

101-104

- **著名计算机**

- **远古：** ABC, ENIAC, EDVAC, EDSAC, UNIVAC
- **大型：** IBM S/360
- **超级：** CDC6600
- **小型：** PDP-8
- **微型：** Altair 8800, Apple II, IBM PC 5150

- **三个约翰：** 冯诺依曼， 莫克利， 埃克特

- **关于EDVAC的报告草案**

- 开关连线→存储程序
- 十进制→二进制
- CA、CC、M、I、O

- **冯诺依曼结构的要点**

- 五个组成部分
- 数据与程序存放在同一存储器
- 自动取指

105-107

- 模型机
 - 控制器：IR、PC、MAR、MDR、指令译码、控制电路
 - 存储器：MAR、MDR、控制逻辑、地址译码器
 - 运算器：通用寄存器、ALU、XYZ、F
 - 控制总线、地址总线、数据总线
 - CPU内部总线
- 指令执行的4个步骤
- 独立芯片
- 南桥芯片
 - 磁盘、键盘、鼠标、音频、网络、USB等设备的控制器

201-202

- Intel 8086

- 16位通用寄存器
- 16根数据线, 20根地址线
- 段加偏移

- 标志寄存器

- 状态标志: A C O P S Z
- 控制标志: D I T

- Intel 80386

- 32位通用寄存器
- 32位地址总线
- 改进了保护模式
- 增加了虚拟8086模式

- 实模式与保护模式的区别

- 寻址方式: 如何取段基址、wrap-around
- 段大小: 固定64KB/4KB对齐
- 段基址的存放地点: 段寄存器/段描述符
- 段保护机制

203-205

- 传送类指令

- MOV **WORD PTR** [BX+2*SI+500], 01H

- 逻辑运算、移位指令

- 立即数不能是唯一的操作数

- 条件转移指令

- JNZ/JNE: ZF=0
 - JB: CF=1
 - JA: CF=0 && ZF=0
 - JG: ZF=0 && SF=OF

- 控制类指令

- STC/CLC/CMC/STD/CLD/STI/CLI
 - HLT/WAIT/ESC/LOCK
 - NOP

- 串操作指令MOVSB

- copy string from **DS:SI** to **ES:DI** for **CX** byte(s)
 - **DF=0** \longleftrightarrow increasingly
 - REP MOVSB

- MIPS指令的特点

- 只有LOAD和STORE可以访存
 - 需要优秀的编译器支持

206-306

- MIPS分支指令

- 条件: $PC = PC + 4 + 4 * imm$
- R型非条件: $PC = R[rs]$
- I型非条件: $PC = \{(PC +$

- 进位/溢出
- MIPS/x86对溢出的处理方式
- 实现减法运算的trick
- RCA/CLA的关键路径、进位输出信号
- 通常把多个小CLA拼起来

401-606

- 两版乘法器、并行优化
- 两版除法器、无法并行优化
- 简单的CPU、IFU、控制信号
- 超标量结构=超标量流水线=超标量
- 超标量处理器
- 单周期→标量流水线
 - 时间并行性，切分现有的硬件资源
- 标量流水线→超标量流水线
 - 空间并行性，成倍增加硬件资源
- 现代的多核CPU通常是在一个CPU芯片中集成了多个超标量处理器核

• 结构冒险

- lw访存时，后面的指令要取指
 - bubble/指令和数据分开存放
- lw写回时，后面的指令要译码
 - 分开读写口，先写后读

• 数据冒险

- 一般把访存/ALU运算结果前递给ALU
- Load-Use Hazard，例如lw+add

• 控制冒险

- I型无条件：取指时确定，无影响
- R型无条件：译码时确定，1 bubble
- 条件：执行时确定，2 bubbles
 - 增加额外电路，译码时确定，1 bubble

701-706

- DRAM/SRAM的写入、读出
 - 存储单元、集成度、功耗、价格、速度、刷新
- tRCD, CL, tRP
- SDRAM读操作的典型访问过程
 - 行地址-tRCD-列地址-CL-数据-预充电-tRP
 - 预充电信号和倒数第二个数据信号同时发出
- 带宽=等效频率*64bit/(8bit/byte)
- 核心频率： I/O频率： 等效频率
 - SDR 1:1:1; DDR 1:2:2; DDR2 1:2:4; DDR3 1:2:8
- 写回、写穿透
- 写分配、写不分配
- 如何计算平均访存时间
- 路、组、tag、offset
- 磁盘1KB=1000B, 内存1KB=1024B

801-806

- 实模式下中断向量表的结构
- 保护模式下如何找到中断向量
- 中断处理的六个步骤
 - 每一步都做些什么？
 - 软硬件如何分工？
 - 保存断点和保护现场的区别是什么？
- 内部中断处理的十步图
- 五个专用中断
 - 区分单步中断和断点中断
 - 单步中断比较特殊（优先级、用TF位屏蔽）
- BIOS中断
- DOS中断（INT 21H）
 - 进一步屏蔽了设备的物理特性及接口特性
 - 功能更加常用

901-907

- I/O接口的六个基本功能
- x86分开编址
- ARM/MIPS/PowerPC统一编址
- IN to cpu, OUT to device
 - 可以用DX指定端口号
- 优缺点
 - 指令功能
 - 内部结构, CPU引脚数目
 - 指令长度, 指令执行时间
 - 占用存储空间
 - 易读性
- 无条件传送
 - 程序简单, 对外设要求低, 但只适合简单外设
- 程序查询方式
 - 准确可靠, 对外设要求低, 但效率低
 - 输入输出的六步
- 中断方式
 - 并行效率高, 外设有主动权, 实时性要求, 但进入和退出中断需要额外的指令
 - 输入输出的六步
- DMA方式的八步
- DMA的配置寄存器
 - 地址初始值 (端口地址? 内存地址?)
 - 地址增减方式 (端口地址一般不变)
 - 数据长度 (可不设置)

第八讲：流水线优化技术

- 流水线的发展变化
 - 没什么理论知识，只是想说明流水线朝着多级、超标量的方向发展了
 - 转移指令的影响
 - 改变指令流向，破坏流水模式
 - 如何减小转移指令的影响
 - 延迟转移
 - 转移预测技术
 - 转移预测技术
 - 要不要转移
 - 转移到哪里
- 为什么可以用RAS?

 - 过程结尾一定有RET
 - RET一定会转移
 - 转移目标地址容易预知

为什么不用BTB?

 - 同一条RET的转移地址往往不同
 - 转移目标地址在存储器中，访问时间较长，且在流水线晚期才能获得
- 要不要转移
 - 硬件总是预测不转移
 - 编译制导：在转移指令中多加一位，由编译器设置是否转移
 - 基于偏移：例如BTFN
 - 基于历史信息
 - 历史信息转移预测器
 - Pentium版本
 - 无倾向性的版本
 - 一般在首次进入循环和退出循环时出错
 - 转移目标缓冲器（BTB）
 - Pentium BTB的工作机制示意图
 - 如果BTB不命中&&发生转移，那么新建表项，设定历史位为11

第九讲：输入输出接口

- 串并行的比较
 - 传输线数量、传输率、串/并转换、信号线串扰
- 差分信号传输技术
 - 抗干扰，时序定位准确
 - 布线难度高
- 可编程并行接口电路8255A
 - 端口ABC、数据、地址、控制、复位、电源、地线
 - ABC的功能区别
 - 端口选择
 - 方式012
 - 方式选择控制字
 - 端口C按位置1/置0控制字
- XLAT指令
 - $AL = [BX+AL]$

第十讲： 中断控制器和定时器

- MIPS CPU的异常处理
 - 在EPC保存出现异常的指令地址
 - 需要将当前保存的指令地址-4???
 - 清空流水线之后的指令
 - 记录产生异常的原因
 - 跳转到特定地址
- x86系统如何获取中断处理程序的入口地址
- 如何确定外部中断优先级
 - 软件查询
 - 硬件中断优先级编码电路（菊花链）
 - 可编程中断控制器
- Local APIC与I/O APIC如何协同工作
- 定时器Intel 8253
 - 计数器、控制字、数据、读写逻辑
 - CLK、GATE、OUT
 - 方式2分频器、方式3方波发生器
- 8253在IBM PC/AT中的连接结构
 - OUT_0 : 系统时钟, 接 IR_0
 - OUT_1 : DRAM刷新
 - OUT_2 : 扬声器
- 南北桥架构演变

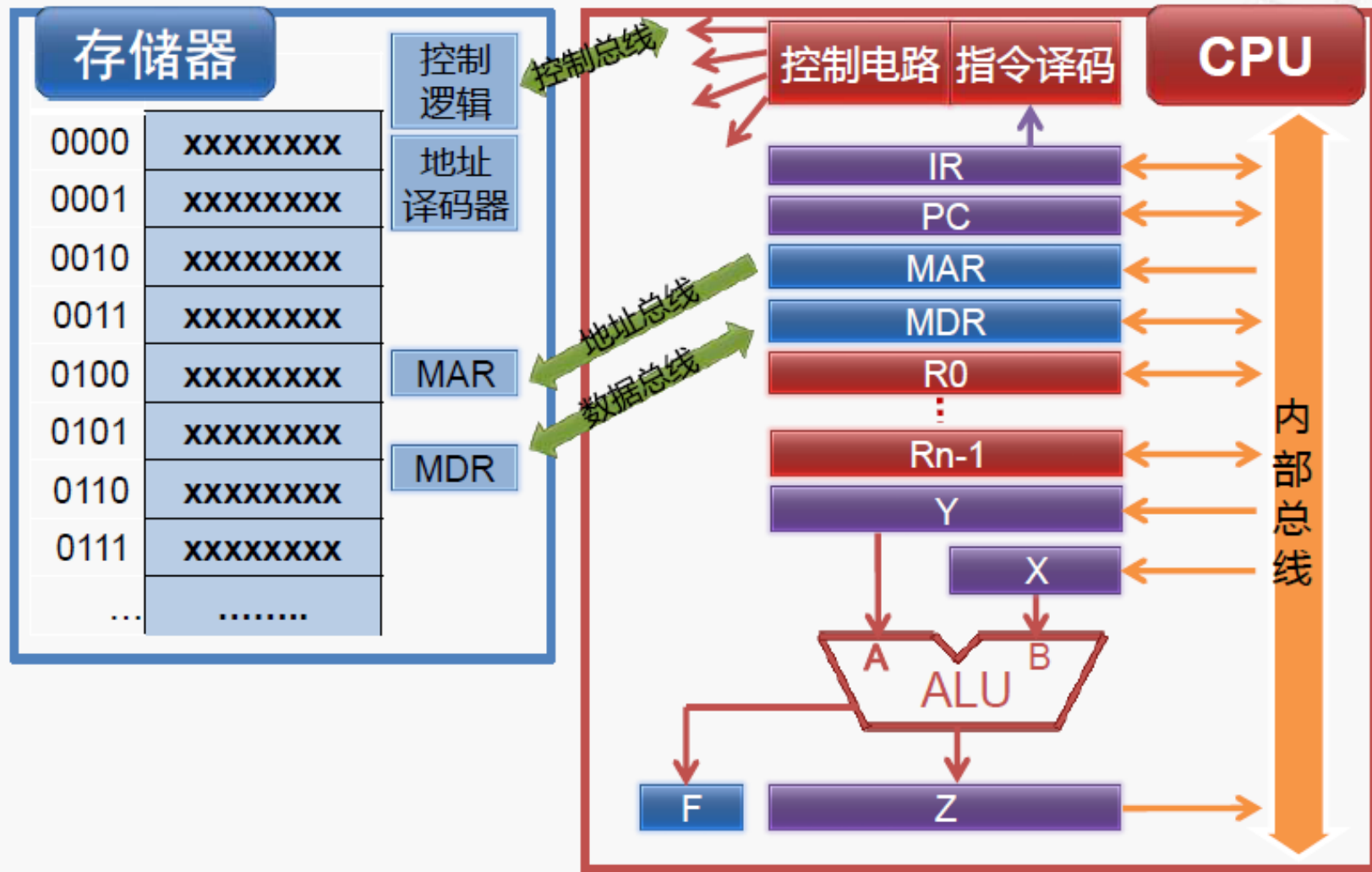
第十一讲：总线与总线标准

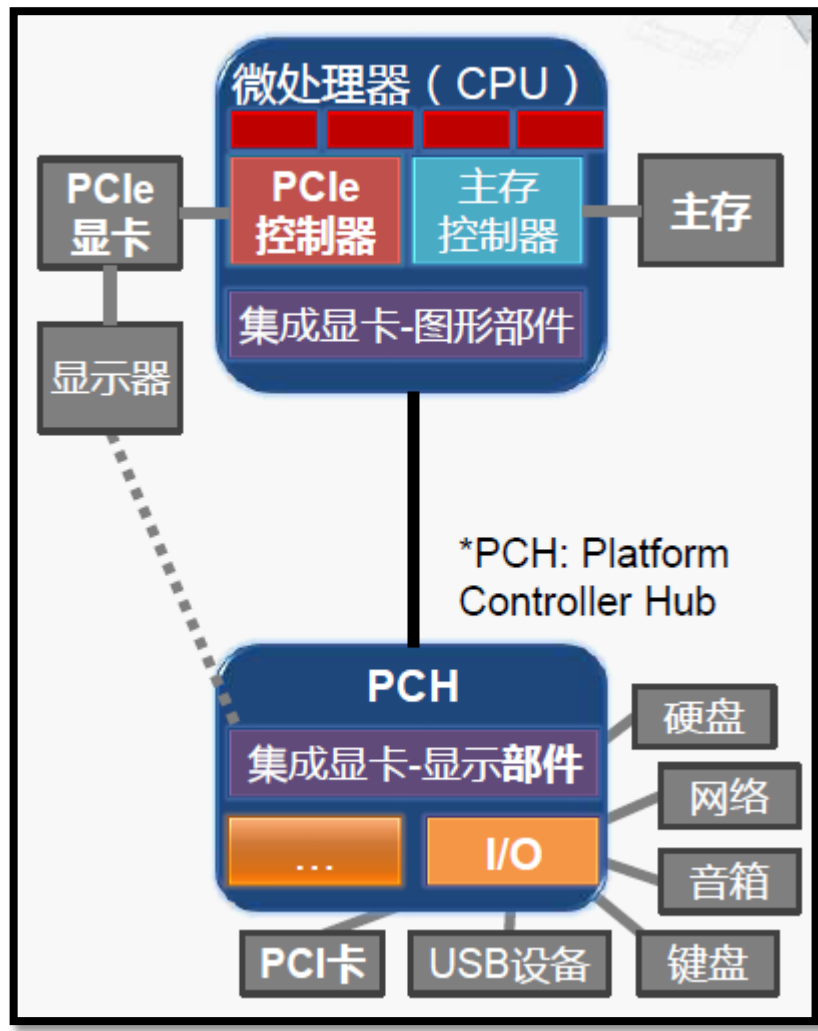
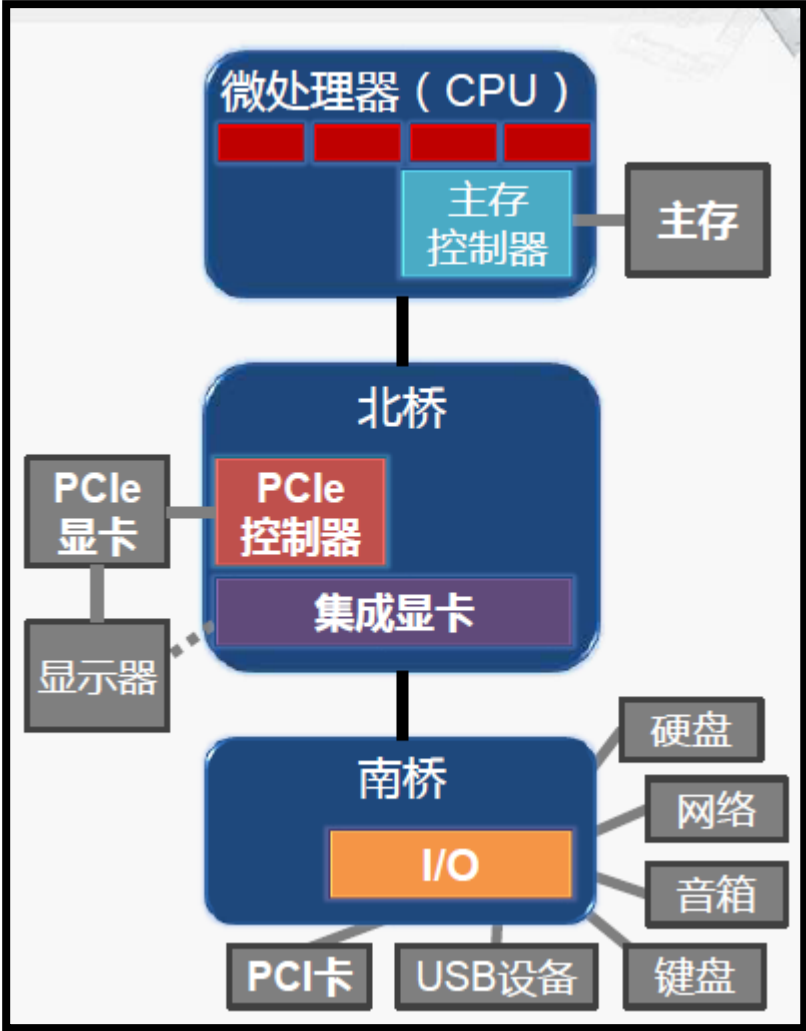
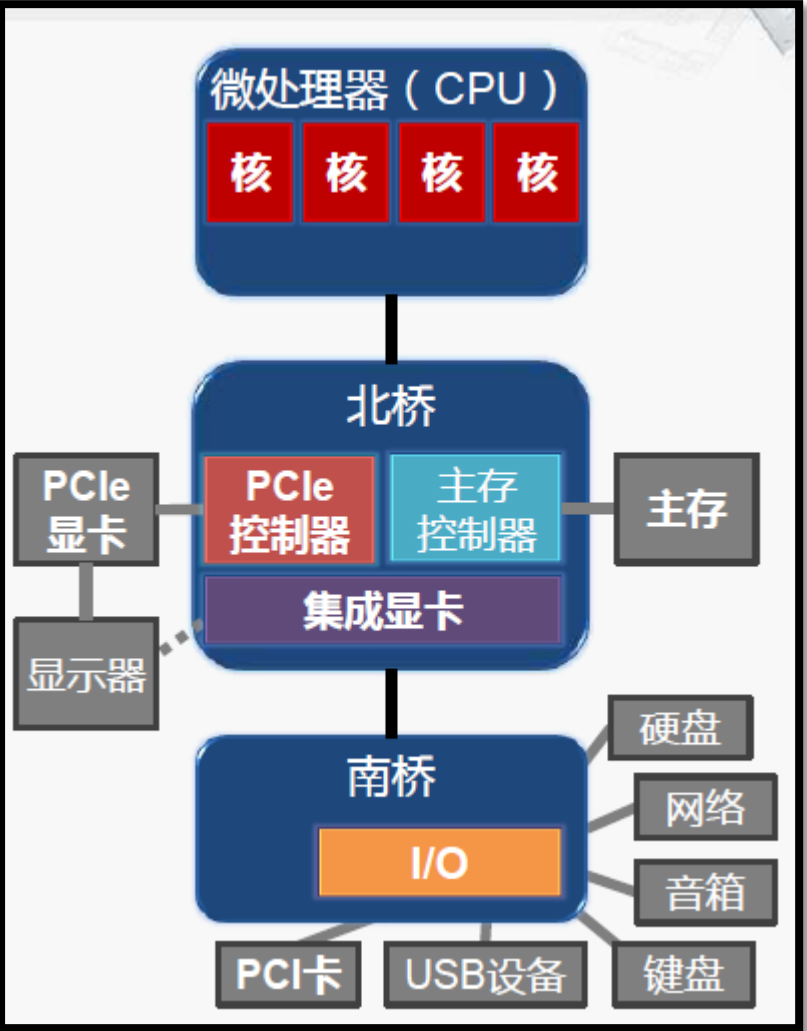
- 片上总线，系统总线，通信总线
- 译码器、仲裁器
- 总线逻辑电路示意图
- PCI Express (PCIe)
 - 串行、高频
 - 全双工、点对点（而不是共享）
 - 差分信号传输
 - 峰值带宽=总线频率*数据位宽*2（GT/s）
 - 端到端，两端都含有RX和TX
- Advanced Microcontroller Bus Architecture
 - Advanced High-performance Bus, from AMBA2
 - AHB的基本结构
- AHB读写一个数据
- 从模块插入等待周期的读写传输
- 三次连续的总线传输
- HTRANS和HBURST信号示例
 - HTRANS: IDLE, BUSY, NONSEQ, SEQ
 - HBURST: SINGLE, INCR(1,4,8,16), WRAP(4,8,16)
 - 高速缓存的行填充（读）、行替换（写）
 - 外设、主存的大量连续数据传输
- 地址回卷的四个数据的传输
 - “关键字优先”的高速缓存行填充

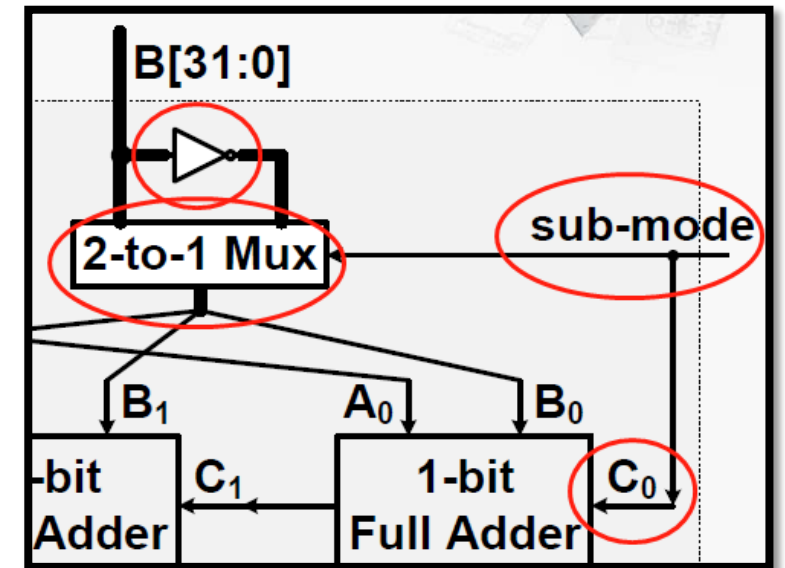
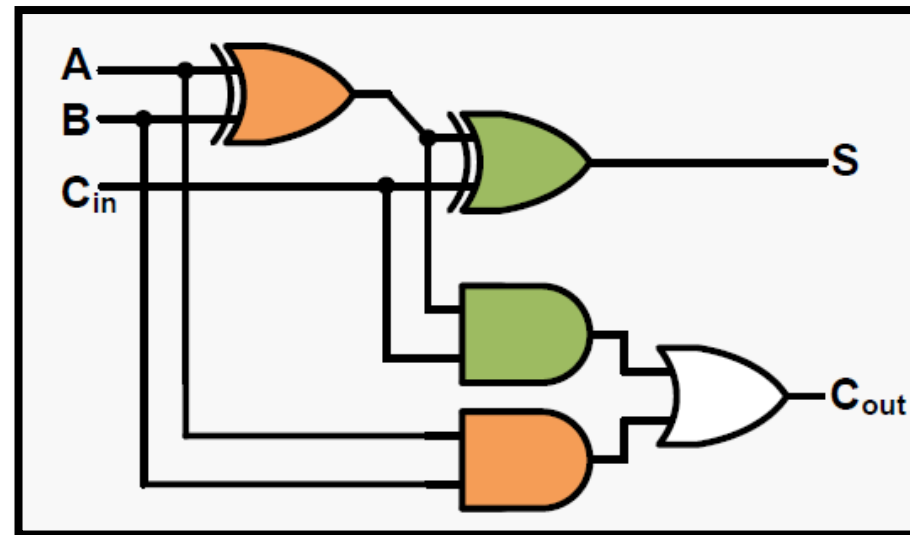
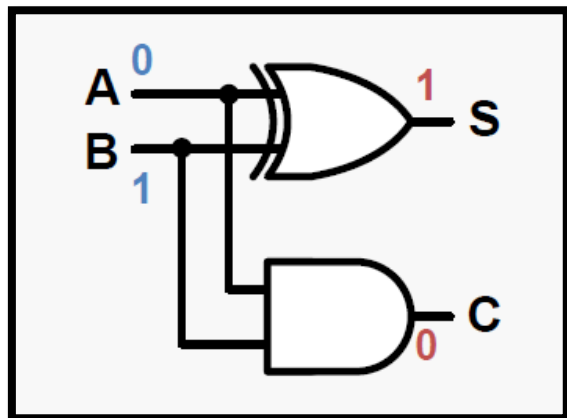
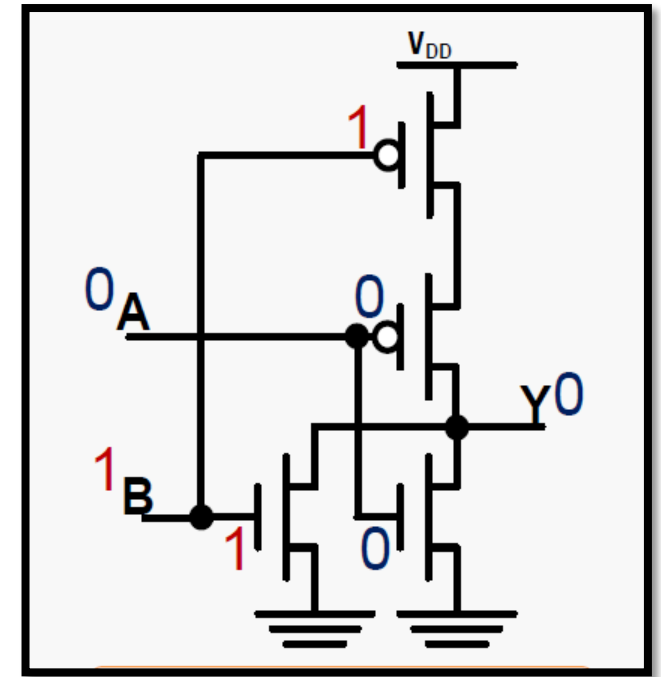
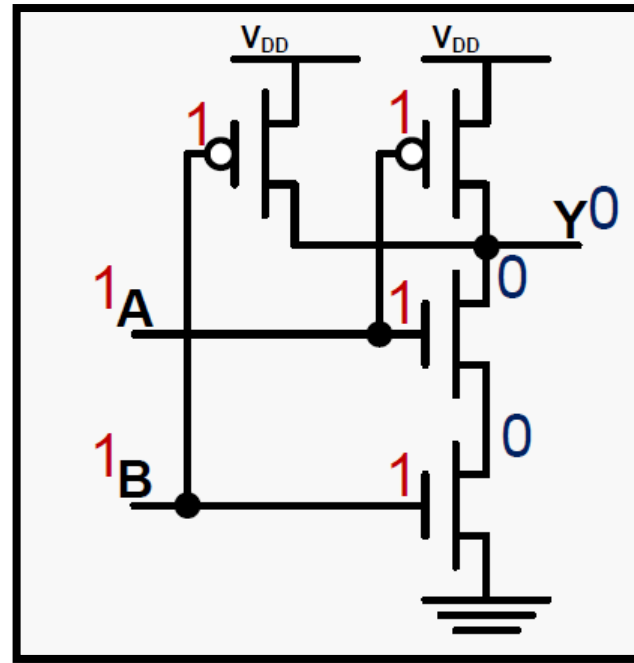
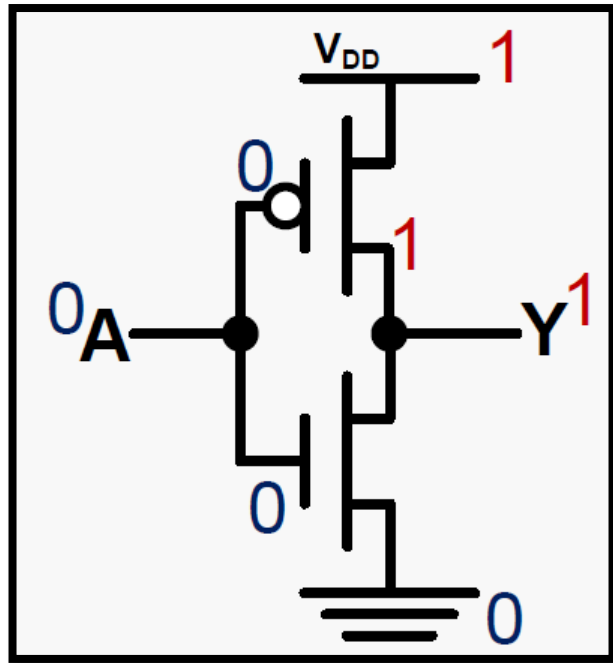
第十二讲：存储结构

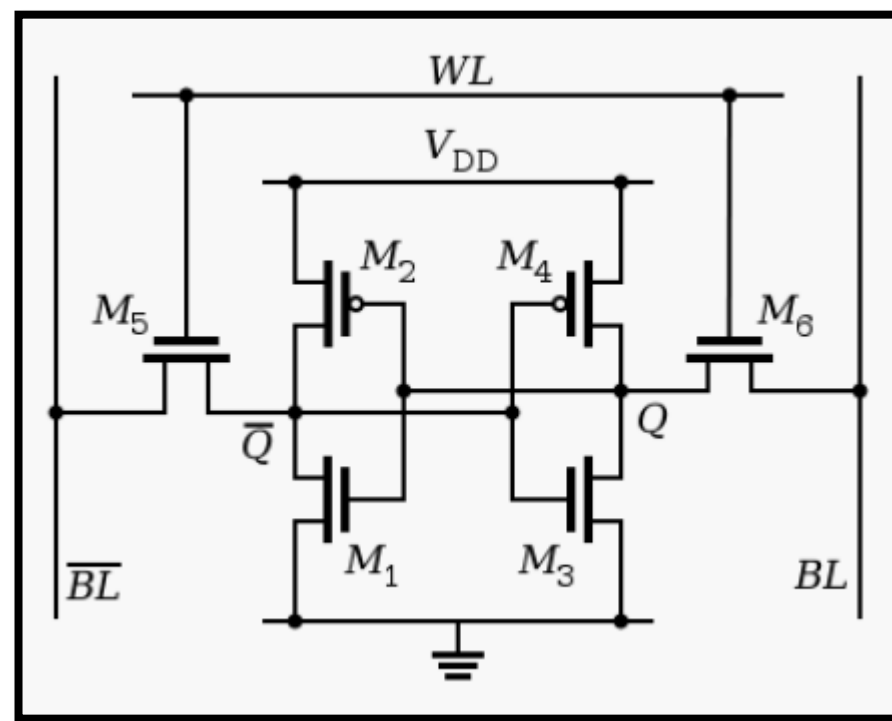
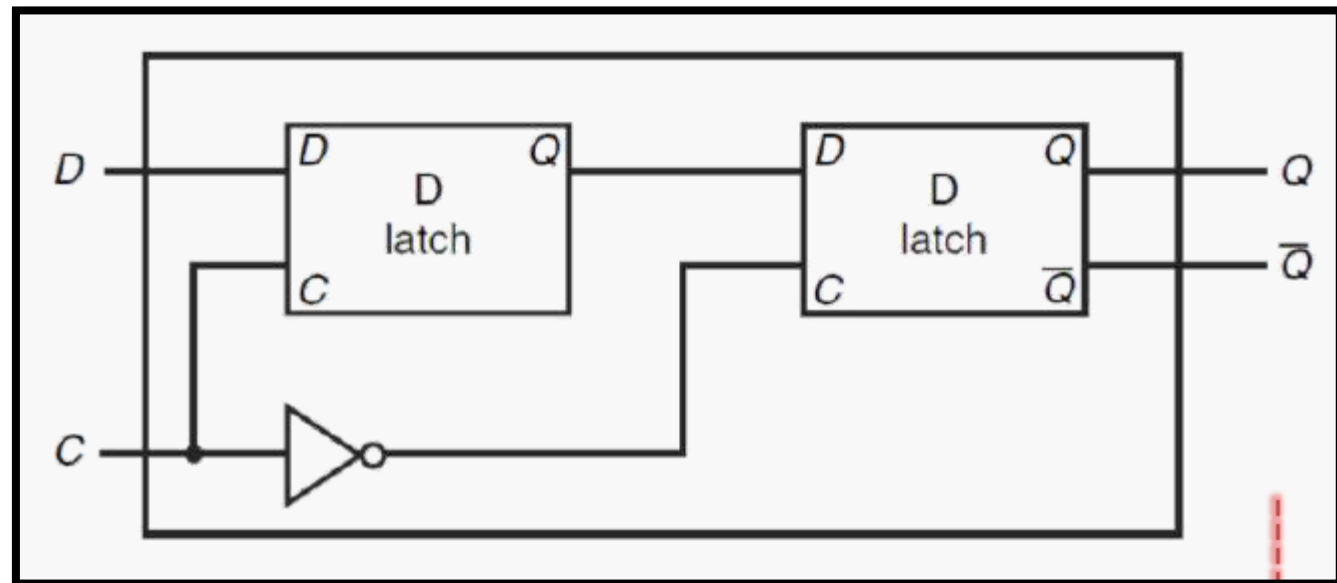
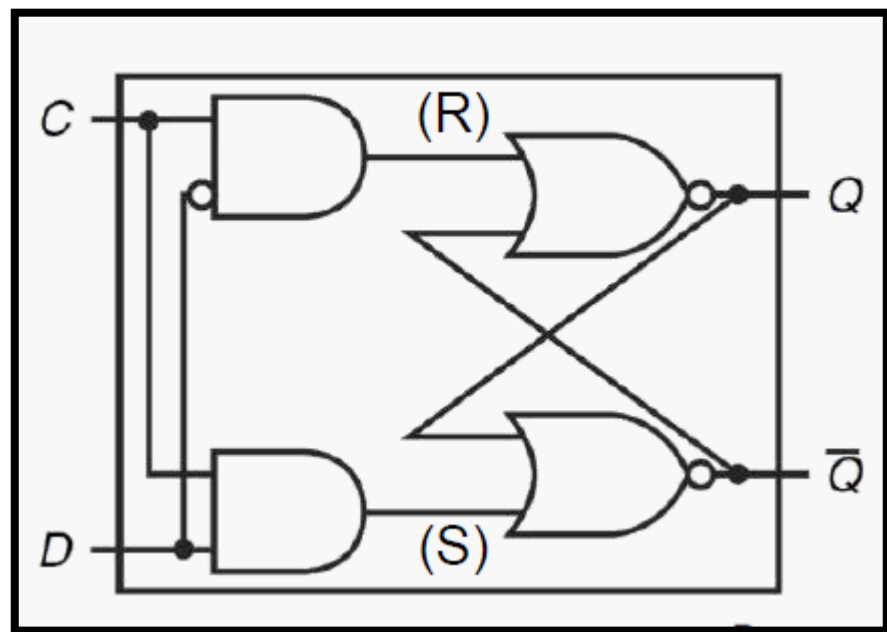
- Page Miss, Page Hit, Page Fast Hit
 - 读操作时序图、示意图
- DDR4的Bank Grouping

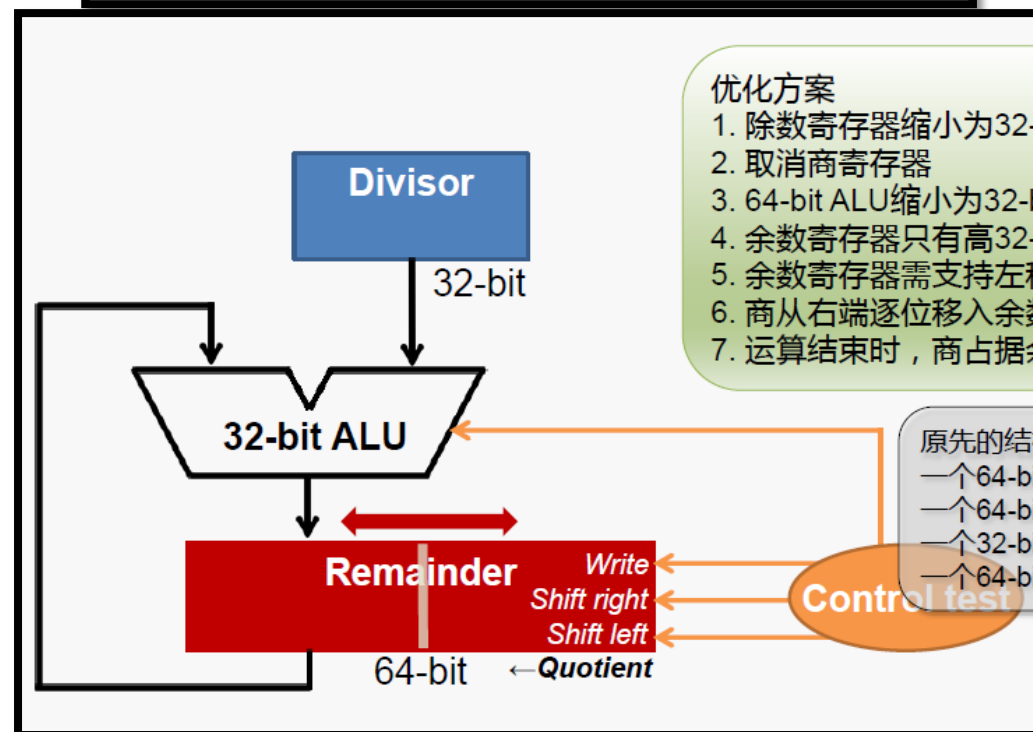
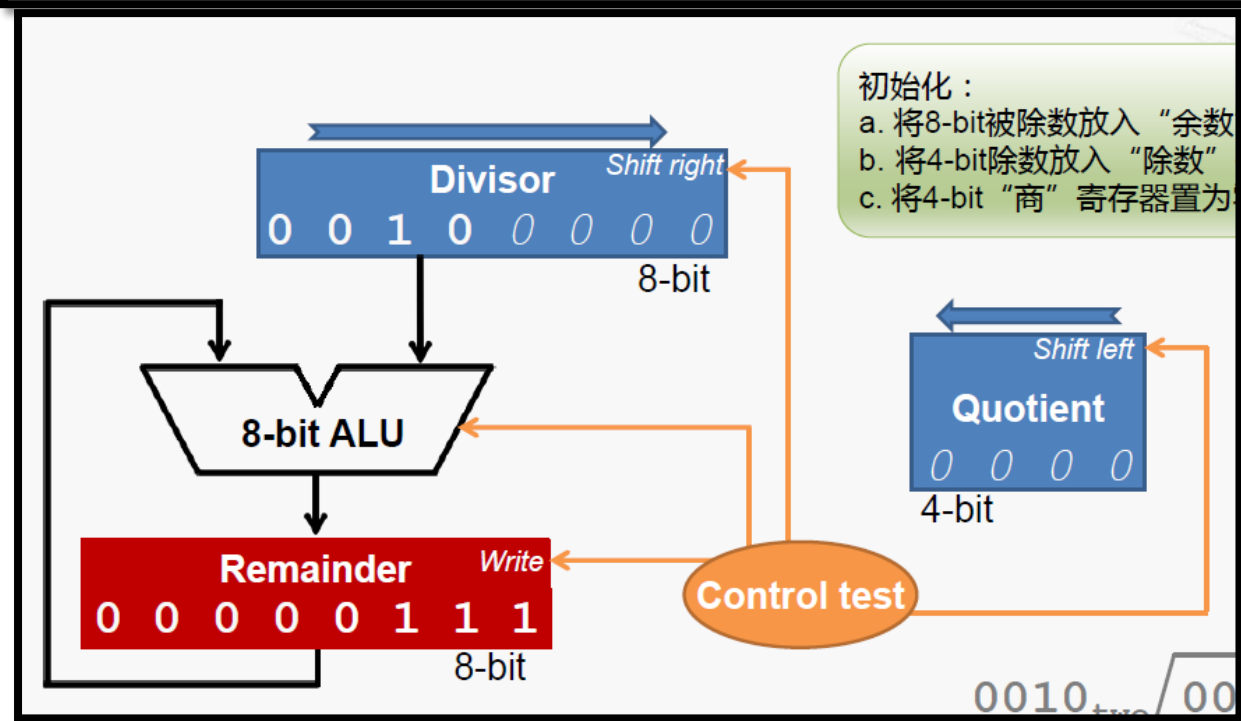
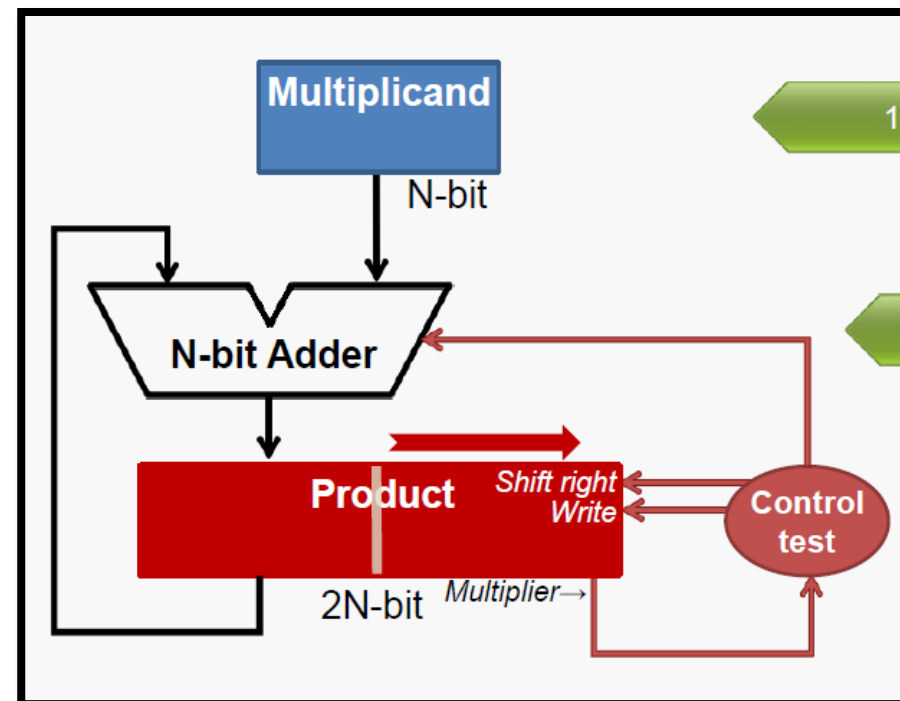
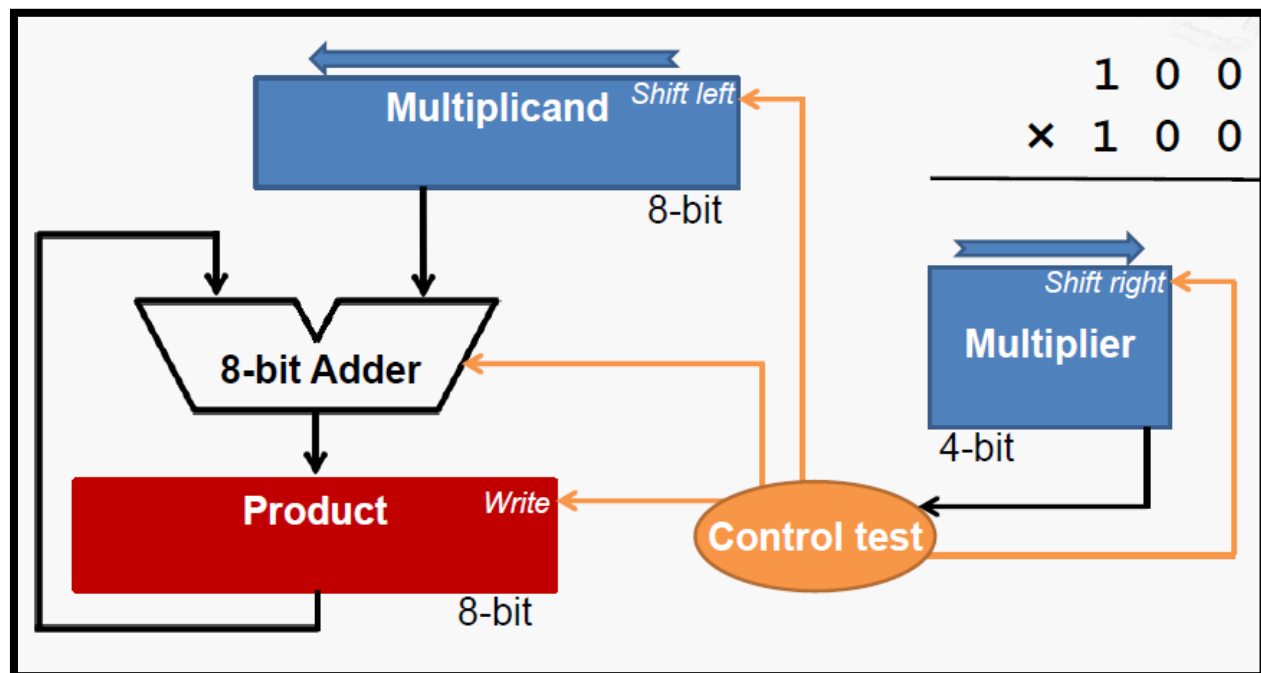
计算机结构的简化模型（模型机）





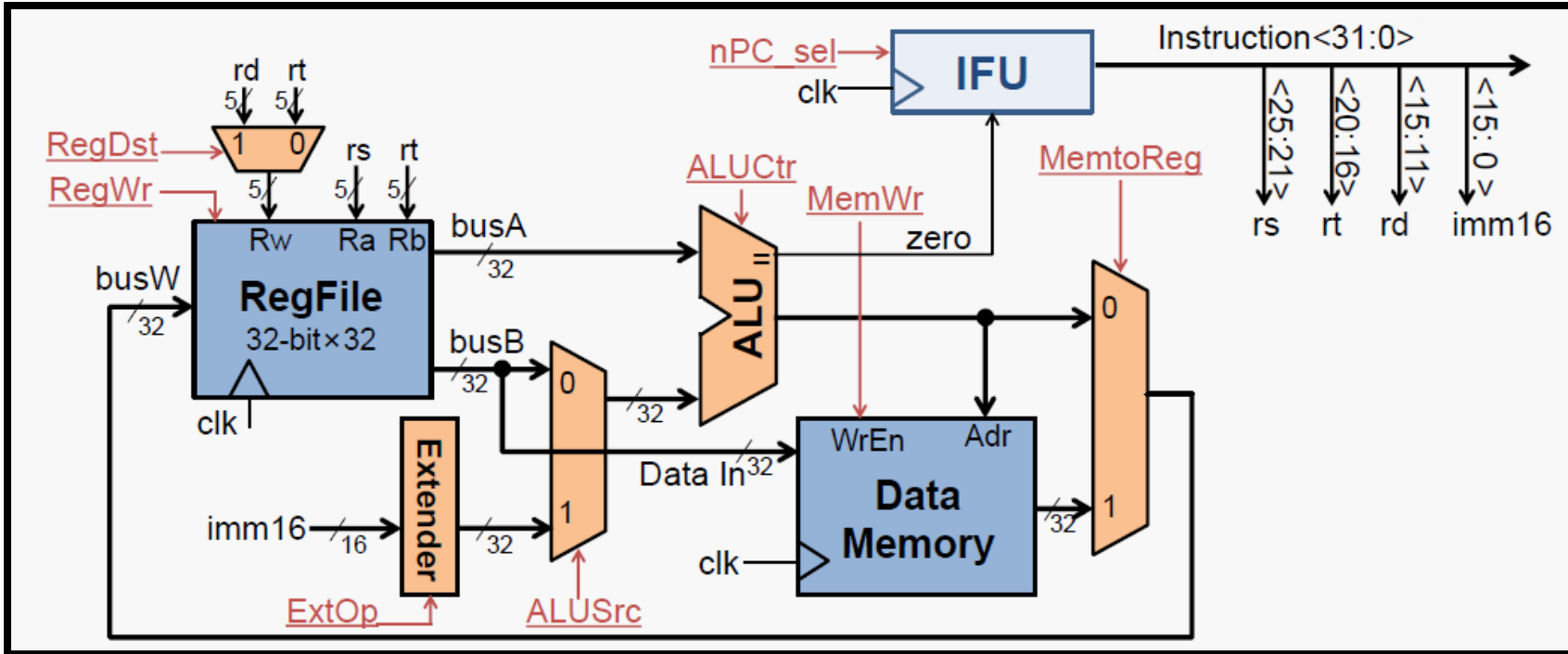
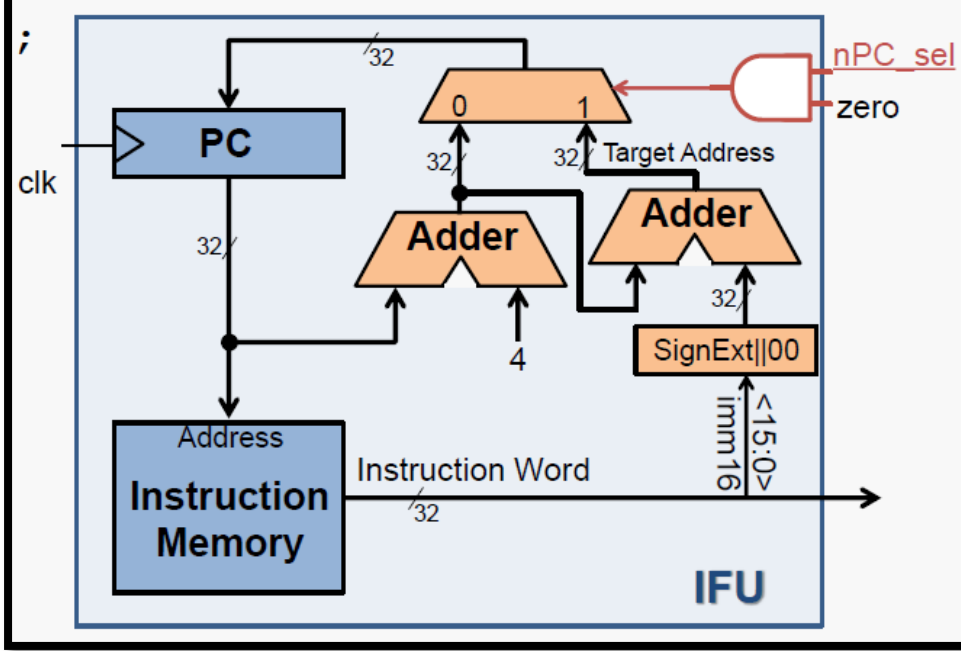


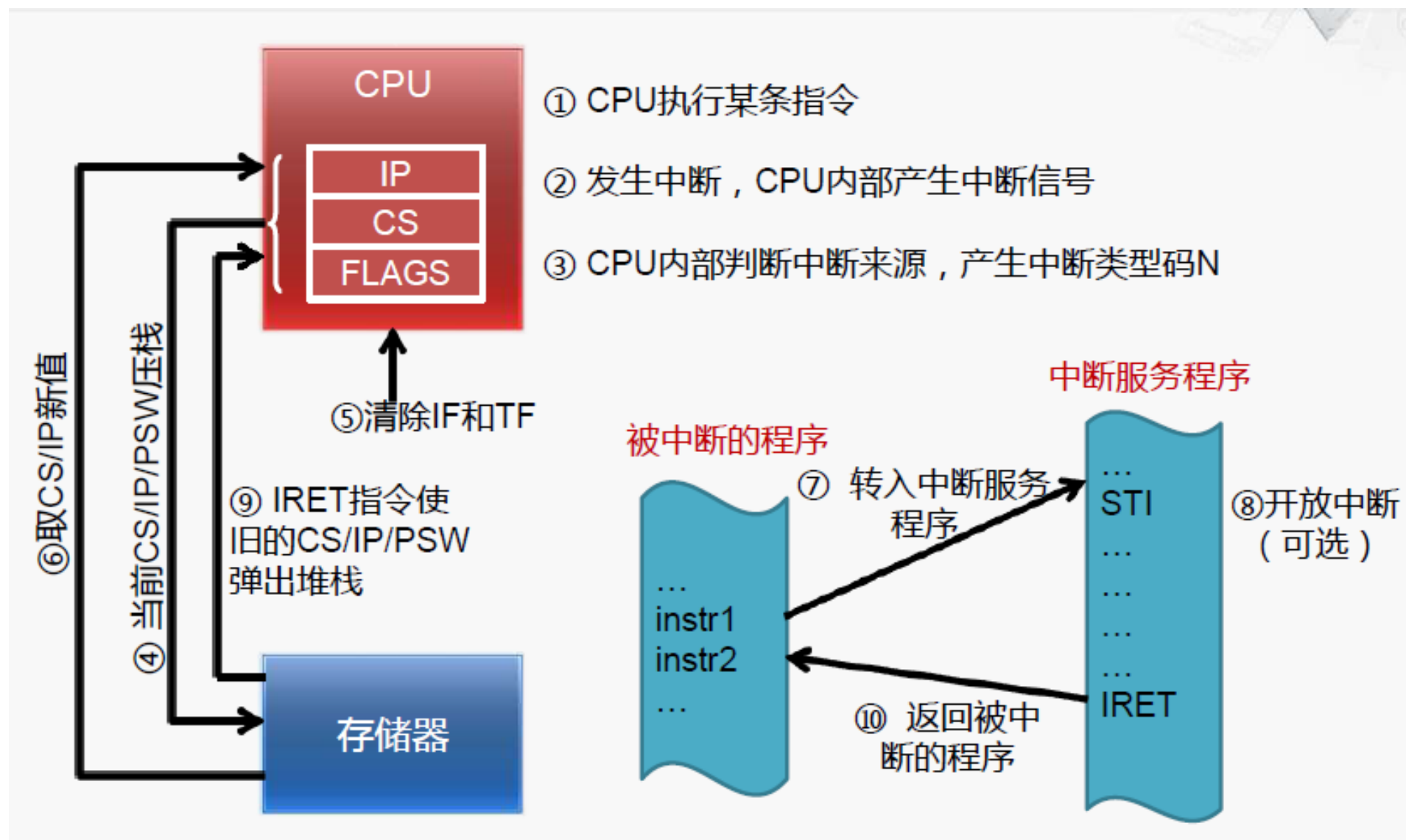


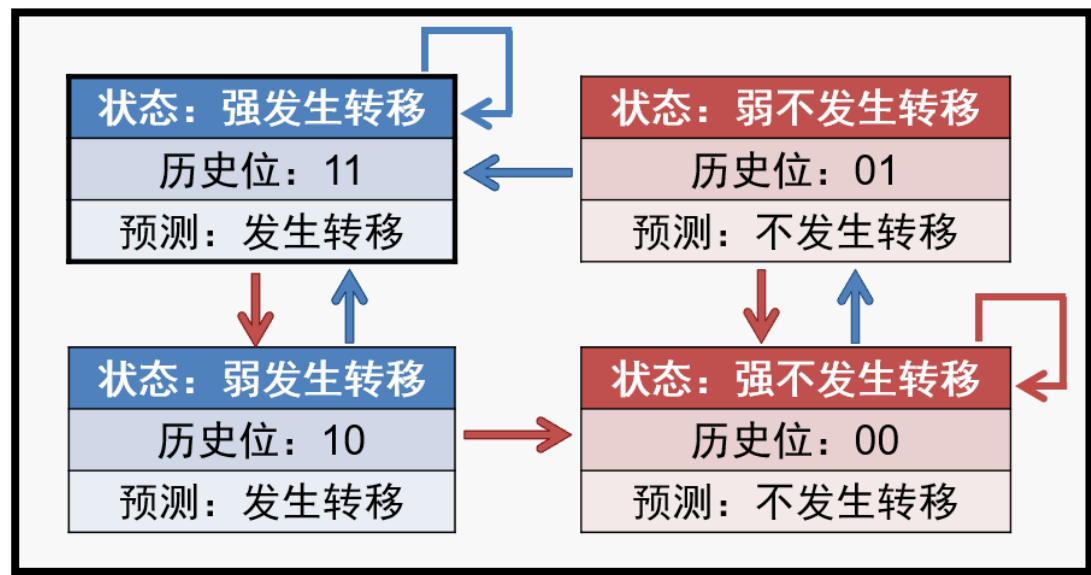
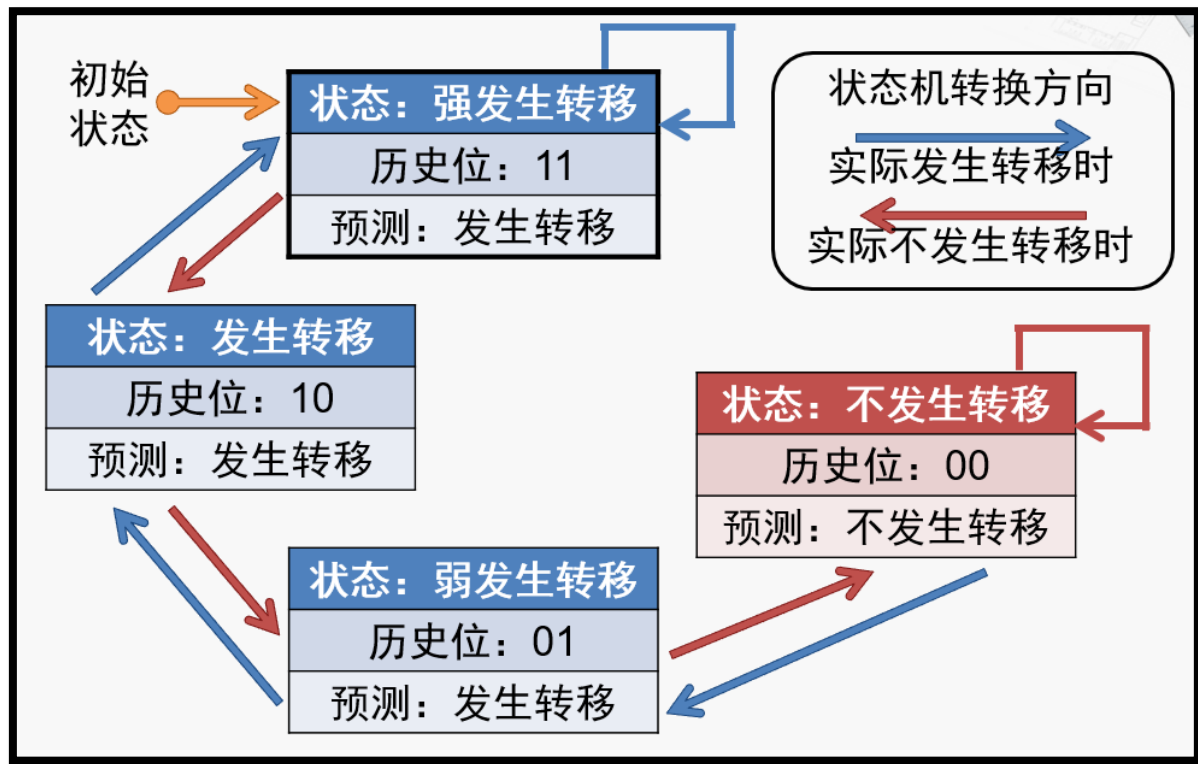


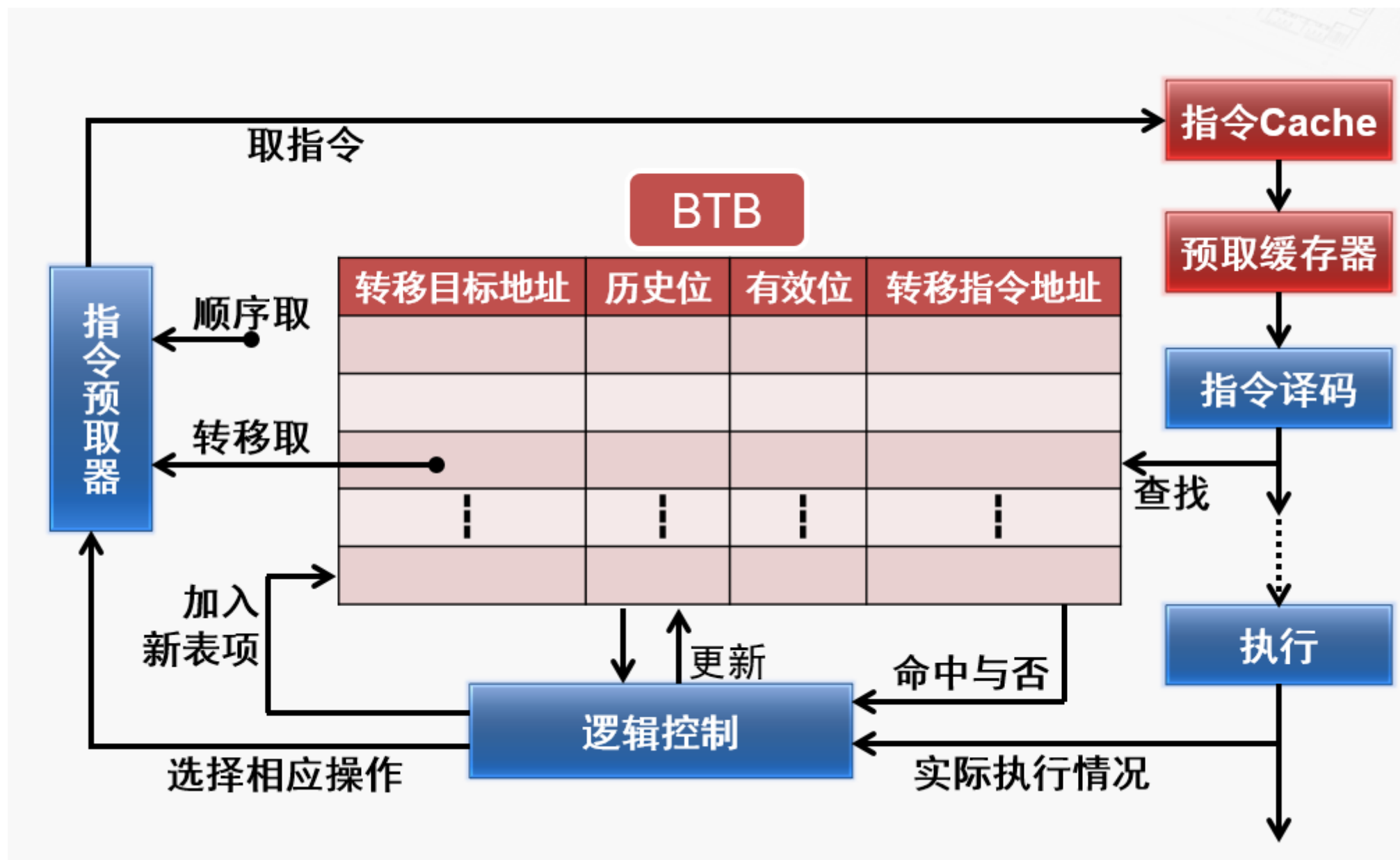
优化方案

1. 除数寄存器缩小为32-bit
2. 取消商寄存器
3. 64-bit ALU缩小为32-bit
4. 余数寄存器只有高32-bit
5. 余数寄存器需支持左移
6. 商从右端逐位移入余数
7. 运算结束时，商占据余数

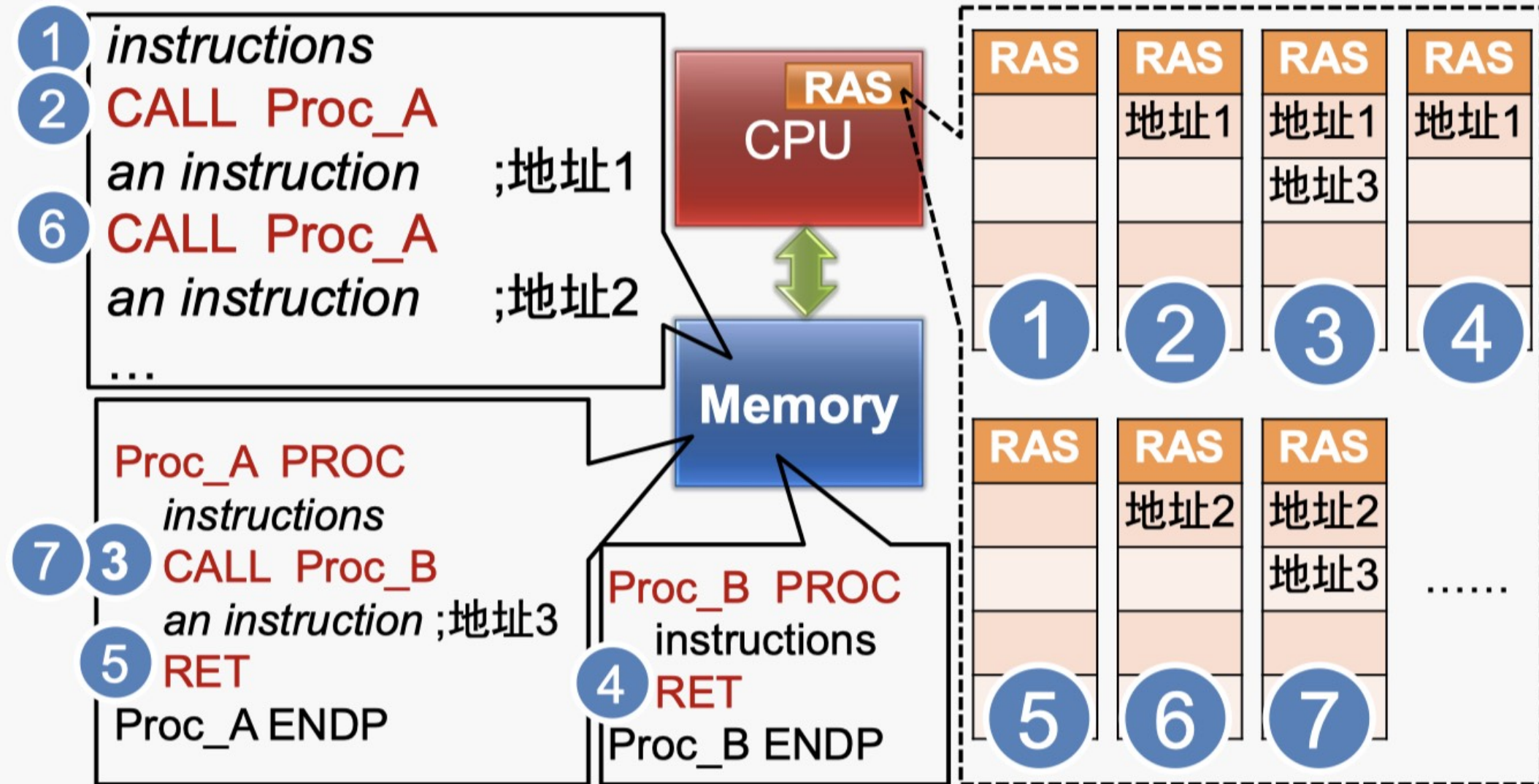




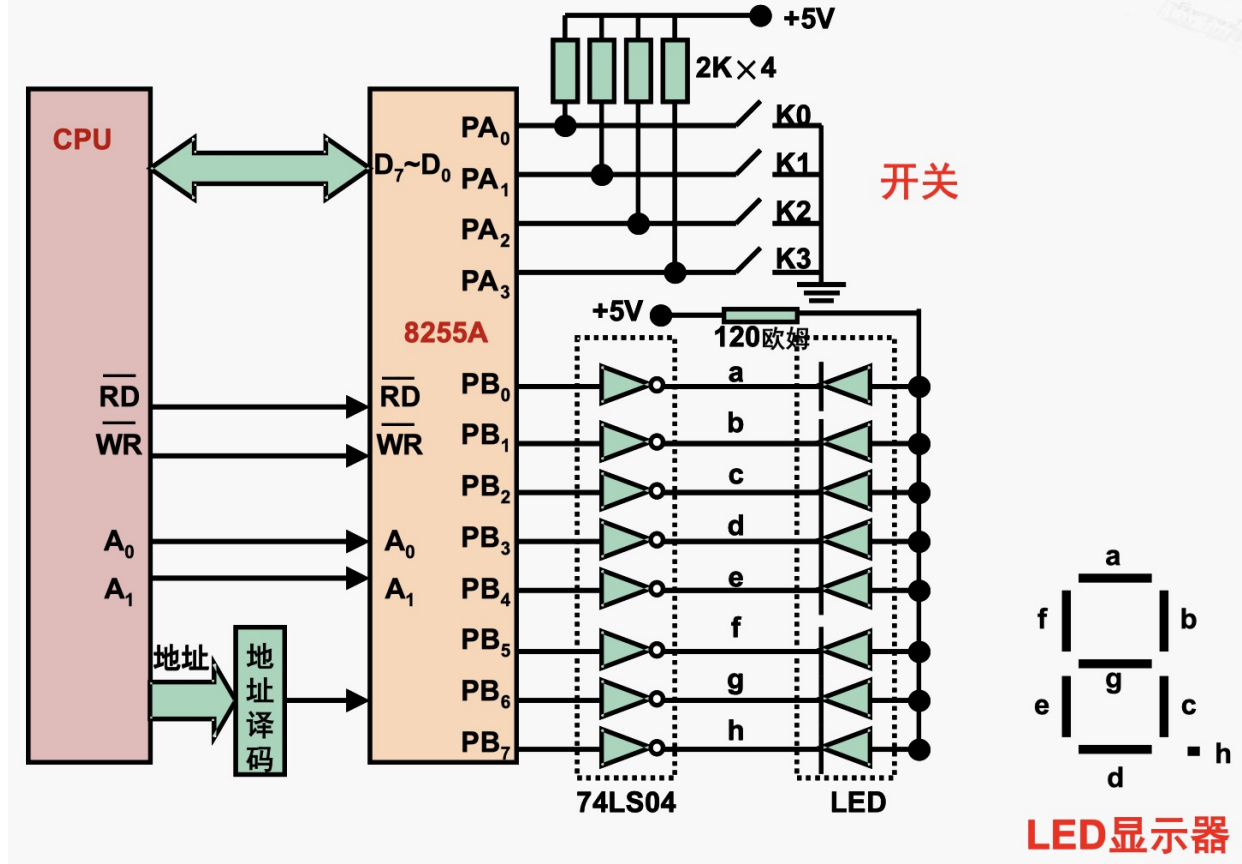
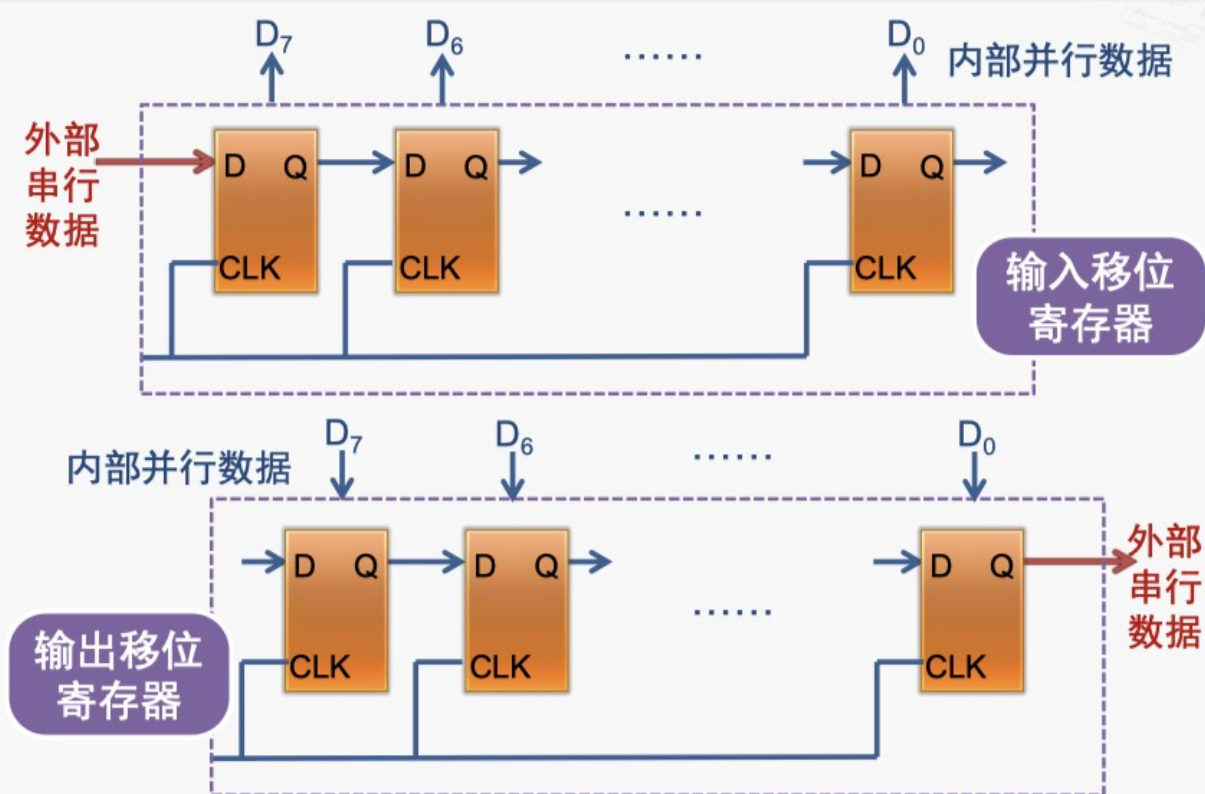




RAS: Return Address Stack



串/并转换

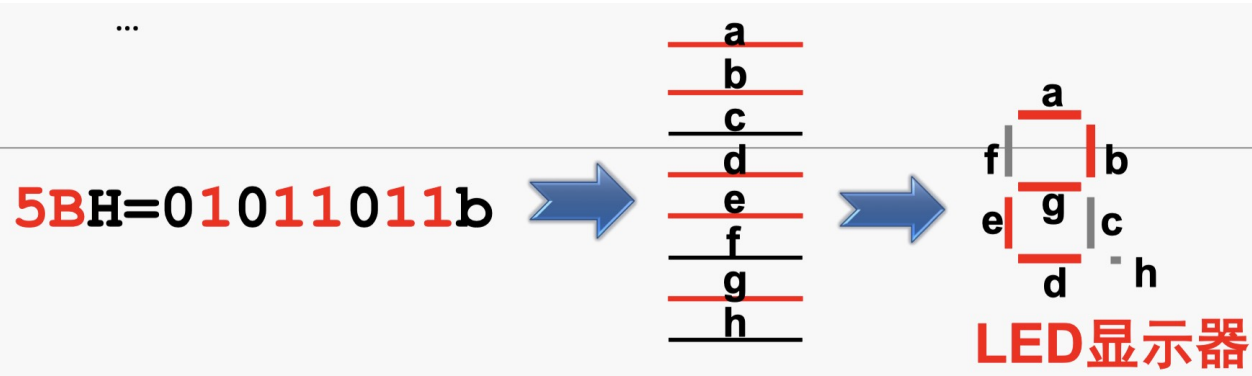


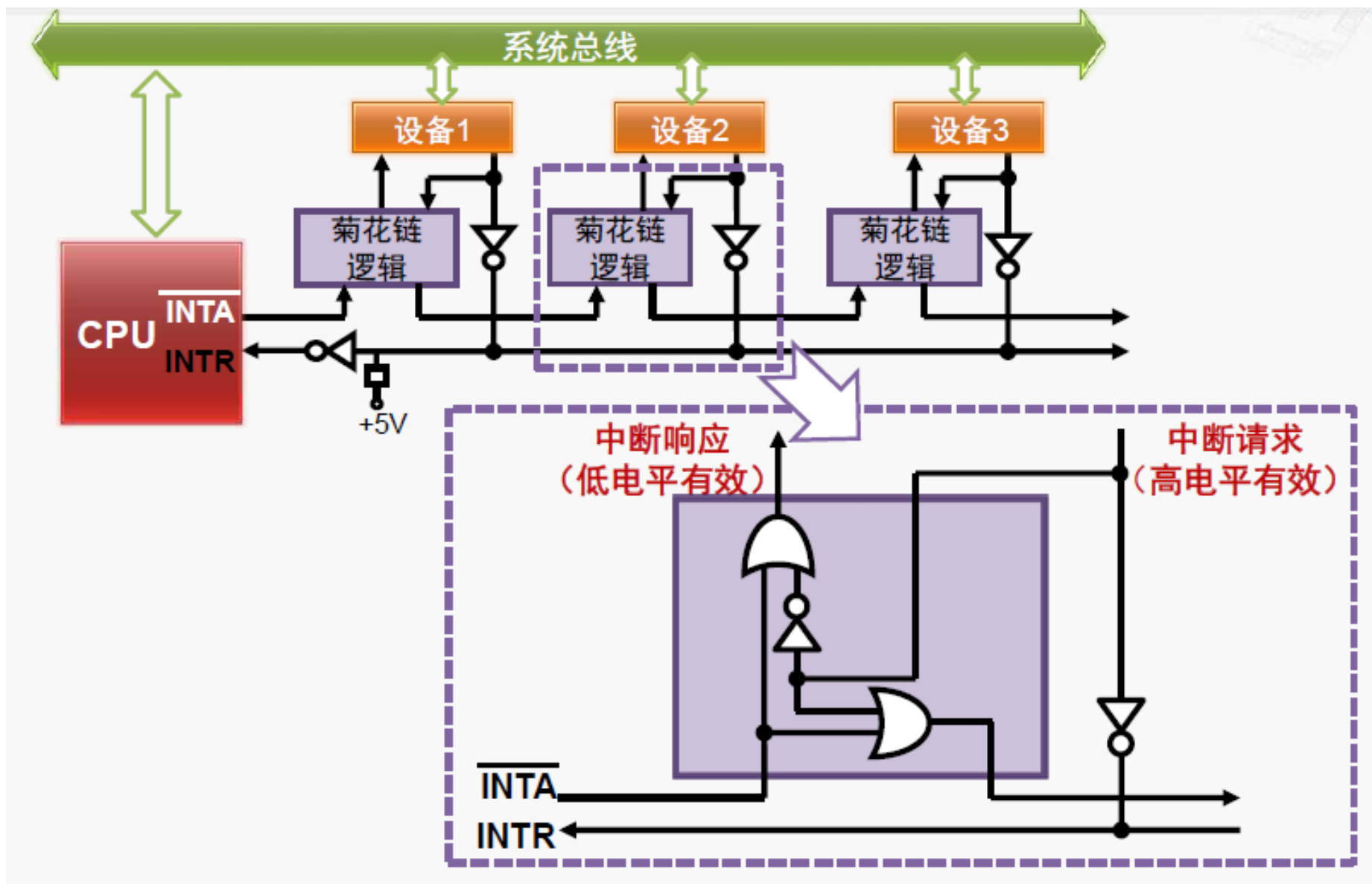
```
DATA SEGMENT
SSEGCODE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
          DB 7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
DATA ENDS
CODE SEGMENT
        ASSUME CS: CODE, DS: DATA
START:  MOV AX, DATA
        MOV DS, AX
        MOV AL, 90H           ; 设置方式选择控制字, A口工作于方式0输入, B口工作于方式0输出
        OUT 0D3H, AL

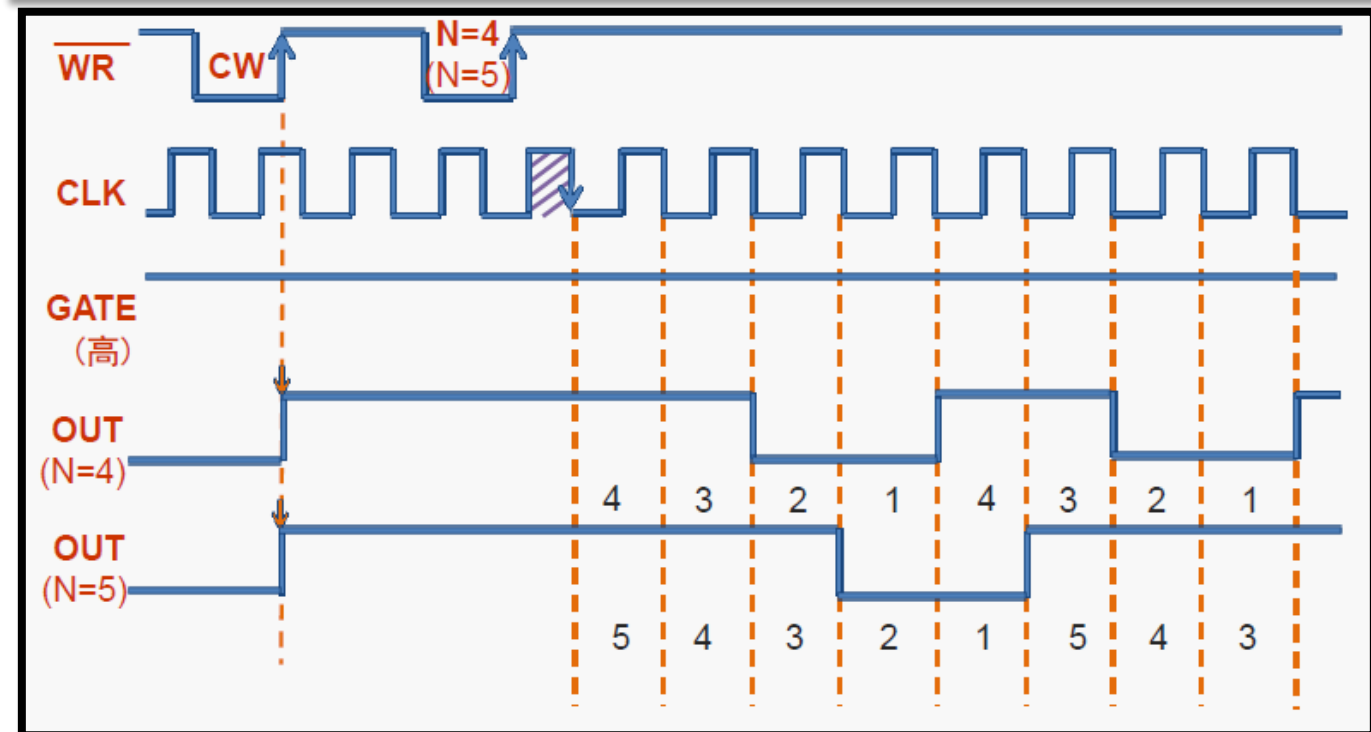
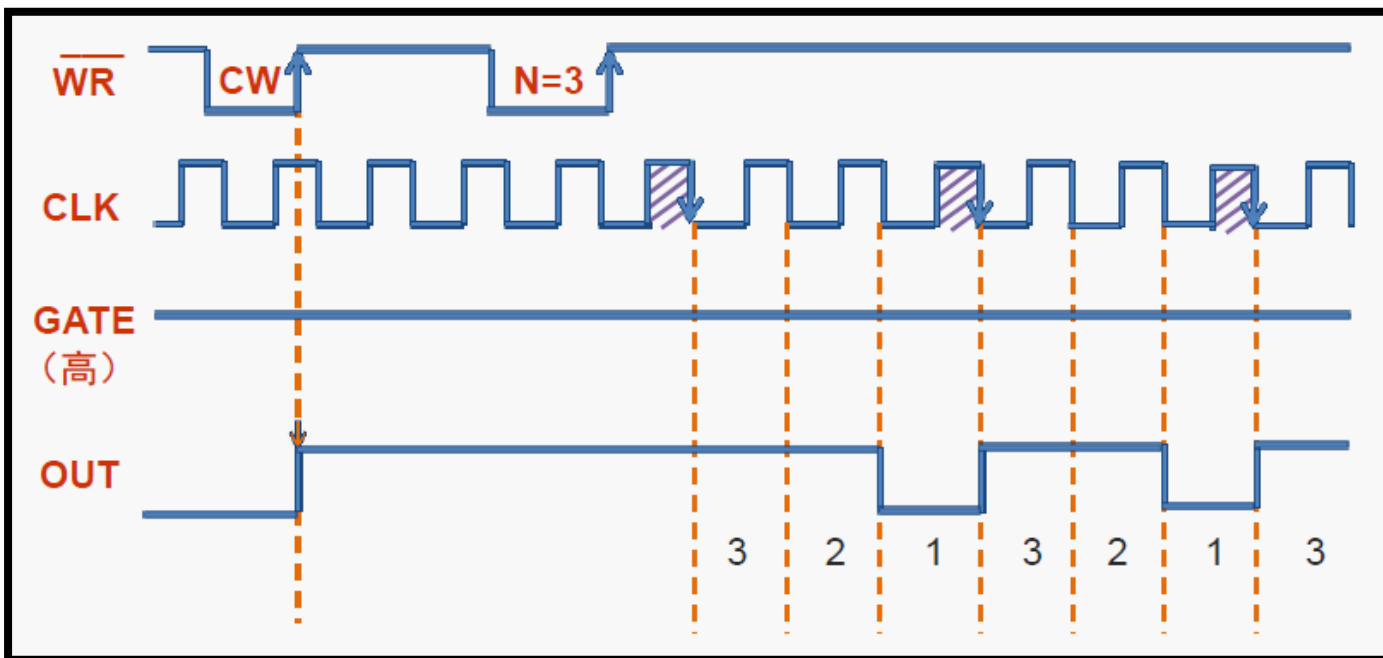
RDPORTA:
        IN  AL, 0D0H          ; 读A口
        AND AL, 0FH           ; 取A口低4位
        MOV BX, OFFSET SSEGCODE ; 取LED段选码表首地址
        XLAT                  ; 查表, AL←(BX+AL)
        OUT 0D1H, AL          ; 从B口输出LED段选码, 显示相应字形符号
        MOV AX, XXXXH         ; 延时
DELAY:  DEC AX
        JNZ DELAY
        MOV AH, 1             ; 判断是否有键按下
        INT 16H
        JZ  RDPORTA           ; 若无, 则继续读端口A
        MOV AH, 4CH           ; 否则返回DOS
        INT 21H
CODE ENDS
        END START
```

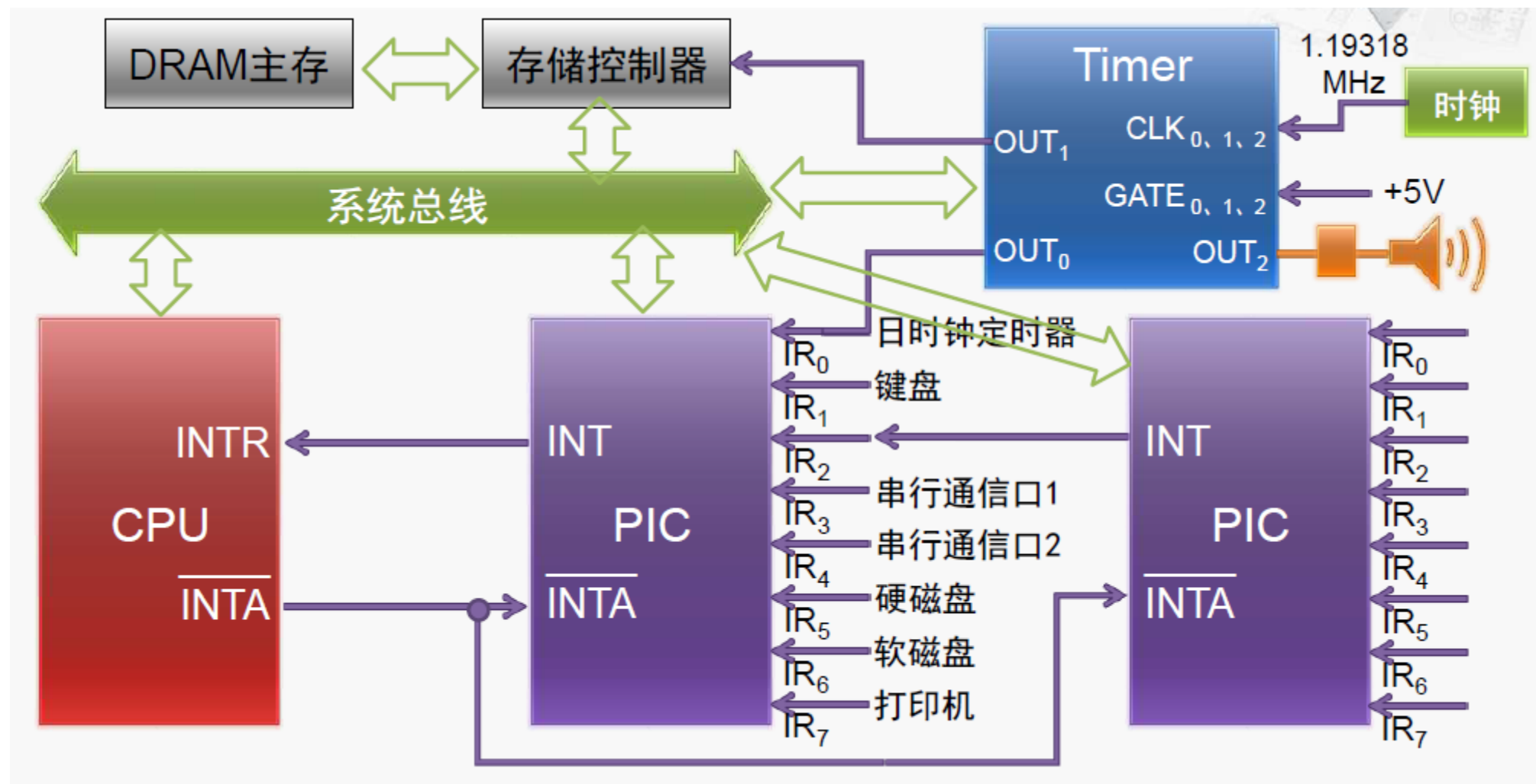
设8255A的端口地址如下:

- 端口A: D0H
- 端口B: D1H
- 端口C: D2H
- 控制口: D3H

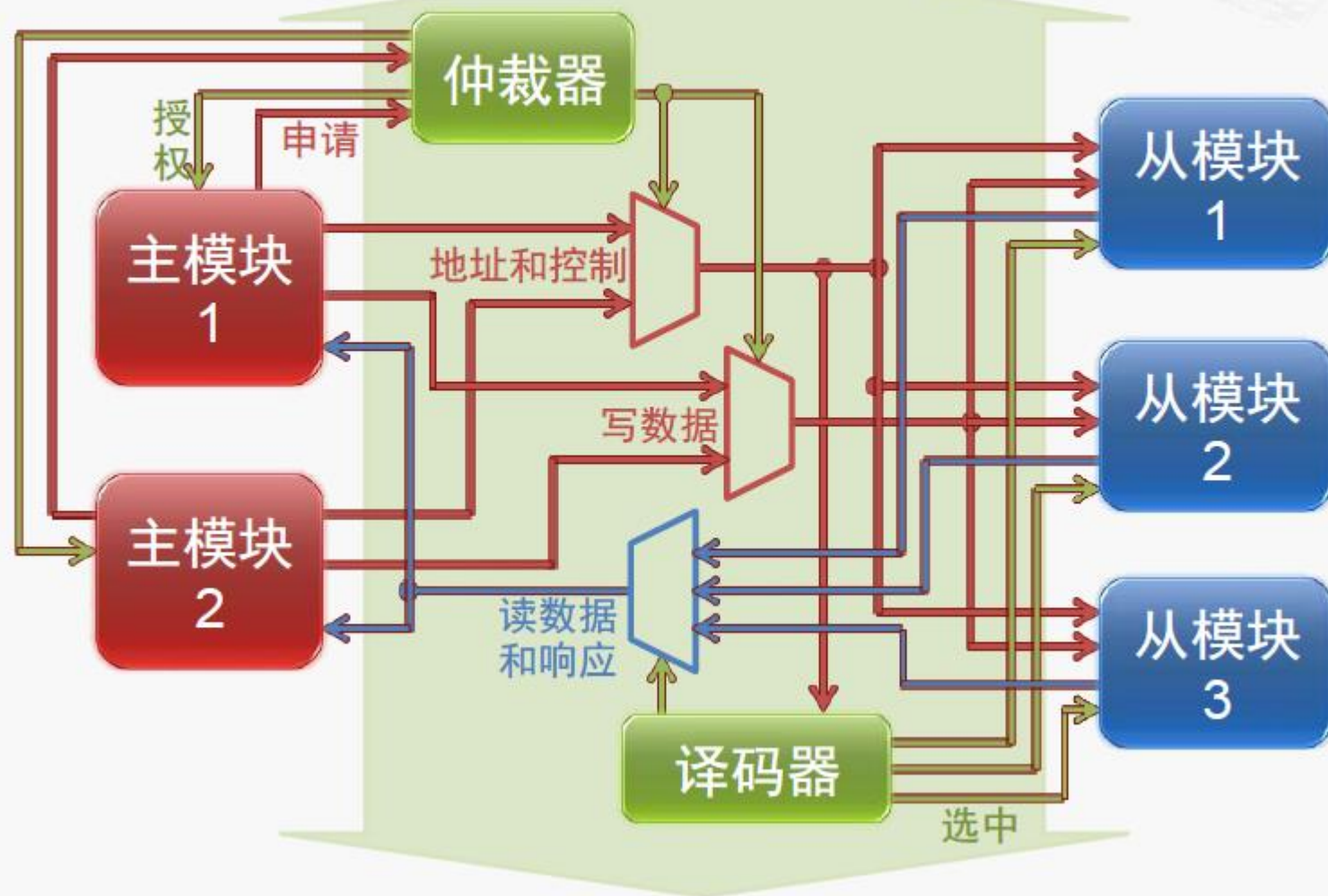




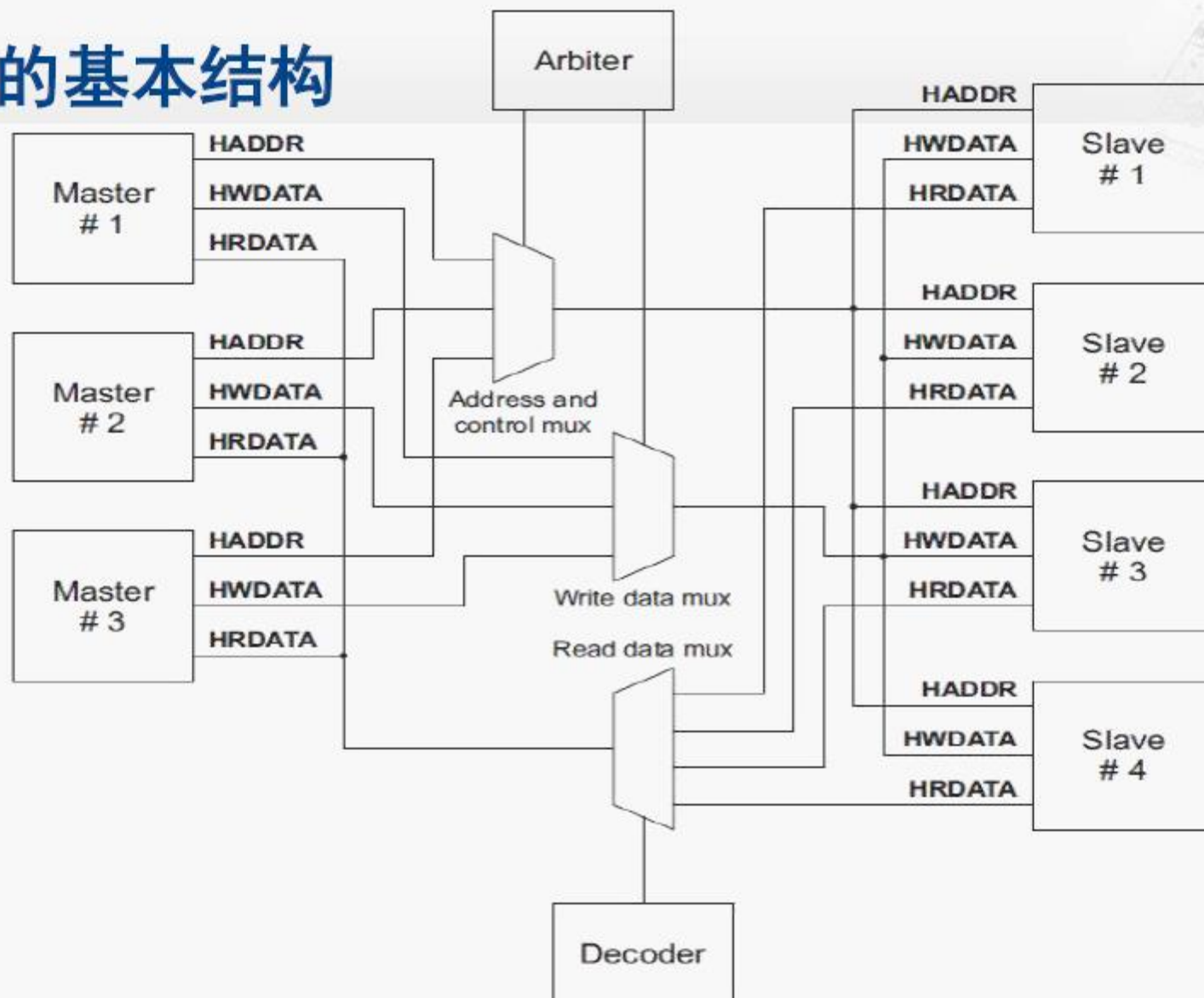




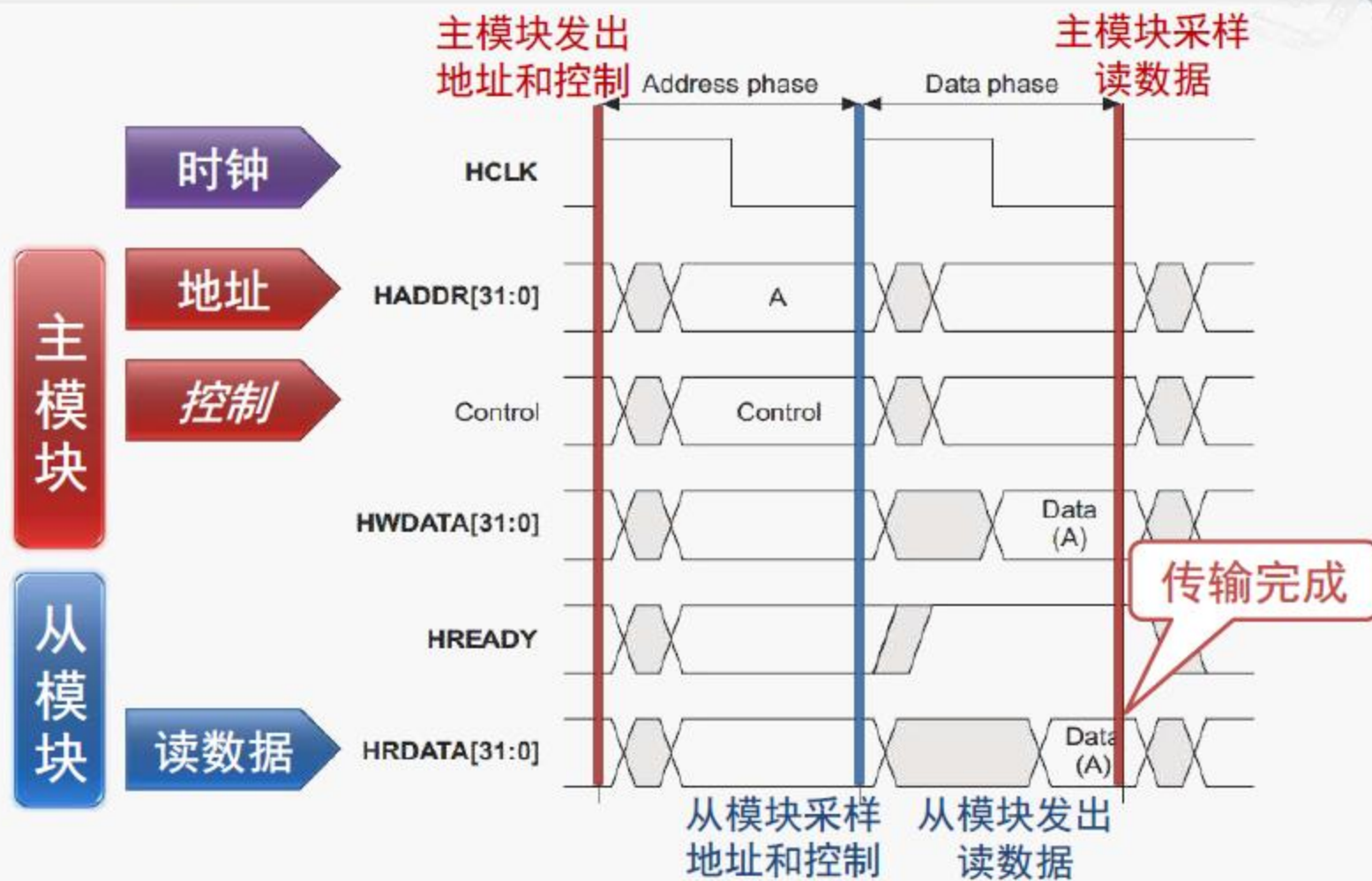
总线逻辑电路示意图



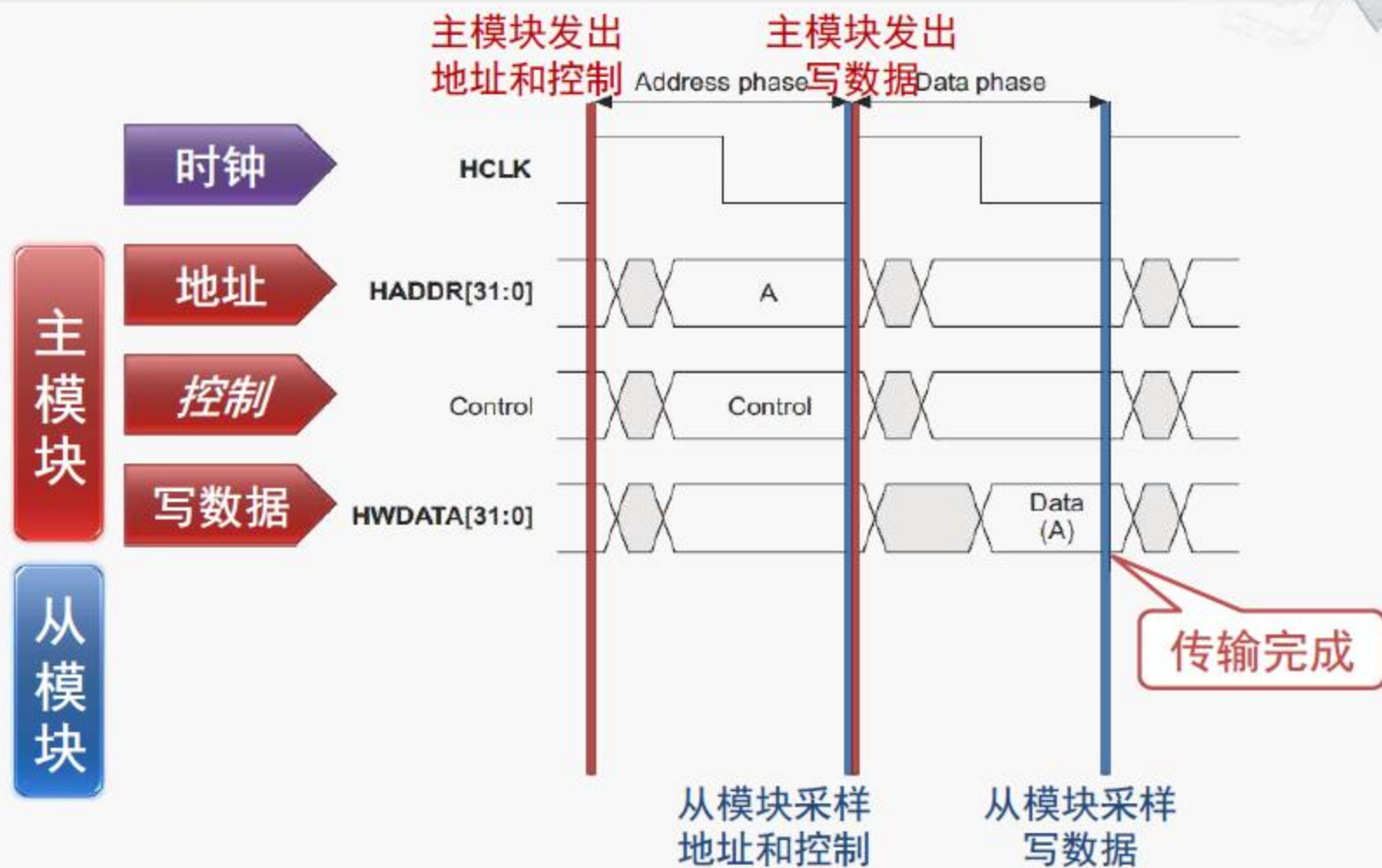
AHB的基本结构



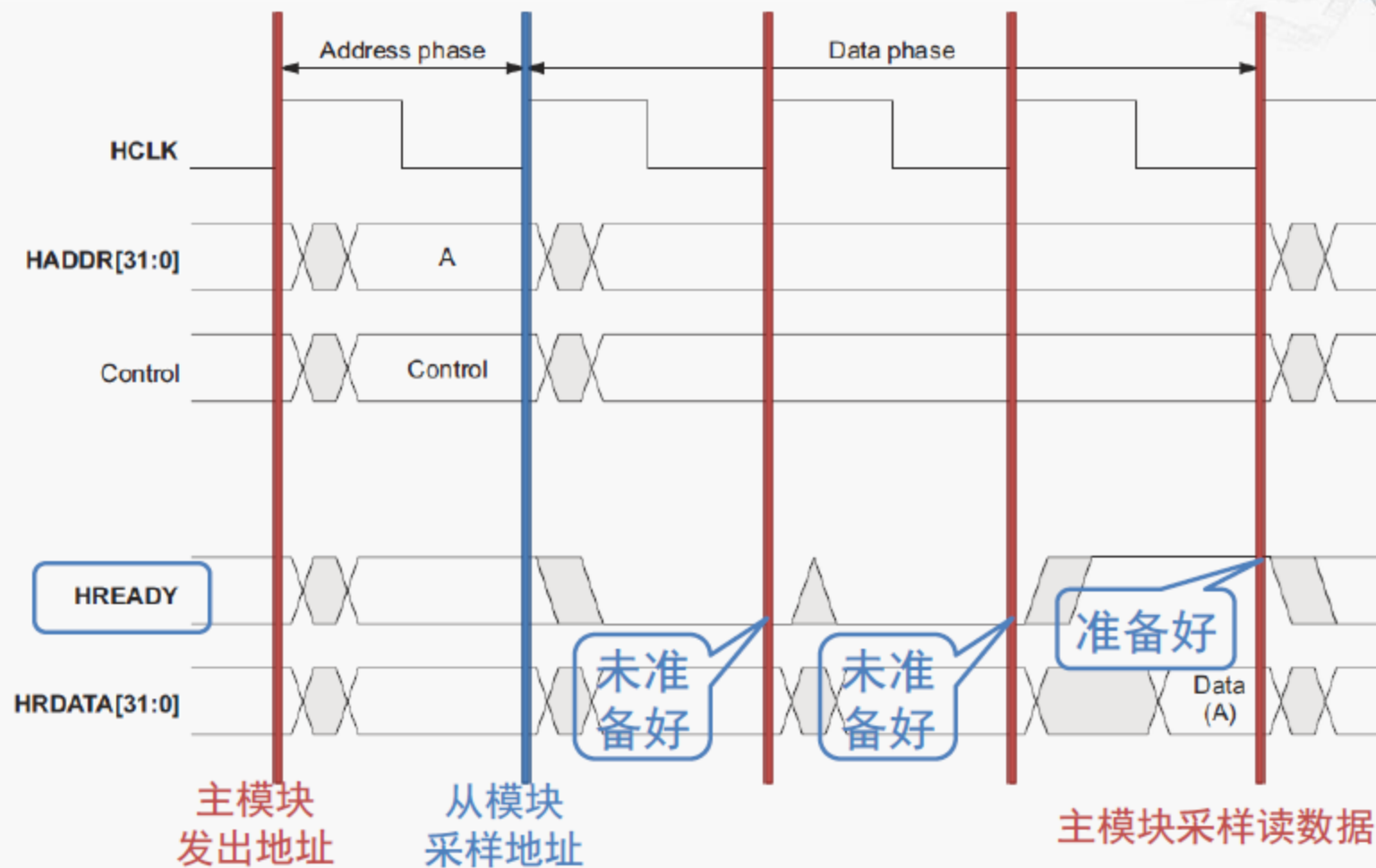
时序图1-a: 读1个数据的传输



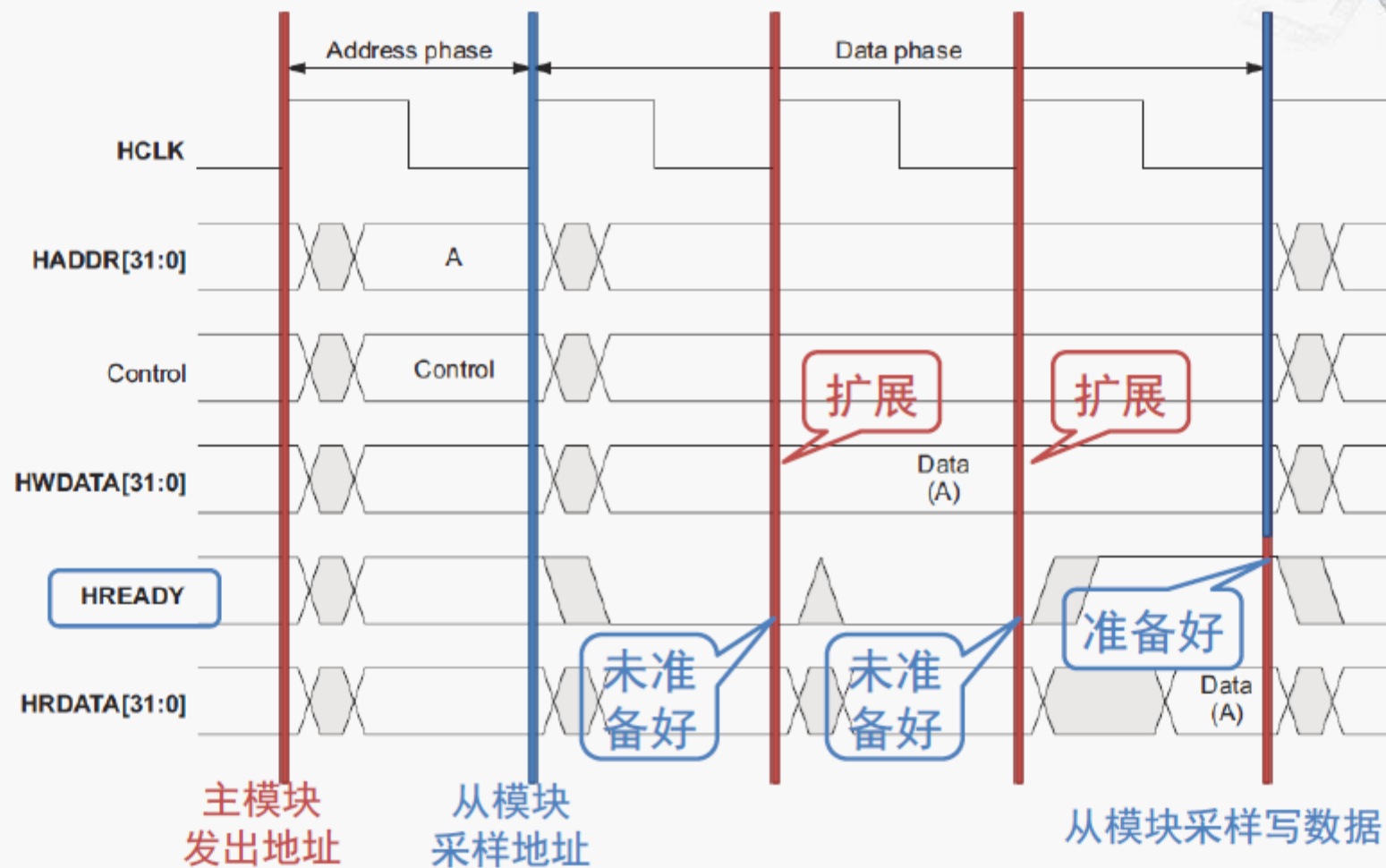
时序图1-b: 写1个数据的传输



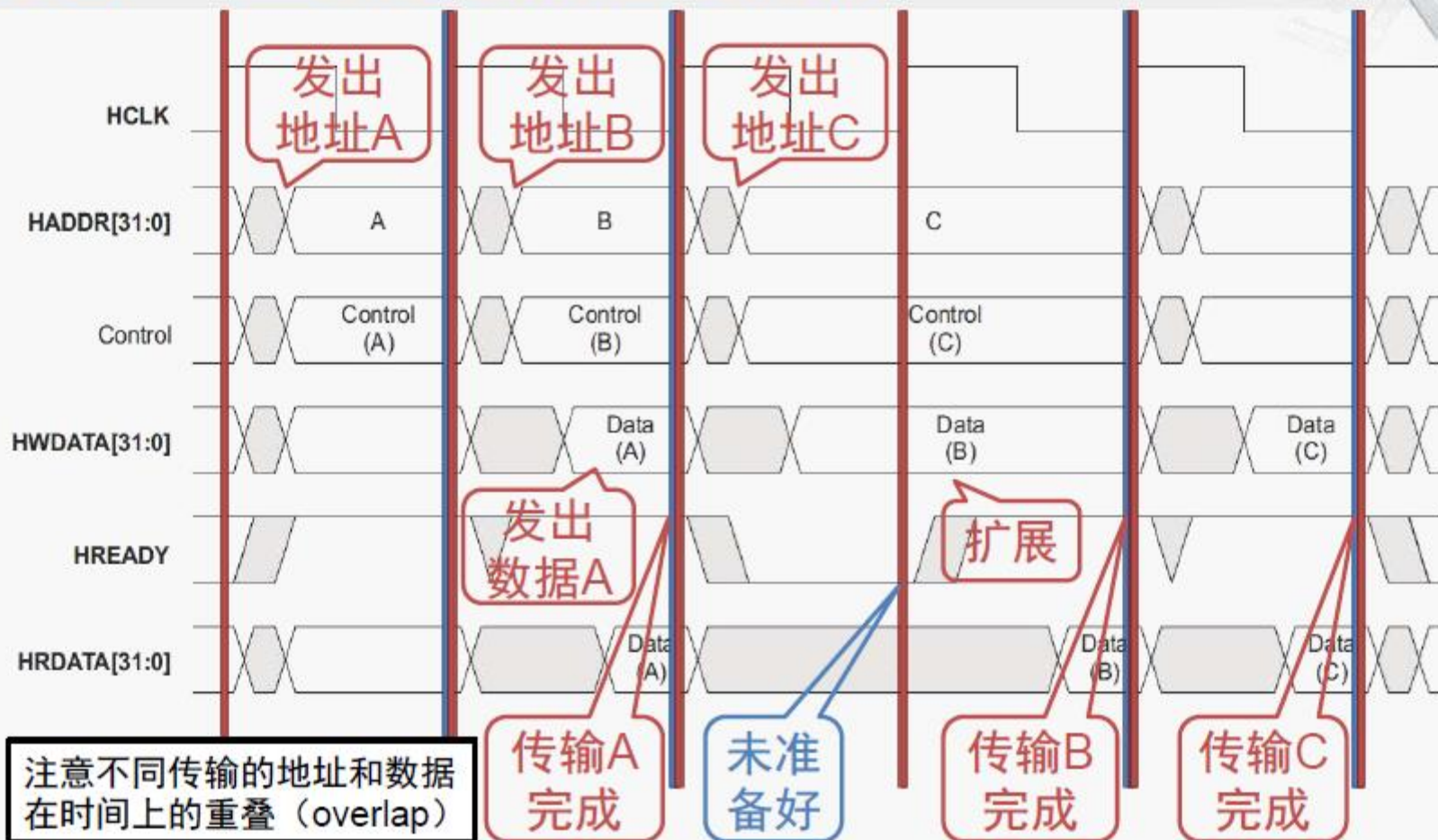
时序图2-a: 从模块插入等待周期的读传输



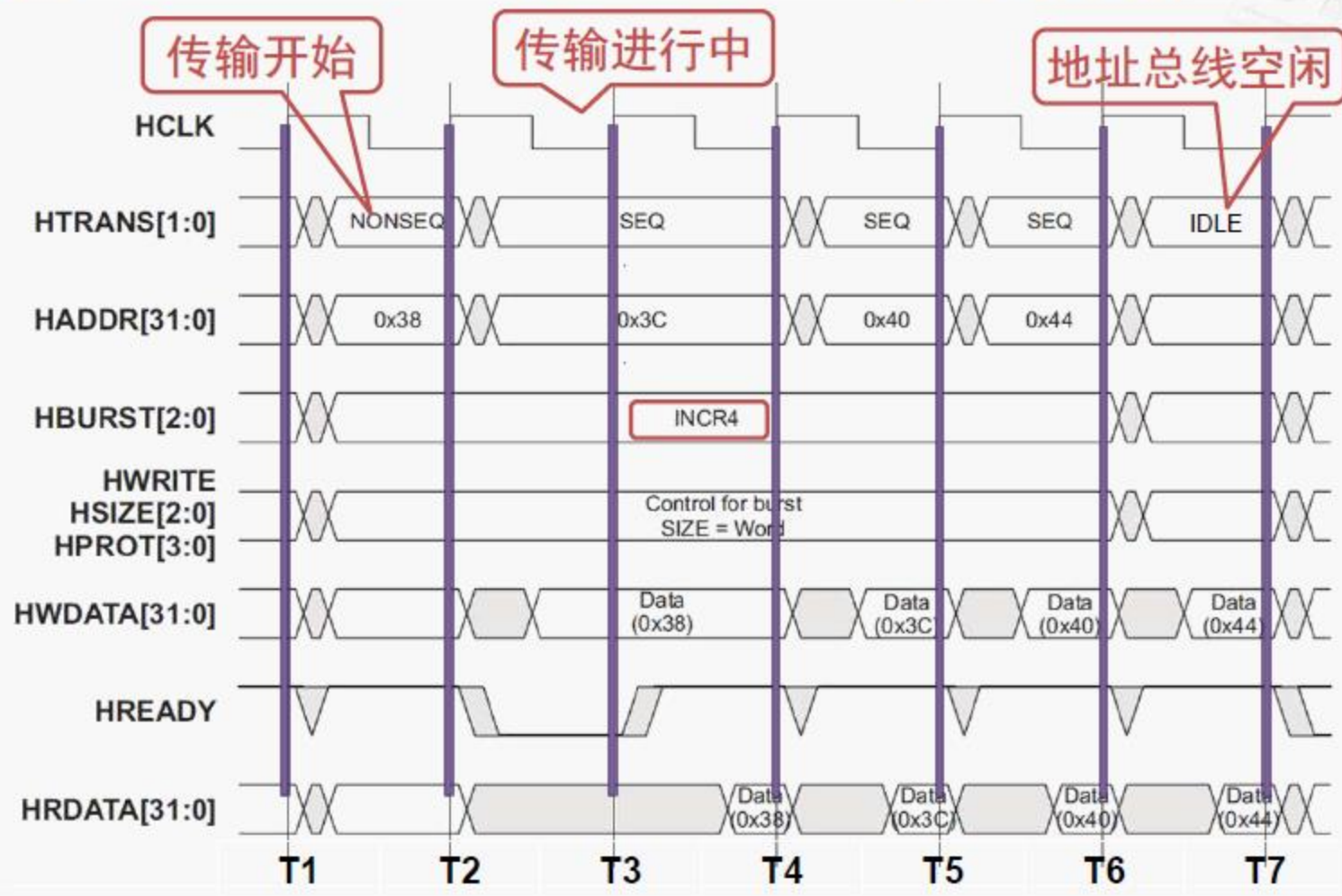
时序图2-a: 从模块插入等待周期的写传输



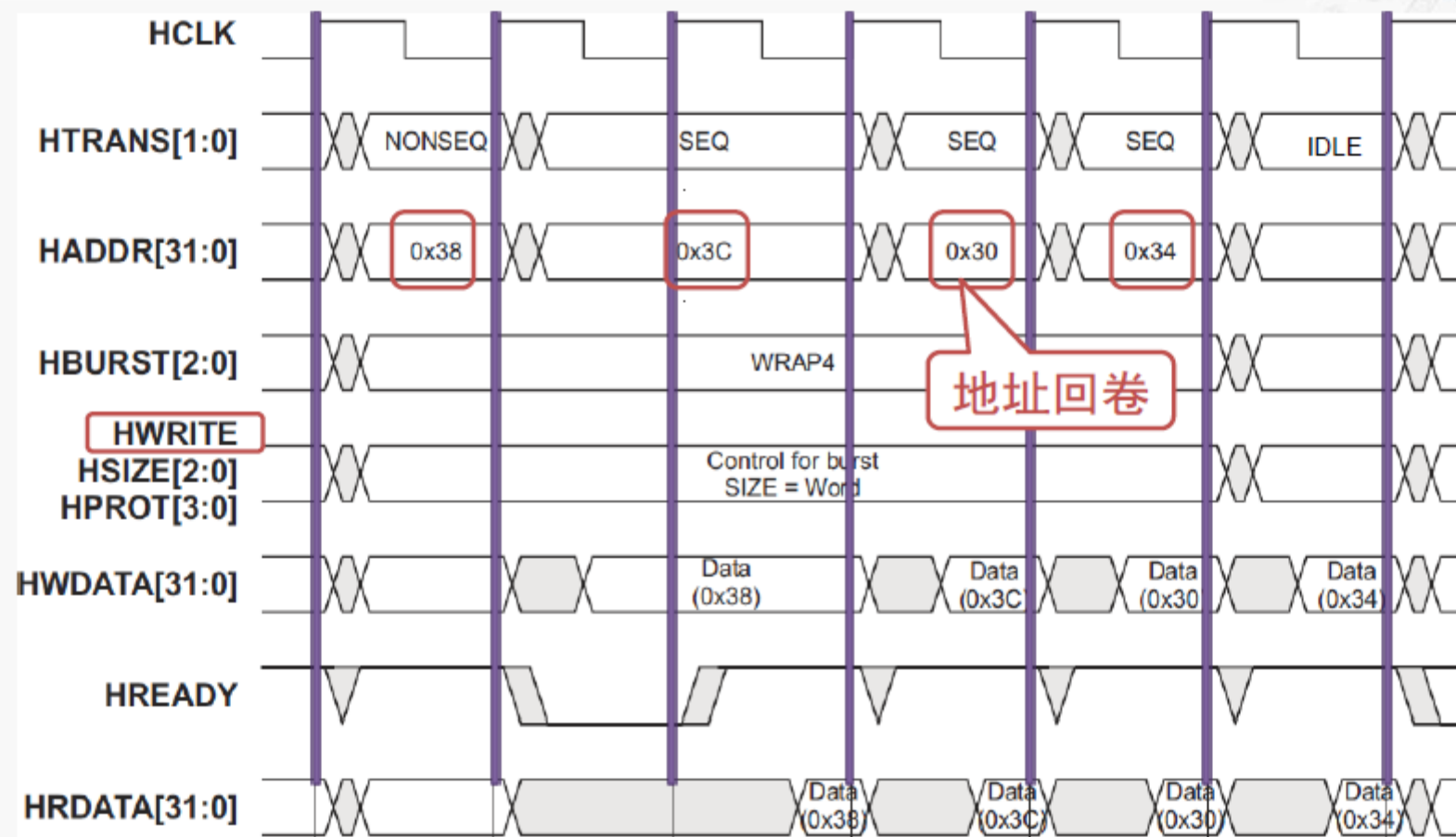
时序图3：三次连续的总线传输



时序图4： HTRANS和HBURST信号示例

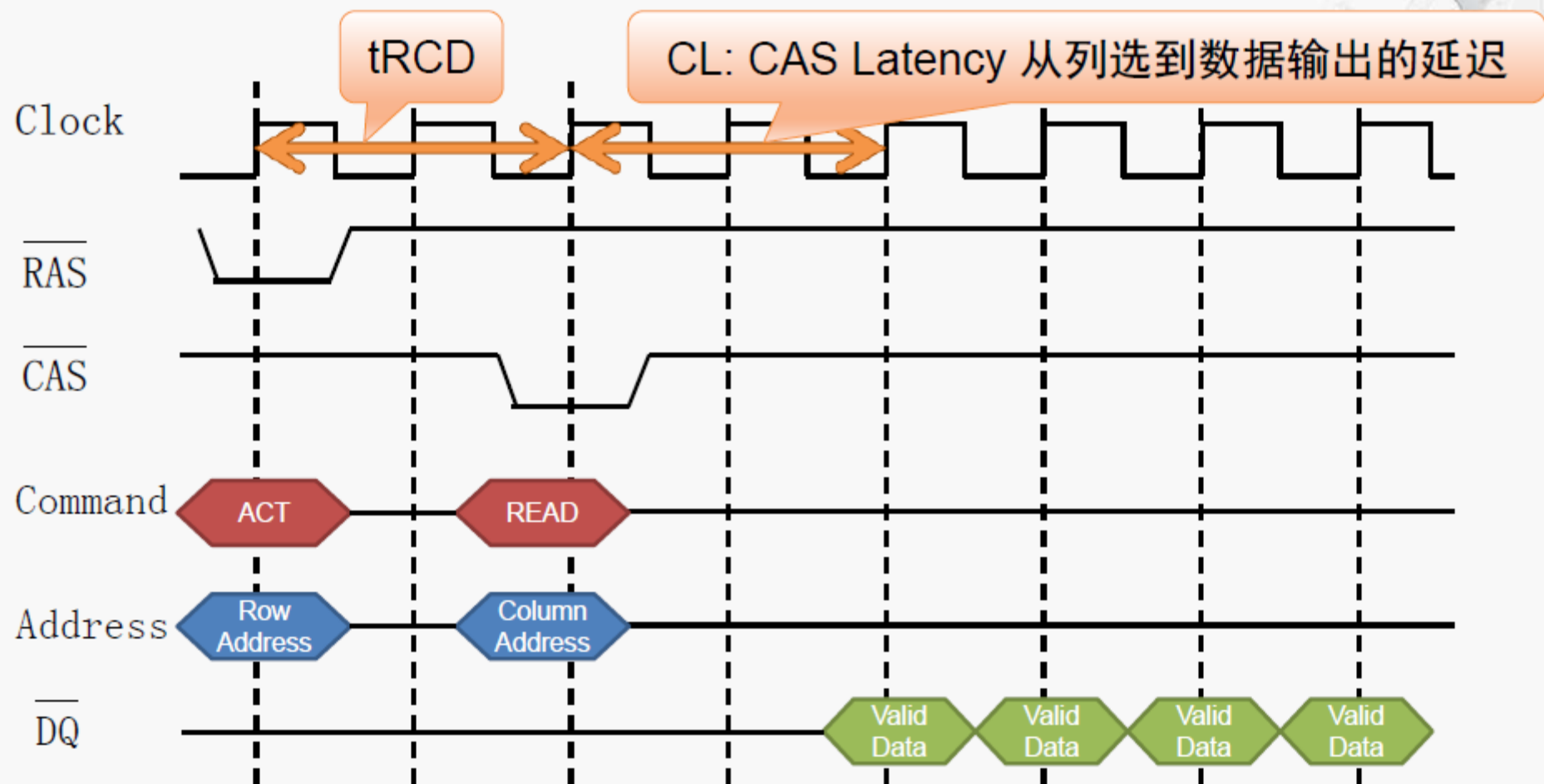


时序图5：地址回卷的四个数据的传输



*HWRITE信号：1表示写传输，0表示读传输

典型SDRAM的读操作时序图



典型SDRAM的读操作时序图

