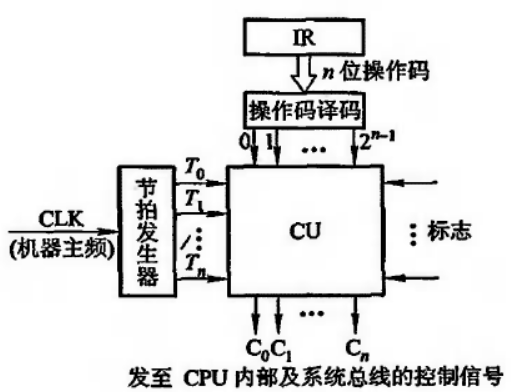
10.7

逻辑控制单元的组成框图



工作原理：

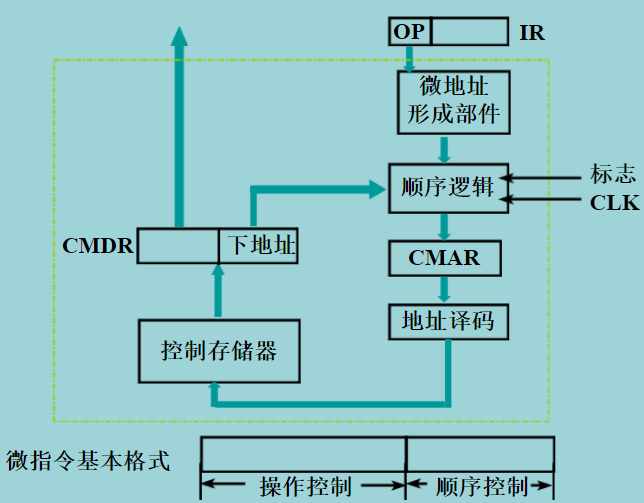
完成一条指令要经过取指阶段和执行阶段

取值阶段完成的任务是：根据PC给定的地址发出读命令，访存后取出相应的指令送至IR，再经指令译码器ID给出信号，控制微操作命令序列形成部件，并修改PC

执行阶段完成的任务是：微操作命令序列形成部件同时还接受时序电路发来的时序信号，在译码输出的共同作用下，按指令操作码的含义，发出一系列微操作命令信号，控制响应部件操作，实现指令功能

10.8

微程序控制单元的组成框图：



取指阶段：

1. 将取值周期微程序首地址M->CMAR
2. 取微指令，CM（CMAR）-> CMDR
3. 产生微操作命令
4. 形成下一条微指令的地址M+1，且M+1 -> CMAR
5. 取出下一条微指令，CM(CMAR) -> CMDR
6. 产生微操作命令；以此类推，知道取出取值周期最后一条微指令，并发出微命令为止。此时一条机器指令已存至指令寄存器中

执行阶段：

1. OP(IR) -> CMAR
2. 取微指令，CM(CMAR) -> CMDR
3. 产生微操作命令
4. 形成下一条微指令的地址，Ad(CMDR) -> CMAR
5. 取微指令，CM(CMAR) -> CMDR
6. 产生微操作命令，以此类推

10.9

从设计步骤上看：

组合逻辑控制器的设计步骤：

* 1. 拟定机器的指令系统
  2. 确定CPU总体结构
  3. 确定时序系统，拟定指令流程
  4. 安排每条指令中微操作的节拍
  5. 列出微操作命令的操作时间表
  6. 写出每一个微操作命令的逻辑表达式并花间
  7. 画出相应的组合逻辑电路图

微程序控制器的设计步骤前三步与组合逻辑控制器相同，后面的步骤如下：

1. 写出对应机器指令的微操作及节拍安排
2. 确定微指令格式（确定微指令的编码方式和后继微地址的形成方式）
3. 编写微指令码点

从硬件组成上看：

组合逻辑控制器由组合逻辑电路提供微指令，其核心部件是各种门电路构成的复杂树形网络

微程序控制器由存储逻辑（微指令）提供微命令，其核心部件是控制存储器

综上，组合逻辑控制器速度更快，因为其微命令全部由硬件（组合逻辑门电路）产生

10.10

主要有三种编码方式：

1. 直接控制，又称直接编码，特点是操作控制字段的每一位代表一个微命令，优点是简单直观，输出直接用于控制，执行速度最快。缺点是微指令字长最长，所需存储空间大
2. 字段直接编码控制，又称显式编码，其特点是将微指令操作控制字段分成几段，并使每个字段经译码后形成各个微操作命令。每个字段中的微命令必须是互斥的。这种编码方式缩短了微指令字长，但增加了译码电路，使微指令的执行速度降低
3. 字段间接编码控制，又称隐式编码，这种方式一个字段的某些微命令还需由另一个字段中的某些微命令解释，更能缩短微指令字长

10.11

垂直型微指令的结构类似于一般机器指令的结构由微操作码译码确定微指令的功能。通常一条微指令只能有1～2个微操作命令。因为它要经过译码后控制对象影响每条微指令的执行时间。而且垂直型微指令字长较短实现一条机器指令的微程序要比水平型微指令编写的微程序长得多它是用较长的微程序结构来换取较短的微指令结构。

水平型微指令一次能定义并执行多个并行操作其并行操作能力强效率高。而且水平型微指令的大多数微命令一般可直接控制对象故执行每条微指令的时间短。又因水平型微指令字长较长故可用较少的微指令数来实现一条机器指令的功能。

10.13

主要有两种方式：

1. 断定方式，直接由微指令的下地址字段指出
2. 增量方式，增设一个微程序计数器，微地址的形成方式类似于机器指令地址的形成方式，要求微程序中的微指令是顺序存储

10.15

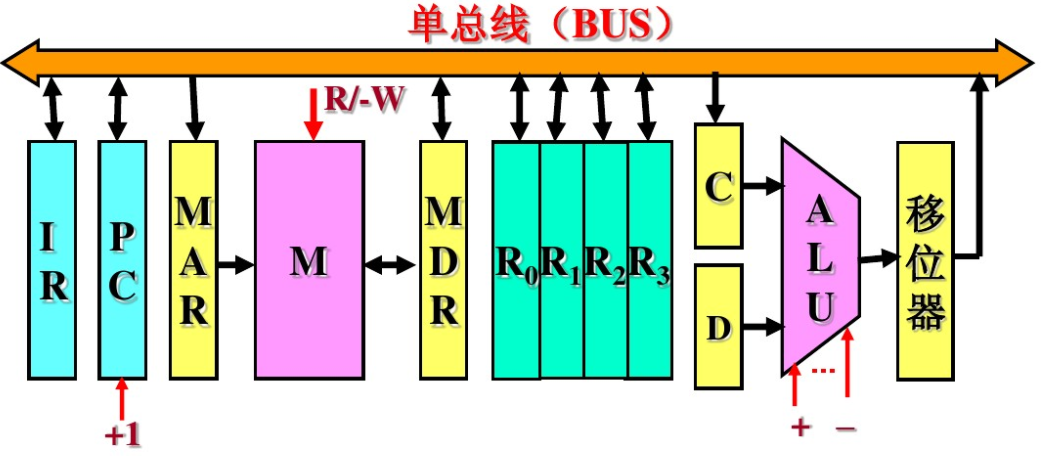
操作控制为35位，转移条件为4位，下地址为9位

微指令格式为水平型微指令

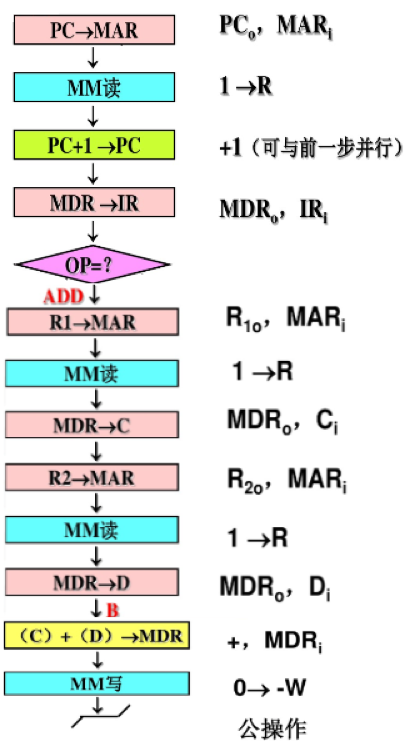
10.19

（1）采用单总线结构的CPU硬件框图如下：

数据流动方向如箭头所示



（2）



（3）对应微操作命令如上图右侧