器是微型计算机的重要组成部分,用来存放程序和数据。计算机有了存储器才具备记忆的能力。如无特殊强调,本书提到的存储器指内部存储器。

存储器是由许多存储单元组成的,存储单元中存放的是二进制数。每一存储器单元可以存放一个 8 位的二进制数,即一个字节(Byte)的二进制信息。一个存储器单元存放的信息称为该存储单元的内容。数据和程序均是以二进制数形式存放。不论是 8 位机还是 32 位机,都以 8 位二进制数作为一个字节存放在存储器中。

为了区分不同的存储单元,微机按一定的规律和顺序对每个存储单元进行排列编号,这个编号称为存储单元的地址。在计算机中,地址也是用二进制数来表示的,每个存储单元具有唯一的地址。对该地址的操作就是对该存储单元的操作。注意存储器单元地址和该单元的内容是两个完全不同的概念,图 1 -3 给出这两个概念的示意图。

地址	内容
0000H	10110010
0001H	10100010
0002H	11000010
	.....
F000H	10111010
	.....
FFFFH	10101110

图 1 -3　存储单元地址与内容

从应用的角度讲,计算机工作时,CPU 对存储器的操作只有“读”和“写”操作。CPU 将数据存入存储器单元的过程称为“写”操作,CPU 从存储器单元中取数据的过程为“读”操作。写入存储单元的数据取代了原数据,而且在下一个新数据写入之前一直保留着,即存储器具有记忆的功能。在执行“读”操作后,存储单元中原有的内容不变,即存储器的读出是非破坏性的。

按工作方式不同,内存可分为两大类:随机存取存储器(Random Access Memory,RAM)和只读存储器(Read Only Memory,ROM)。RAM 可以被 MPU 随机地读和写,所以又称为读写存储器。这种存储器用于存放用户装入的程序、数据及部分系统信息。当机器断电后,所存信息消失。

ROM 中的信息只能被 CPU 随机读取,而不能由 CPU 任意写入。机器断电后,信息并不丢失。所以,这种存储器主要用来存放各种程序,如汇编程序、各种高级语言的解释或编译程序、监控程序、基本 I/O 程序等标准子程序,也用来存放各种常用数据和表格等。ROM 中的内容一般是由生产厂家或用户使用专用设备写入固化的。

有关存储器的详细内容将在第 3 章中详细叙述。

3. I/O 接口电路

I 的意思是 Input,O 的意思是 Output。I/O 接口即输入或输出电路。它是微型计算机的重要组成部件,是微型计算机连接外部设备与外部设备进行信息交换的逻辑控制电路。

由于外部设备种类繁多,结构、原理各异,其运行速度、数据格式、电平等各不相同,常常与 CPU 不一致。因此,CPU 与外部设备间的连接及信息交换不能直接进行,而必须通过一个“接口电路”作为两者之间的桥梁,在 CPU 与外部设备之间起到信息转换、传递与协调的作用。

由图 1 - 1 可见 CPU 只有通过 I/O 电路才能与外部设备传输信息,只要选中 I/O 接口就能找到相应的外围设备。

从应用的角度讲,计算机工作时,CPU 对 I/O 口的操作只有“读”和“写”。CPU 对输入接口的操作称为“读操作”,对输出口的操作称为“写操作”。在操作时是对所选择的 I/O 接口的口地址的操作,因此 CPU 必须对 I/O 接口电路编址,这种编址就是 I/O 接口的地址。每个 I/O 接口所拥有的“地址”是唯一的和固定的。

I/O 接口电路种类繁多,有并行接口、串行接口、定时芯片、A/D 转换器、D/A 转换器、中断控制器、DMA 控制器等。

有关 I/O 接口的详细内容将在第 4 章中详细叙述。

4. 外部设备

外部设备简称外设。典型外设有输入设备:键盘、鼠标、扫描仪等;输出设备:显示器、打印机、绘图仪等;磁带、磁盘则既是输入设备也是输出设备。

有关外设的内容将在第 9 章中详细叙述。

5. 总线 Bus

Bus 即总线,是一组公共通信线。微型计算机系统采用总线结构后,芯片之间不需要单独走线,这就大大减少了连接线的数量。采用总线结构后,系统中各芯片间的相互关系转变为各芯片面向总线的单一关系。符合总线标准的设备都可以连接到系统中,使系统功能得到扩展。

总线可一分为三:地址总线 AB (Address Bus)、数据总线 DB (Data Bus)、控制总线 CB (Control Bus)。它们来自微处理器 CPU,参见图 1 - 2。

(1)地址总线 AB

AB 是三态单向总线,由 CPU 输出。CPU 通过 AB 向存储器或 I/O 接口传输地址信息,此地址信息是某存储单元或某 I/O 接口的地址,表示 CPU 将对此地址进行读操作或写操作。地址总线的宽度决定了 CPU 可直接寻址的容量,n 条地址线用 An - 1 ~ A0 表示,An - 1 为最高位地址线,A0 为最低位地址线,最大寻址范围为 2n。

(2)数据总线 DB

数据总线是中央处理器与存储器及外设交换数据的通路,是三态双向总线。数据总线的位数与微处理器的位数相同,一般有 8 位、16 位、32 位、64 位等。N 位(N = 8,16,32,64)数据总线用 DN - 1 ~ D0 表示,DN - 1 为最高有效位,D0 为最低有效位。

(3)控制总线 CB

控制总线是用来传输控制信号的,传输方向就具体控制信号而定:

1)微处理器向从器件输出的控制信号:读信号、写信号、"存储器地址有效"信号、"I/O 口地址有效"信号等。

2)I/O 接口向微处理器输出的信号:复位信号、中断请求信号等。

如没有特别交代,本书所提控制信号专指上述两类型信号。这些控制信号将在专门的章节讨论。

由此可见就某一根控制线而言其传输方向是确定的,单向的。但控制总线作为一个整体则认为是双向线。所以在各种结构框图中,凡把控制总线为一个整体画出时,均以双向线表示。

1.2 指令与程序的概念

1. 指令

指令是规定计算机执行特定操作的命令。具体而言,指令由 CPU 执行。CPU 就是根据指令来指挥和控制微机各部分协调地动作,以完成规定的操作。计算机全部指令的集合叫作计算机指令系统,指令系统准确定义了计算机的处理能力。不同型号的计算机有不同的指令系统,从而形成各种型号计算机的特点和相互间差异。

2. 程序

程序则是为解决某一问题而编写在一起的指令序列。目前微机系统中使用着 3 个层次、3 种形式的程序。

(1)机器语言程序

指令是以二进制代码形式存在的,这种指令叫做机器码指令。机器码指令构成的指令系统叫机器语言。用机器语言编写的程序叫机器语言程序。

机器语言程序的优点是能被 CPU 直接理解和执行。缺点是编程烦琐、不直观、难记忆、易出错。

(2)汇编语言程序

为克服机器语言程序的缺点,人们通常用助记符(几个字母构成的符号)来代替机器码指令。助记符与机器码指令之间有一一对应的关系。这种用助记符构成的指令系统称为汇编语言,用汇编语言编写的程序称为汇编语言程序。

汇编语言与机器语言并无本质区别。由于 CPU 只能执行机器码指令,即只能执行机器语言程序,汇编语言程序必须先"汇编"成机器语言程序才能被 CPU 执行。

(3)高级语言程序

为了使编写的程序更直观、易懂,更易于面向问题和对象,人们设计和推出了多种多样的

更接近于习惯用的自然语言和数学语言的高级语言，如 BASIC、C、FORTRAN、COBOL、PASCAL、Turbo C、C + +、VB、VC 等。各种高级语言尽管各有其特点，但都是以语句和数据的定义为基础，且通常一个语句都是由一组机器语言指令（亦即一组汇编指令）构成的。所谓高级语言程序就是用高级语言编写的程序。

同样由于 CPU 只能执行机器语言程序，高级语言程序必须先“编译”成机器语言程序才能被 CPU 执行。

3. 程序在计算机系统中的安放

存储器存放的程序一定是 CPU 能直接采取、直接理解和执行的机器语言程序。而其他语言的程序，由于不能被 CPU 直接执行，只能存放在外部存储器。外部存储器是图 1 - 1 中外部设备的一种。

1.3 微型计算机工作原理

1.3.1 CPU 工作原理

定义图 1 - 1 中 CPU 为微机系统主芯片（主片），总线上的其他芯片为从芯片（从片）。

无论哪种主片，其工作原理大同小异，如图 1 - 4。特别强调的是图 1 - 4 中提及的指令是机器语言指令。

1. CPU 工作原理

图 1 - 4 中 PC 是程序计数器，是管理程序执行次序的特殊功能寄存器。CPU 复位时 PC 一般为 0，之后按图 1 - 4 变化。一般来说指令在存储器中的存放方式是从地址 0 开始顺序存放的。我们结合图 1 - 1 来理解 CPU 工作原理

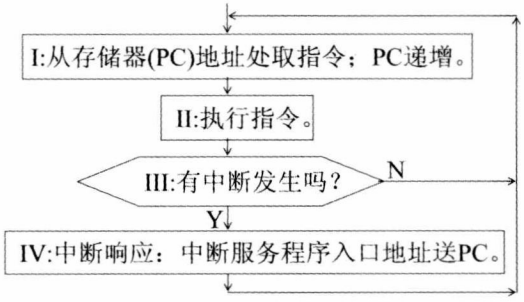


图 1 - 4　CPU 工作原理图

Ⅰ：CPU 从存储器读取一条指令时，总是将 PC 的内容作为指令地址，并经地址总线送到存储器，从而从存储器该地址中取得指令，送入 CPU 指令寄存器 IR。同时每取回指令的一个字节 PC 的内容自动加 1。因此，在 CPU 取得一条指令后，PC 的内容已是下一条指令的地址。

Ⅱ：CPU 执行指令。各种指令实现各自的功能，使微型计算机系统功能丰富多彩。有人称“计算机由软件决定一切”，其依据就是这一步，因为软件即为指令的有序集合。

Ⅲ：在不发生中断的情况下，CPU 返回 Ⅰ，保证了程序按在存储器中的存储顺序依次执行。