



# 计算机组成与系统结构

## 第五章 中央处理机（1）

吕昕晨

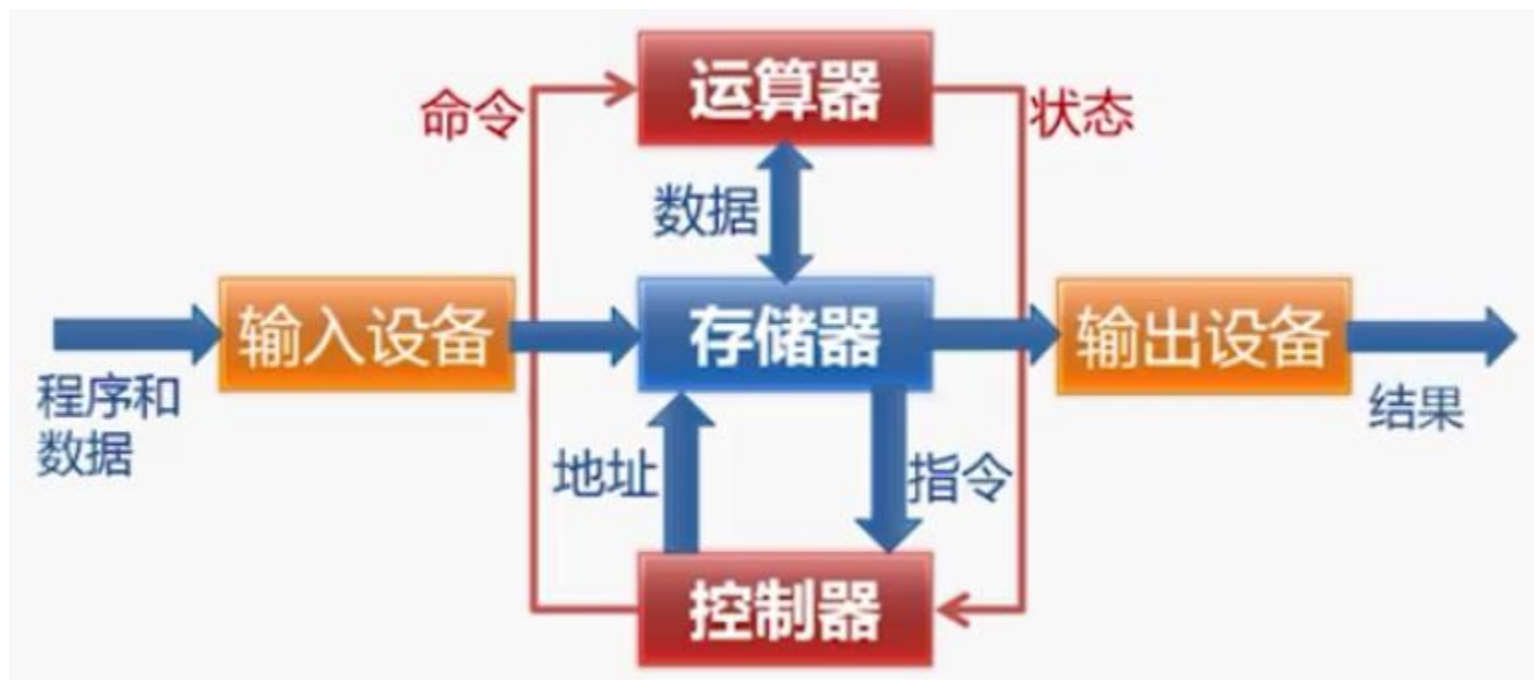
[lvxinchen@bupt.edu.cn](mailto:lvxinchen@bupt.edu.cn)

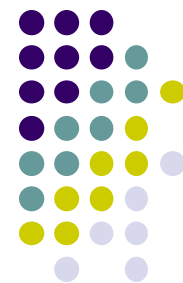
网络空间安全学院



# 回顾：冯·诺依曼结构

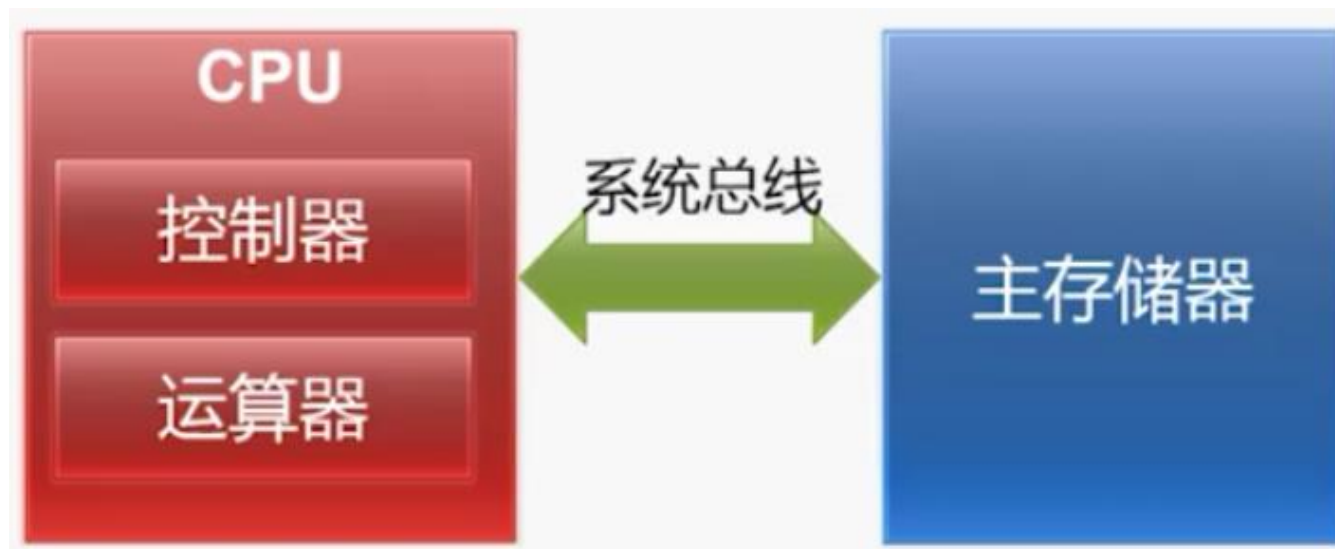
- 核心思想：存储程序，共享数据，顺序执行
  - 数据与程序均以二进制代码形式不加区别地存放在存储器中，存储位置由存储器地址制定
  - 计算机可自动从存储器中取出指令加以执行





# 冯·诺依曼结构核心

- 冯诺依曼结构
  - 运算器与控制器 ↔
  - 存储器
- 现代计算机
  - CPU（中央处理器）
  - 主存储器（内存）

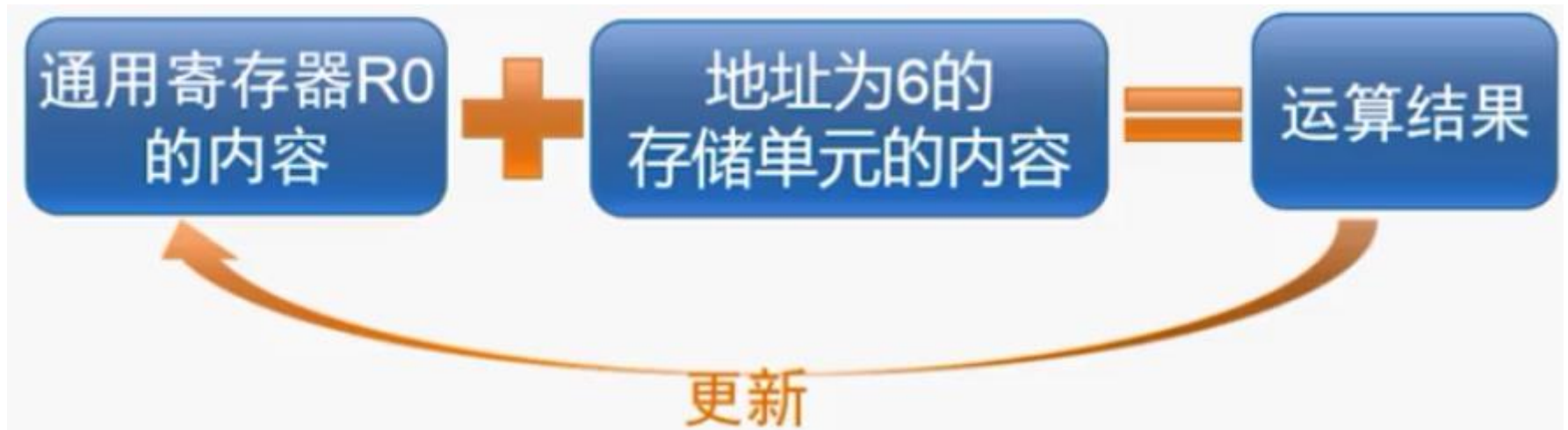


- 功能：CPU可自动从存储器中取出指令加以执行



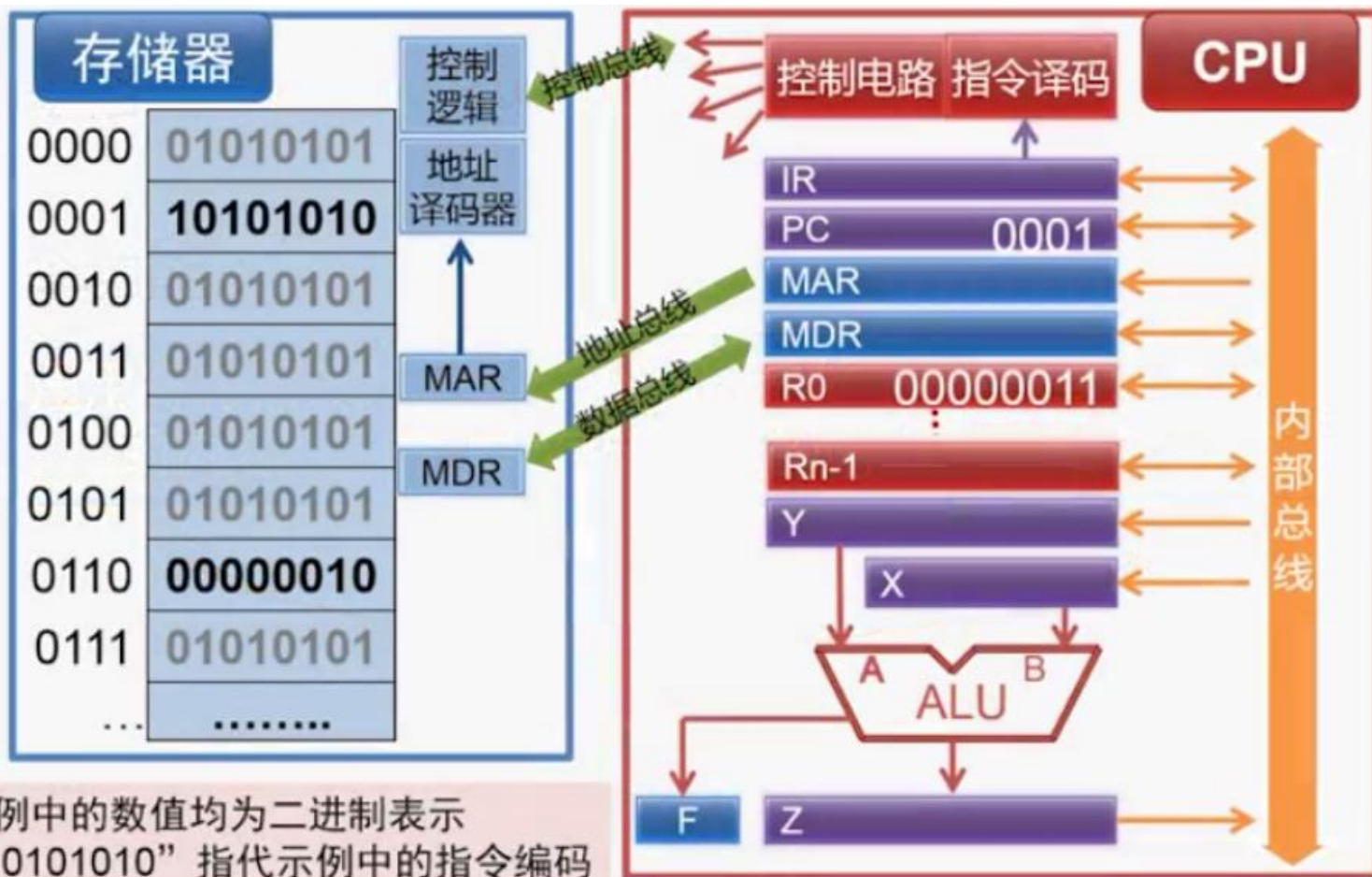
# 指令示例——第四章内容

- 指令示例
  - 加法指令：ADD R0, [6]
- 指令功能





# 指令执行流程——第五章内容

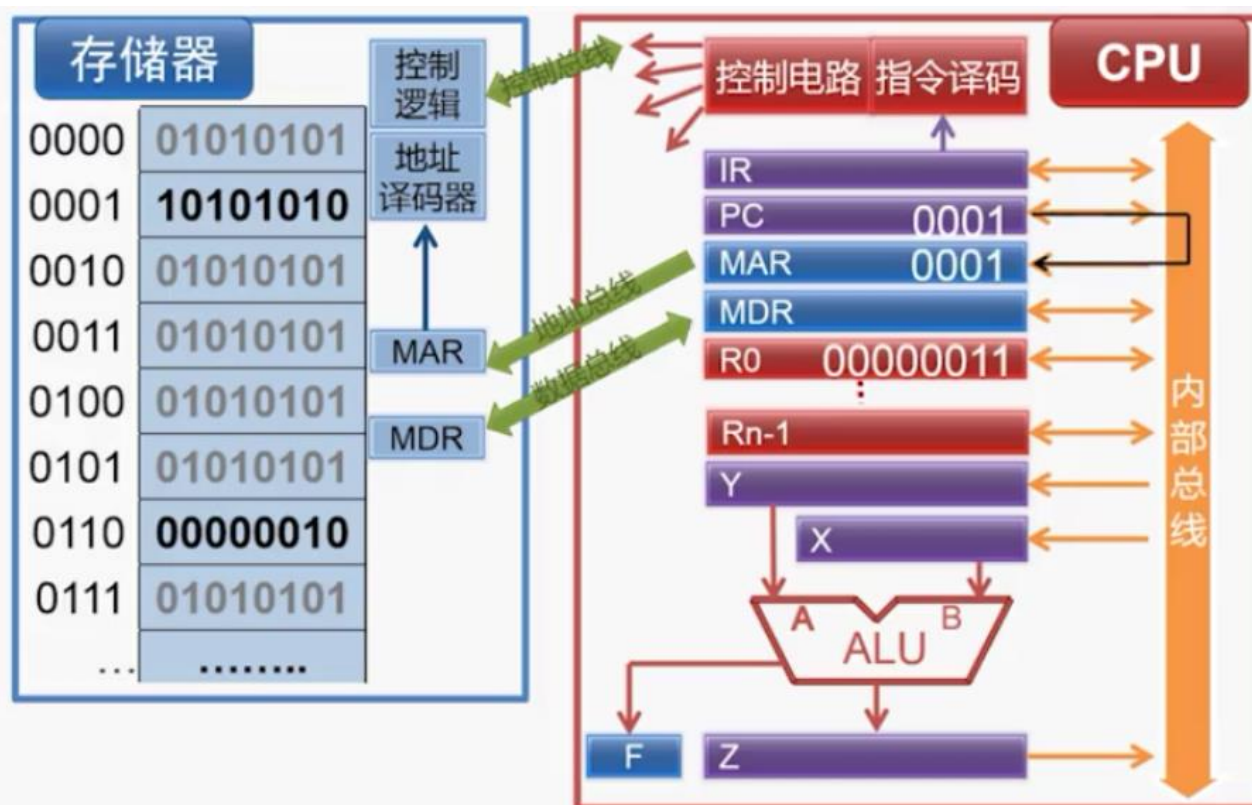


注1: 本例中的数值均为二进制表示  
注2: “10101010” 指代示例中的指令编码

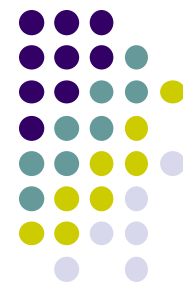


# 指令执行流程——取指 (1)

- 控制器将指令地址 (PC) 送往存储器
- 存储器按给定地址读出指令内容，送回控制器
- 更新下一条指令地址：PC自加

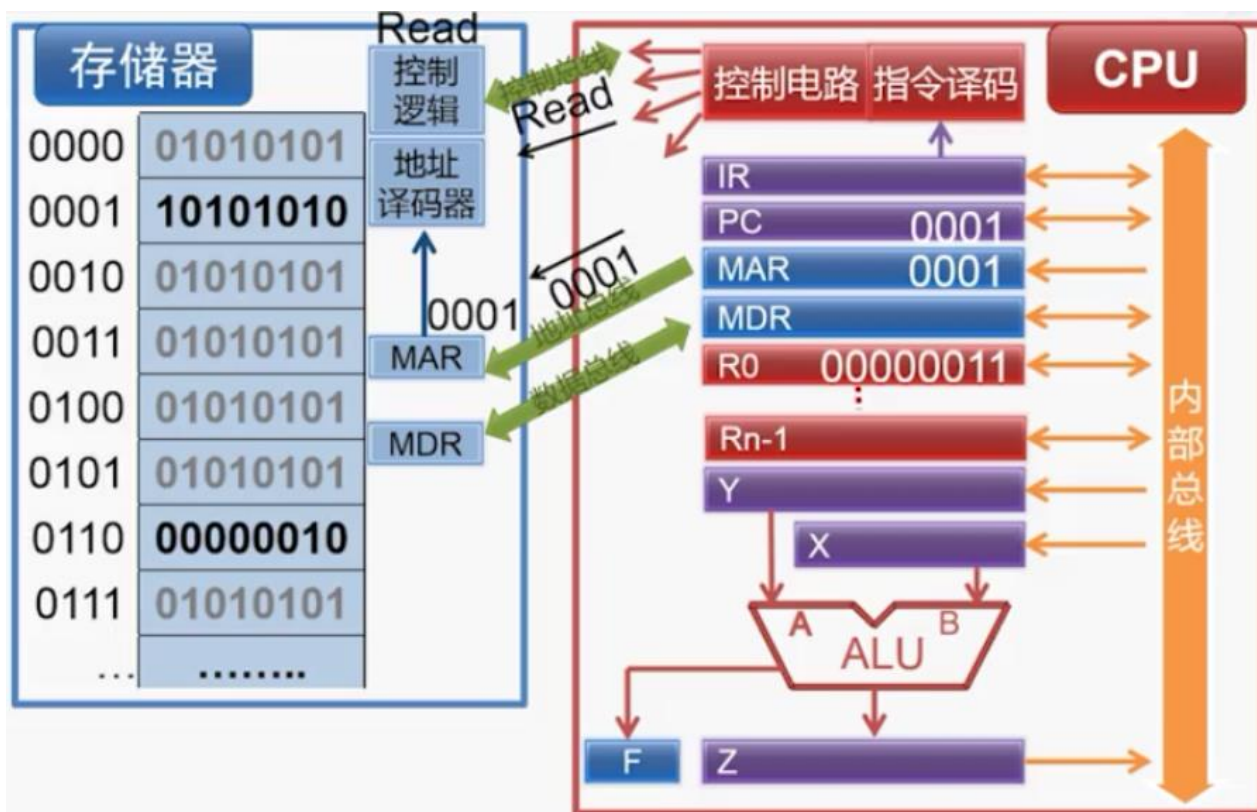






## 指令执行流程——取指 (2)

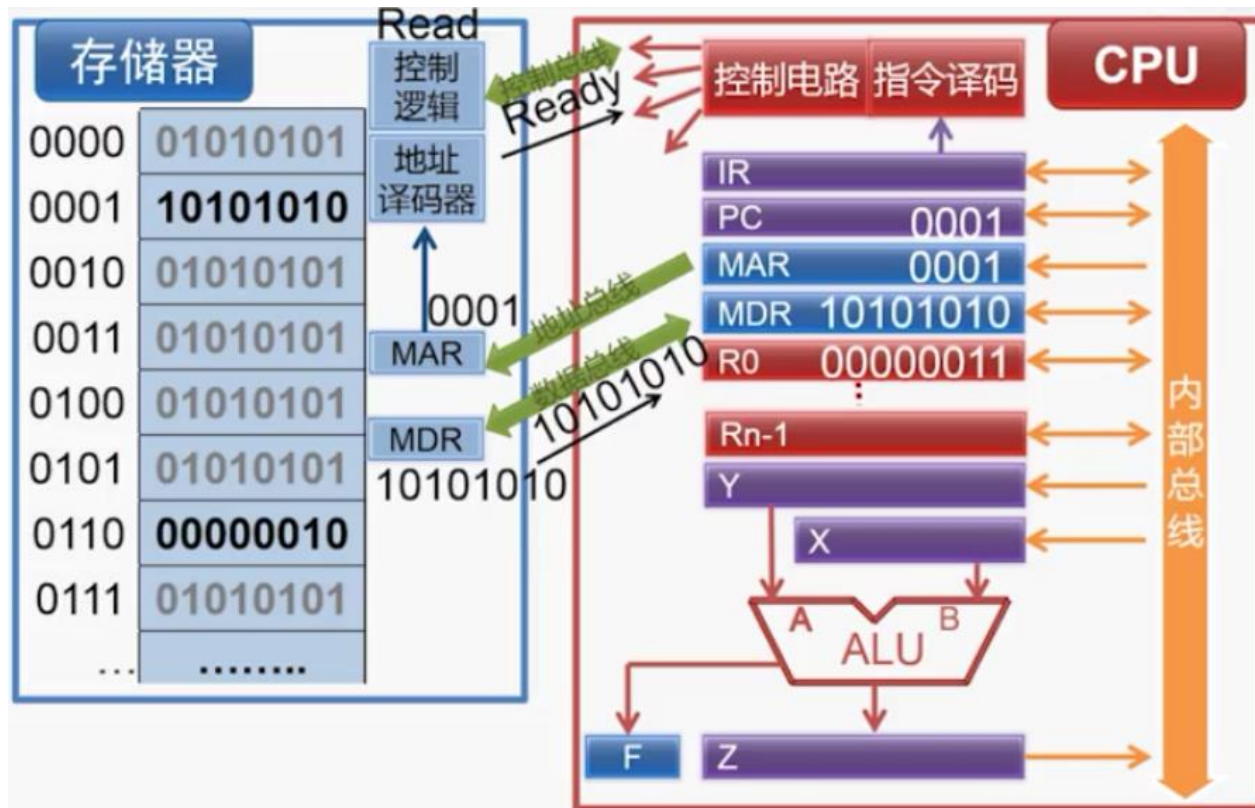
- 控制器将指令地址 (PC) 送往存储器
- 存储器按给定地址读出指令内容，送回控制器
- 更新下一条指令地址：PC自加





# 指令执行流程——取指 (3)

- 控制器将指令地址 (PC) 送往存储器
- 存储器按给定地址读出指令内容，送回控制器
- 更新下一条指令地址：PC自加

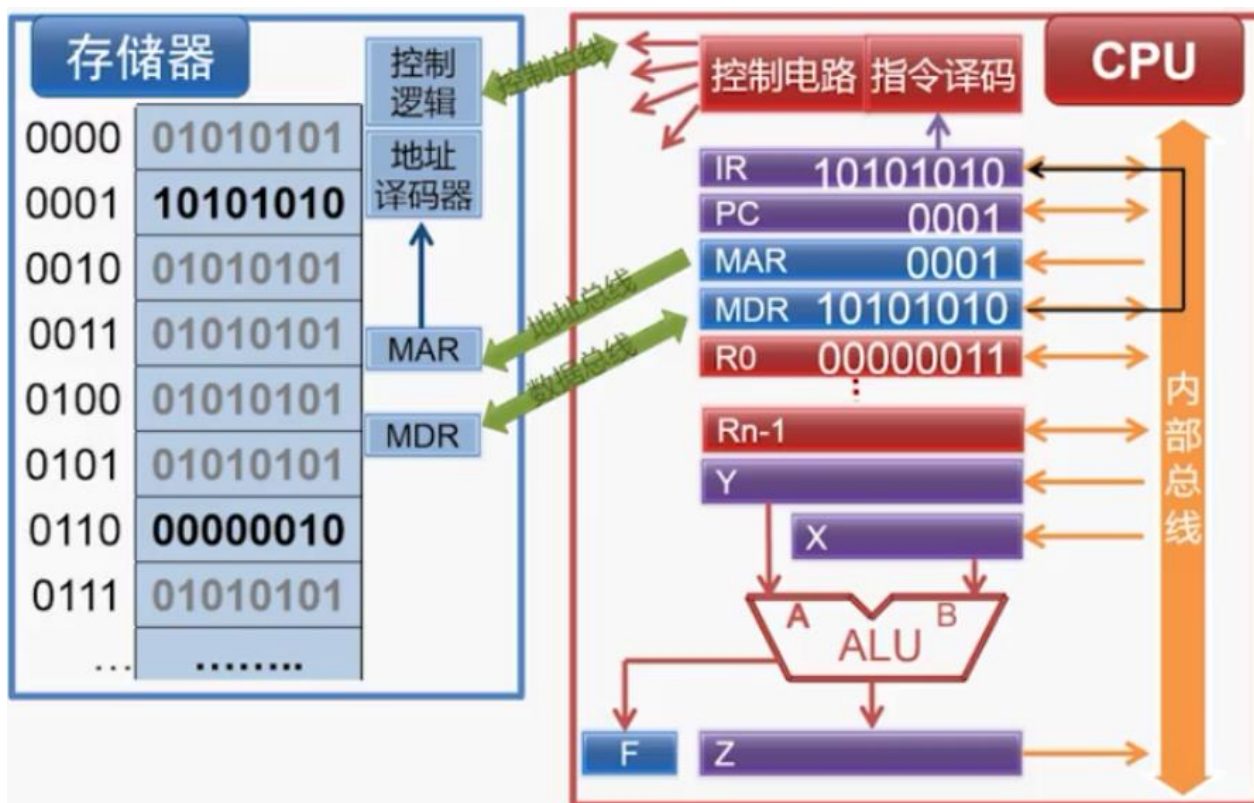


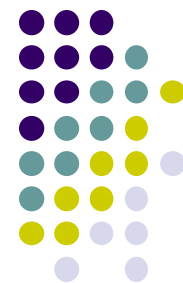




# 指令执行流程——取指（4）

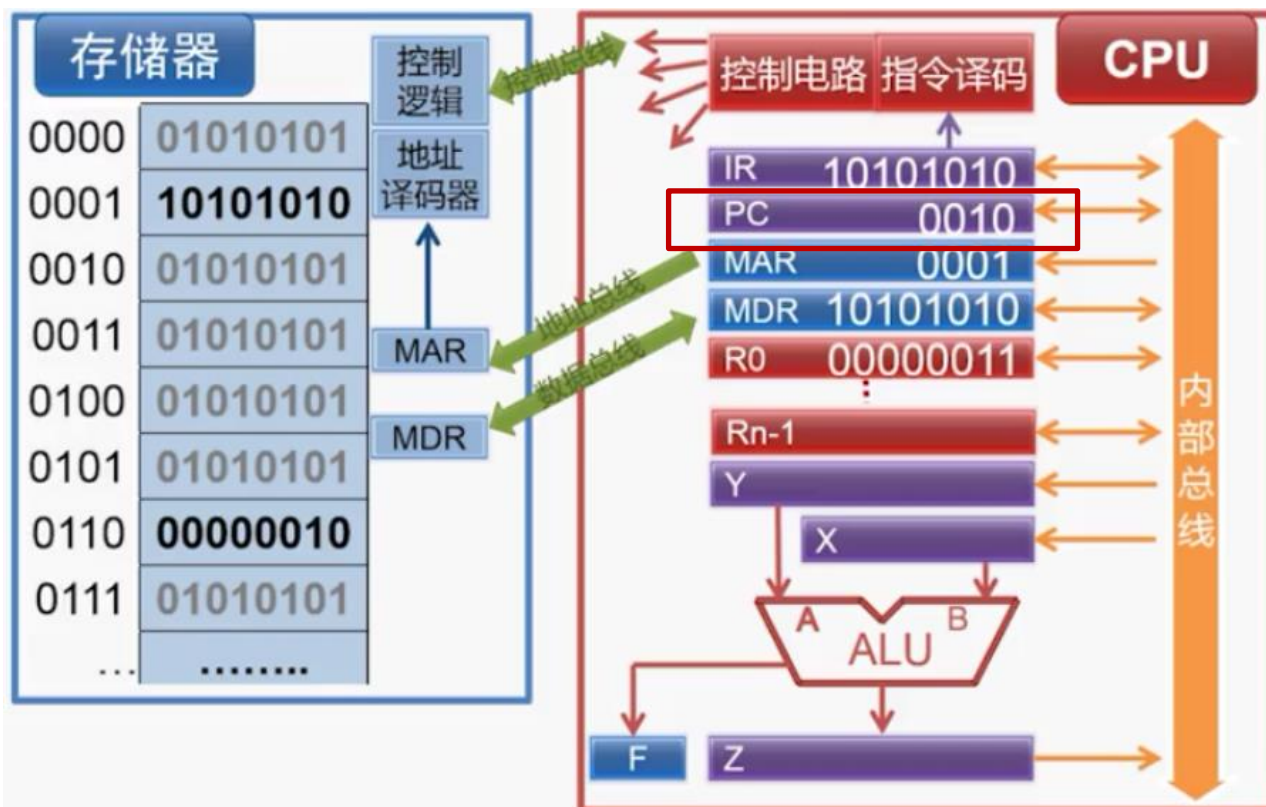
- 控制器将指令地址（PC）送往存储器
- 存储器按给定地址读出指令内容，送回控制器
- 更新下一条指令地址：PC自加

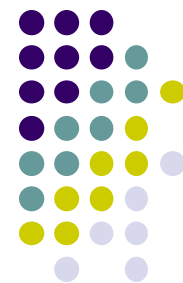




# 指令执行流程——取指 (5)

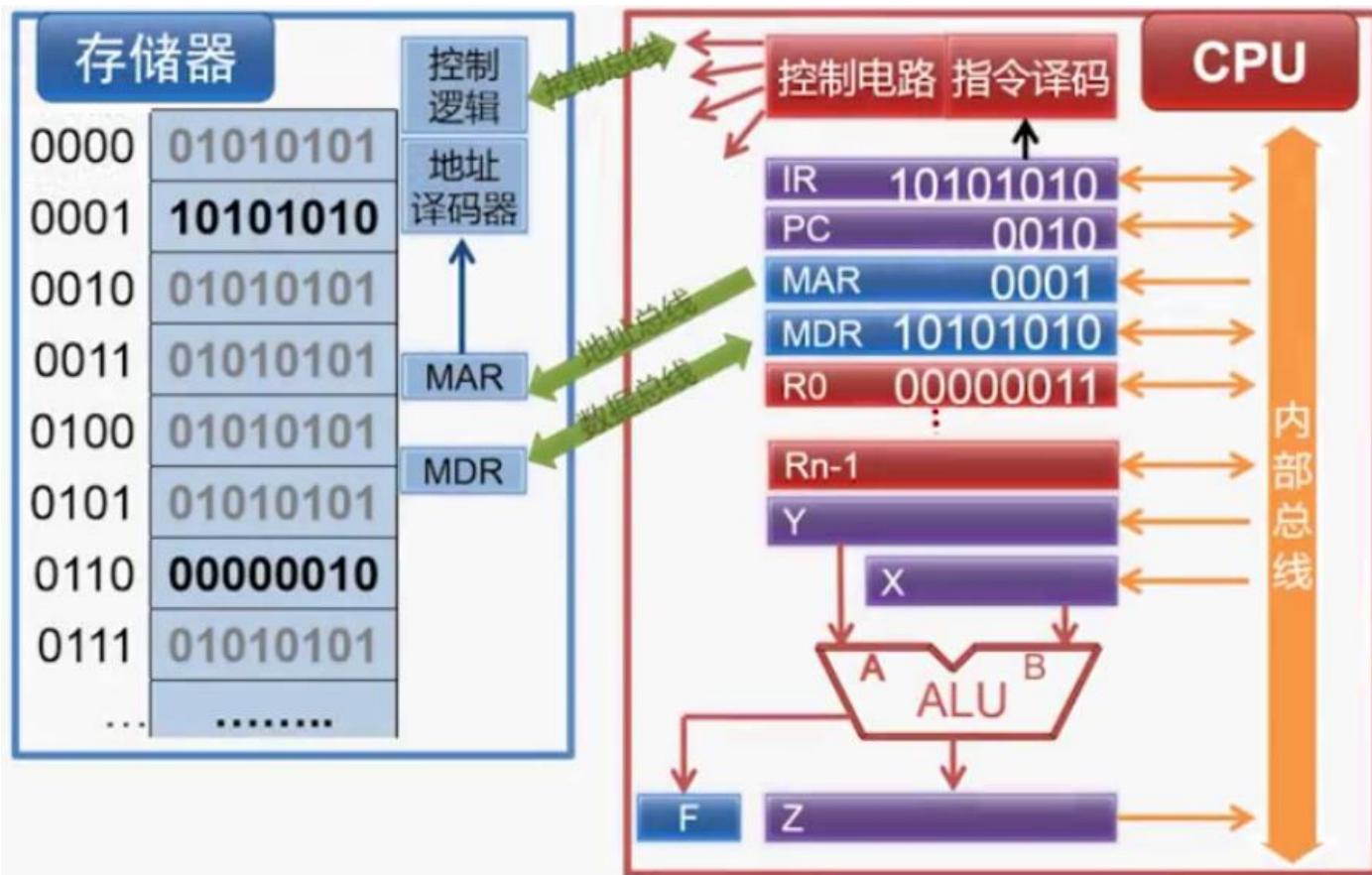
- 控制器将指令地址（PC）送往存储器
- 存储器按给定地址独出指令内容，送回控制器
- 更新下一条指令地址：PC自加





# 指令执行流程——译码

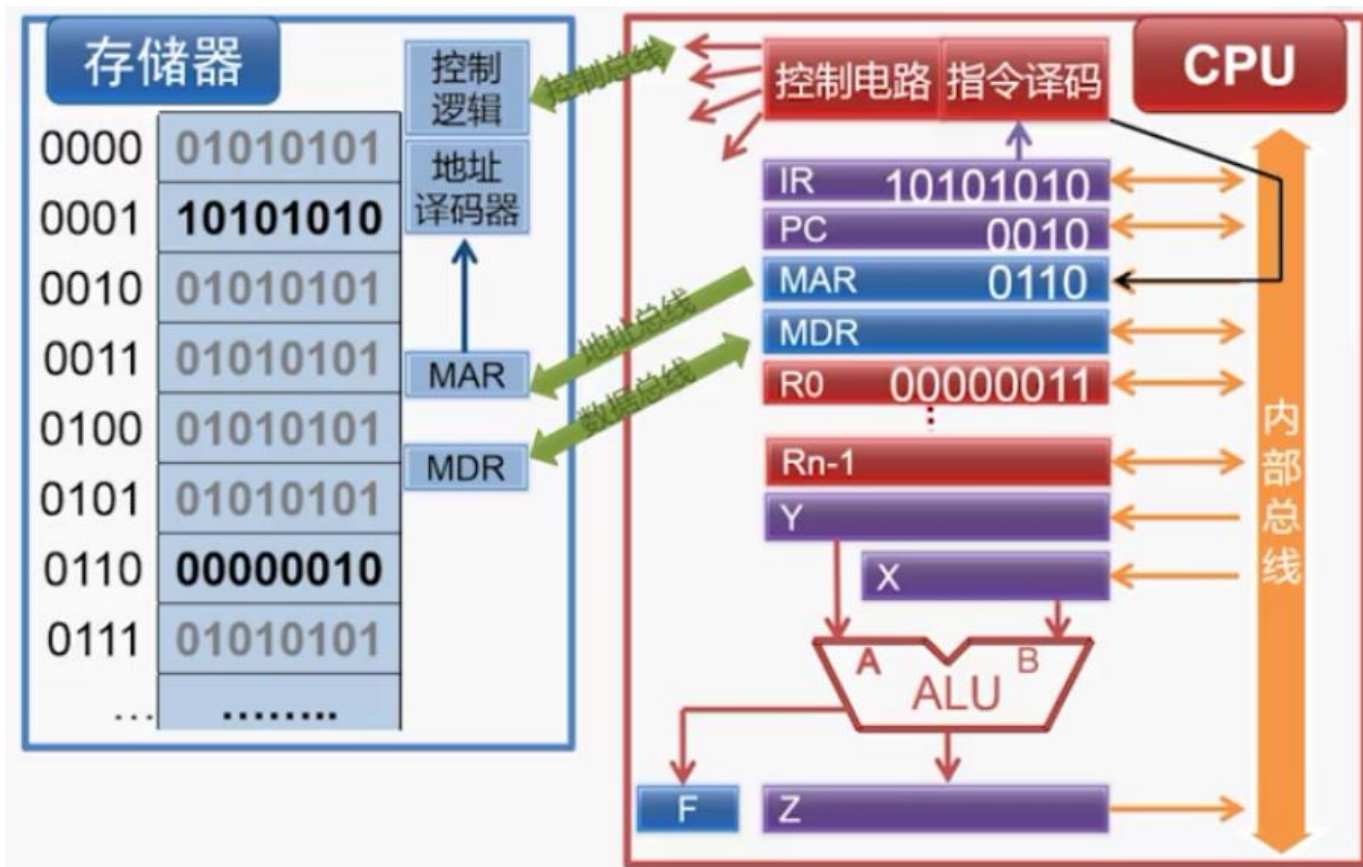
- 控制器分析指令的操作性质
- 根据时钟信号，向各部件发出指令所需控制信号





# 指令执行流程——执行（1）

- 控制器从通用寄存器或存储器中取出操作数
- 命令运算器对操作数进行规定运算（ADD R0, [6]）

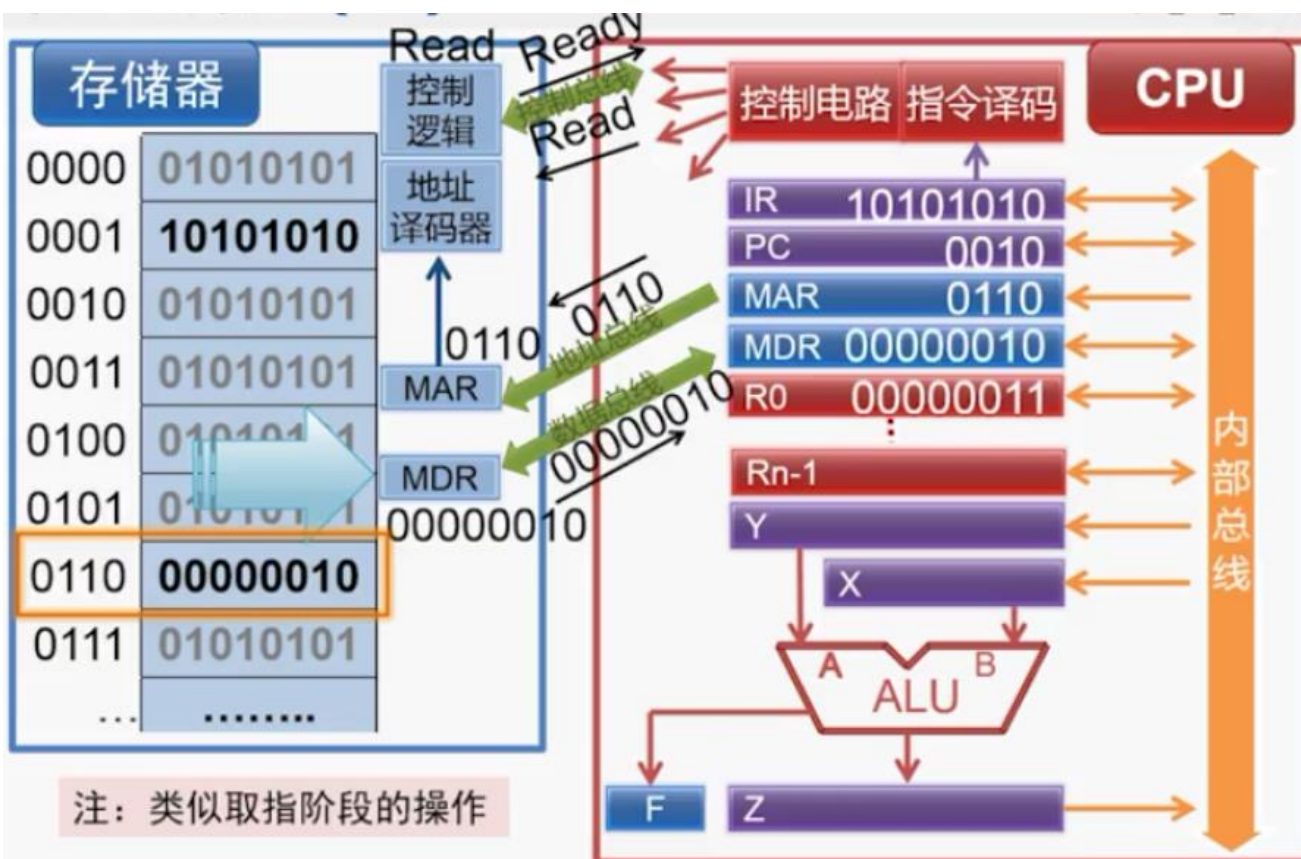






## 指令示例——执行 (2)

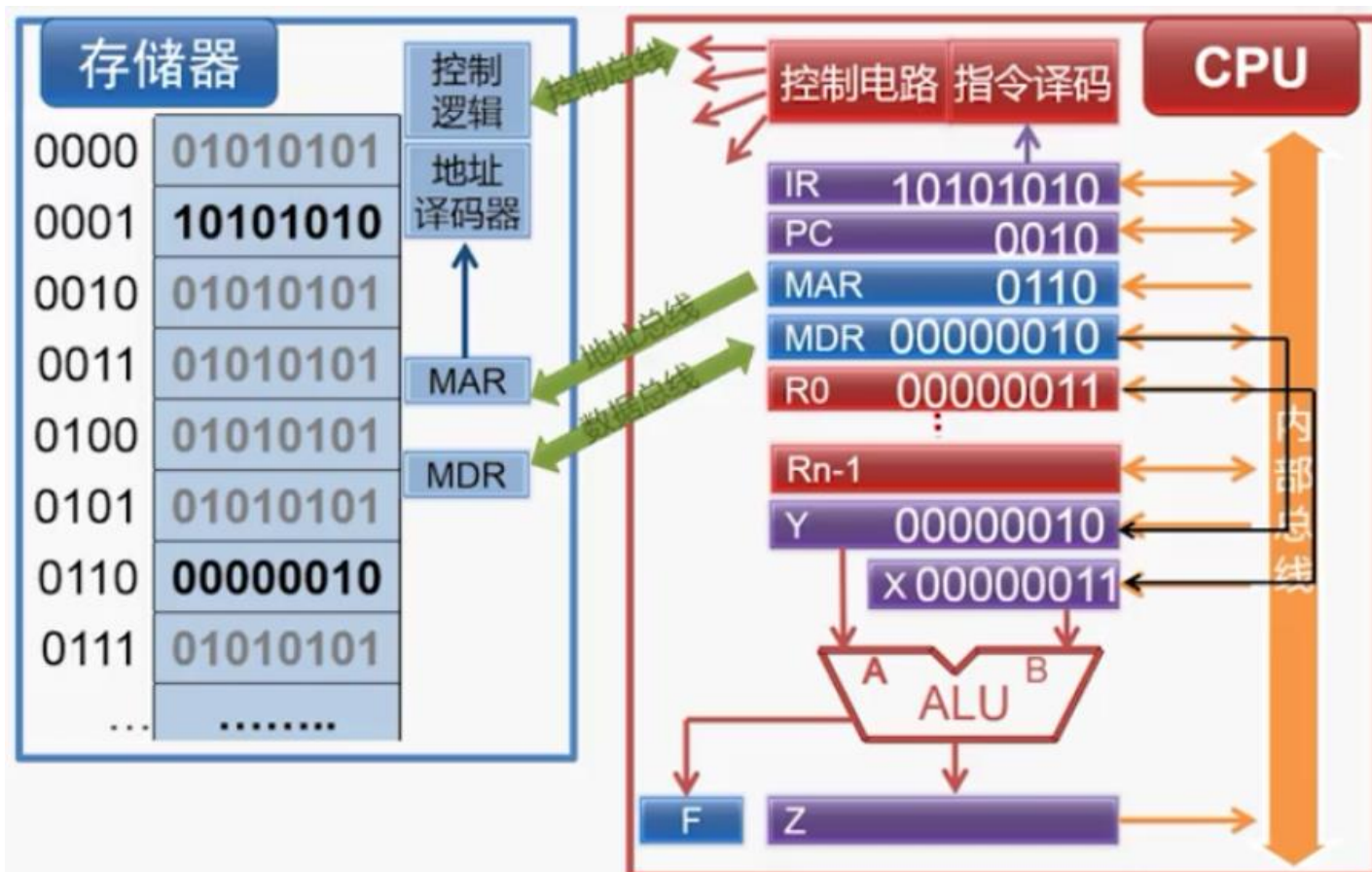
- 控制器从通用寄存器或存储器中取出操作数
- 命令运算器对操作数进行规定运算 (ADD R0, [6])





## 指令示例——执行 (3)

- 控制器从通用寄存器或存储器中取出操作数
- 命令运算器对操作数进行规定运算 (ADD R0, [6])

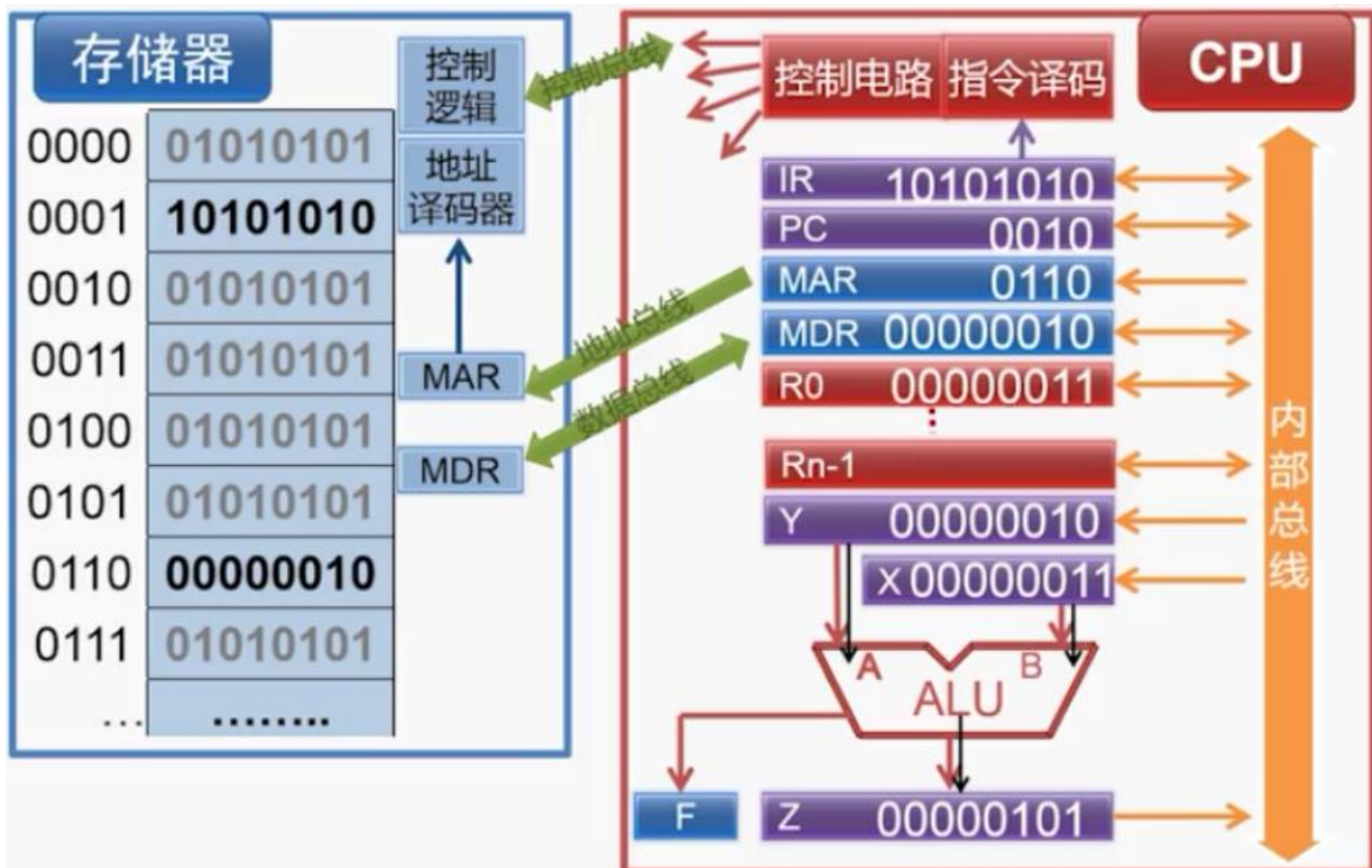






## 指令示例——执行（4）

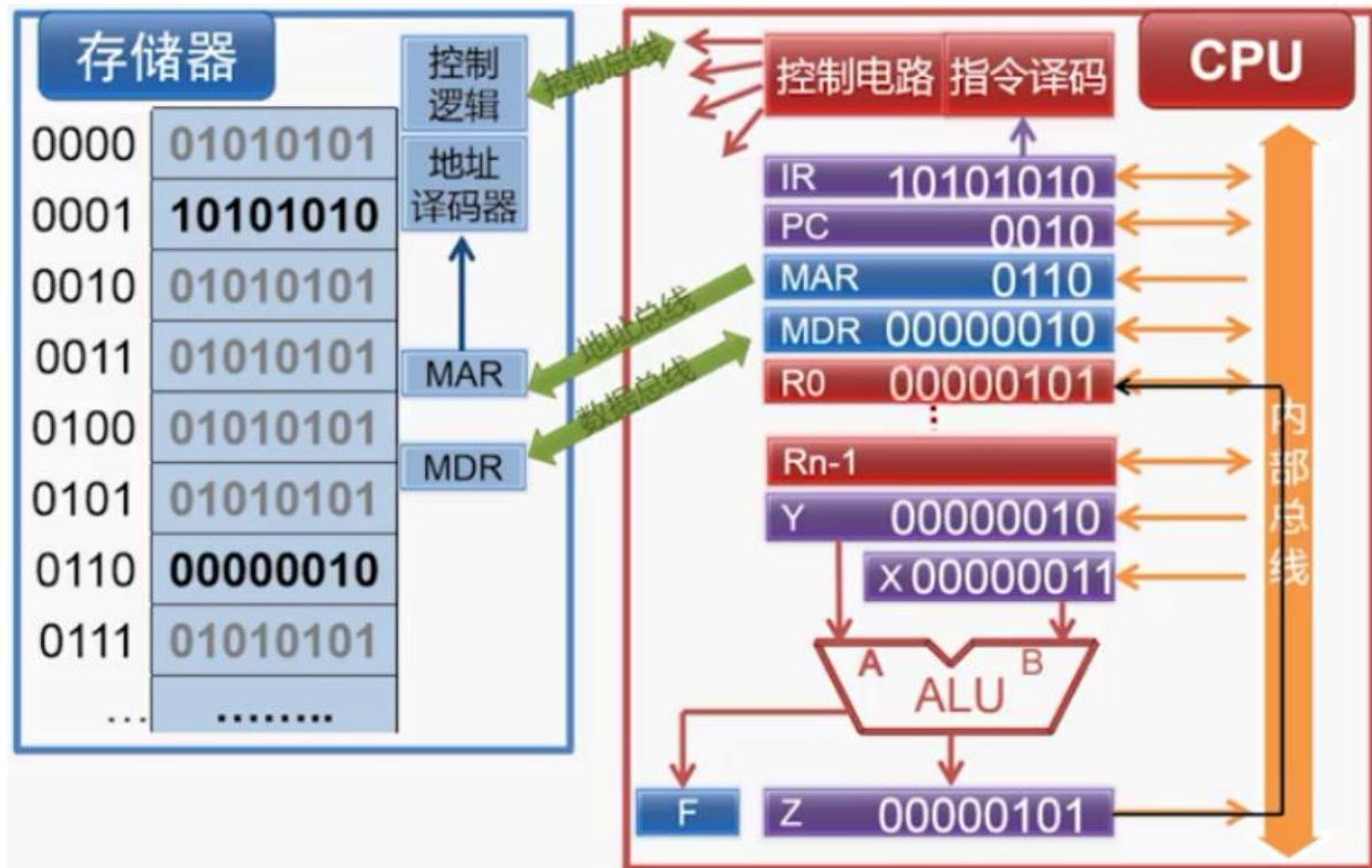
- 控制器从通用寄存器或存储器中取出操作数
- 命令运算器对操作数进行规定运算 (ADD R0, [6])

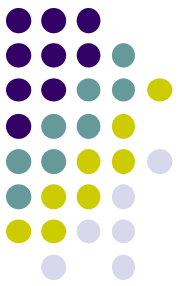




# 指令示例——回写

- 将运算结果写入通用寄存器或存储器 (ADD R0, [6])





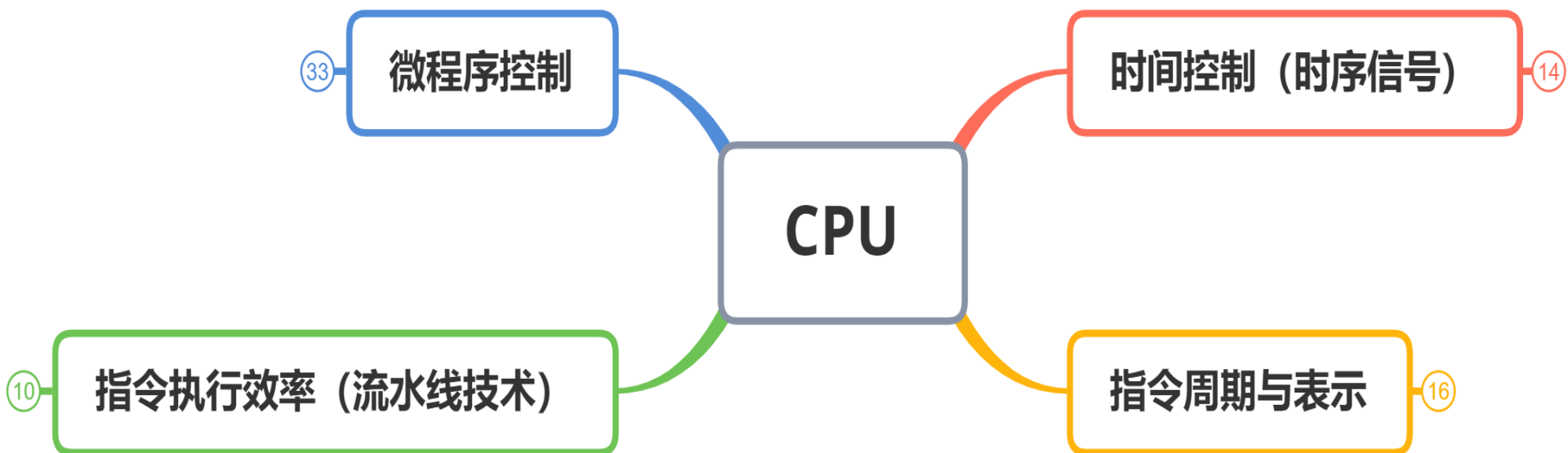
# CPU的功能总结

- CPU：中央处理器
  - 根据编写的程序，**自动从存储器中取出指令，并完成指令操作**
- 指令操作——复杂的控制过程
  - 核心：多个操作信号在指定时间完成对应控制
  - 关键：描述时间、事件
  - **操作控制**（一条指令有若干操作信号实现）
  - **时间控制**（指令各个操作实施时间的定时）

**如何  
控制**

取指令  $\xrightarrow{\text{操作控制、时间控制}}$  执行指令

# 中央处理机





# 第五章 中央处理器

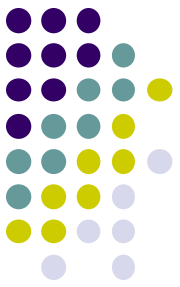
- CPU时序系统
  - 多级时序信号体系
  - 时序信号控制方式
- CPU基本组成
- CPU数据通路



# 单周期、多周期CPU

- 单周期CPU
  - 一个周期内完成从指令取出到得到结果的所有工作
  - 指令系统中所有指令执行时间都以**最长时间的指令为准**，因而效率低，**当前较少采用**
  - 未访存指令 v.s. 单次访存指令 v.s. 多次访存指令
- 多周期CPU
  - 指令的执行分成多个阶段，每个阶段一个周期
  - 例如：取指→译码→执行→回写
  - 因而时钟周期短，不同指令所用周期数可以不同
  - 优点：**灵活性高**、降低等待开销，被广泛采用
  - 核心：**设计多级（多层次）的时序系统**





# 为什么/什么是多级时序?

回顾：指令执行过程由 [填空1] 、 [填空2] 、 [填空3] 、 [填空4] 四步组成。

- 取指，译码，执行，回写
- 思考：
  - 控制器如何确定取指/译码/执行/回写先后顺序（时序）？
  - 单一步骤内部又有复杂细微操作，如何控制细微操作先后顺序（时序）？
- 计数？方式1：1, 2, 3, 1, 2, 3, ... 多级时序  
                  方式2：1, 2, 3, 4, 5, ..... 单级时序

细微操作

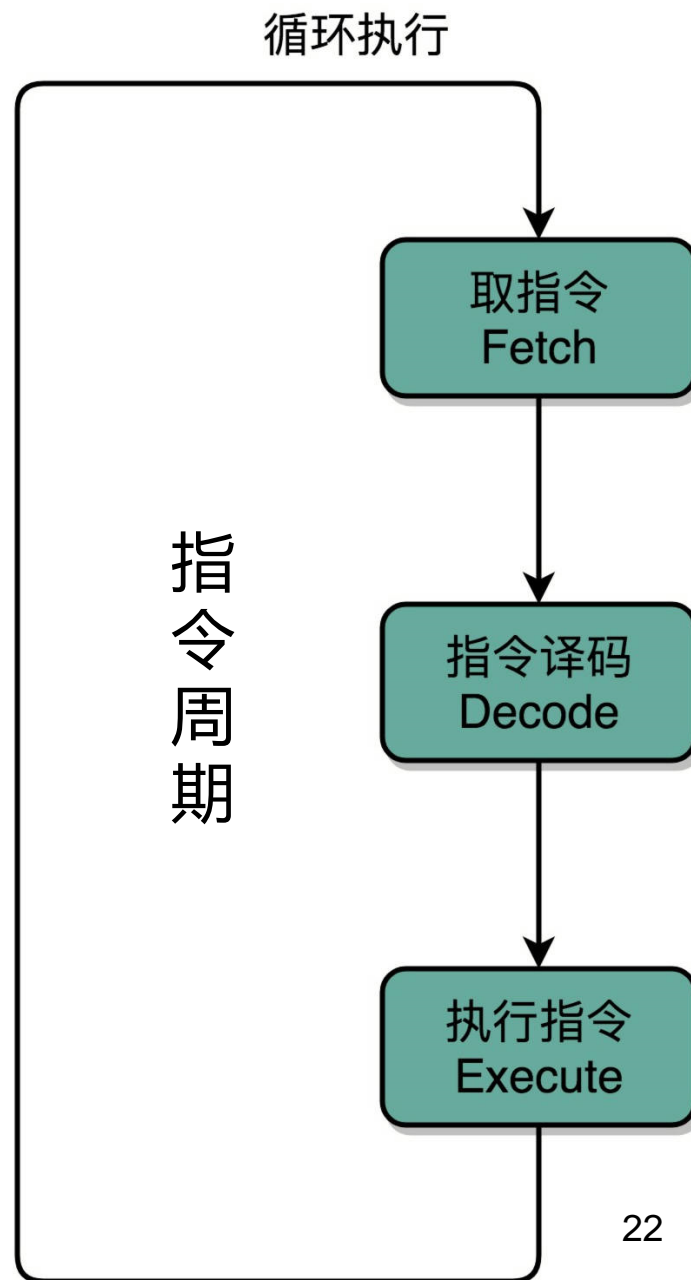
一、取指

二、译码

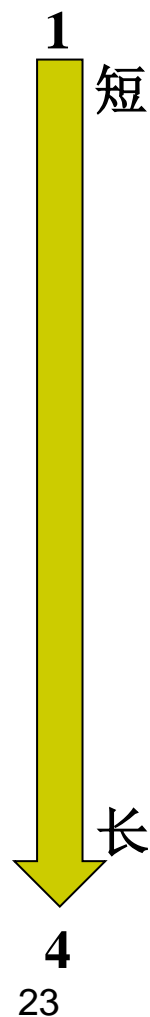
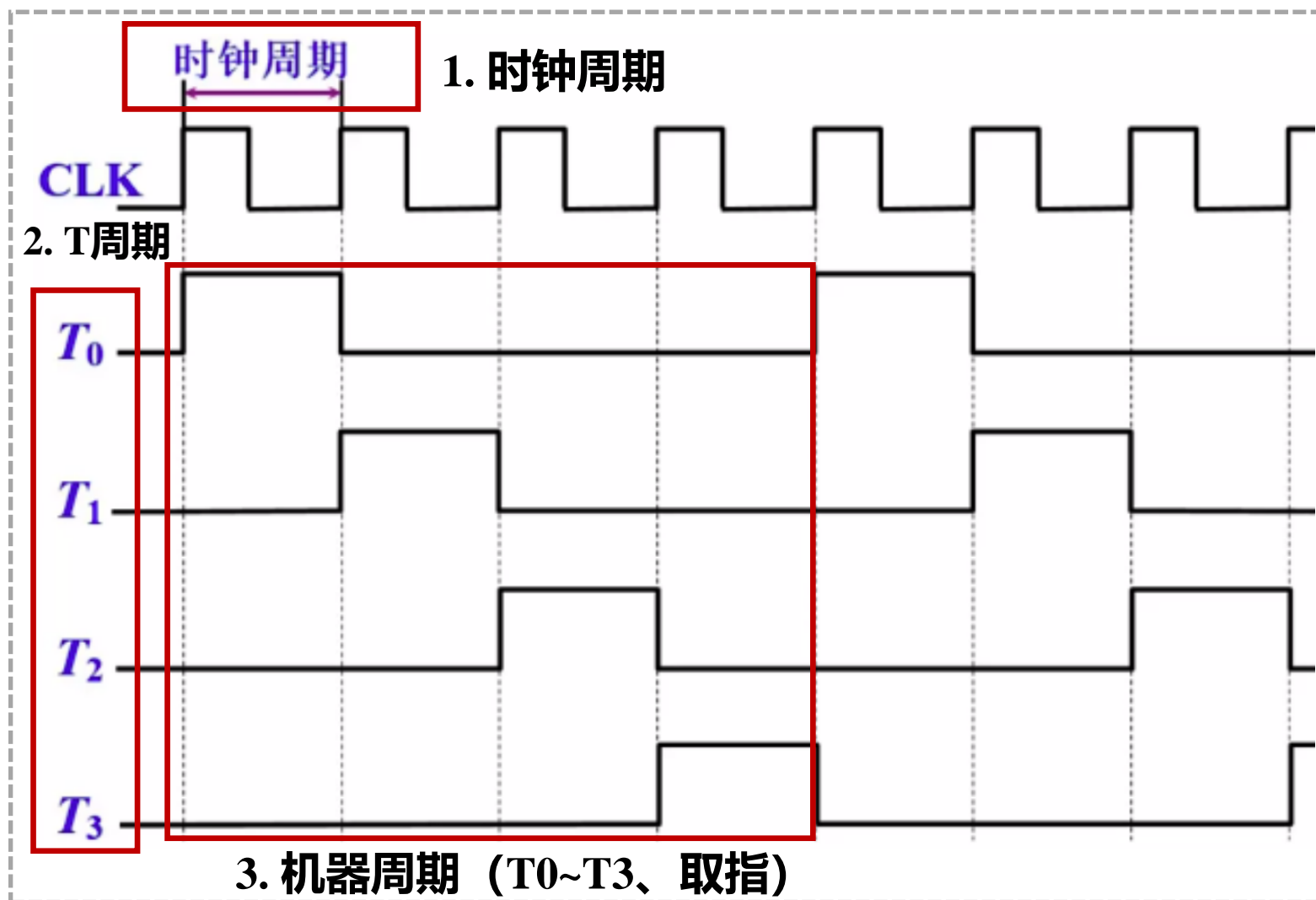
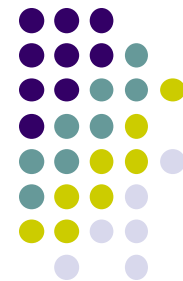


# 指令周期的基本概念

- CPU功能
  - 根据存储程序自动取出指令，译码、并执行
- 指令周期
  - 取出指令、分析指令到执行完该指令所需的**全部时间**
  - 各种指令功能不同（访存次数/执行复杂度）
  - 各种指令的指令周期不同



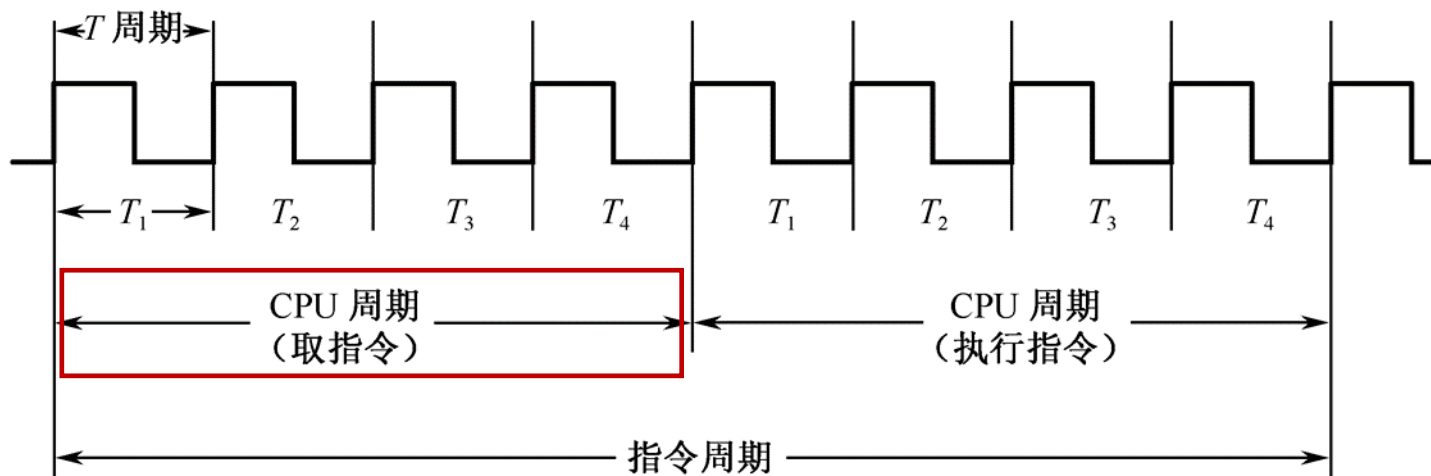
# 多级时序系统

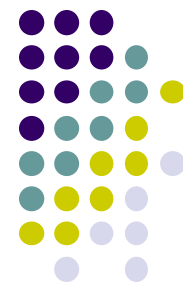




# 各类周期概念——机器/CPU周期

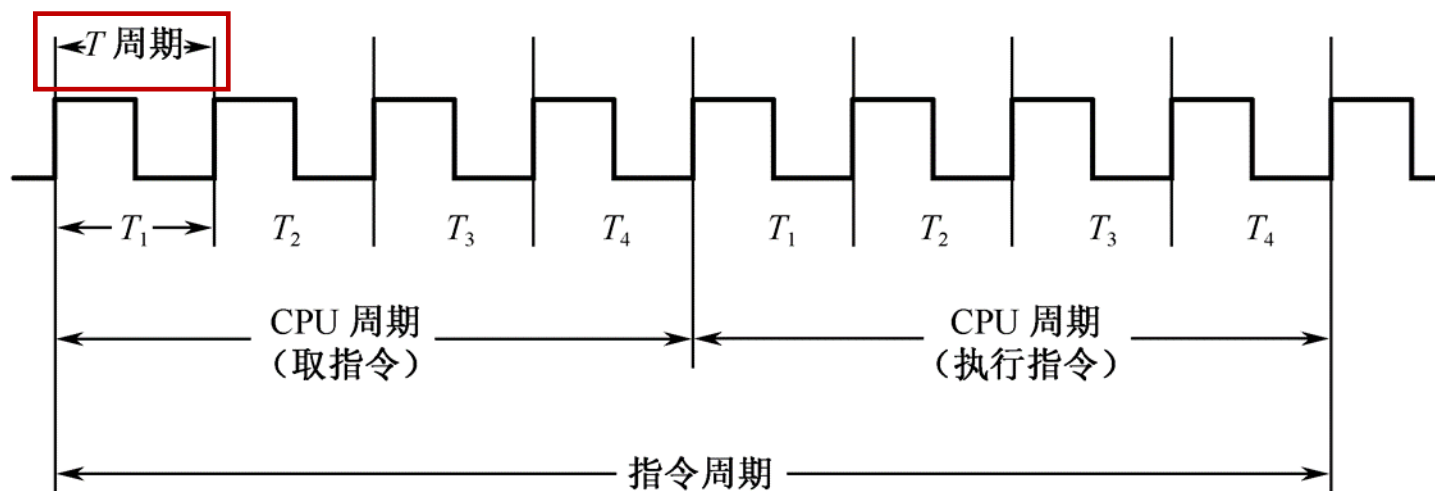
- **指令周期**划分为若干个CPU周期/机器周期
- **机器/CPU周期**
  - 又称**机器周期**，每个机器周期完成一个基本操作。
  - 一般，以**主存的工作周期（存取周期）为基础**来规定CPU周期（例如，读取一个指令字的最短时间）





# 各类周期概念——T周期/节拍脉冲

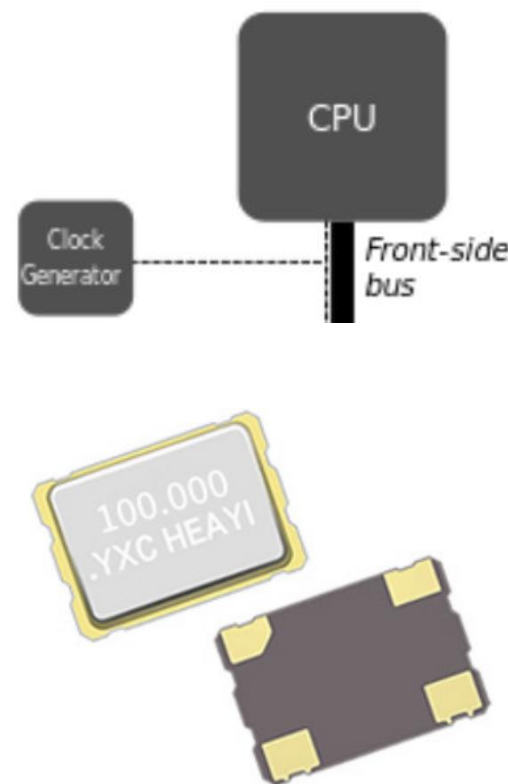
- 机器周期由多个T周期组成
- **T周期**
  - 又称**节拍脉冲**，是处理操作的基本单位
- 在一个机器周期内，要完成若干个微操作
  - 因而需要把一个CPU周期分为若干个相等的时间段，每一个时间段称为一个节拍脉冲或T周期



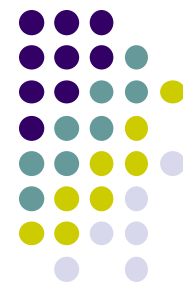


# 各类周期概念——时钟周期

- 时钟源：标准晶振
  - 例如，100MHz有源贴片晶振（主板）
  - CPU内部通过倍频器，提升晶振频率，生成CPU主频（CLK）
  - 酷睿i7 3770处理器：倍频范围16~39，空闲CPU主频1.6GHz，最高睿频3.9GHz
- 时钟周期：
  - 控制计算机操作的**最小时间单位**
  - 是生成CPU时序信号的基准



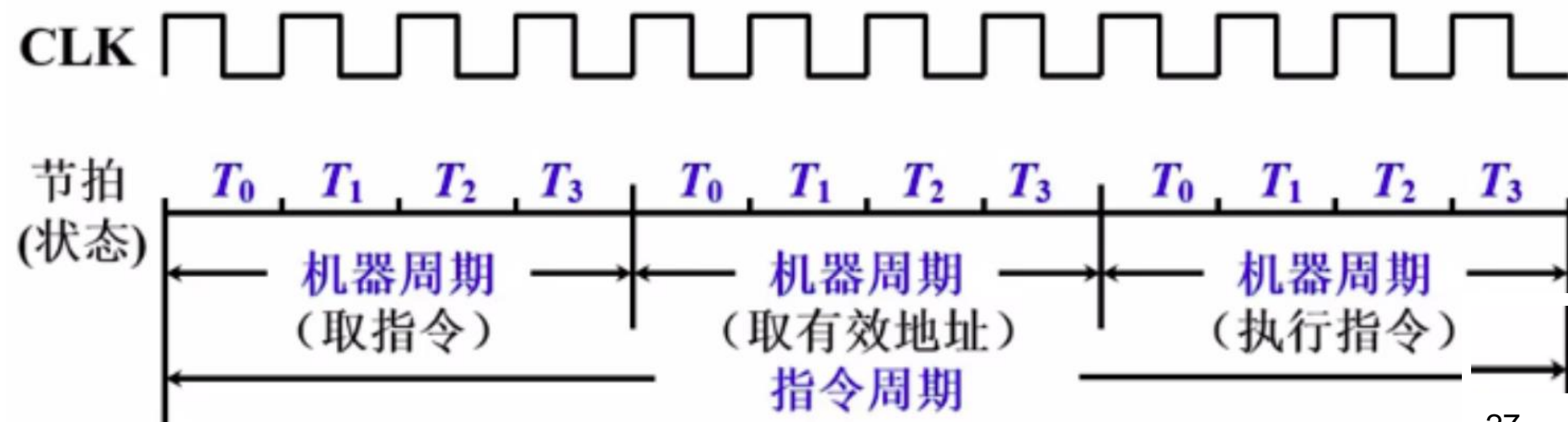




# 多级时序系统——总结

多级时序系统组成

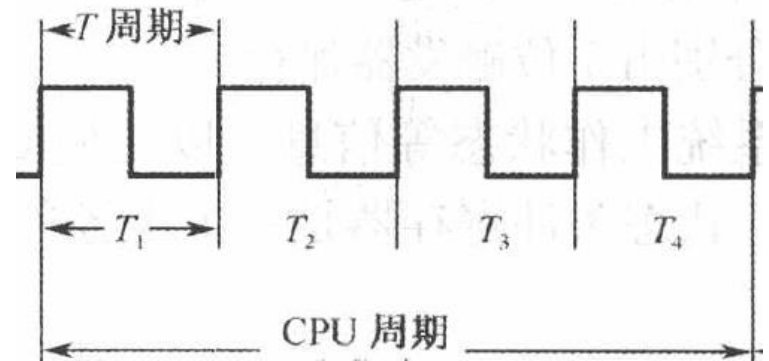
- 一个指令周期包含若干个机器/CPU周期
- 一个机器周期包含若干个T周期
- T周期由时钟周期生成
- 时钟周期（主频）是控制CPU操作的最小时间单位

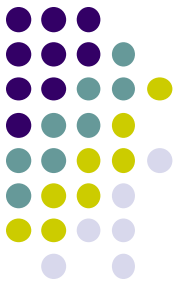




# 时序信号体制

- 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：机器周期
  - 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（倍频）**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）



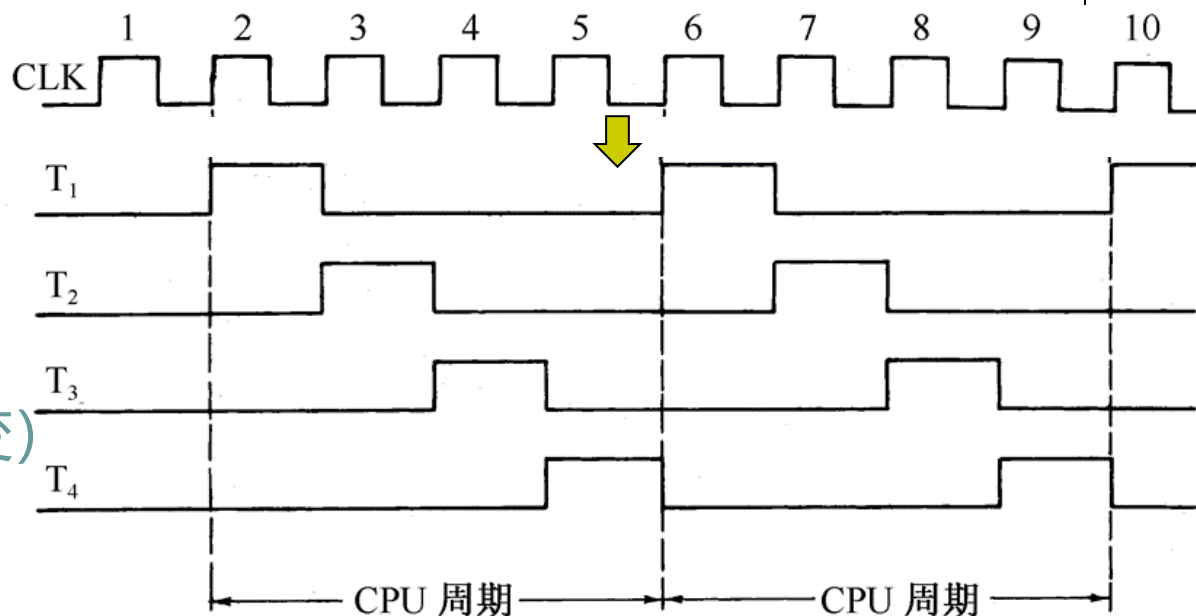


# 时序信号关系

- 晶振 (倍频)

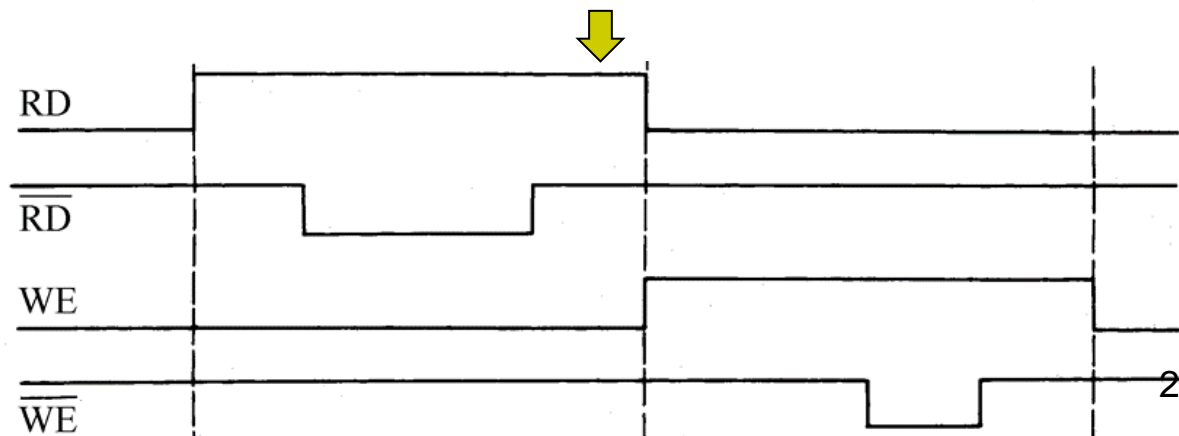
- 节拍脉冲

- CNT: 0~3
- 四分频 (可变)
- 占空比: 25%



- 读写控制信号

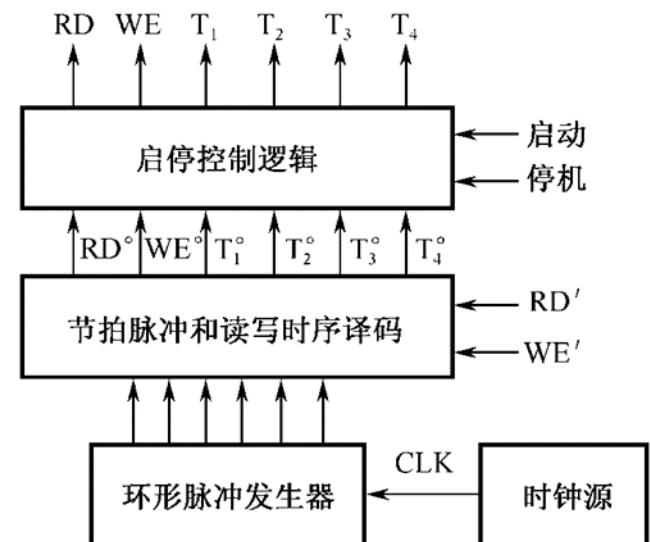
- 读写控制功能



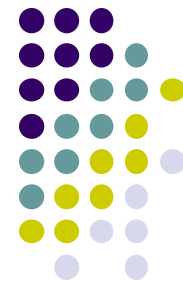


# 硬件：时序信号产生器（了解）

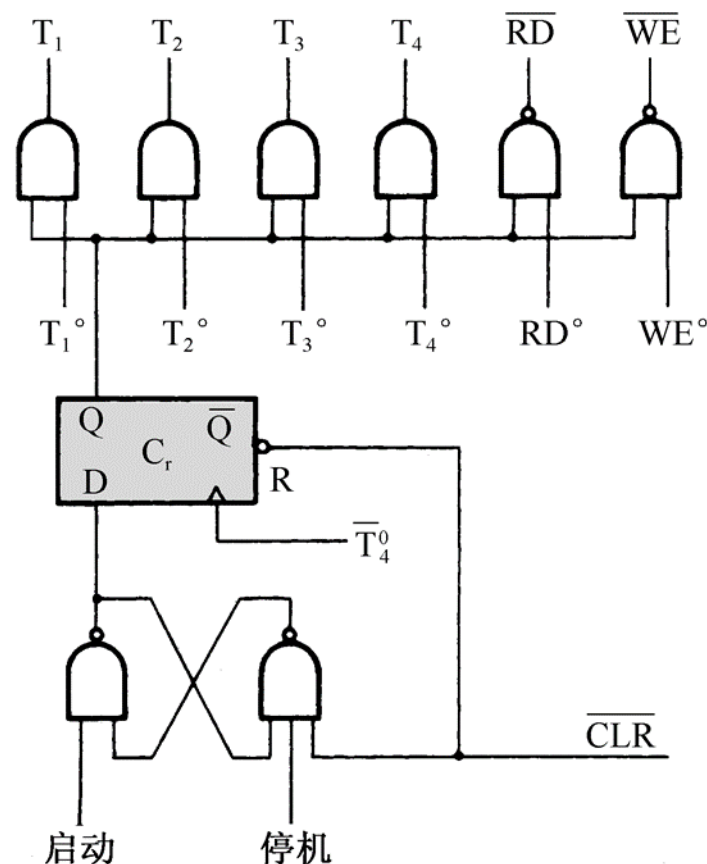
- 功能
  - 产生时序信号
    - 各型计算机产生时序电路不相同
    - 大、中型计算机的时序电路复杂，微型计算机的时序电路简单
- 构成
  - 时钟源
  - 环形脉冲发生器
  - 节拍脉冲和读写时序译码逻辑
  - 启停控制逻辑

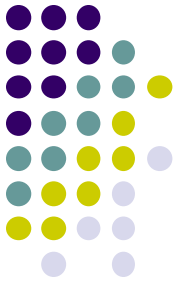


# 硬件：启停控制逻辑



- 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
- 当运行触发器为0时
  - 关闭时序产生器





# 第五章 中央处理机

- CPU时序系统
  - 多级时序信号体系
  - 时序信号控制方式
- CPU基本组成
- CPU数据通路





# 控制方式

- 指令所包含的机器周期数反映了指令的复杂程度，不同机器周期的操作信号数目和出现的先后次序也不同
- 控制方式
  - 控制不同操作序列时序信号的方法
  - 节拍脉冲（T周期）-节拍电位（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



# 什么是多种控制方式?

- 指令每个阶段复杂程度不同, 如何考虑?

同步控制

- 方式1: 1, 2, 3, 1, 2, 3..... 定长
- 方式2: 1, 2, 3, (4) , 1, 2, 3, ..... 不定长
- 方式3: 1, 2, (3.1, 3.2, 3.3) , 1, 2, 3.....

中央  
局部  
融合

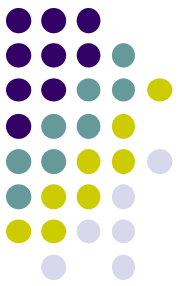
e.g., Ready信号

异步控制

- 方式4: 1, — (等待) , 2, —, 3, —, 1, —, .....

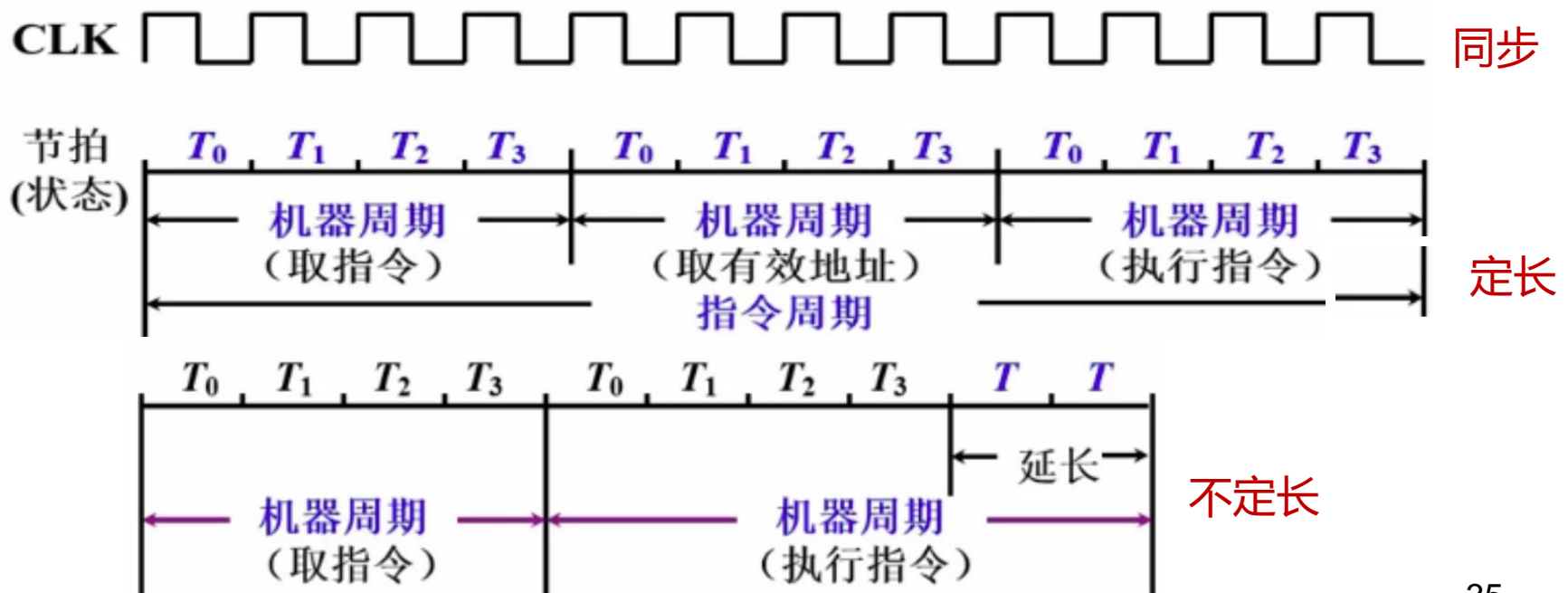
联合控制

- 方式5: 1, 2, 3, —, 1, —, 2, 3, .....



# 控制方式——同步控制 (1)

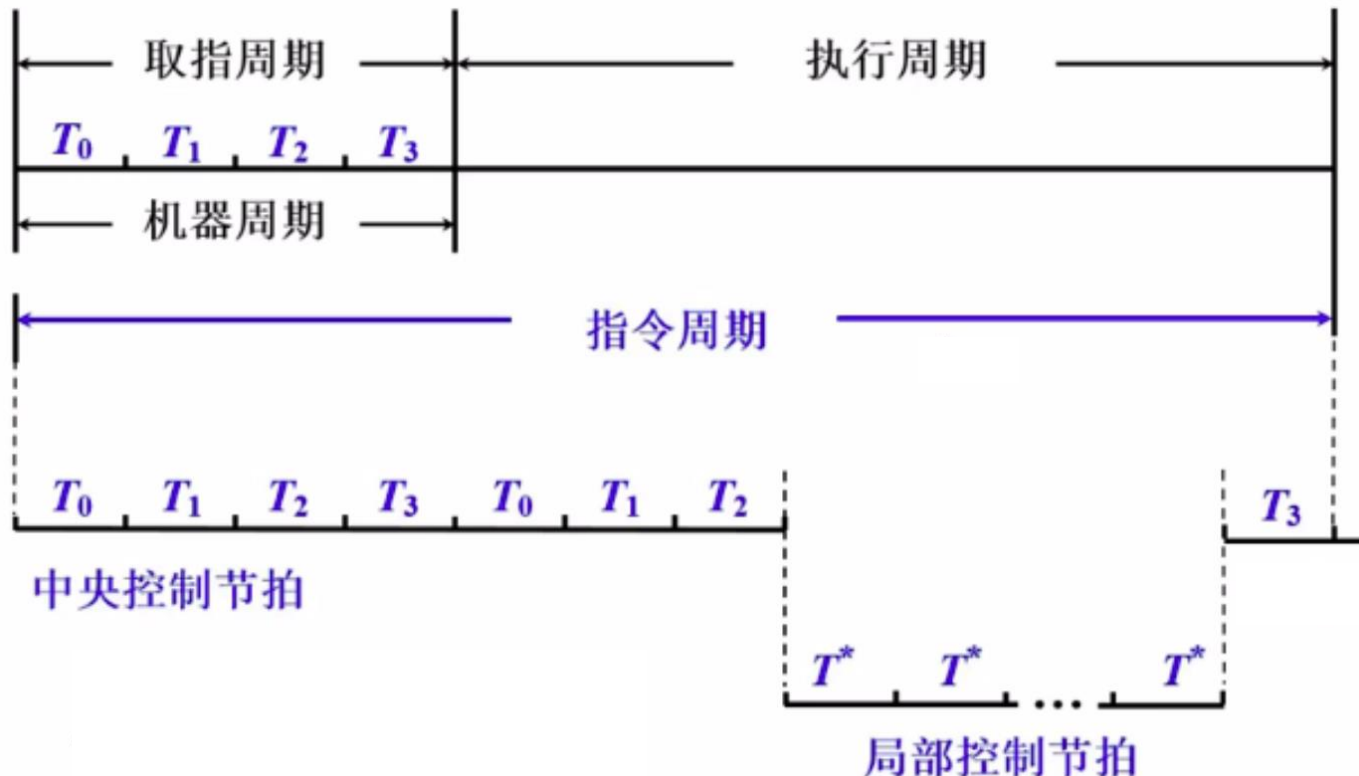
- 同步控制：由统一基准时标的时序控制方式 (CLK)
  - 方式1：完全统一的 (定长) 机器周期执行不同的指令
  - 方式2：采用不定长机器周期





# 控制方式——同步控制 (2)

- 同步控制：由统一基准时标的时序控制方式 (CLK)
  - 方式3：中央控制与局部控制的结合





# 控制方式——异步/联合控制

- 异步控制方式
  - 无基准时间信号
  - 每条指令需要多长时间就占多长时间，回答信号 (Ready)
  - 优点：不需要时间同步，灵活，适用于不定时、大规模网络场景
  - 缺点：效率问题，不适用于超高速率场景
- 联合控制方式（同步与异步结合）
  - 主要使用同步方式，部分耗时较长指令（IO指令）异步
  - 融合同步与异步的优点
  - 微程序控制器采用



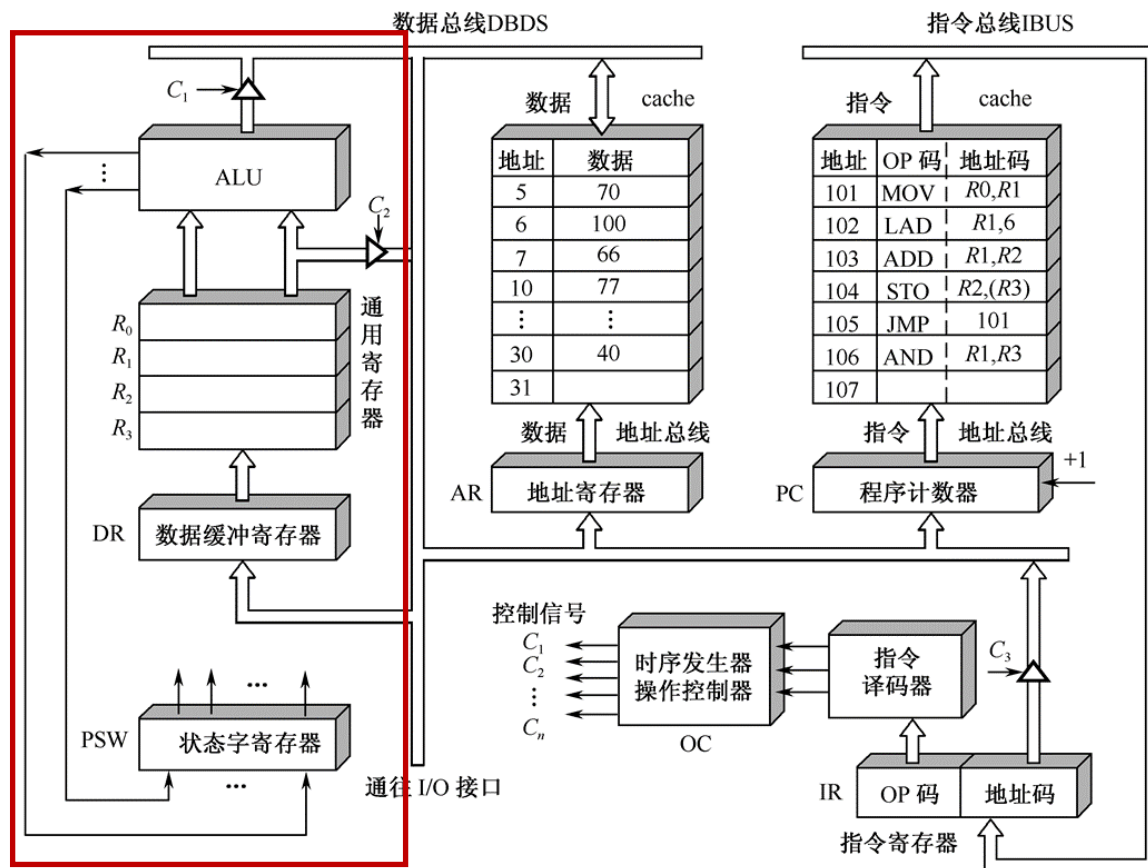
# 第五章 中央处理机

- CPU时序系统
- CPU基本组成
- CPU数据通路

# CPU基本组成——运算器



- CPU组成
  - 运算器
  - I-Cache/D-Cache
  - 控制器
- 运算器
  - ALU
  - 通用寄存器
  - 数据缓冲器：DR
  - 状态字寄存器：PSWR

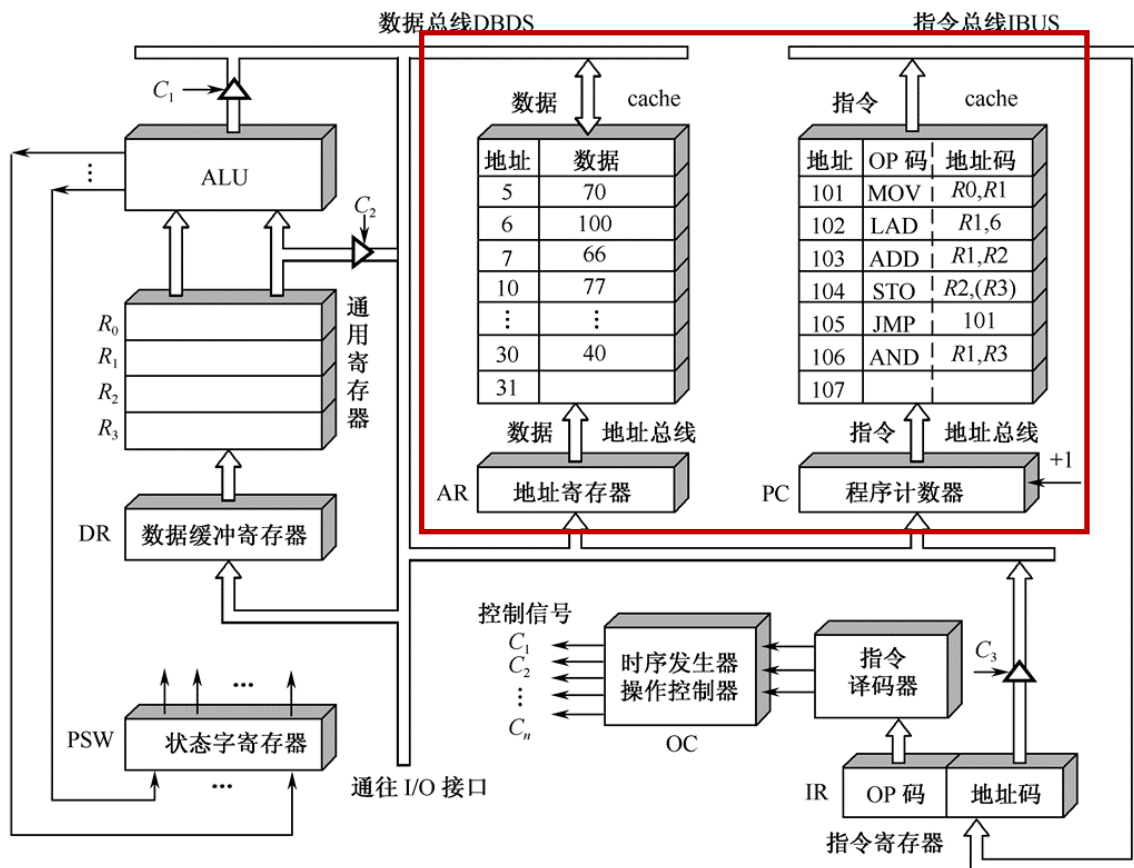




# CPU基本组成——高速缓存



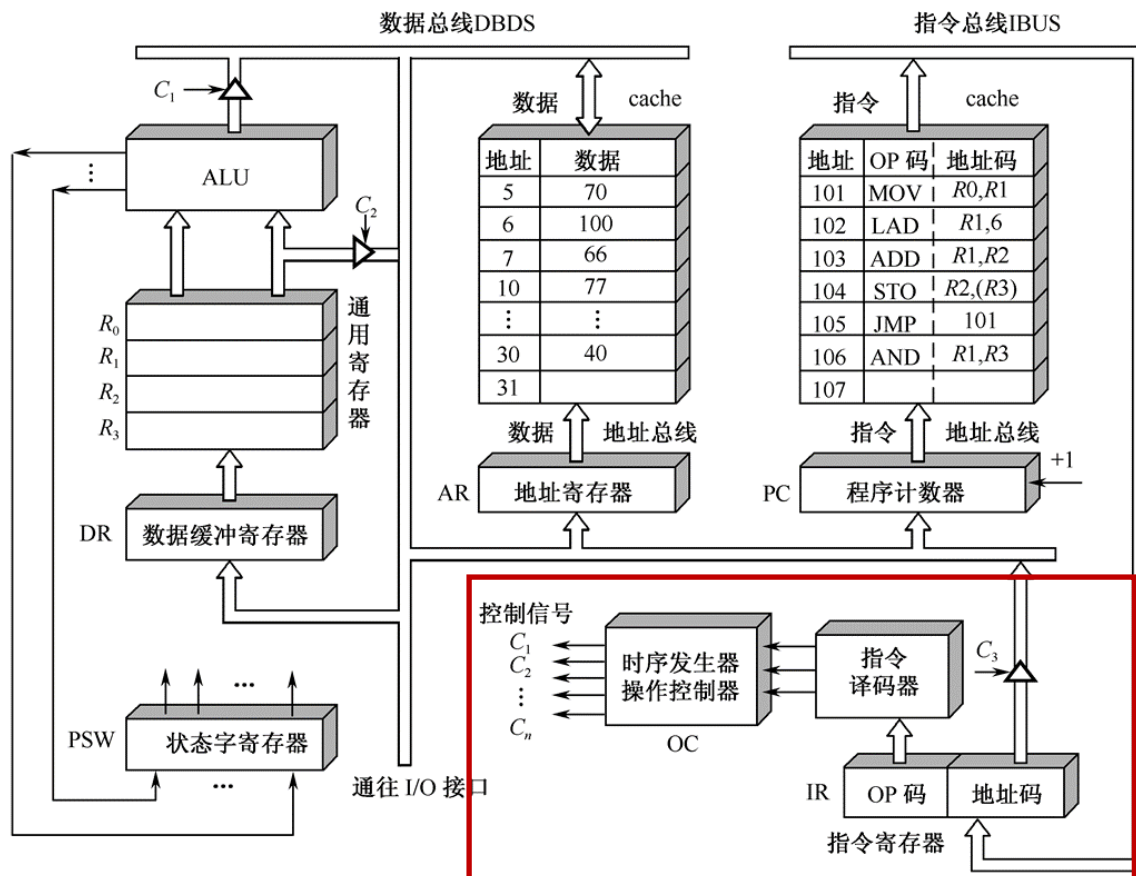
- CPU组成
  - 运算器
  - I-Cache/D-Cache
  - 控制器
- 高速缓存
  - I-Cache
    - PC寄存器
    - I-BUS
  - D-Cache
    - AR寄存器
    - D-BUS



# CPU基本组成——控制器



- CPU组成
  - 运算器
  - I-Cache/D-Cache
  - 控制器
- 控制器
  - 指令寄存器IR
  - 指令译码器
  - 时序发生器
  - 操作控制器
  - 数据通路





# CPU基本组成——控制器

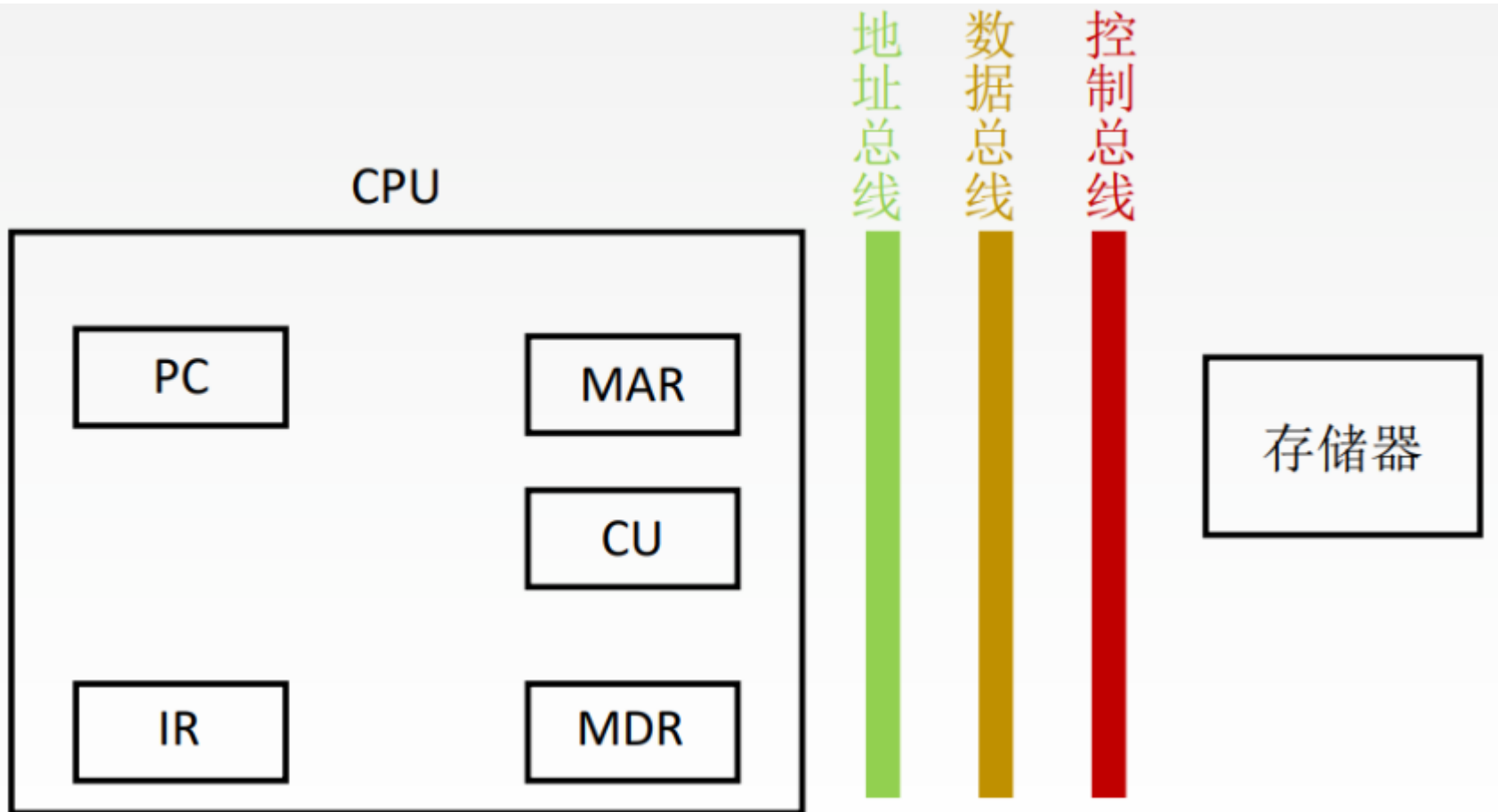
- 数据通路
  - 各组件间通路
- 操作控制器
  - 为数据通路的建立提供各种操作信号。根据设计方法不同
    - 硬布线控制器
    - 微程序控制器
- 时序产生器
  - 提供定时和时序信号
- 其他功能部件：中断系统、总线接口等



# 第五章 中央处理机

- CPU时序系统
- CPU基本组成
- CPU数据通路

# 数据通路





# 数据通路分类

- **数据通路**

- 数据在功能部件之间传送的路径，建立数据通路的任务是由“控制部件”来完成的 **(控制初步)**

- **数据通路分类**

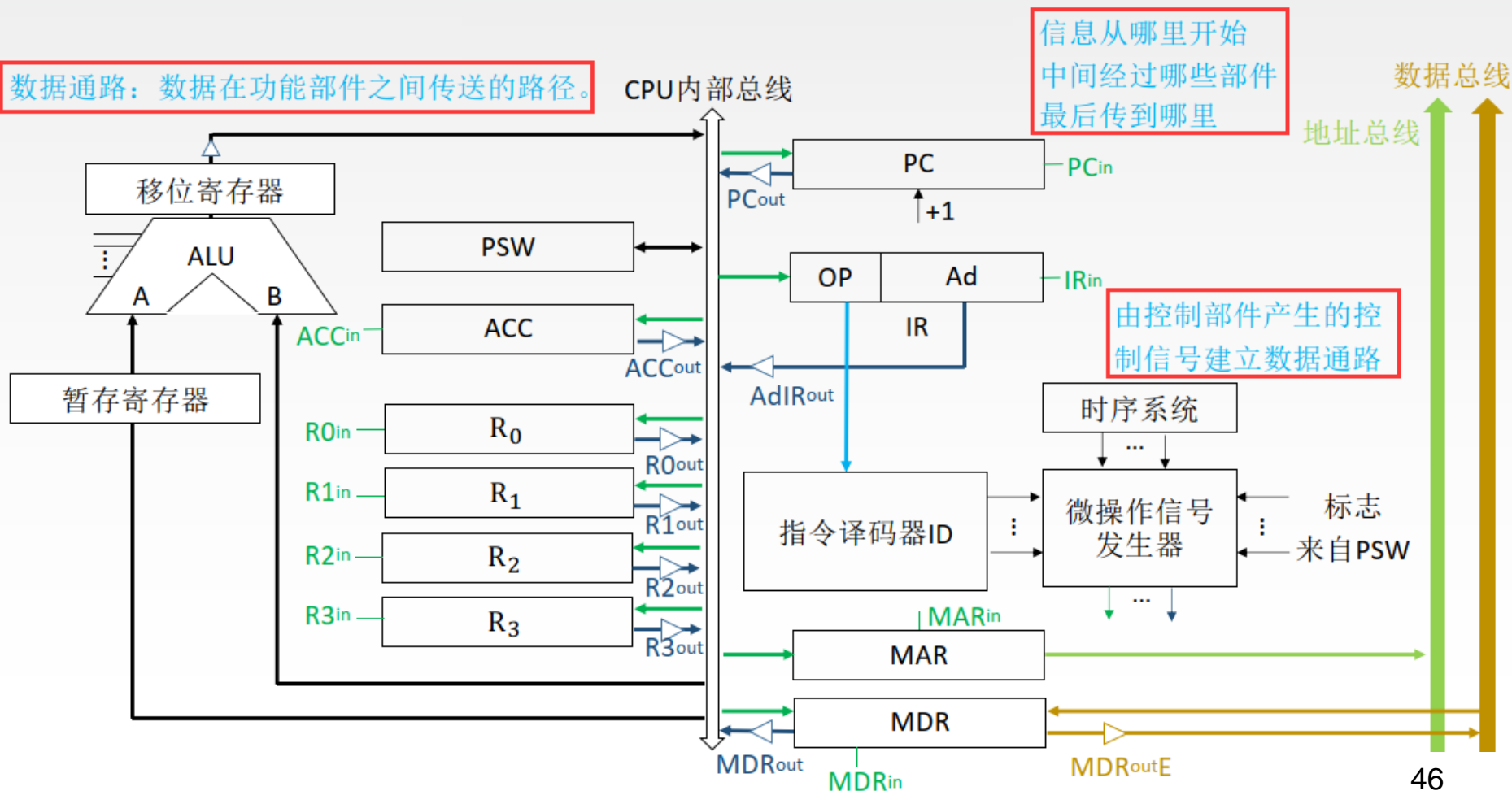
- 内部数据通路：CPU内部的数据传送通路，用于在寄存器之间或寄存器与ALU之间传送数据
- 外部数据通路：一般借助系统总线，将寄存器（MAR和MDR）与存储器和I/O模块连接起来

- **关键：阅读数据通路图**

# 数据通路图——示例1

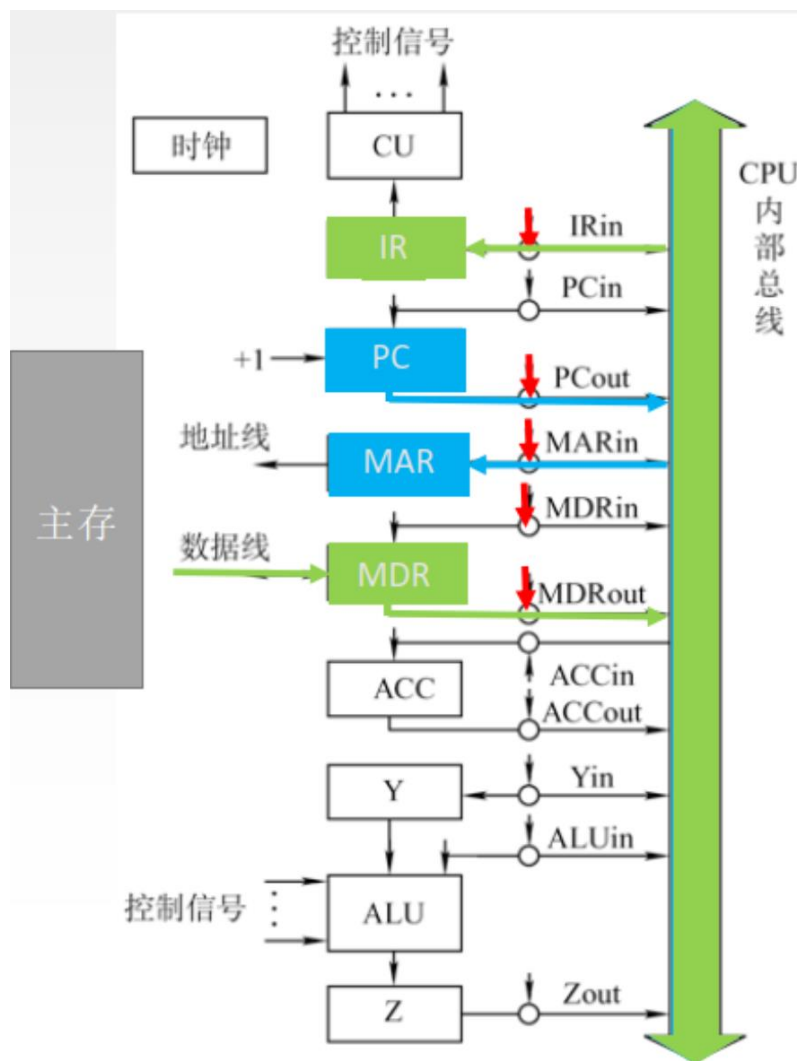


数据通路：数据在功能部件之间传送的路径。



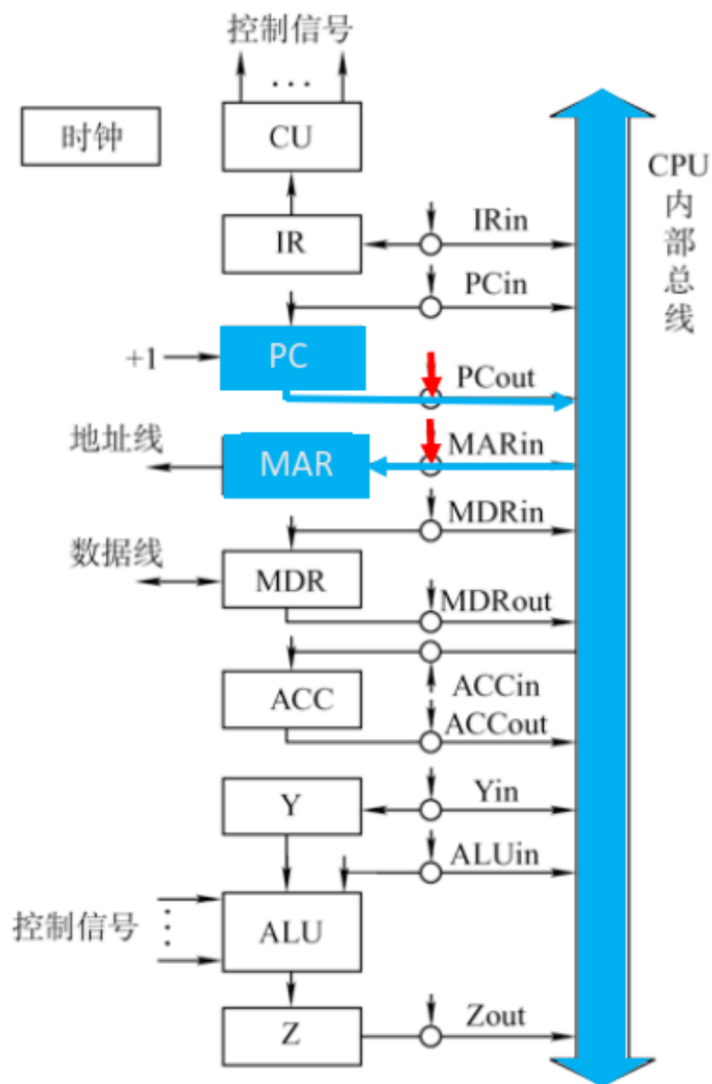


# 数据通路图——示例2



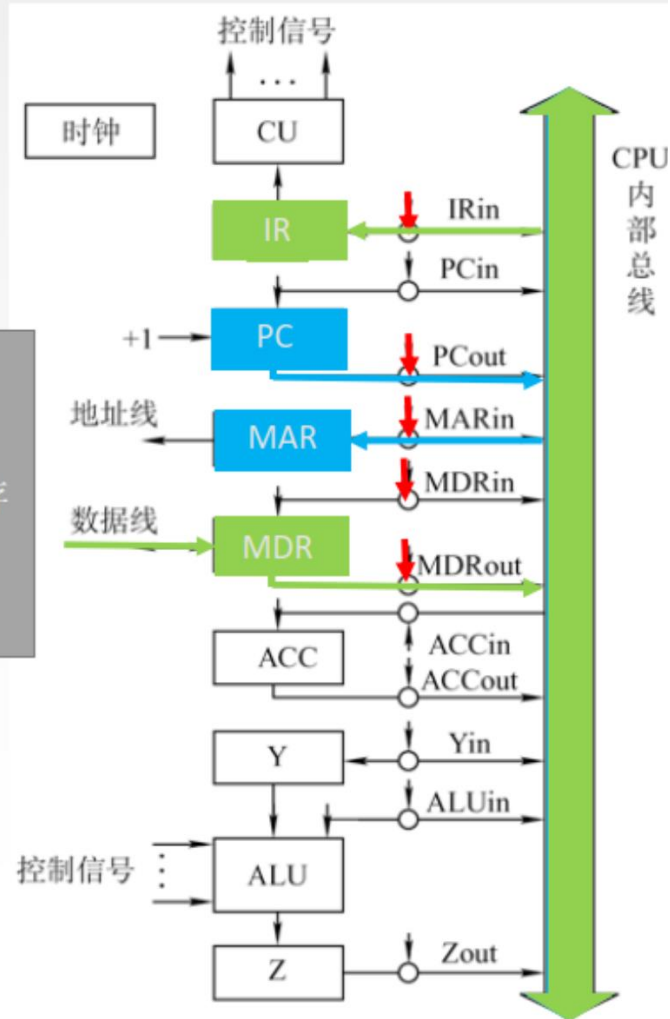


# 1. 操作—寄存器间



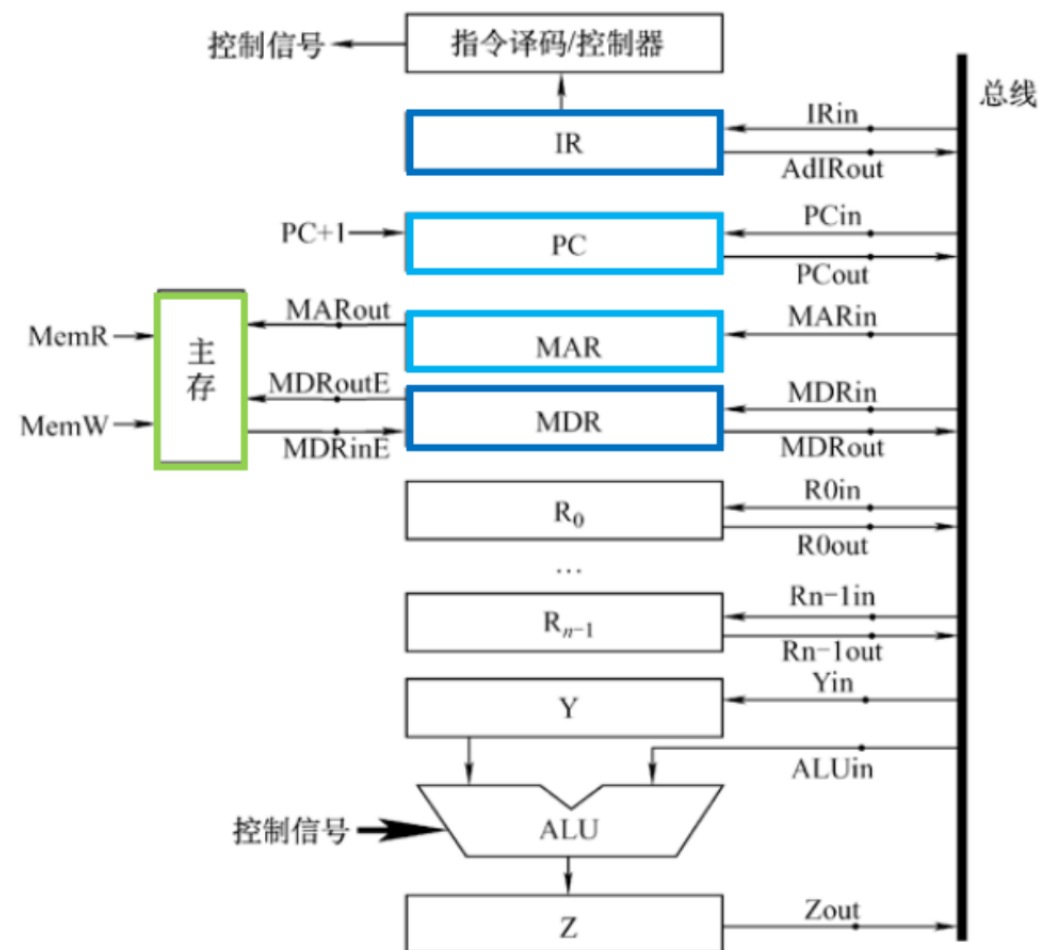
- 寄存器间数据通路
- 例如, PC内容送至MAR
- 写作:
  - PC→Bus→MAR
  - 标明控制信号
    - PC→BUS, PCout
    - Bus→MAR, MARin

## 2. 操作—主存→CPU



- 主存与CPU间数据传送
- 方式：通过MAR/MDR寄存器
- 例如，取指流程写作：
  - PC→BUS→MAR
  - MEM(MAR)→MDR
  - MDR→BUS→IR

# 数据通路例题

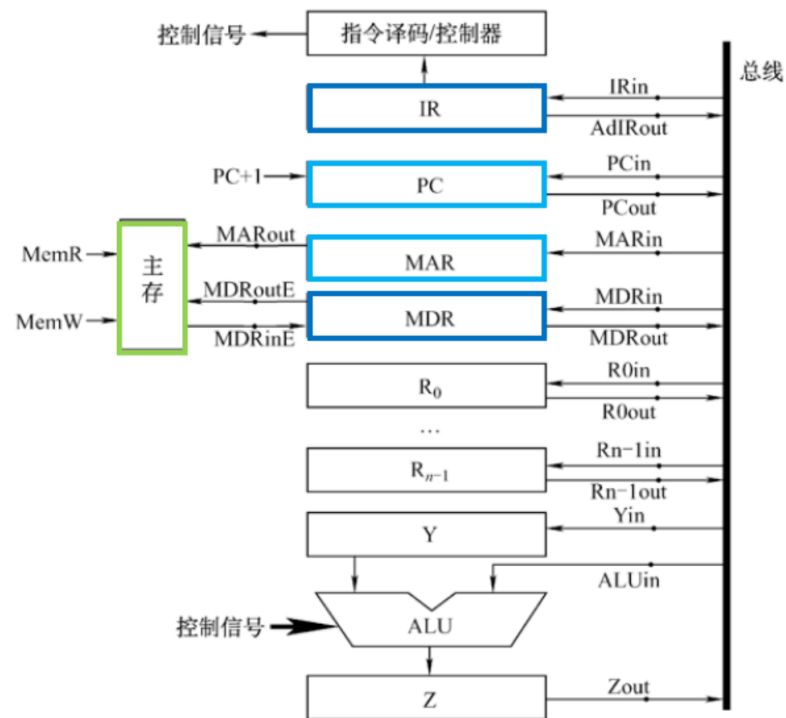


- 数据通路如图所示
- 请写出ADD [R0], R1指令的控制信号
- 功能:  $[R0] + R1 \rightarrow [R0]$

# 数据通路例题—取指



时序	微操作	有效控制信号
1	$(PC) \rightarrow MAR$	PCout, MARin
2	$M(MAR) \rightarrow MDR$	MemR, MARout, MDRinE
3	$(MDR) \rightarrow IR$	MDRout, IRin
4	指令译码	-
5	$(PC)+1 \rightarrow PC$	-

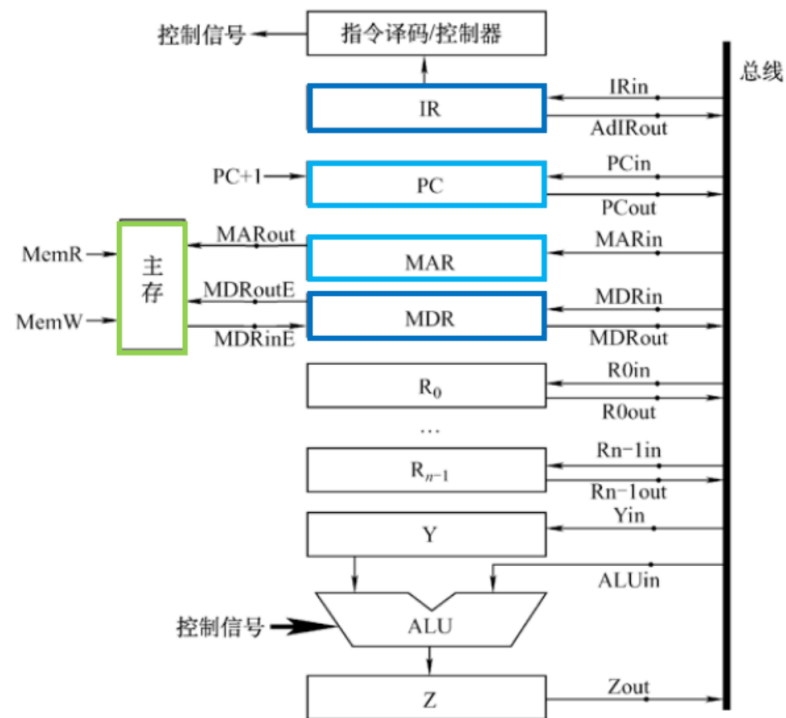


# 数据通路例题—执行



时序	微操作	有效控制信号
1	$(R0) \rightarrow MAR$	$R0_{out}, MAR_{in}$
2	$M(MAR) \rightarrow MDR$	$MemR, MAR_{out}, MDR_{inE}$
3	$(MDR) \rightarrow Y$	$MDR_{out}, Y_{in}$

时序	微操作	有效控制信号
1	$(R1) + (Y) \rightarrow Z$	$R1_{out}, ALU_{in}, CU \text{ 向 } ALU \text{ 发 } ADD \text{ 控制信号}$
2	$(Z) \rightarrow MDR$	$Z_{out}, MDR_{in}$
3	$(MDR) \rightarrow M(MAR)$	$MemW, MDR_{outE}, MAR_{out}$





# 总结

CPU功能：自动取出指令，译码、并执行

## 电位-脉冲制

### 多级时序系统

指令周期

CPU周期/机器周期-电位

T周期/节拍脉冲-脉冲

时钟周期

## 时间控制（时序信号）

## 结合课件时序图理解

定长控制

同步控制

不定长控制

中央与局部控制结合

### 控制方式

异步控制

无CLK基准时钟、大规模系统

联合控制

理解CPU各部分功能与数据通路构建

模型机/CPU基本组成

PC、IR、DR、AR、D-Cache、I-Cache、M...