1 冯·诺伊曼计算机 ---------- 2
2 系统总线 ---------- 3
2.1 总线的基本概念（P41） ---------- 3
2.2 总线的分类（P43-44） ---------- 3
2.3 总线的特性（P45-46） ---------- 3
2.4 总线的性能指标（P46） ---------- 3
3 存储器 ---------- 4
3.1 概述（P68-70） ---------- 4
3.2 存储器层次结构（P71） ---------- 5
3.3 主存储器 ---------- 5
3.4 高速缓存存储器（P109，P117-120） ---------- 6
3.5 辅助存储器（P123-125） ---------- 6
4 I/O 设备与 I/O 接口 ---------- 7
4.1 I/O 设备 ---------- 7
4.2 I/O 接口 ---------- 7
5 程序查询方式 ---------- 8
6 程序中断方式 ---------- 9
7 DMA 方式 ---------- 9
8 计算机运算方式 ---------- 10
8.1 三种码制 ---------- 10
8.2 浮点表示 ---------- 10
8.3 补码的加减法 ---------- 10
8.4 乘法运算 ---------- 11
9 机器指令 ---------- 11
10 寻址方式 ---------- 12
10.1 指令寻址（P310） ---------- 12
10.2 数据寻址（P310-318） ---------- 12
11 指令周期 ---------- 13
12 中断屏蔽字（P366） ---------- 14
13 微程序设计 ---------- 14

# 1 冯·诺伊曼计算机

（1） 冯•诺依曼机的特点（P8-9）

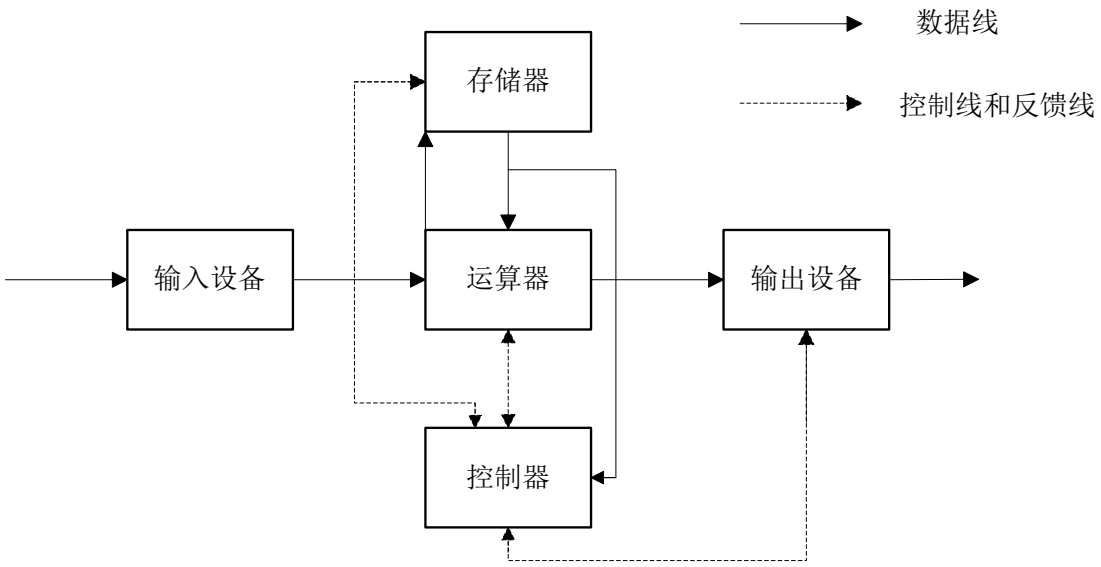



**图 1 典型的冯·诺伊曼计算机的结构框图**

1) 计算机由运算器、存储器、控制器、输入设备和输出设备五大部件组成。
2) 指令和数据以同等地位存放于存储器中，并可按地址寻访。
3) 指令和数据均用二进制数表示。
4) 指令由操作码和地址码组成。
5) 指令在存储器内按顺序存放。
6) 机器以运算器为中心，输入输出设备与存储器间的数据传送通过运算器完成。

（2） 计算机各部件的功能（P9）

1) 运算器用来完成算术运算和逻辑运算，并将运算的中间结果暂存在运算器内。
2) 存储器用来存放数据和程序。
3) 控制器用来控制、指挥程序和数据输入、运行以及处理运算结果。
4) 输入设备可将人们熟悉的信息形式转换为机器能识别的信息形式。
5) 输出设备可将机器运算结果转换为人们熟悉的信息形式。

# 2 系统总线

## 2.1 总线的基本概念（P41）

总线是连接多个部件的信息传输线，是各个部件共享的传输介质。

## 2.2 总线的分类（P43-44）

（1） 数据总线

数据总线用来传输各功能部件之间的数据信息，它是双向传输总线，其位数与机器字长、存储字长有关。

（2） 地址总线

地址总线用来指出数据总线上的源数据或目的数据在主存单元的地址或 I/O 设备的地址，它是单向传输总线，与存储地址、I/O 设备地址有关。

（3） 控制总线

控制总线用来发出各种控制信号，它是逻辑双向，物理单向的总线。

## 2.3 总线的特性（P45-46）

（1） 机械特性

机械特性是指总线在机械连接方式上的一些性能。

（2） 电气特性

电气特性是指总线的每一根传输线上信号的传输方向和有效的电平范围。

（3） 功能特性

功能特性是指总线中每根传输线的功能，其分为地址、数据和控制三种总线。

（4） 时间特性

时间特性是指总线中的任一根线在什么时间内有效。

## 2.4 总线的性能指标（P46）

（1） 总线宽度

通常指数据总线的根数，即多少 bit（位）对应多少根。

（2） 总线带宽

总线的数据传输速率，即单位时间内总线上传输数据的位数。

（3） 时钟同步/异步

总线上的数据是否与时钟同步。

（4） 总线复用

一条信号线上分时传送两种信号（地址线和数据线复用）。

（5） 信号线数

地址、数据、控制三种总线数的总和。

（6） 总线控制方式

突发工作、自动配置、仲裁方式、逻辑方式、计数方式等。

# 3 存储器

## 3.1 概述（P68-70）

（1） 按存储介质分类

半导体存储器、磁表面存储器、磁芯存储器、光盘存储器、纸带。

（2） 按存取方式分类

随机存储器（RAM）、只读存储器（ROM）、串行访问存储器。

（3） 按在计算机中的作用分类

主存储器（内存）、辅助存储器、缓冲存储器（放在速度不同的部件之间用作提速）。

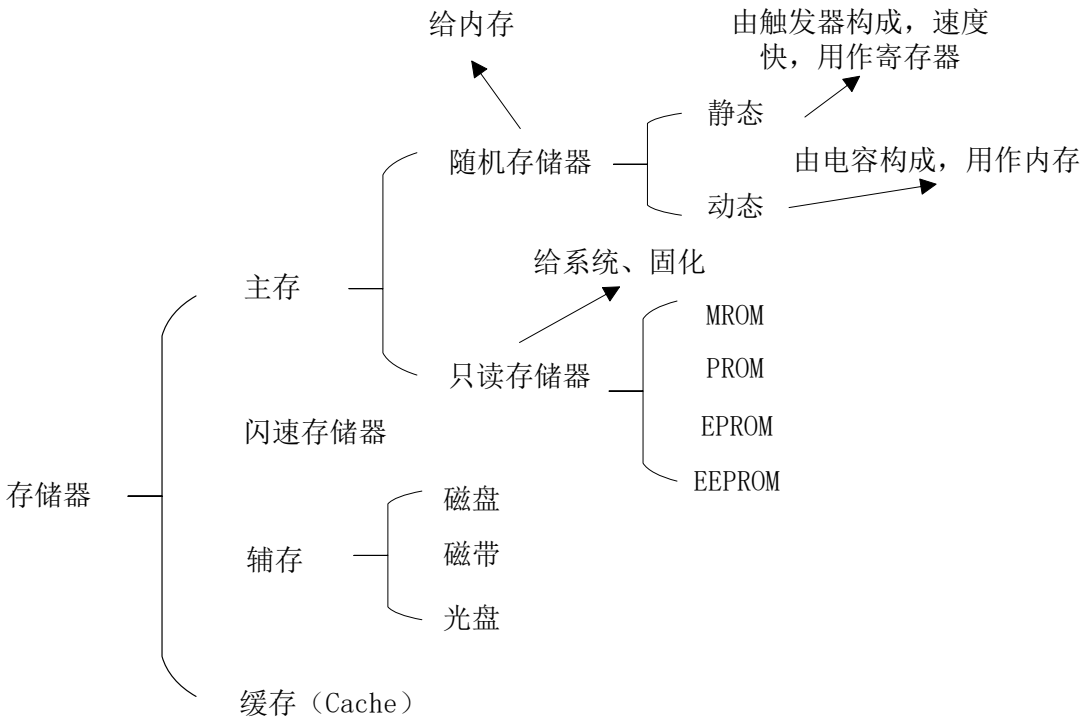



**图 2 存储器分类**

## 3.2 存储器层次结构（P71）

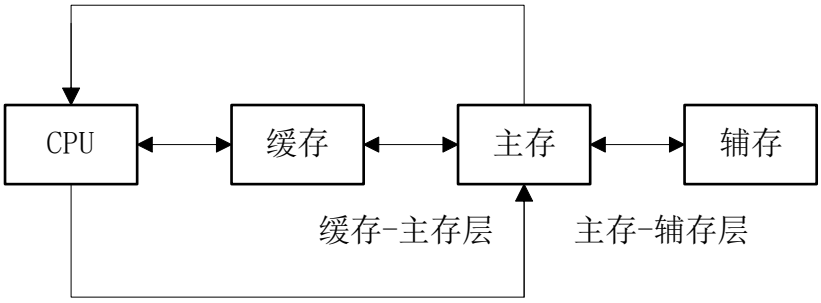



图 3 缓存-主存层次和主存-辅存层次

（1） 缓存-主存层次主要解决 CPU 和主存速度不匹配的问题（提速）。

（2） 主存-辅存层次主要解决存储系统的容量问题（扩容）。

## 3.3 主存储器

（1） 主存的技术指标（P74）

1) 存储容量

存储容量是指主存能存放的二进制代码的总位数，即

存储容量=存储单元个数×存储字长

2) 存储速度

存储速度由存取时间和存取周期来表示。

① 存储时间又称为存储器的访问时间，是指启动一次存储器操作到完成该操作所需的全部时间，其又分为读出时间和写入时间两种。

② 存储周期是指存储器进行连续两次独立的存储操作所需的最小间隔时间。

3) 存储器带宽

它表示的是单位时间内存储器存取的信息量。

（2） 半导体存储芯片的译码驱动方式：线选法和重合法。

（3） 存储器容量的扩展

存储容量扩容的公式如下：

$$\text{个数}=\frac{\text{目标容量}}{\text{芯片容量}}$$

1) 位扩展

位扩展指增加存储字长，即使用多个芯片来存储目标存储器的高低各个位，其中芯片的地址线顺序连接，地址线按对应位进行连接。

2) 字扩展

字扩展指增加存储字的数量，即使用多个芯片来存储目标存储器的地址个数，通过高位地址线来区分所使用的芯片或芯片组。

## 3.4 高速缓存存储器（P109，P117-120）

（1） 程序访问的局部性

这是由于指令和数据在主存内都是连续存放的，并且有些指令和数据往往会被多次调用，即指令和数据在主存的地址分布不是随机的，而是相对的蔟聚，使得 CPU 在执行程序时，访存具有相对的局部性。

（2） 直接映射

每个缓存块 $i$ 可以和若干个主存块对应，每个主存块 $j$ 只能和一个缓存块对应，即某一主存块只能固定映射到某一缓存块中（不灵活）。

（3） 全相联映射

主存中的任一块可以映射到缓存中的任一块，即某一主存块能映射到任一缓存块中（成本高）。

（4） 组相联映射

某一主存块 $j$ 按模 $Q$ 映射到缓存的第 $i$ 组中的任一块，即某一主存块只能映射到某一缓存组中的任一块。

## 3.5 辅助存储器（P123-125）

（1） 概念及特点

辅助存储器作为主存的后援设备又称为外部存储器，简称外存，它与主存一起组成了存储器系统的主存-辅存层次，其主要特点为不直接与 CPU 交换信息。

（2） 存储容量

$$C=n\times k\times s$$

$C$ 为存储总容量，$n$ 为存放信息的盘面数，$k$ 为每个盘面的磁道数，$s$ 为每条磁道上记录的二进制代码数。

（3） 平均寻址时间

$$T_a = t_{sa} + t_{wa} = \frac{t_{smax} + t_{smin}}{2} + \frac{t_{wmax} + t_{wmin}}{2}$$

$T_a$ 为平均寻址时间，$t_{sa}$ 为磁头寻找目标磁道的平均找道时间，$t_w$ 为磁头等待欲读/写的磁道区段旋转到磁头下方所需的平均等待时间。

（4） 数据传输率

$$D_r = D_b \times V$$

数据传输速率 $D_r$ 与记录密度 $D_b$ 和记录介质的运动速度 $V$ 有关。

# 4 I/O 设备与 I/O 接口

## 4.1 I/O 设备

（1） I/O 设备的分类

大致可分为三类：人机交互设备、计算机信息的存储设备、机-机通信设备。

（2） I/O 设备与主机的联系方式

1) I/O 设备编址方式：统一编址、不统一编址。

2) 设备选址：用设备选择电路识别是否被选中。

3) 传送方式：并行、串行。

4) I/O 指令：操作码+命令码+设备码。

## 4.2 I/O 接口

（1） 接口的功能和组成

1) 功能

选址功能、传送命令的功能、传送数据的功能、反映设备状态的功能。其中反映设备状态的功能由完成触发器 D、工作触发器 B、中断请求触发器 INTR 和屏蔽触发器 MASK 组成。

2) 组成

由设备选择电路、命令寄存器、命令译码器、数据缓冲寄存器和设备状态标记组成。

（2） 接口类型

接口类型可以分别按照传送方式、选择的灵活性、通用性和控制方式分类，分为并行和串行、可编程和不可编程、通用和专用、中断和 DMA。

# 5 程序查询方式

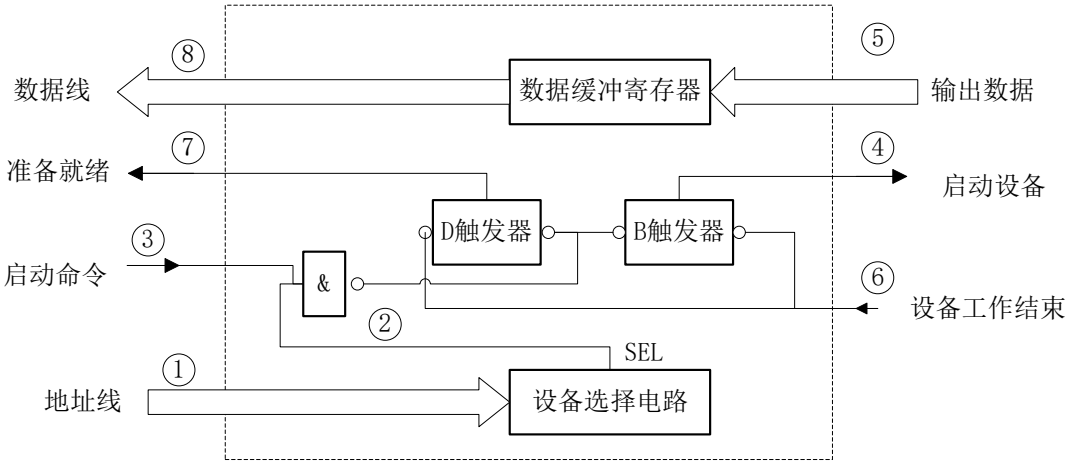



**图 4 程序查询方式接口电路的基本结构**

以输入设备为例，该接口的工作过程如下：

① 当 CPU 通过 I/O 指令启动输入该设备时，指令的设备码字段通过地址线送至设备选择电路。、

② 若该接口的设备码与地址线上的代码吻合，其输出 SEL 有效。

③ I/O 指令的启动命令经过“与非”门将工作触发器 B 置“1”，将完成触发器 D 置“0”。

④ 由 B 触发器启动设备工作。

⑤ 输入设备将数据送至数据缓冲寄存器。

⑥ 由设备发送设备工作结束信号，将 D 置“1”，B 置“0”，表示外设准备就绪。

⑦ D 触发器以“准备就绪”状态通知 CPU，表示“数据缓冲满”。

⑧ CPU 执行输入指令，将数据缓冲寄存器中的数据送至 CPU 的通用寄存器（ACC 累加器），再存入主存相关单元。

## 6 程序中断方式

（1） 概念

计算机在执行程序的过程中，当出现异常情况或特殊请求时，计算机停止现行程序的运行，转向对这些异常情况或特殊请求的处理，处理结束后再返回先行程序的尖端出，继续执行源程序的过程。

（2） 中断服务程序的流程

一般的中断服务程序的流程分四部分：保护现场、中断服务、恢复现场和中断返回。

1) 保护现场

程序断点的保护（由中断隐指令完成），寄存器内容的保护（由进栈指令完成）。

2) 中断服务（设备服务）

对不同的 I/O 设备具有不同内容的设备服务。

3) 恢复现场（由出栈指令完成）

4) 中断返回（由中断返回指令完成）

（3） 单重中断和多重中断

1) 单重中断：不允许中断现行的中断服务程序。

2) 多重中断：允许级别更高的中断源中断现行的中断服务程序。

## 7 DMA 方式

（1） 特点

由于主存和 DMA 接口之间有一条数据通道，因此主存和设备交换信息时，不通过 CPU，也不需要 CPU 暂停现行程序为设备服务，省去了保护现场和恢复现场，因此工作速度比程序中断方式的工作速度高。

（2） DMA 与主存交换数据的三种方式

1) 停止 CPU 访问主存：控制简单，但 CPU 处于不工作状态或保持状态，未充分发挥 CPU 对主存的利用率。

2) 周期挪用（窃取）：仅当 CPU 和 DMA 同时请求访存时，CPU 将总线控制权让给 DMA。

（3） D 与 C 交替访问：交替进行，无需申请建立和归还总线的使用权。

# 8 计算机运算方式

## 8.1 三种码制

（1） 正数：正数的原码、补码和反码一致。

（2） 负数

1) 原码：高位为符号位。

2) 补码：除符号位外按位取反，末位加一。

3) 反码：除符号位外按位取反。

## 8.2 浮点表示

（1） 规格化数：尾数最高位为 1 的浮点数。

（2） 浮点数的表示形式

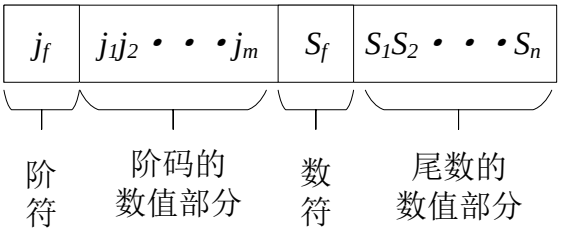



图 5 浮点数在机器中的形式

阶码是整数，阶符和阶码的位数 $m$ 合起来反映浮点数的表示范围及小数点的实际位置；尾数是小数，其位数 $n$ 反映了浮点数的精度；尾数的符号 $S_f$ 代表浮点数的正负。

（3） 浮点数表示范围的溢出

1) 上溢：超过大小范围。

2) 下溢：超过精度范围。

## 8.3 补码的加减法

（1） 加法

1) 整数：$[A]_{补}+[B]_{补}=[A+B]_{补}$（mod $2^{n+1}$）

2) 小数：$[A]_{补}+[B]_{补}=[A+B]_{补}$（mod 2）

（2） 减法

实质为 A-B=A+（-B）连同符号位一起相加，符号位产生的进位自然丢掉。

（3） 溢出判断

使用两位符号位判断溢出，结果的双符号位相同则未溢出，反之则溢出，其中最高符号位代表其真正的符号。

## 8.4 乘法运算

（1） 原码一位乘法

1) 符号位单独处理：$x_f \oplus y_f = S_f$

2) $x^* \cdot y^*$

① 判别位为 1：（部分积）Z+x*，右移一位。

② 判别位为 0：部分积）Z+0，右移一位。

③ 结束条件：移位次数等于乘数位数。

（2） Booth 算法（补码比较法）

相比于原码一位乘法，在运算表中添加一列乘数的附加位 $\mathbf{y_{n+1}}$，初始为 0，后按照乘数列的尾数位和附加位共同作为判别位，令附加位为 $\mathbf{y_{n+1}}$，乘数列的尾数位为 $\mathbf{y_n}$。

1) $\mathbf{y_{n+1}-y_n=0}$：部分积右移一位。

2) $\mathbf{y_{n+1}-y_n=1}$：部分积加上$\mathbf{[x]_{补}}$后右移一位。

3) $\mathbf{y_{n+1}-y_n=-1}$：部分积加上$\mathbf{[-x]_{补}}$右移一位。

# 9 机器指令

（1） 指令的一般格式

| 操作码字段 | 地址码字段 |
|---|---|

**图 6 指令的一般格式**

1) 操作码：用于指明指令所要完成的操作。

2) 地址码：用于指出该指令的源操作数的地址、结果的地址以及下一条指令的地址。

（2） 扩展操作码（P301，直接看书）

# 10 寻址方式

## 10.1 指令寻址（P310）

（1） 简述：指令寻址是取下一条指令的过程。

（2） 分类：顺序寻址和跳跃寻址。

## 10.2 数据寻址（P310-318）

（1） 简述（含义）：从形式地址出发，按寻址特征找到有效地址的过程。

（2） 分类及特点

1) 立即寻址：操作数本身设在指令字内。
2) 直接寻址：指令字中的形式地址就是操作数的真实地址。
3) 隐含寻址：指令字中不明显地给出操作数的地址，其操作数的地址隐含在操作码或某个寄存器内。
4) 间接寻址：有效地址是由形式地址间接提供的。
5) 寄存器寻址：地址码字段直接指出了寄存器的编号。
6) 寄存器间接寻址：地址码字段指出的寄存器内存放的是操作数所在主存单元的地址号。
7) 基址寻址：操作数的有效地址等于指令字中的形式地址与基址寄存器中的内容（基地址）相加。
8) 变址寻址：操作数的有效地址等于指令字中的形式地址与变址寄存器中的内容相加。
9) 相对寻址：其有效地址是将程序计数器 PC 的内容（即当前指令的地址）与指令地址中的形式地址相加而成。
10) 堆栈寻址：遵循先进后出，其操作数只能从栈顶地址指示的存储单元存或取，硬堆栈由寄存器组实现，软堆栈由指定的存储空间实现。

# 11 指令周期

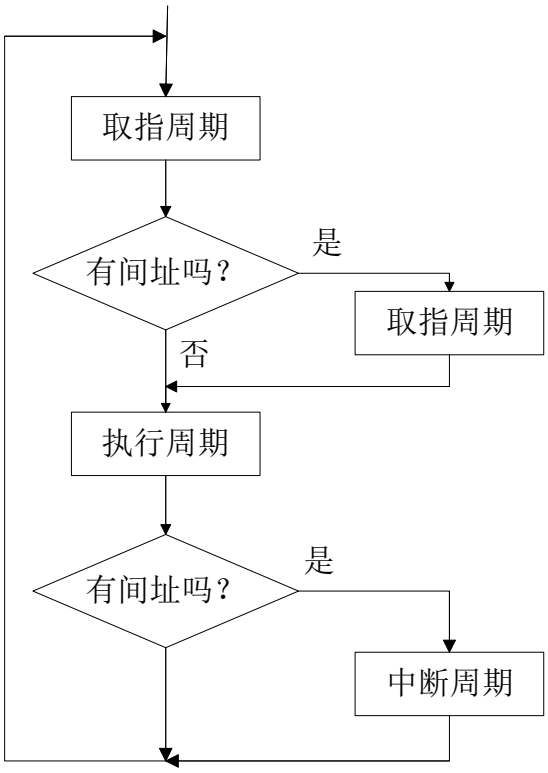



图 7 指令周期流程

（1） 概念：CPU 取出并执行一条指令所需的全部时间。

（2） 分类

1) 取指周期：完成取指令和分析指令的操作。

2) 间址周期：取出有效地址。

3) 执行周期：取出操作数。

4) 中断周期：保存程序断点。

## 12 中断屏蔽字（P366）

对应每个中断请求触发器就会有一个屏蔽触发器，将所有的屏蔽触发器组合在一起，便构成了一个屏蔽寄存器，屏蔽寄存器的内容称为屏蔽字。中断优先级与屏蔽字的关系为高优先级的屏蔽字从右往左的 1 多，低优先级的屏蔽字从左往右的 0 多。

## 13 微程序设计

简述：将一条机器指令编写成一个微程序，每一个微程序包含若干条微指令，每一条微指令对应一个或几个微操作命令。然后将这些微程序存到一个控制存储器中，用寻址用户程序机器指令的方法来寻找每个微程序中的微指令。逐条执行每一条微指令，就相应地完成了一条机器指令的全部操作。