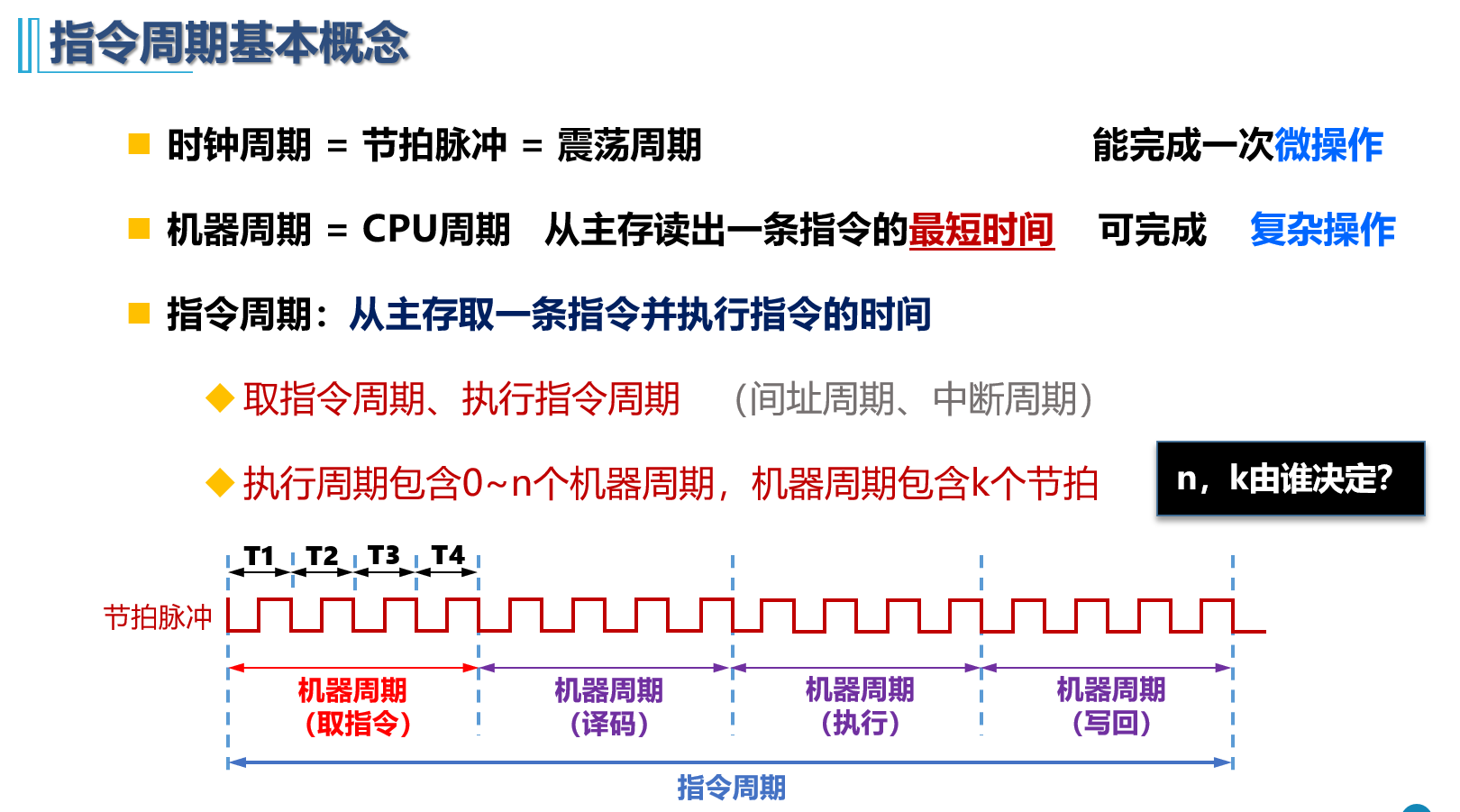
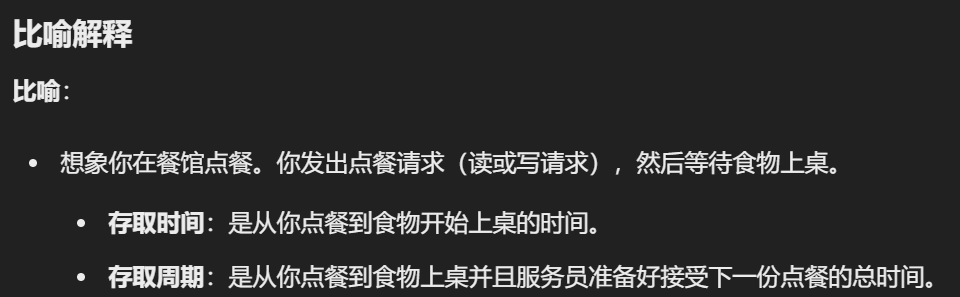
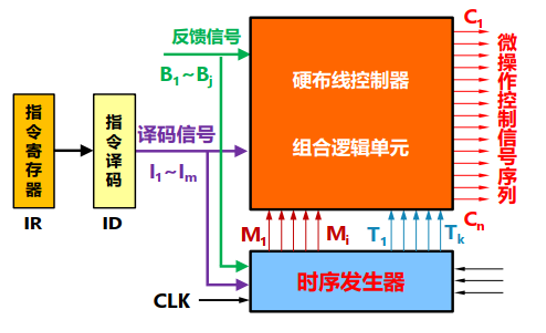
1. CPU由运算器和控制器组成
2. CPU的主要功能有：程序控制、操作控制、时序控制、数据加工、中断处理
3. CPU中主要的寄存器有：
4. 程序计数器（PC）
5. 地址寄存器（AR）
6. 数据寄存器（DR）
7. 指令寄存器（IR）
8. 通用寄存器组（GR）
9. 程序状态字寄存器（PSW/PSR）
10. 非必须的寄存器：DR、AR、IR
11. 用户可见存储器：GR、PSW、PC
12. 指令寄存器（PC）存放的是将要执行指令的字节地址，即指令指针寄存器，指令字要到主存中去找
13. 指令周期等周期的基本概念：
14. 指令的执行流程：

* 取指令，执行指令
* 取指周期，译码/取操作数周期，执行周期，写回周期

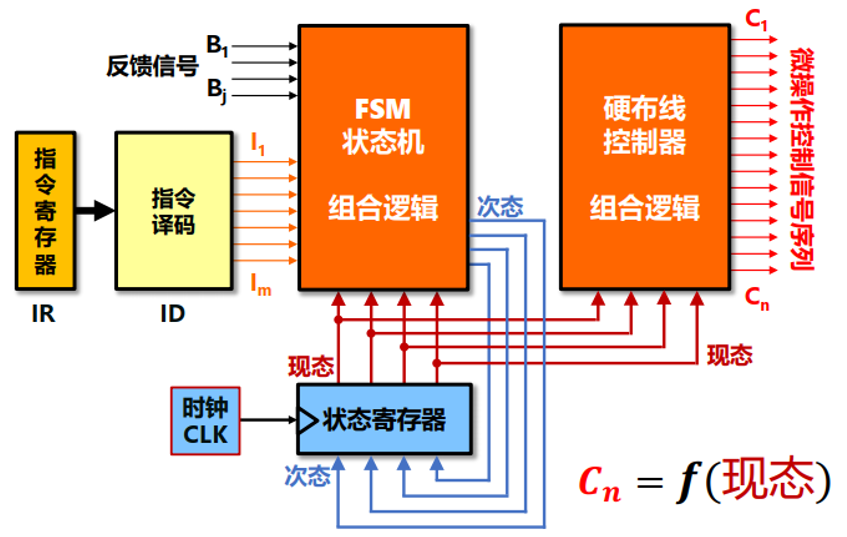
1. 存取时间是从发出存取请求（例如读或写）到数据开始传输之间的时间间隔。反映的是存储器响应请求的速度。
2. 存取周期是完成一次存储器操作所需的总时间，包括请求发出、数据传输和准备下一次操作的时间。它表示存储器完成一次操作后准备好接受下一次操作的时间间隔。
3. 存取周期包含存取时间



1. 数据通路的定时：PPT， P20~22
2. 单总线常见指令的数据通路：PPT， P39~46
3. 时序发生器的Input：节拍（时钟）脉冲；Output：节拍电位、周期电位
4. 时序发生器将时钟脉冲转化为节拍电位、状态周期，进而为控制信号的产生做准备
5. 传统三级时序硬布线控制器通过节拍电位、状态周期、指令类型、反馈信号产生控制信号，纯组合逻辑电路



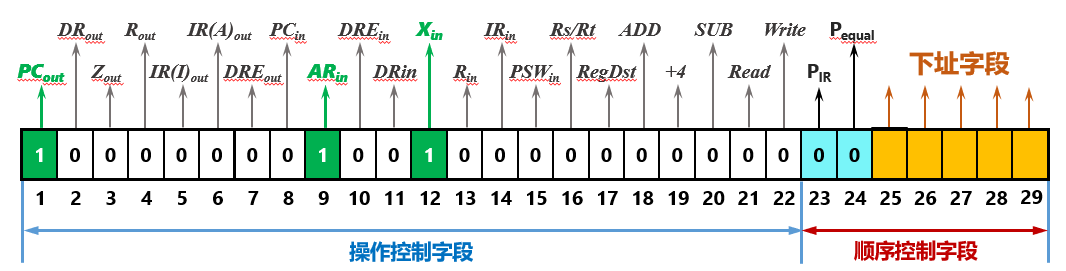
1. 现代时序硬布线控制器对每一个状态进行编号，每一个状态对应单独的控制信号序列，因此执行一条指令需要多少时钟周期就安排多少时钟周期，不会存在任何时钟节拍的浪费。控制信号由现态产生，而次态由状态机产生。



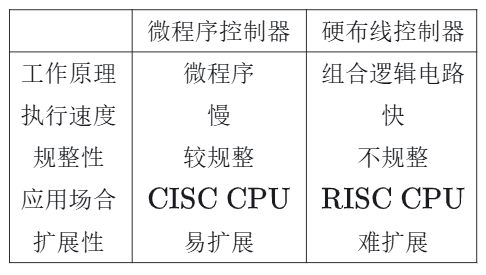
1. 微指令包括操作控制字段和顺序控制字段，操作控制字段就是控制信号的序列，顺序控制字段包括判别测试字段P和下址字段，P判断指令执行过程中的外部条件，下址字段给出下条微指令的地址。
2. 程序是一系列机器指令的有序集合；微程序是完成一条指令的若干条微指令的集合，微指令存放于控制存储器中。控制存储器在CPU内，主存在CPU外。都是按地址访问。控制存储器不一定要求读写存储器，用EPROM（可擦除可编程只读存储器）

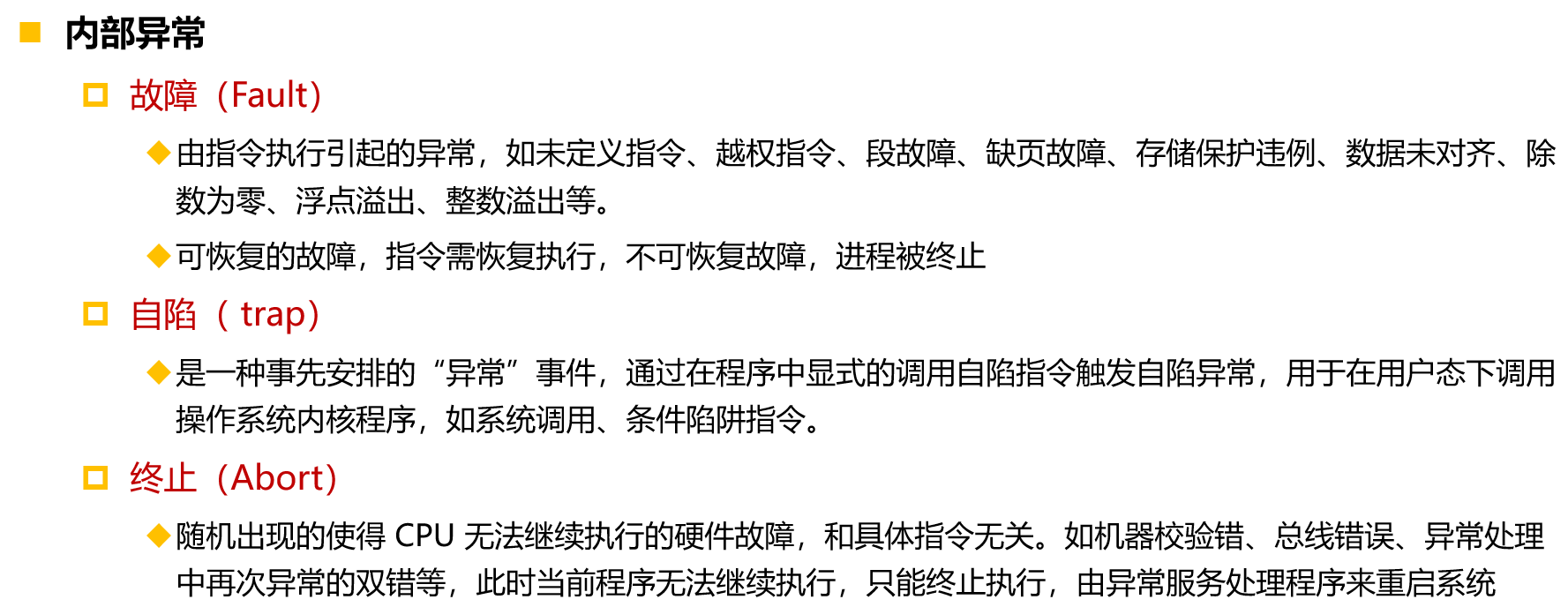


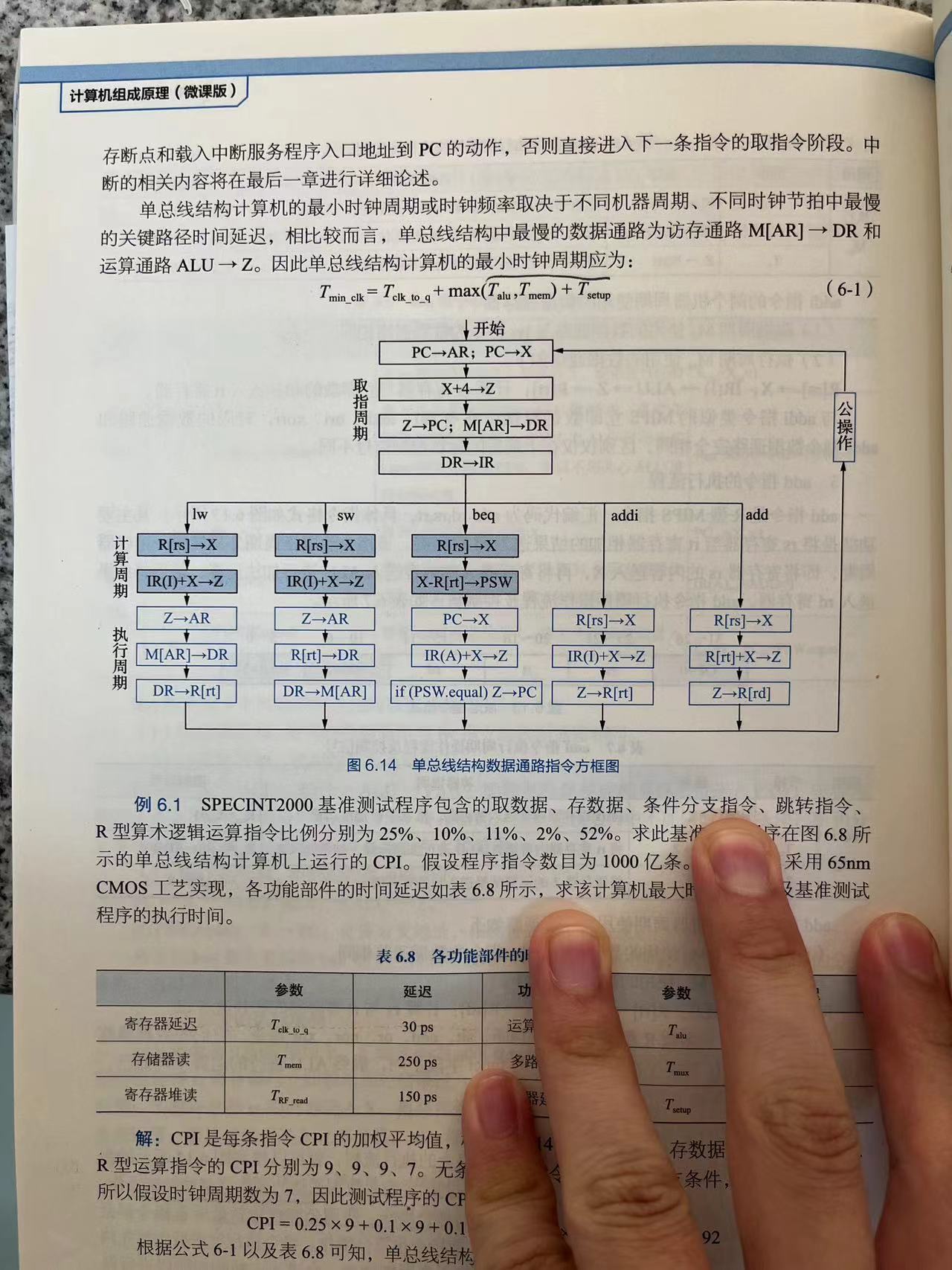
1. 微指令的三个字段：操作控制、判别、下址

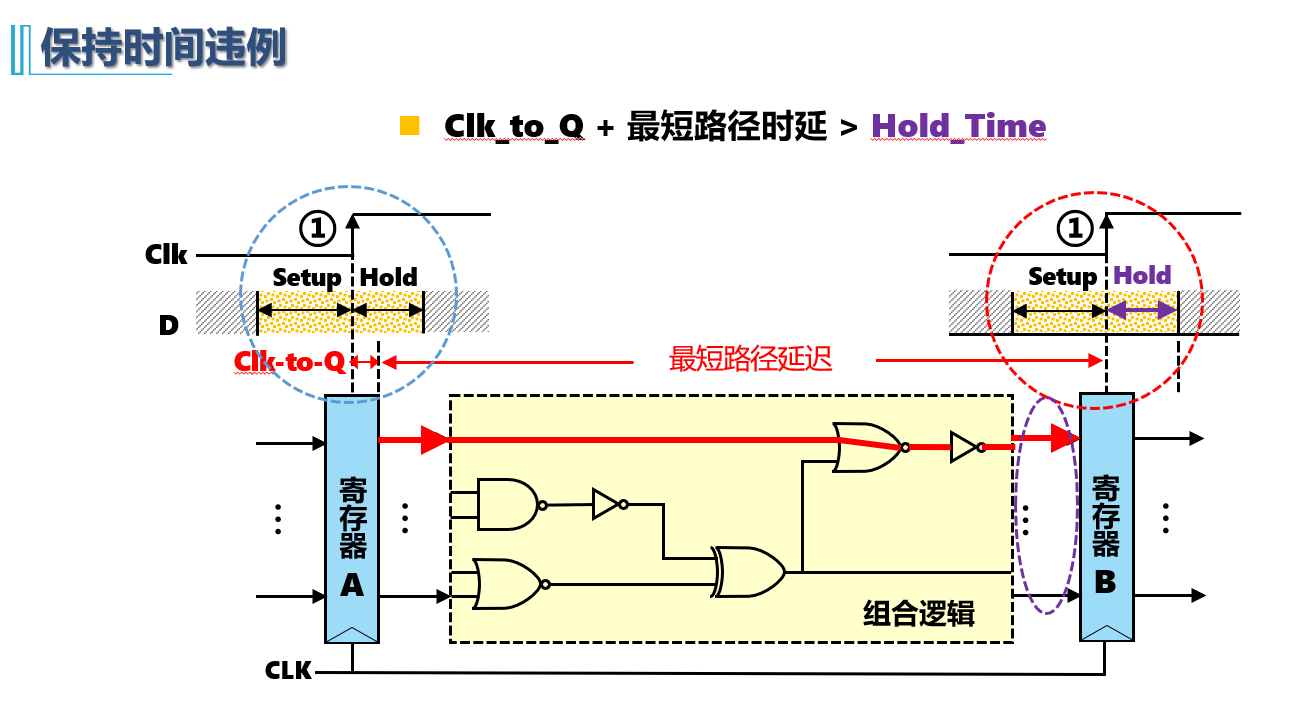


1. 微指令的编码表示法要注意，译码器的输出端需预留一个来表示所有的互斥控制信号均不使用，比如，编码后的3位字段只能表示原来的7个互斥控制信号，8个控制信号需要4位编码。
2. 微命令就是微指令的1位控制信号，比如PCin
3. 微程序控制器对应CISC，硬布线控制器对应RISC。



1. 单周期不能设置AR,DR,IR寄存器，程序和数据分开存放---哈佛结构
2. 异常（内部中断）：包含故障、自陷、终止。指的是CPU内部引起的异常事件。
3. 可恢复故障修复后回到当前指令，不可恢复的故障结束进程。
4. 自陷很像“断点”，调用操作系统内核程序并执行，完毕后回到自陷指令的下一条指令执行
5. 外部中断：由外部设备向CPU提出中断请求，要求CPU暂停当前执行程序，转去执行为某个外部设备事件服务的中断服务程序，处理完毕后返回断点继续执行。
6. 外部设备中断的时机是一条指令结束之后，故外部中断保存的断点是下一条指令的地址。
7. 单总线最小时钟周期的计算





1. 单周期最小时钟周期的计算

6项，存储器时间2倍



