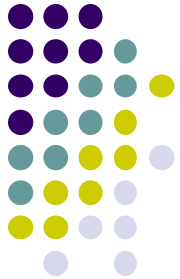


计算机组成与系统结构

第五章 中央处理器

吕昕晨

lvxinchen@bupt.edu.cn

网络空间安全学院

第五章 中央处理器

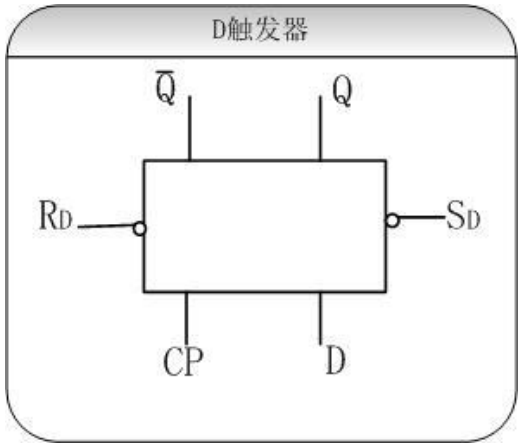
- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
- 微程序控制技术

时序信号

- 指令与微操作
  - 每一条计算机指令微操作
  - 各微操作的执行是有顺序的，控制器只会发出微操作指令，叫各个部件去完成
  - 问题：怎么知道这个操作什么时候完成?
- 时序信号
  - 控制管理上述微操作的“作息时间”，确定时间段执行哪些微操作
  - CPU可以用时序信号/周期信息来辨认从内存中取出的是指令（取指）还是数据（执行）
  - 操作控制器发出的各种信号是时间（时序信号）和空间（部件操作信号）的函数

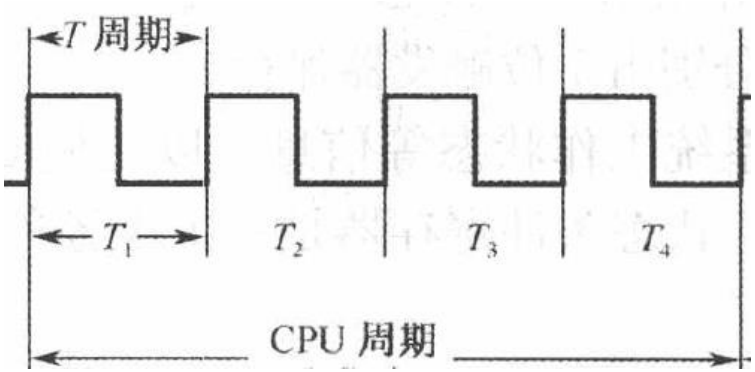
时序信号体制——电位脉冲

- 时序信号体制
  - 组成计算机硬件的器件特性决定了时序信号的基本体制是电位—脉冲制
- 以触发器为例
  - D为电位输入端，CP（Clock Pulse）为脉冲输入端
  - R,S为电位输入端
  - 特性方程如下
    - D=0时，CP上升沿到来时
      - D触发器状态置0
    - D=1时，CP上升沿到来时
      - D触发器状态置1



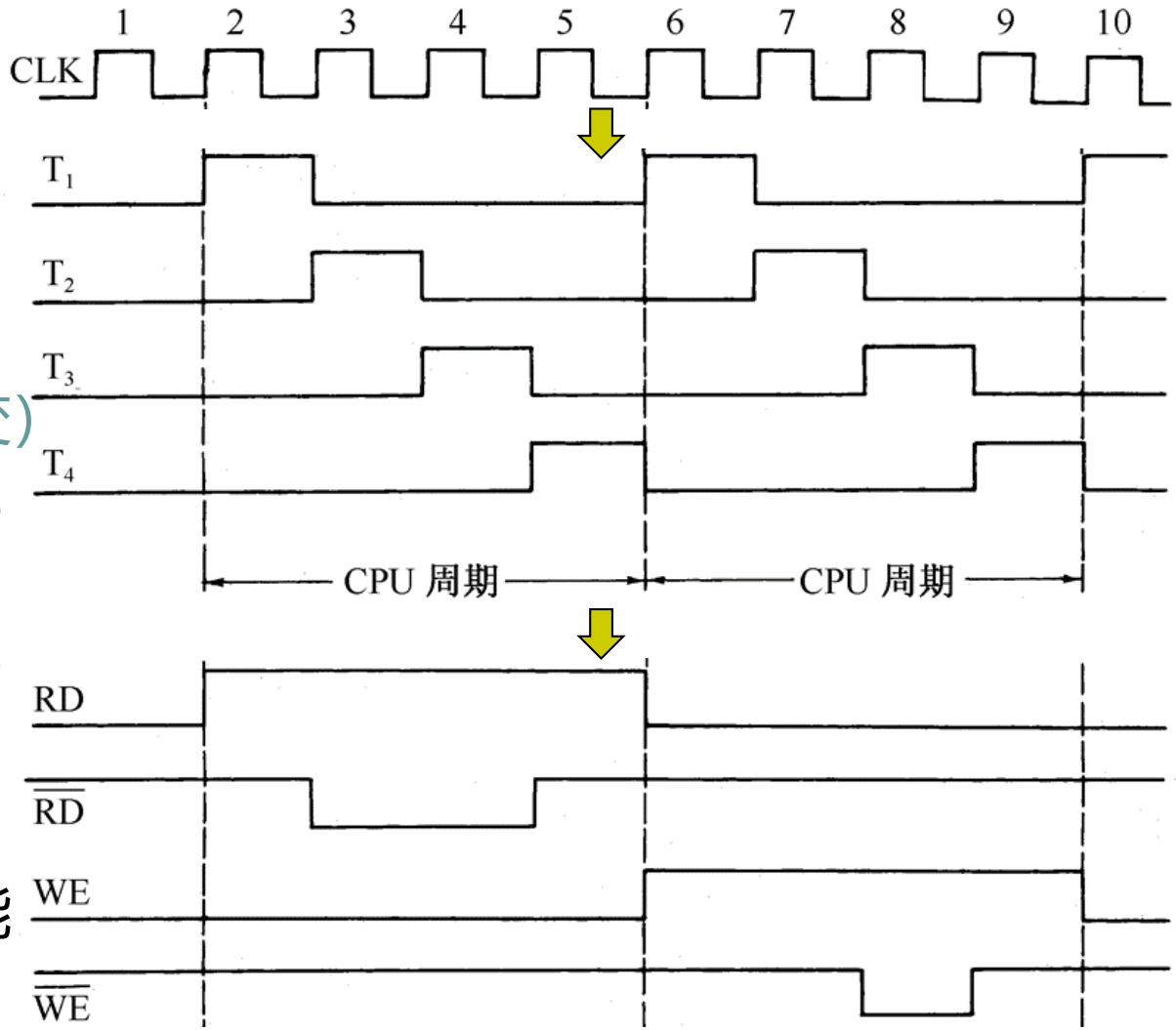
时序信号体制

- 硬布线控制器——三级体制
  - 主状态周期
  - 节拍电位
  - 节拍脉冲
- 微程序控制器——二级体制
  - 节拍电位：CPU周期时间
  - 节拍脉冲：T周期
- 实现方式：分频器
  - 输出信号频率为输入信号频率整数分之一的电路
  - 以高精度晶振为主时钟，变换输出各种频率成分
  - 脉冲分频器，调整频率与占空比（Counter/Verilog）



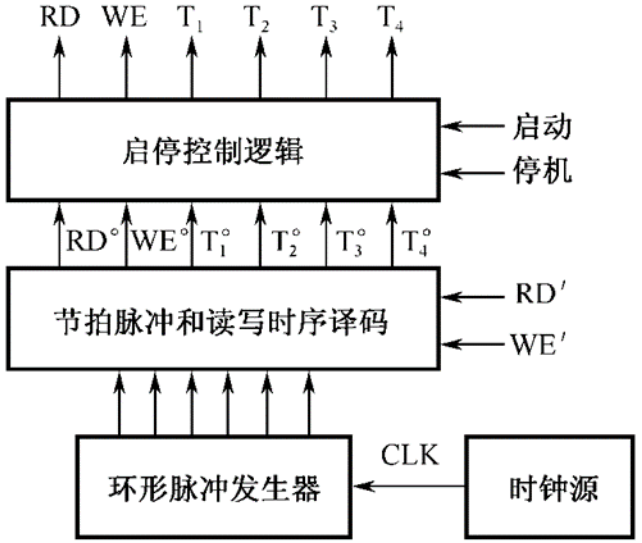
时序信号关系

- 主频：晶振
- 节拍脉冲
  - CNT：0~3
  - 四分频（可变）
  - 占空比：25%
- 读写控制信号
  - 读写控制功能

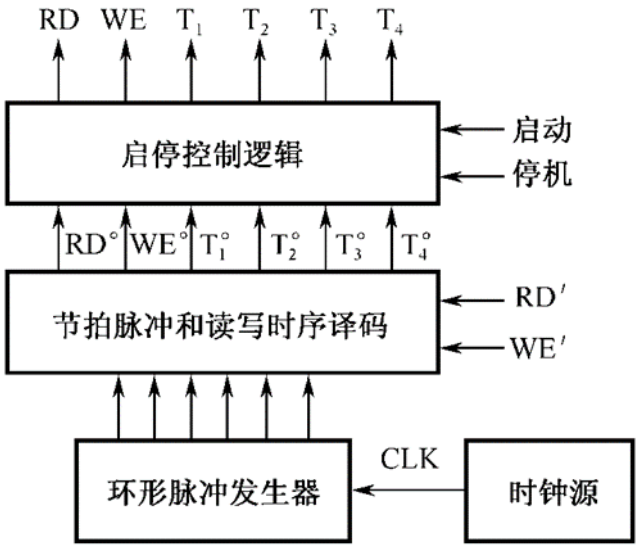


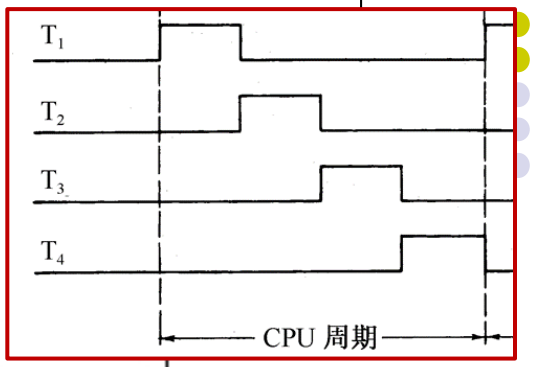
时序信号产生器

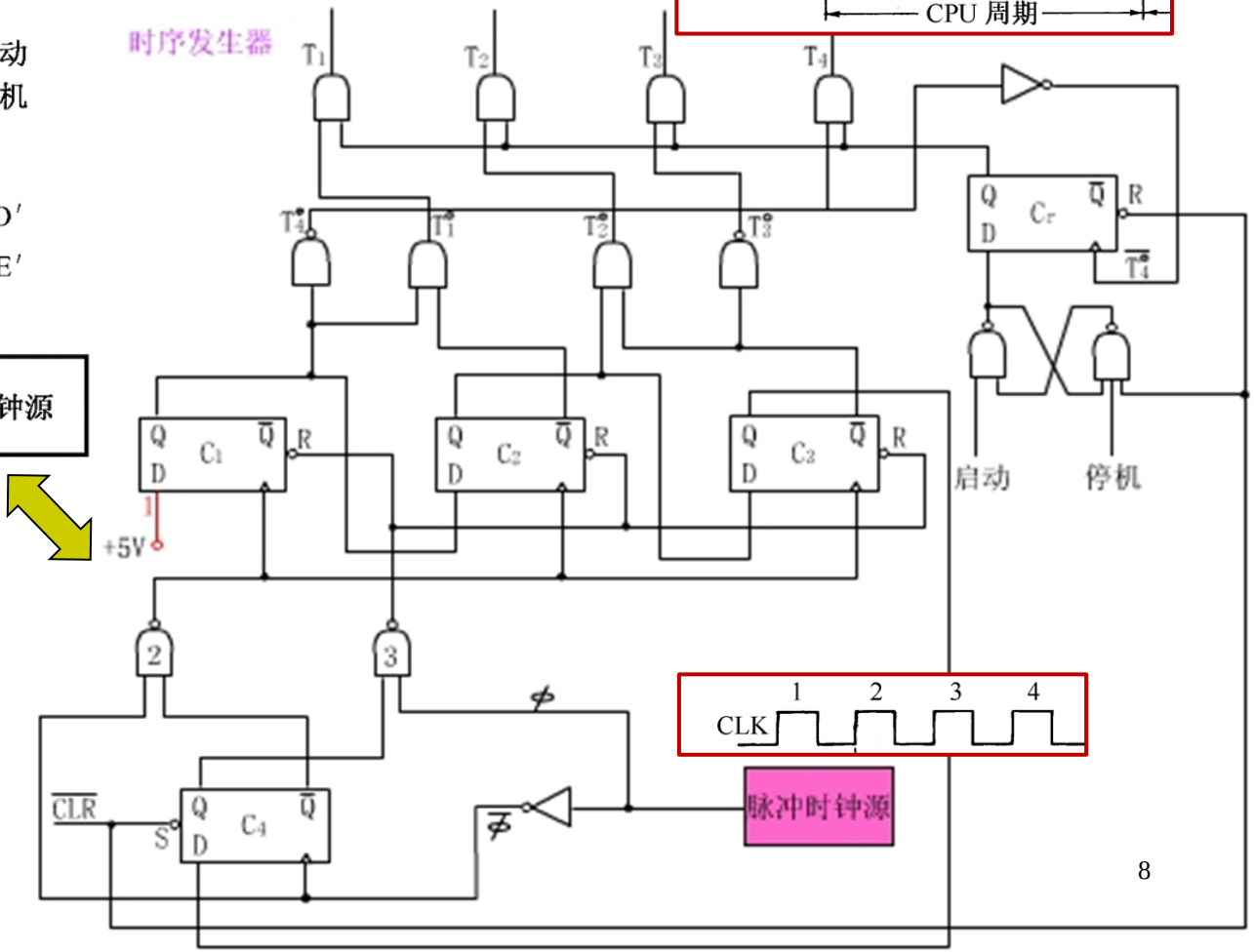
- 功能
  - 产生时序信号
    - 各型计算机产生时序电路不相同
    - 大、中型计算机的时序电路复杂，微型计算机的时序电路简单
- 构成
  - 时钟源
  - 环形脉冲发生器
  - 节拍脉冲和读写时序译码逻辑
  - 启停控制逻辑



时序信号产生器电路结构

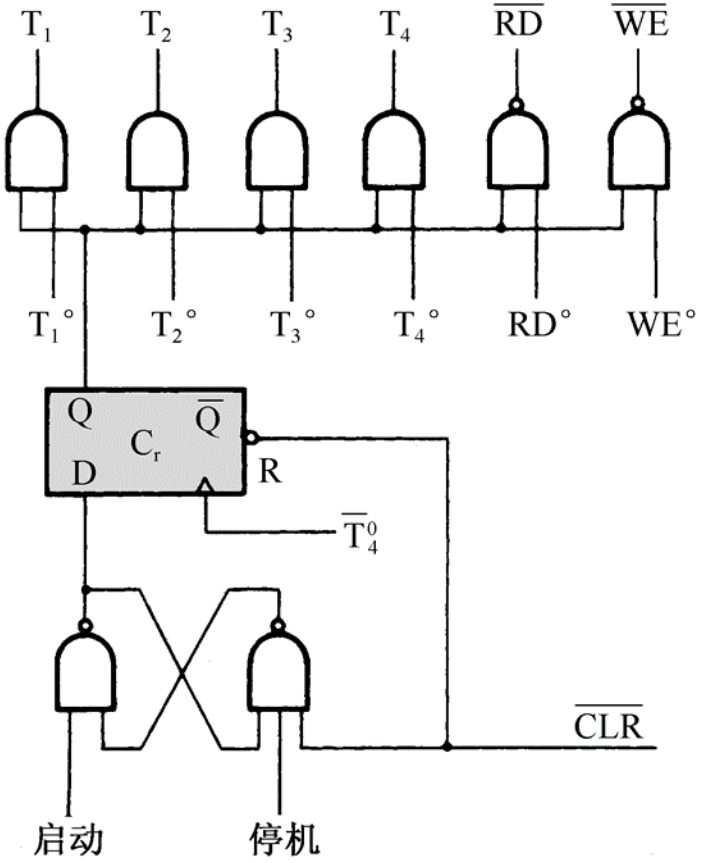






启停控制逻辑

- 启停控制逻辑
  - 启动、停机是随机的
  - 读/写时序信号也需要由启停逻辑加以控制
- 当运行触发器为1时
  - 打开时序电路
- 当运行触发器0时
  - 关闭时序产生器



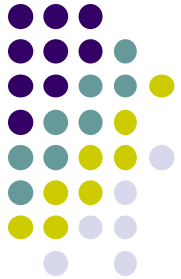
控制方式

- 机器指令所包含的CPU周期数反映了指令的复杂程度，不同CPU周期的操作信号的数目和出现的先后次序也不相同
- 控制方式
  - 控制不同操作序列时序信号的方法
  - 节拍脉冲—CPU周期
- 分为以下几种：
  - 同步控制方式
  - 异步控制方式
  - 联合控制方式

控制方式

- 同步控制（给定指令的机器周期和时钟周期数不变）
  - 方式1：完全统一的机器周期执行各种不同的指令
  - 方式2：采用不定长机器周期（大多数短/少数长）
  - 方式3：中央控制与局部控制的结合（大多数固定/少数局部）
- 异步控制方式
  - 每条指令需要多长时间就占多长时间，回答信号（Ready）
- 联合控制方式（同步与异步结合）
  - 方式1：大部分指令在固定的周期内完成，少数难以确定的操作采用异步方式
  - 方式2：机器周期的节拍脉冲固定，但是各指令的机器周期数不固定（微程序控制器采用）

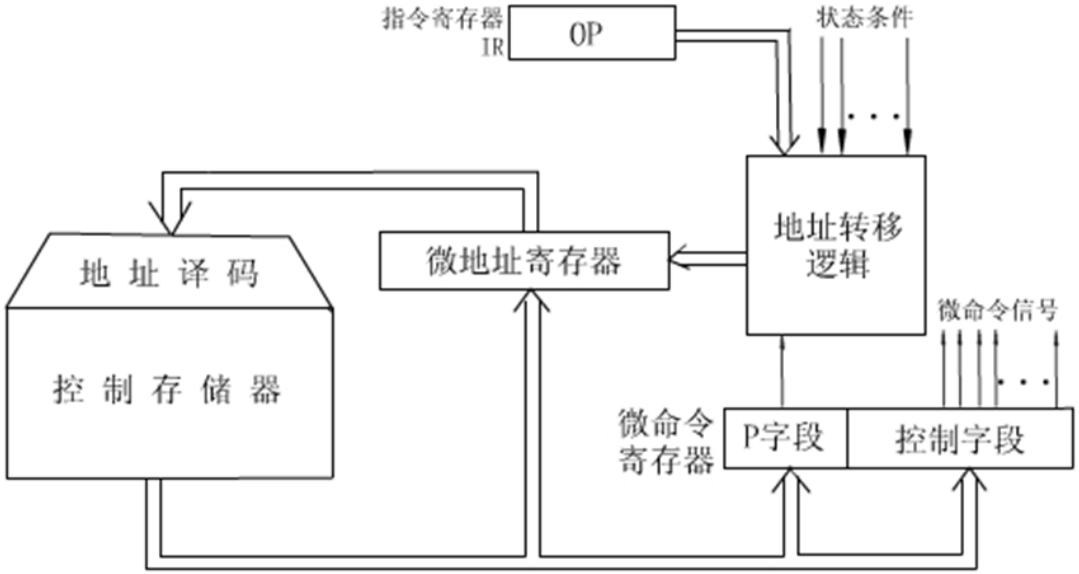
第五章 中央处理器



- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
- 微程序控制技术

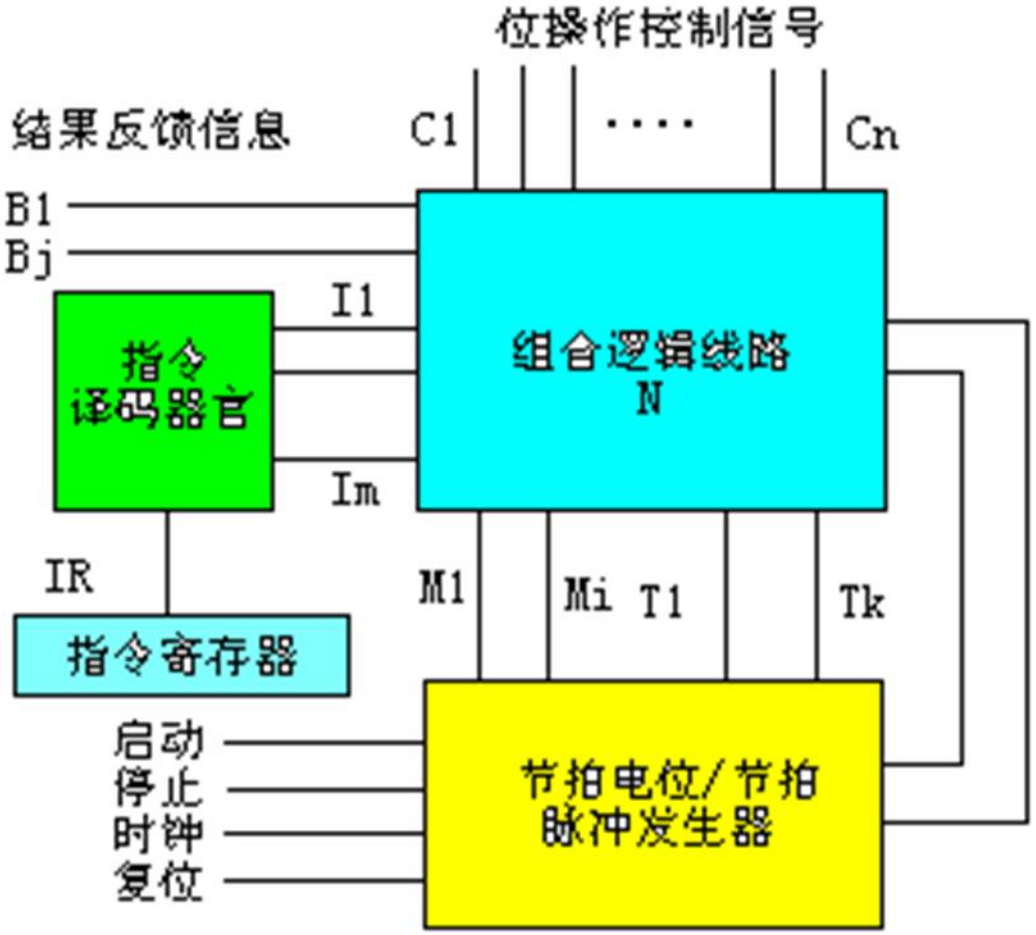
微程序控制器

- 基本思想与实现方法
  - 把操作控制信号编制成微指令，存放到控制存储器里，运行时，从控存中取出微指令，产生指令运行所需的操作控制信号
  - 微程序设计技术是用软件方法来设计硬件的技术



硬件布线控制器

- 基本思想
  - 通过逻辑电路直接连线而产生的，又称为组合逻辑控制方式
- 设计目标
  - 使用最少元件（复杂的树形网络）
  - 速度最高



微操作控制信号产生

- 微程序控制器
  - 微操作控制信号由微指令产生，并且可以重复使用
  - 优点：灵活、设计方便
- 硬件布线控制器
  - 某一微操作控制信号由布尔代数表达式描述的输出函数产生
  - 树形网络结构复杂，设计困难，但适合VLSI实现
  - 设计微操作控制信号流程
    - 根据所有机器指令流程图，寻找出产生同一个微操作信号的所有条件，并与适当的节拍电位和节拍脉冲组合
    - 从而写出其布尔代数表达式并进行简化
    - 用门电路或可编程器件来实现

硬件布线控制信号原理示例

- 示例，MIPS指令系统简化版本（六条指令）
  - addu rd, rs, rt
  - subu rd, rs, rt
  - ori rt, rs, imm16
  - sw rt, imm16(rs)
  - lw rt, imm16(rs)
  - beq rs, rt, imm16
- 列出指令编码（opcode/func）与控制信号关系

func	100000	100010	/			
opcode(op)	000000	000000	001101	100011	101011	000100
	add	sub	ori	lw	sw	beq
RegDst	1	1	0	0	x	x
ALUSrc	0	0	1	1	1	0
MemtoReg	0	0	0	1	x	x
RegWr	1	1	1	1	0	0
MemWr	0	0	0	0	1	0
nPC_sel	0	0	0	0	0	1
ExtOp	x	x	0	1	1	x
ALUctr<1:0>	00(ADD)	01(SUB)	10(OR)	00(ADD)	00(ADD)	01(SUB)

硬件布线控制信号原理示例

- 列出微操作控制信号逻辑表达式（以RegDst为例）

func	100000	100010	/			
opcode (op)	000000	000000	001101	100011	101011	000100
	add	sub	ori	lw	sw	beq
RegDst	1	1	0	0	x	x

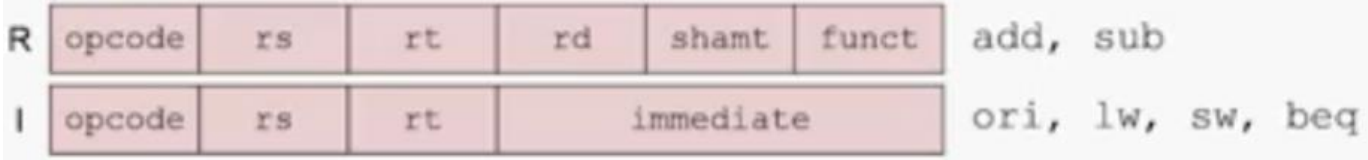
- 写出布尔代数表达式并化简（忽略T节拍信号）

RegDst = add + sub

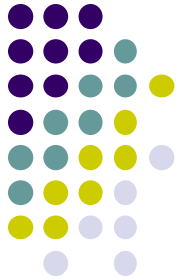
add =rtype · func5 ·~func4 ·~func3 ·~func2 ·~func1 ·~func0

sub =rtype · func5 ·~func4 ·~func3 ·~func2 · func1 ·~func0

rtype =~op5 ·~op4 ·~op3 ·~op2 ·~op1 ·~op0



第五章 中央处理器



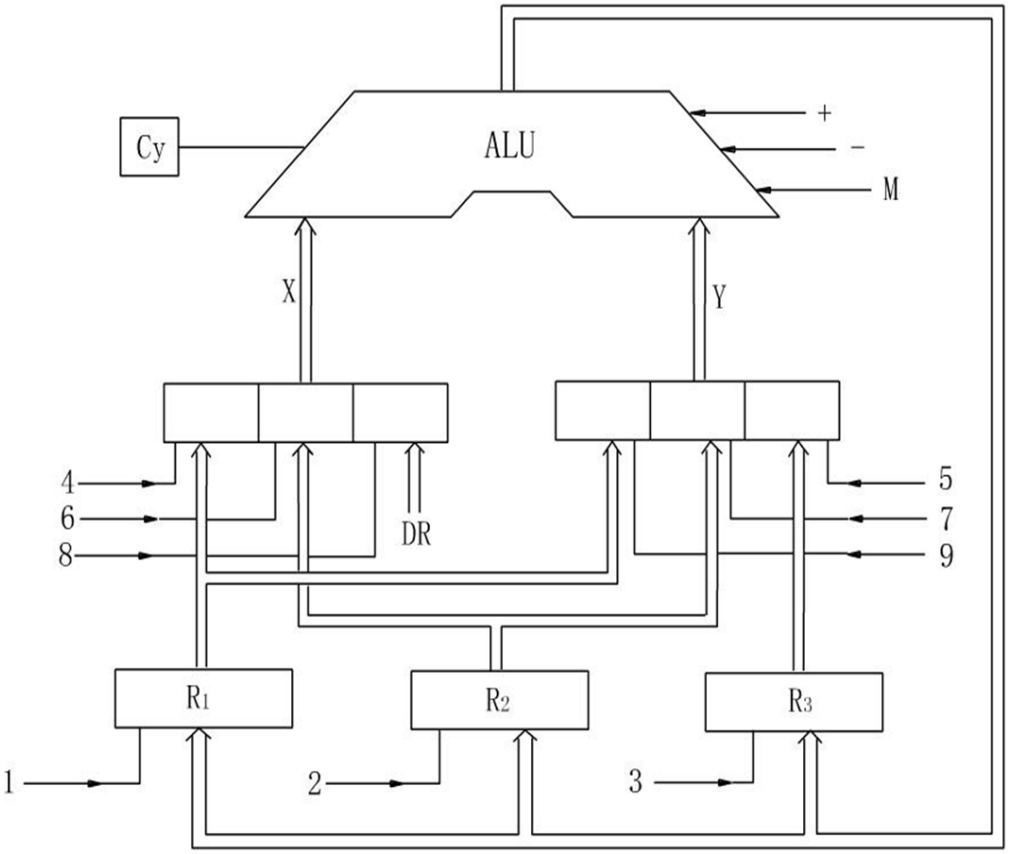
- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
  - 微命令/微操作/微指令/微程序
  - 微程序控制原理与示例
  - 机器指令、微指令、CPU周期
- 微程序控制技术

微命令和微操作

- 微命令
  - 控制部件向执行部件发出的各种控制命令
  - 构成控制序列的最小单位
  - 例如：打开或关闭某个控制门的电位信号、某个寄存器的打入脉冲等
- 微操作：微命令的操作过程
  - 微命令和微操作是一一对应的
  - 关系：微命令是微操作的控制信号，微操作是微命令的操作过程
  - 微操作是执行部件中最基本的操作

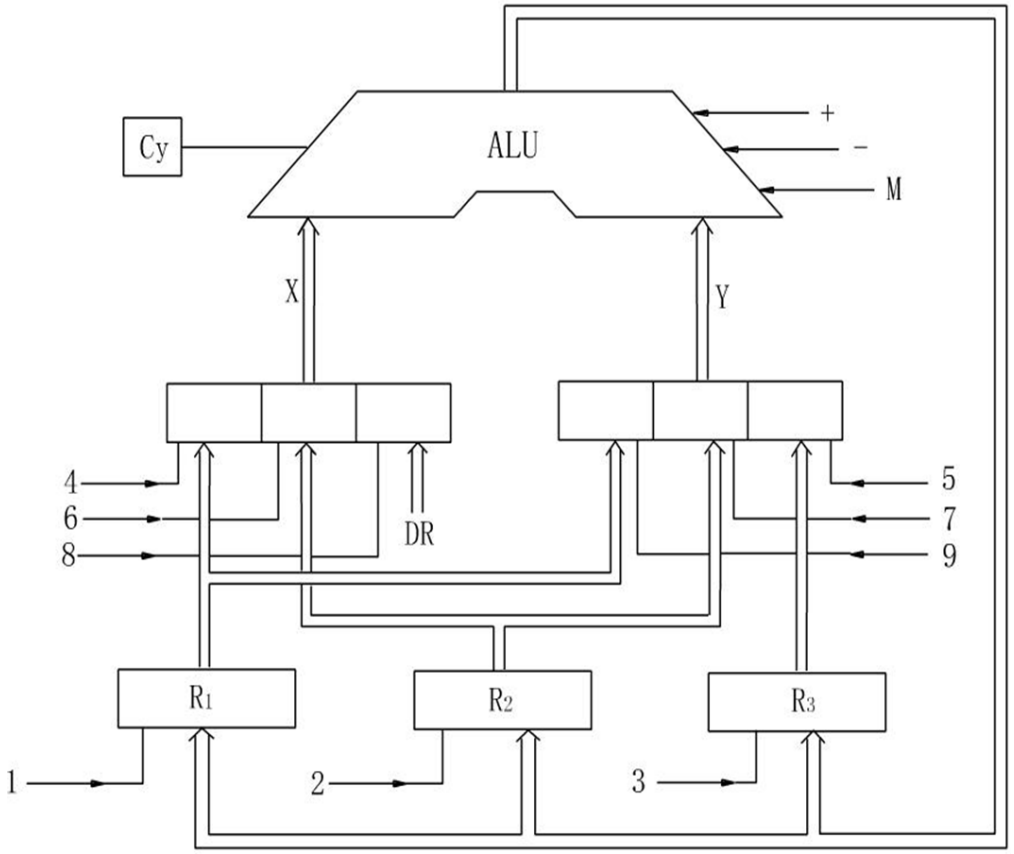
微操作相容、互斥性质

- 由于数据通路的结构关系，微操作可分为：
  - 互斥微操作
    - 不能同时或不能在同一个CPU周期内并行执行的微操作
  - 相容微操作
    - 是指能够同时或在同一个CPU周期内并行执行的微操作



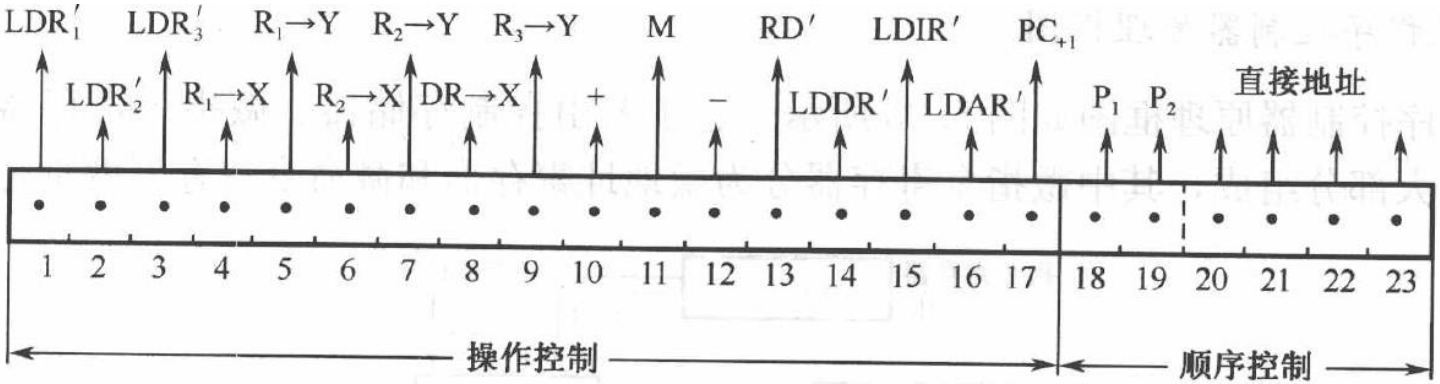
微操作相容、互斥性质

- 互斥微操作
  - ALU控制信号
    - +、-、M（传送）
  - ALU的X/Y输入端
    - X：4、6、8
- 相容微操作
  - 寄存器R1~R3
    - 1、2、3
  - X端与Y端
    - 4、7（任意组合）



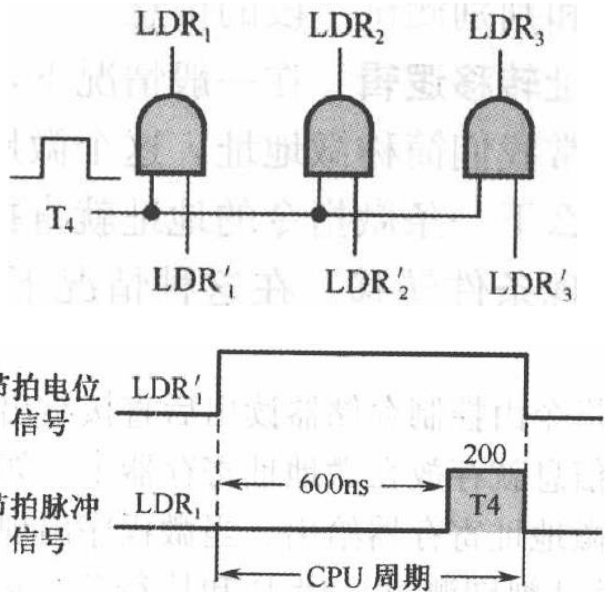
微指令结构

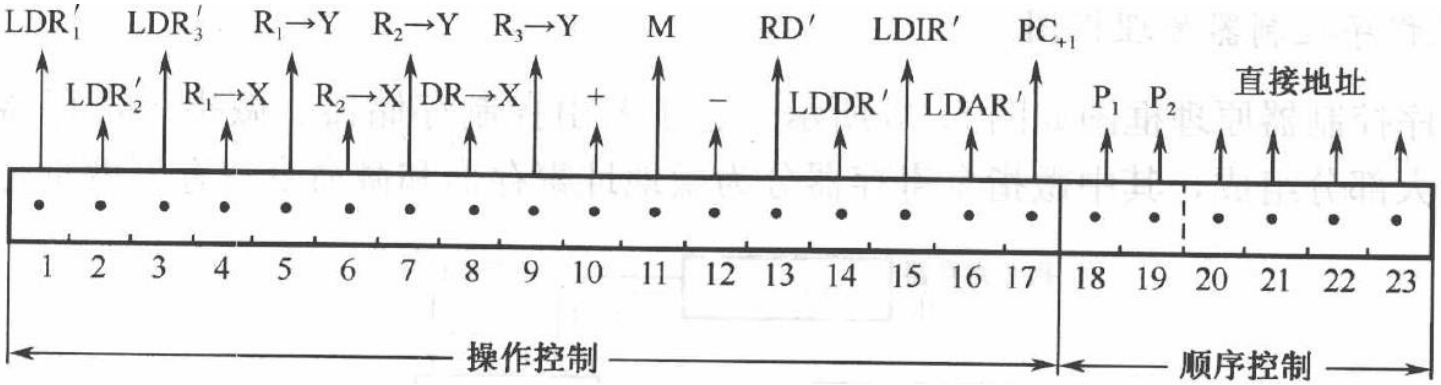
- 微指令
  - 一个CPU周期可执行多条微操作（相容微操作）
  - 这些微操作控制信息，存储在控制存储器里，就是微指令
  - 微命令的组合，微指令存储在控制器的控制存储器
- 组成
  - 操作控制字段（1~17位）
  - 顺序控制字段（18~23位）



微指令——操作/顺序控制字段

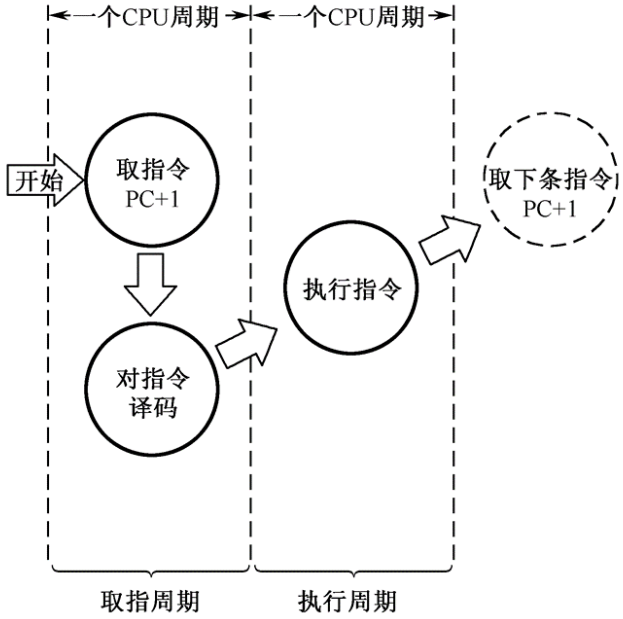
- 操作控制字段，又称微操作码字段
  - 产生各个微操作控制信号
    - 某位为1，表明发微指令
    - 微指令发出的控制信号都是节拍电位信号，持续时间为一个CPU周期
    - 引入节拍脉冲信号作时间控制
- 顺序控制字段，又称微地址码字段
  - 控制产生下一条要执行的微指令地址





微程序

- 机器指令包含多个CPU周期
  - 一个CPU周期微操作对应一条微指令
  - 一条机器指令包含多条微指令
- 微程序
  - 一系列微指令的有序集合
  - 一段微程序对应一条机器指令

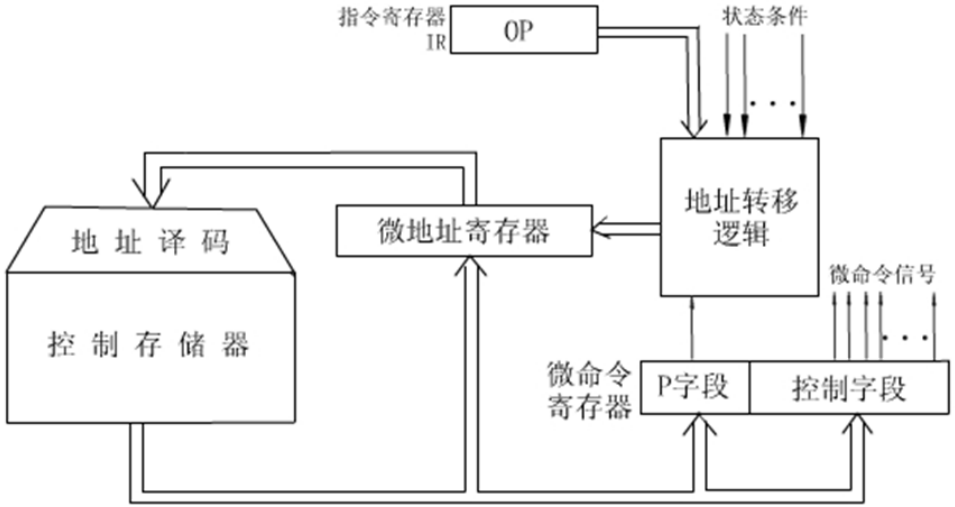


第五章 中央处理器

- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
  - 微命令/微操作/微指令/微程序
  - 微程序控制原理与示例
  - 机器指令、微指令、CPU周期
- 微程序控制技术

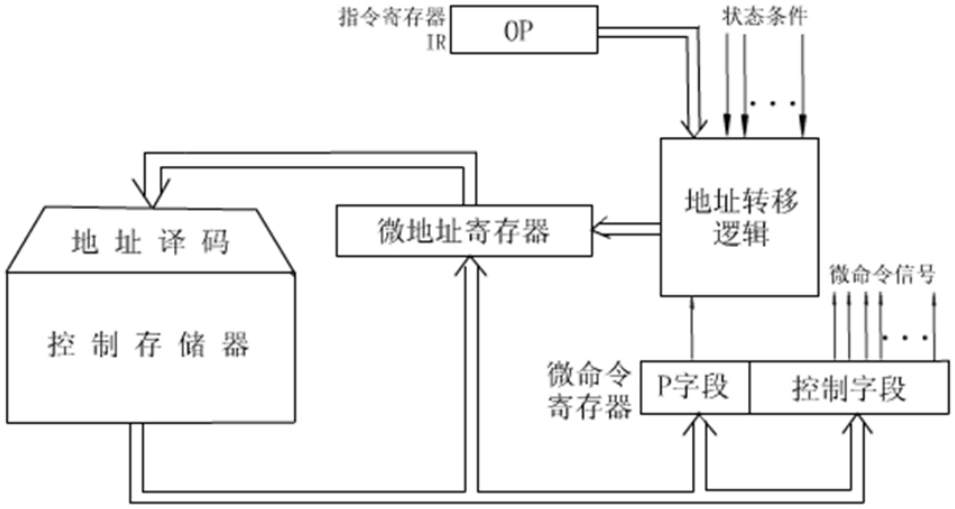
微程序控制原理框图

- 控制存储器(μCM)
  - 微程序控制器的核心部件，用来存放微程序
  - 性能(容量、速度、可靠性等)与计算机性能密切相关
- 微指令寄存器(μIR)
  - 用来存放从μCM取出的正在执行的微指令
  - 位数同微指令字长相等



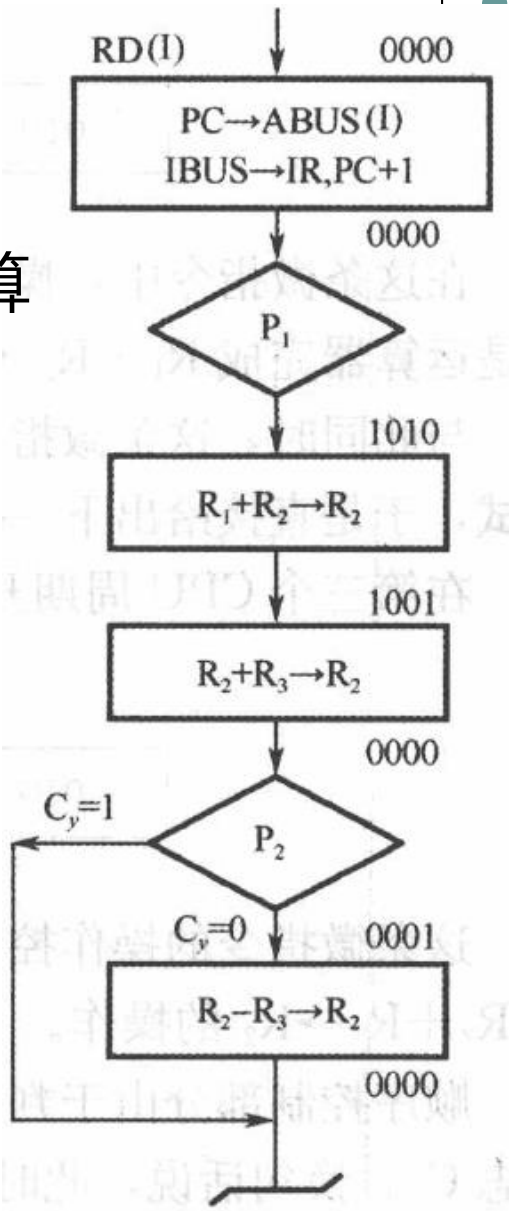
微程序控制原理框图

- 微地址形成部件
  - 用来产生初始微地址和后继微地址
  - 保证微指令的连续执行
- 微地址寄存器(μMAR)
  - 接受微地址形成部件送来的微地址
  - 为下一步从μCM中读取微指令作准备



微程序控制示例

- 十进制加法指令（一条指令）
  - 功能：用BCD码完成十进制加法运算
  - 流程框图（R3=6）
    - 取指、译码、正常加法
    - 加6修正
    - 若产生进位
      - 保持不变
    - 若未产生进位
      - 退回，减6



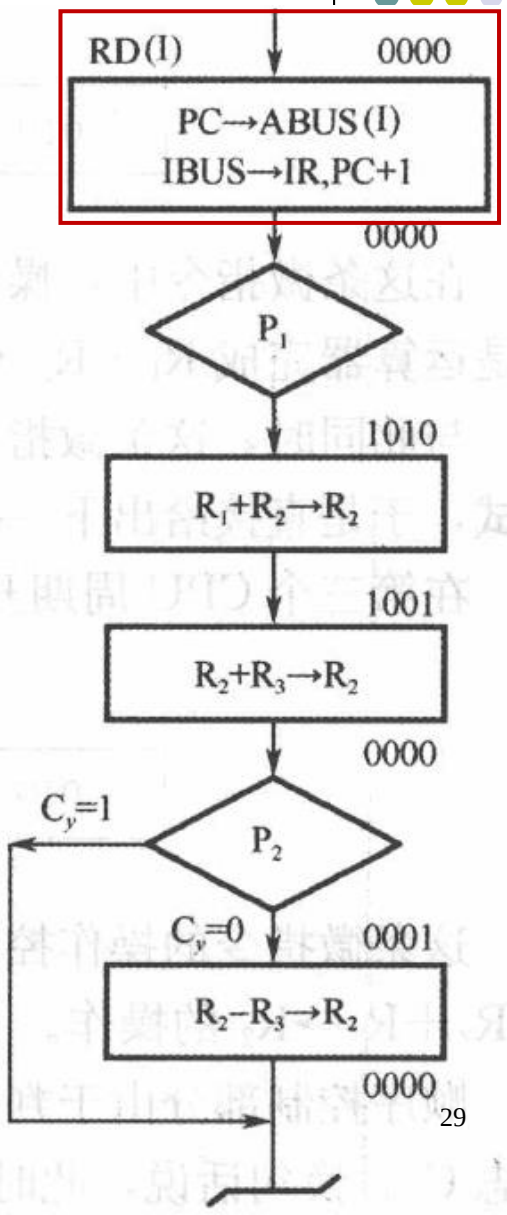
微程序控制示例

- 取指令——微指令编码

000	000	000	000	11110	10	0000

- 第13位：PC→ABUS（I）
- 第14位：RD（I），将指令放入IBUS
- 第15位：LDIR，将指令放入IR
- 第16位：PC+1

- 第18位，P1测试，用IR寄存器OP段作为地址



微程序控制示例

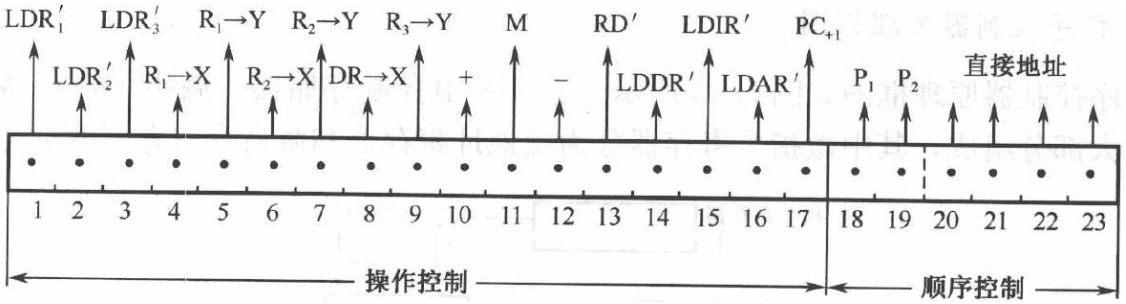
- 加法指令——微指令编码

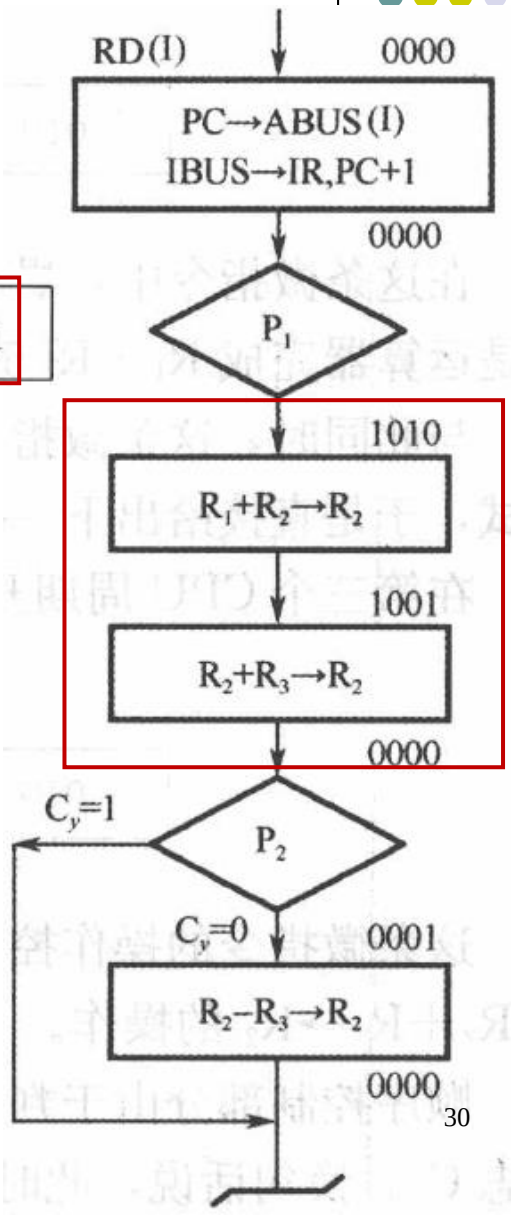
010	100	100	100	00000	00	1001

- R1+R2→R2，下一条微地址1001

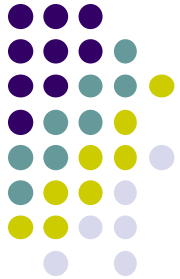
010	001	001	100	00000	01	0000

- R2+R3→R2
- LDR2、R2→X、R3→Y、+
- 执行P2测试，测试进位标志Cy





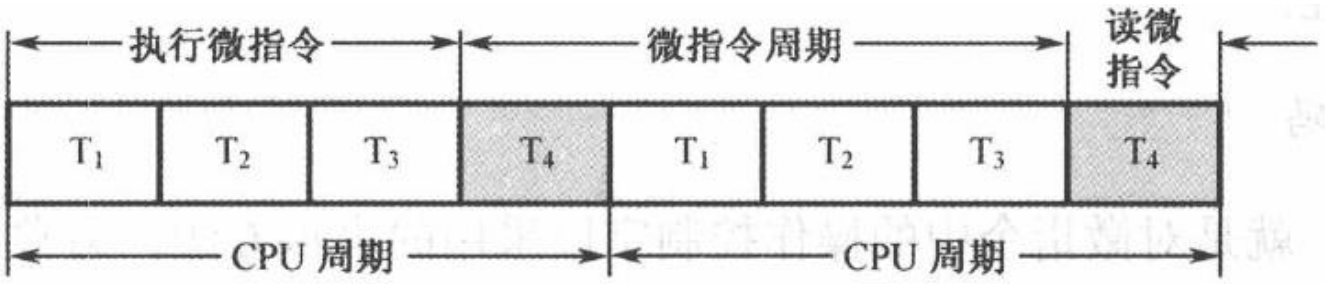
第五章 中央处理器



- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
  - 微命令/微操作/微指令/微程序
  - 微程序控制原理与示例
  - 机器指令、微指令、CPU周期
- 微程序控制技术

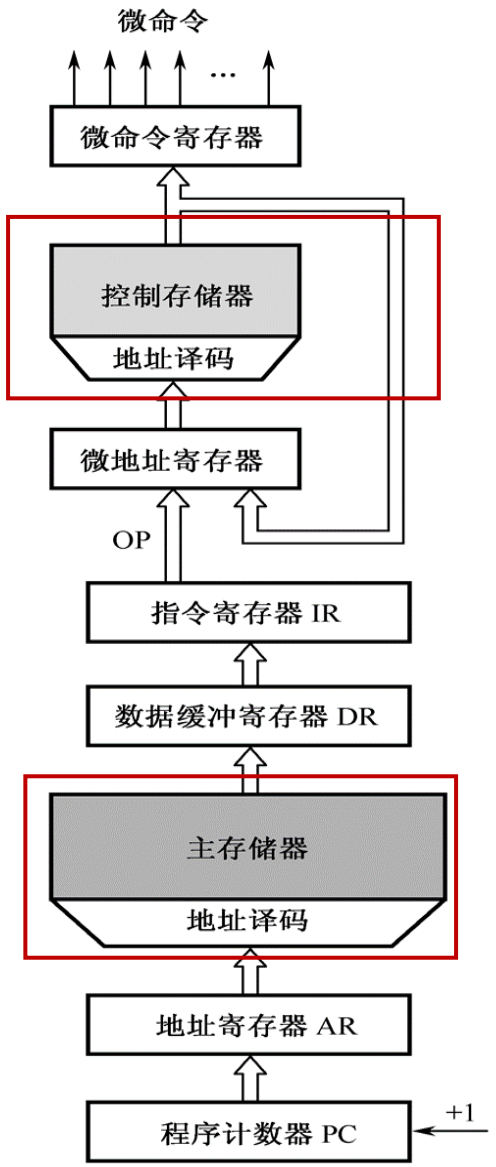
CPU周期与微指令周期关系

- 微指令周期
  - 微指令周期：T4（读），T1~T3（执行）
  - CPU周期：T1~T4
  - 微指令周期设计成等于CPU周期长度
  - 保障机器控制信号的同步

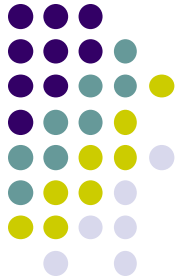


机器指令与微指令的关系

- 机器指令对应微程序
  - 微程序由一系列机器指令组成
  - 例如，BCD加法
- 存储位置与地址
  - 机器指令存储在主存储器，对应PC
  - 微指令存储在控制存储器，对应微地址
- 作用
  - 机器指令明确指令功能
  - 微指令明确微操作信号

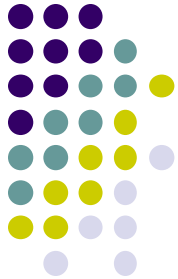


第五章 中央处理器



- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
- 微程序控制技术
  - 微指令结构
  - 微地址形成
  - 微指令格式

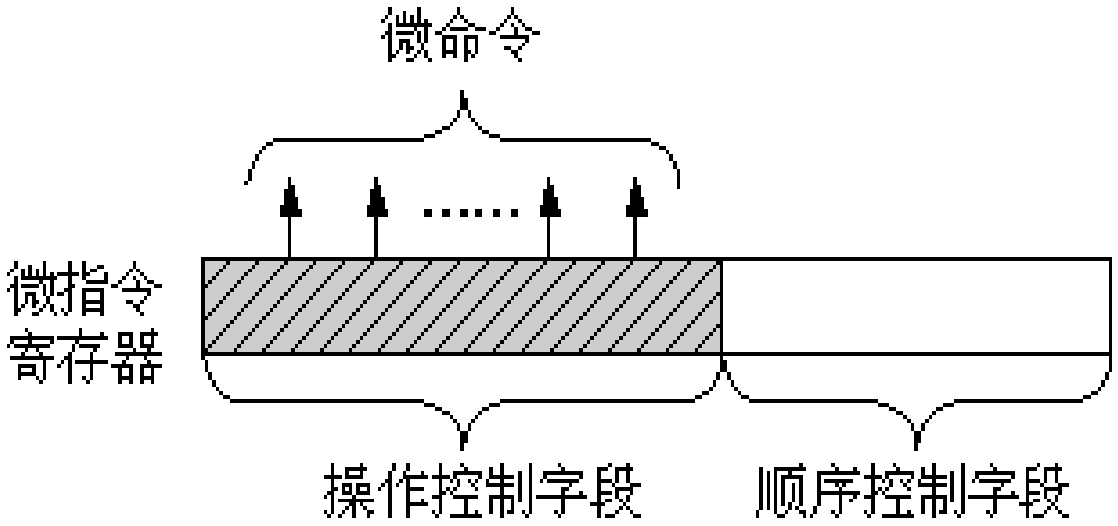
微程序设计——微指令结构



- 微指令结构是微程序设计的关键
- 设计微指令应当追求的目标
  - 有利于缩短微指令的长度
  - 有利于缩小控制存储器的容量
  - 有利于提高微程序的执行速度
  - 有利于对微指令的修改
  - 有利于提高微程序设计的灵活性
- 微命令编码
  - 直接表示法
  - 编码表示法
  - 混合表示法

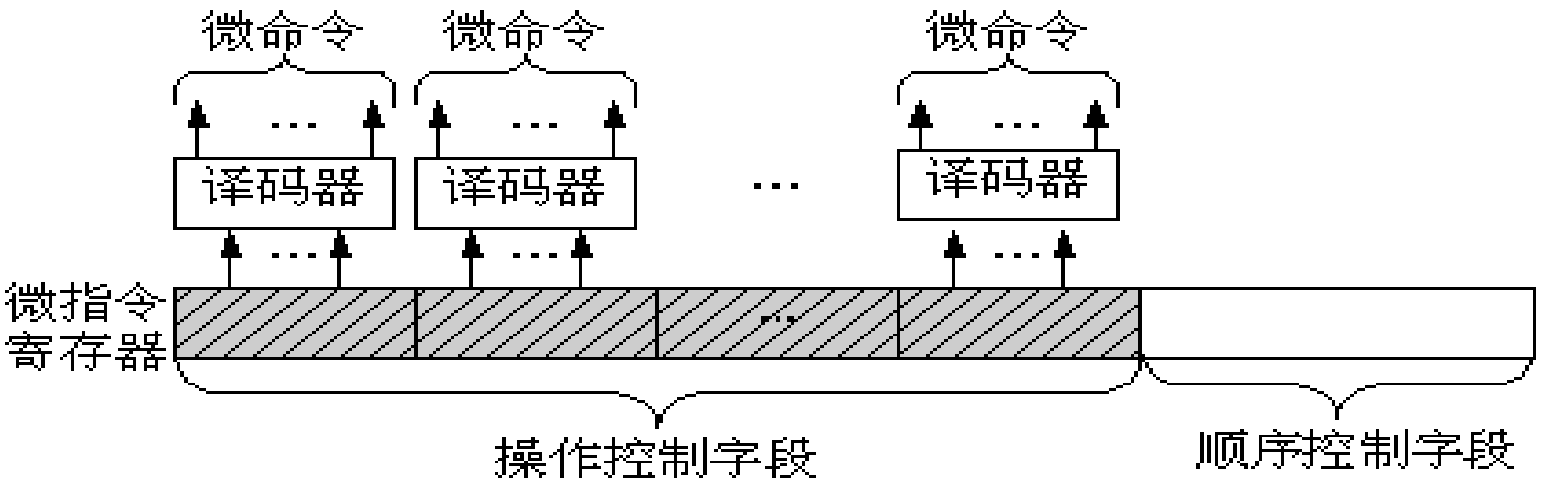
微命令编码——直接表示法

- 直接表示法
  - 操作控制字段中的各位分别可以直接控制计算机，不需要进行译码
  - 优点：简单直观，输出直接进行控制
  - 缺点：微指令字较长，控制存储器容量大



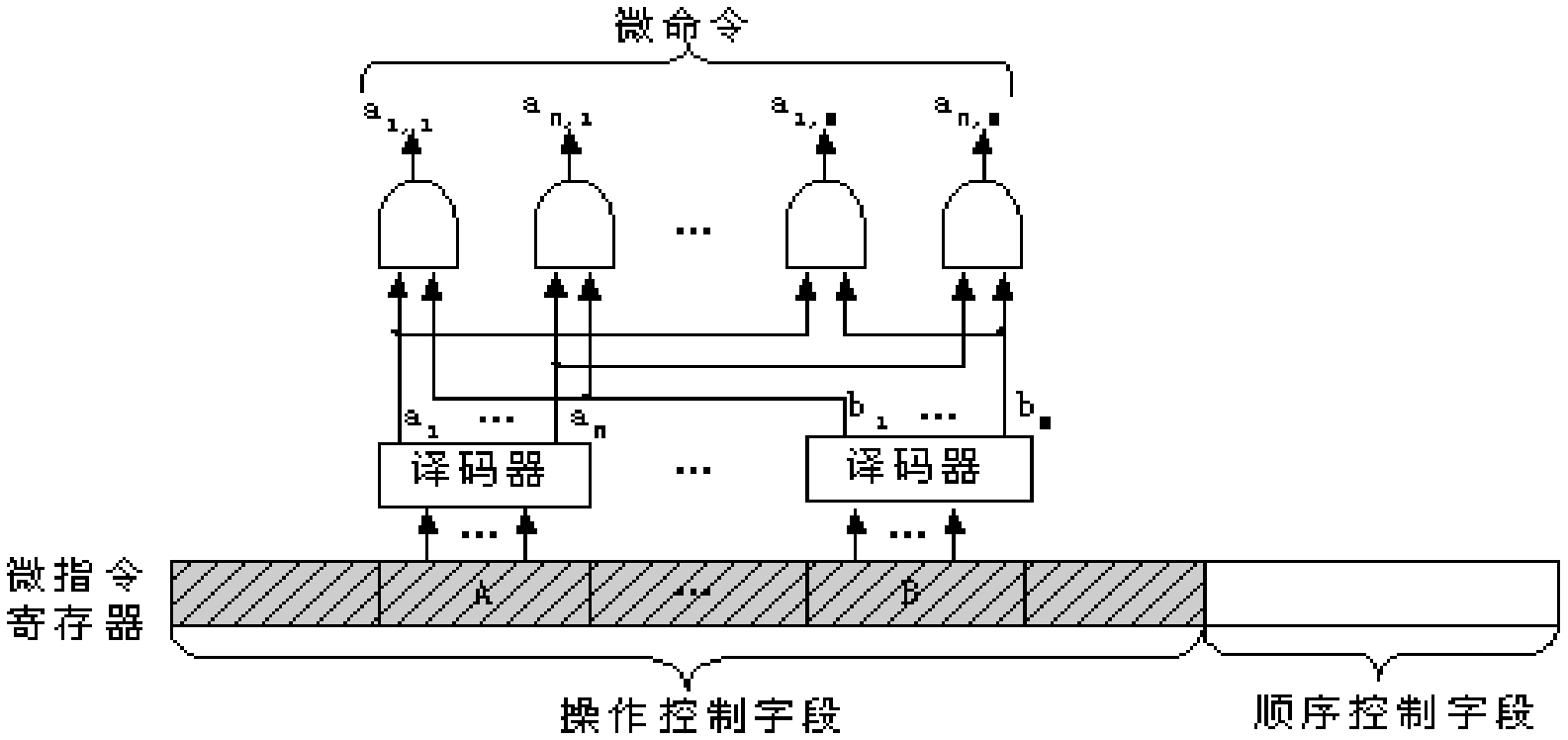
微命令编码——编码表示法

- 编码表示法
  - 将操作控制字段分为若干个小段（段内互斥），每段内采用最短编码法，段与段之间采用直接控制法
  - 优点：利用互斥性，缩短微指令字长
  - 缺点：额外译码电路，执行速度较慢

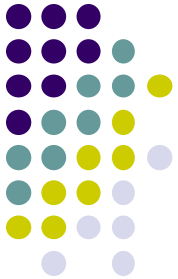


微命令编码——混合表示法

- 混合表示法
  - 结合直接表示法与编码表示法
  - 综合指令字长、灵活性、执行速度等方面要求



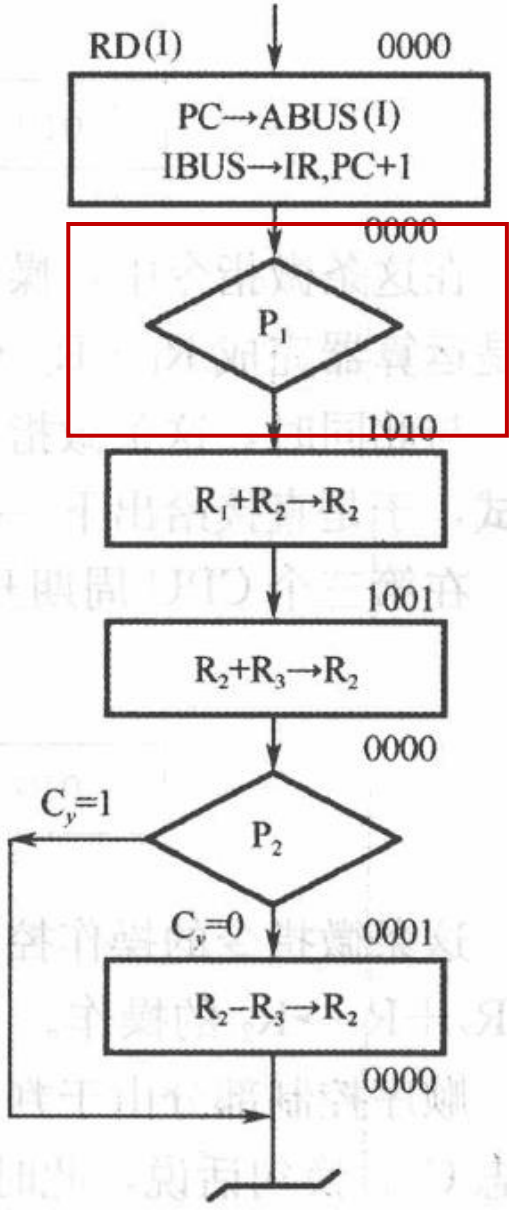
第五章 中央处理器



- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
- 微程序控制技术
  - 微指令结构
  - 微地址形成
  - 微指令格式

微地址形成方法

- 入口地址
  - 每条机器指令对应一段微程序，当公用的取指微程序从主存中取出机器指令
  - 通过P1测试，机器指令的操作码字段指出各段微程序的入口地址，这是一种多分支(或多路转移)的情况
- 机器指令操作码转换成初始微地址方式
  - 计数器方式
  - 多路转移方式

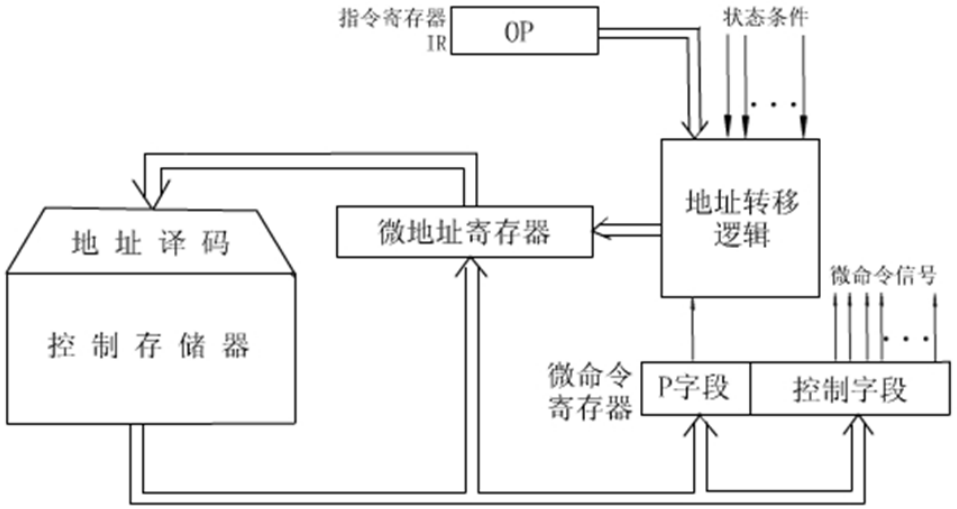


计数器方式

- 计数器方式
  - 微程序顺序执行时，其后继微地址就是现行微地址加上一个增量（通常为1）
  - 当微程序遇到转移或转子程序时，由微指令的转移地址段来形成转移微地址
  - 在微程序控制器中也有一个微程序计数器μPC，一般情况下都是将微地址寄存器μMAR作为μPC
- 性能分析
  - 优点：简单、易于掌握，编制微程序容易
  - 缺点：这种方式不能实现两路以上的并行微程序转移，不利于提高微程序的执行速度

多路转移方式

- 多路转移方式
  - 基本原理
    - 若不产生分支，顺序执行
    - 产生分支，根据状态条件，产生“备选”地址（IR OP段）
    - N位状态条件标志→2^N路转移
  - 性能分析：灵活性好、并行转移；需设计地址转移逻辑



多路转移逻辑设计例题

- [例2] 微地址寄存器有6位 (μA5-μA0)，当需要修改其内容时，可通过某一位触发器的强置端S将其置1。
- 现有三种情况：
  - 执行“取指”微指令后，微程序按IR的OP字段 (IR3-IR0) 进行16路分支；
  - 执行条件转移指令微程序时，按进位标志C的状态进行2路分支；
  - 执行控制台指令微程序时，按IR4，IR5的状态进行4路分支。
- 请按多路转移方法设计微地址转移逻辑。

[例2] 解：按所给设计条件，微程序有三种判别测试，分别为P1，P2，P3。

由于修改μA5-μA0内容具有很大灵活性，现分配如下：

(1)用P1和IR3-IR0修改μA3-μA0；

(2)用P2和C修改μA0；

(3)用P3和IR5，IR4修改μA5，μA4。

另外还要考虑时间因素T4 (假设CPU周期最后一个节拍脉冲)，故转移逻辑表达式如下：

$\mu A5=P3\cdot IR5\cdot T4$

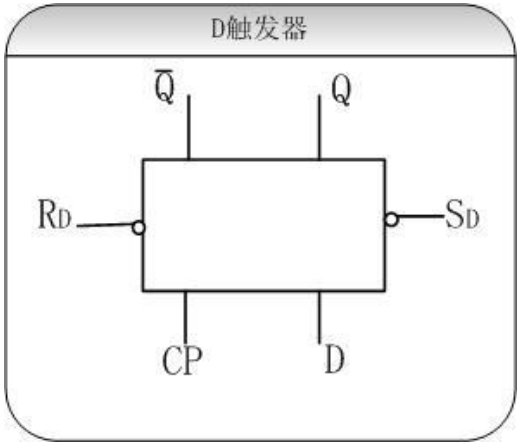
$\mu A4=P3\cdot IR4\cdot T4$

$\mu A3=P1\cdot IR3\cdot T4$

$\mu A2=P1\cdot IR2\cdot T4$

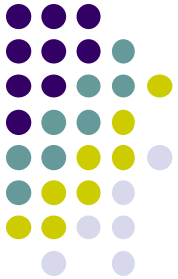
$\mu A1=P1\cdot IR1\cdot T4$

$\mu A0=P1\cdot IR0\cdot T4+P2\cdot C\cdot T4$



由于从触发器强置端修改，故前5个表达式可用“与非”门实现，最后一个用“与或非”门实现。

第五章 中央处理器

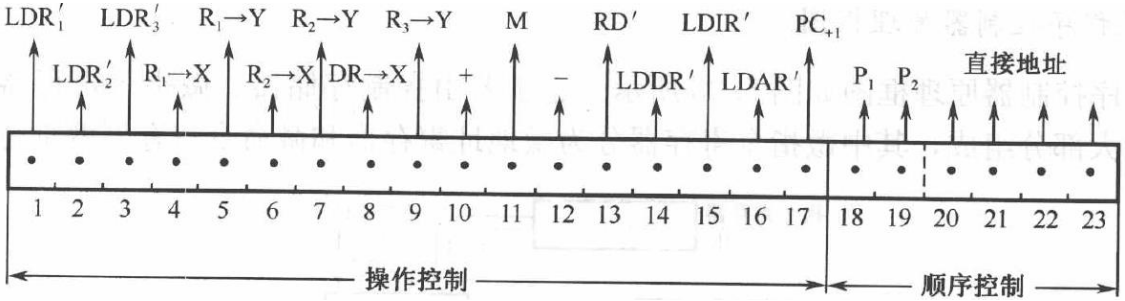


- 时序信号作用与时序信号产生器
- 微程序/硬件布线控制器简介
- 微程序控制原理
- 微程序控制技术
  - 微指令结构
  - 微地址形成
  - 微指令格式

水平型微指令

- 微指令格式分类
  - 水平型微指令
  - 垂直型微指令
- 水平型微指令
  - 水平型微指令是指一次能定义并能并行执行多个微命令的微指令

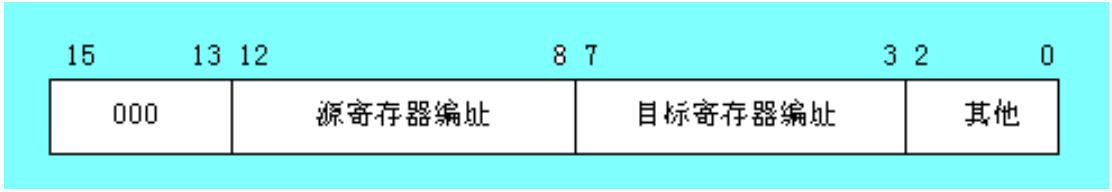
控制字段	判别测试字段	下地址字段



垂直型微指令

- 垂直型微指令
  - 采用编码方式
  - 设置微操作控制字段时，一次只能执行一到二个微命令的微指令

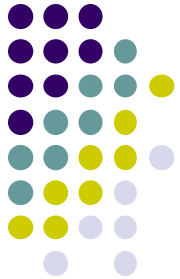
- 寄存器—寄存器传送型微指令



水平型/垂直型微指令比较

- 水平型微指令和垂直型微指令的比较
  - 并行能力
    - 水平型微指令并行操作能力强，效率高，灵活性强
    - 垂直型微指令则较差
  - 执行时间
    - 水平型微指令执行一条指令的时间短
    - 垂直型微指令执行时间长
  - 微指令字长
    - 水平型微指令：微指令字较长而微程序短的特点
    - 垂直型微指令则相反。
  - 掌握难度
    - 水平型微指令用户难以掌握
    - 垂直型微指令与机器指令比较相似，比较容易掌握。

动态微程序设计



- 静态微程序设计
  - 对应于一台计算机的机器指令只有一组微程序，这一组微程序设计好之后，一般无须改变而且也不好改变
- 动态微程序设计
  - 采用EPROM作为控制存储器，可以通过改变微指令和微程序来改变机器的指令系统

第五章作业

- 5-10，5-12，5-13