



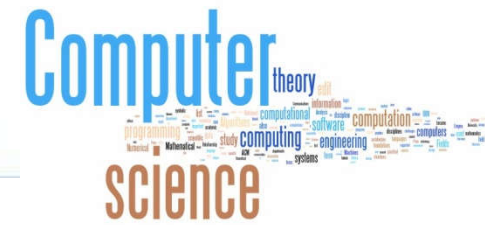
计算机组成原理

总结——图说COD

llxx@ustc.edu.cn



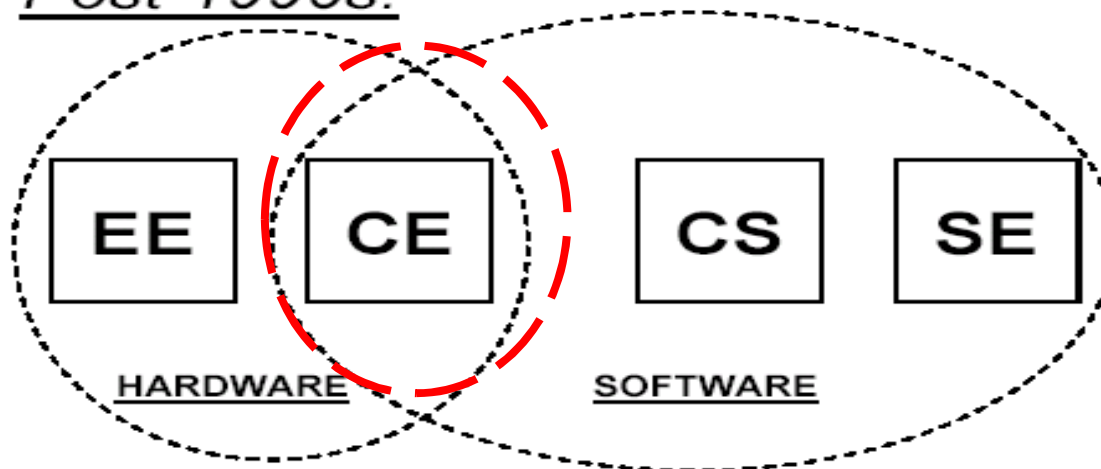
计算机科学 (Computing)



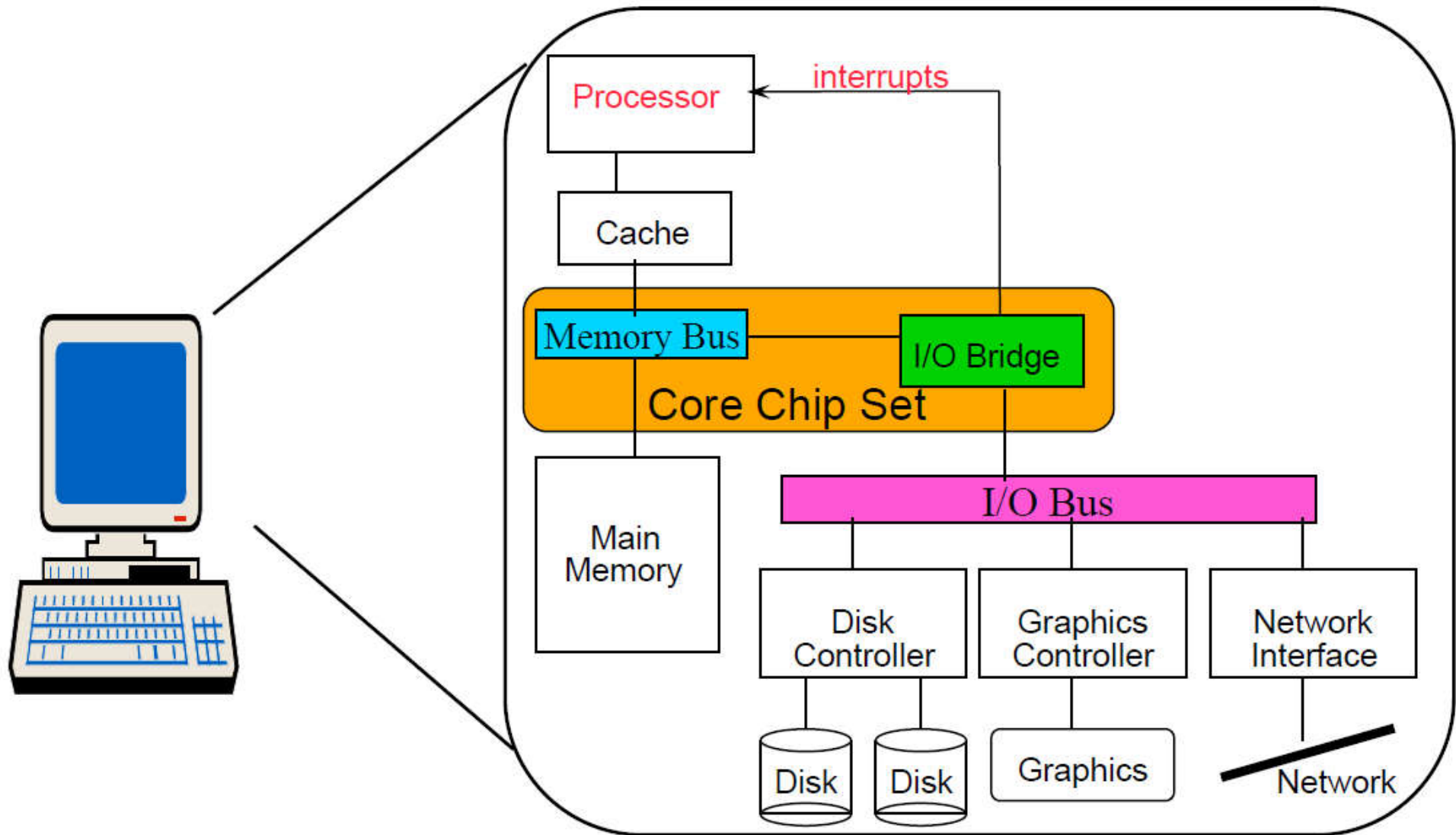
Pre-1990s:



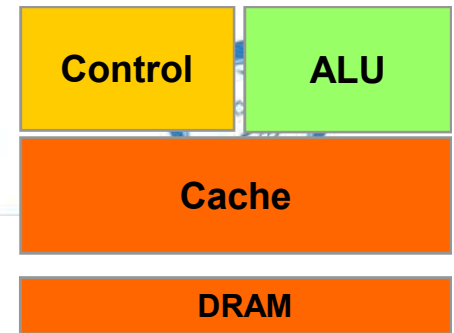
Post-1990s:



System Organization



冯.诺依曼机器

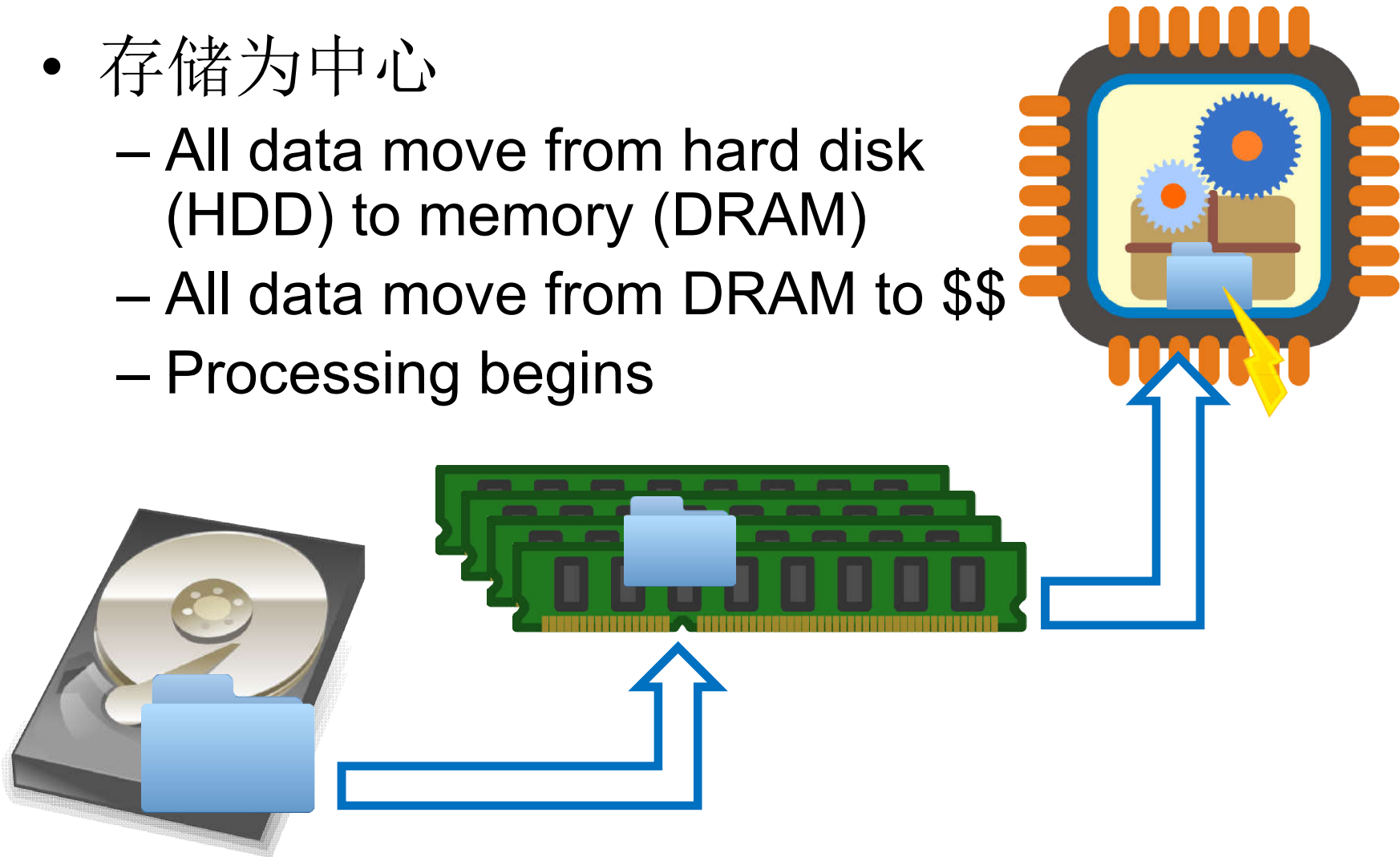


- 系统由五大功能部件构成
 - 以运算器为中心，I/O设备与存储器间的数据传送要经过运算器。
- 指令和数据以二进制表示。
- 指令由操作码和地址组成。
 - 操作数本身无数据类型的标志，其数据类型由操作码确定。
- 控制驱动：存储程序，顺序执行
 - 一维存储模型：存储器是按地址访问的线性编址的一维结构
 - 每个单元的位数是固定的。
 - 指令和数据不加区别存储在同一个存储器中
 - 指令和数据都可以送到运算器进行运算，即程序是可以修改的。
 - 一维计算模型：执行指令，发出控制信号，控制计算机的操作。
 - 指令在存储器中按其执行顺序存放，由指令计数器指明要执行的指令所在的单元地址。
 - 指令计数器只有一个，一般按顺序递增，但执行顺序可按运算结果或当时的外界条件而改变。
- 机器语言语义与高级语言差距很大

Data processing, a bird's eye view



- 存储为中心
 - All data move from hard disk (HDD) to memory (DRAM)
 - All data move from DRAM to \$\$
 - Processing begins



metrics

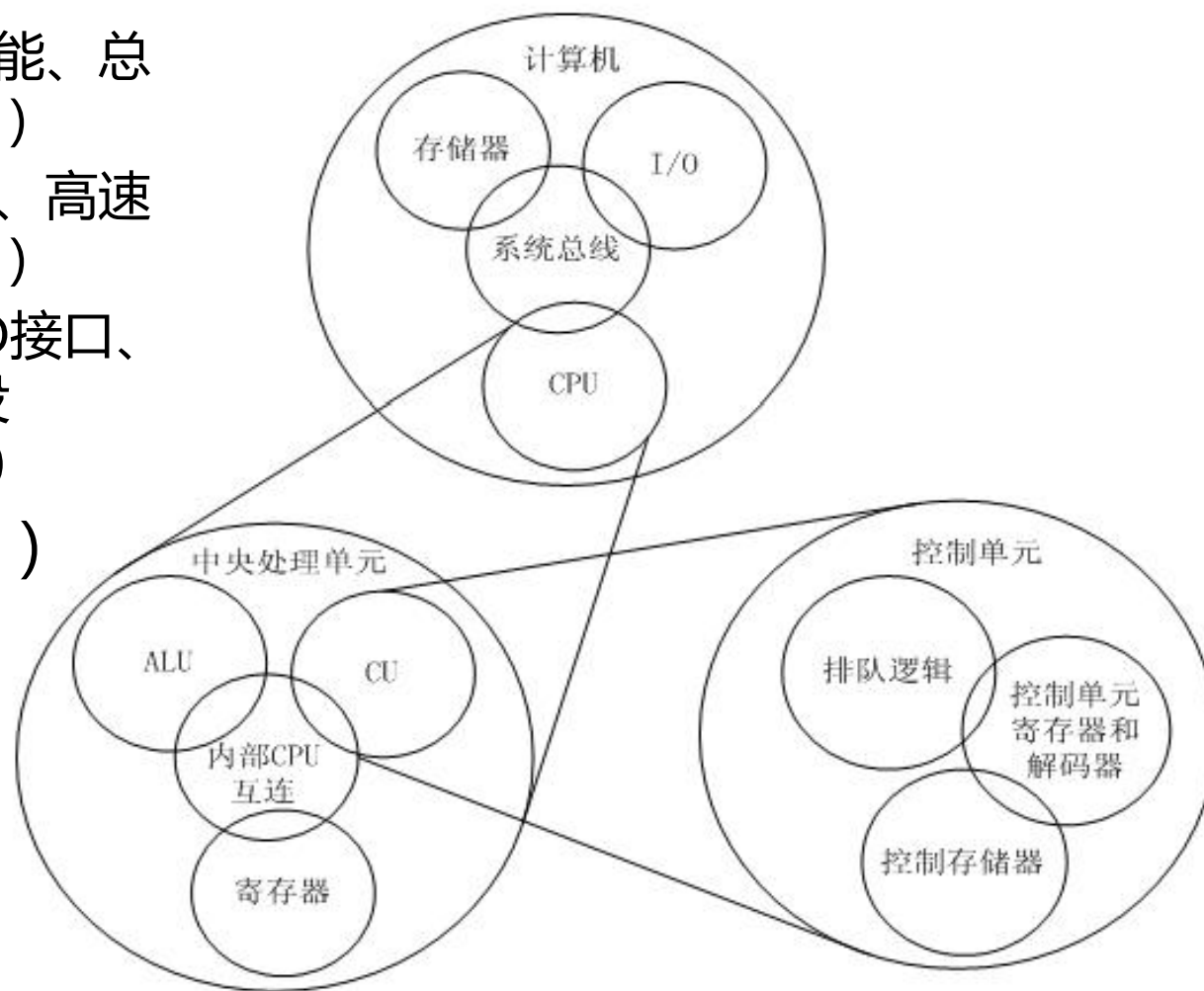


- performance
 - **Program-centric** (conventional)
 - **TIME = IC × CPI × CCT**
 - IC = “instruction count”, CPI = “clocks per instruction”, CCT = “clock cycle time”
 - **Data-centric**
 - **TIME = DC × CPB × CCT**
 - DC = “data count”, CPB = “clocks per byte”
 - CPB = IPB × CPI
- 功耗

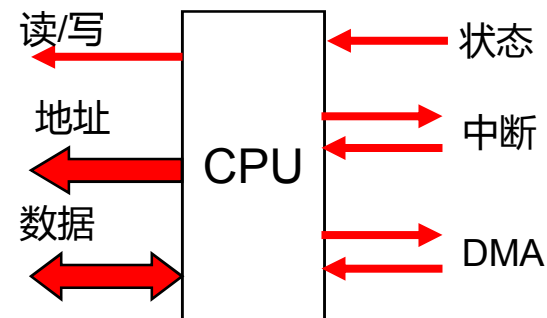
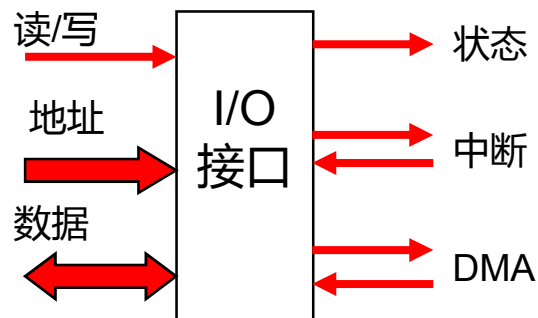
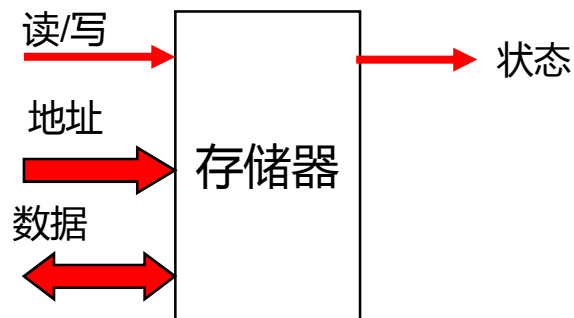
课程内容



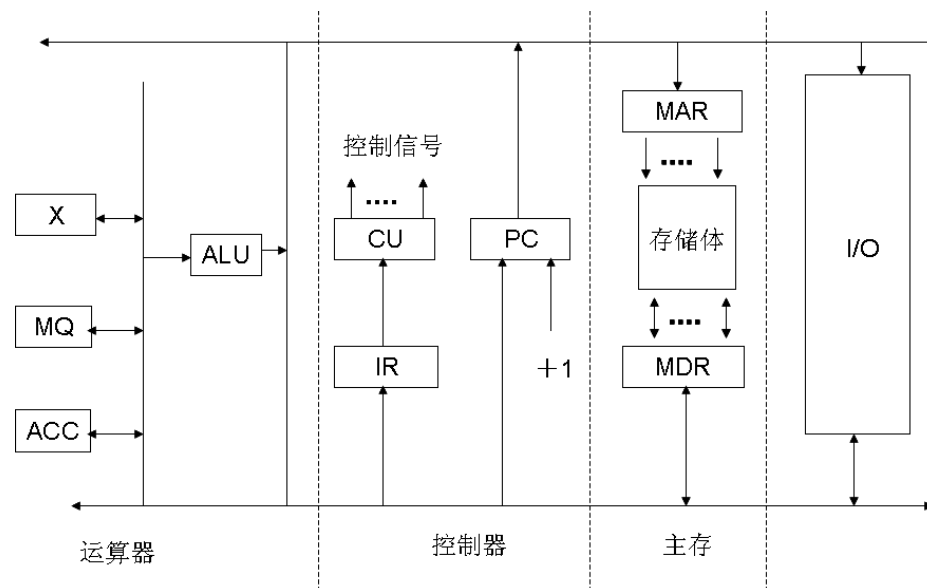
- 外围部件
 - **系统总线**（总线性能、总线结构、总线控制）
 - **存储器**（主存储器、高速缓存、辅助存储器）
 - **输入输出系统**（I/O接口、I/O控制方式、外设 peripheral device）
- 中央处理器（CPU）
 - 计算机的运算方法
 - 指令系统
 - CPU的结构
 - 控制单元设计



计算机部件



- **读写控制信号**
- **地址线**
- **数据线**
- **握手信号**
 - **状态、中断、DMA**
- **时钟、电源、地线**



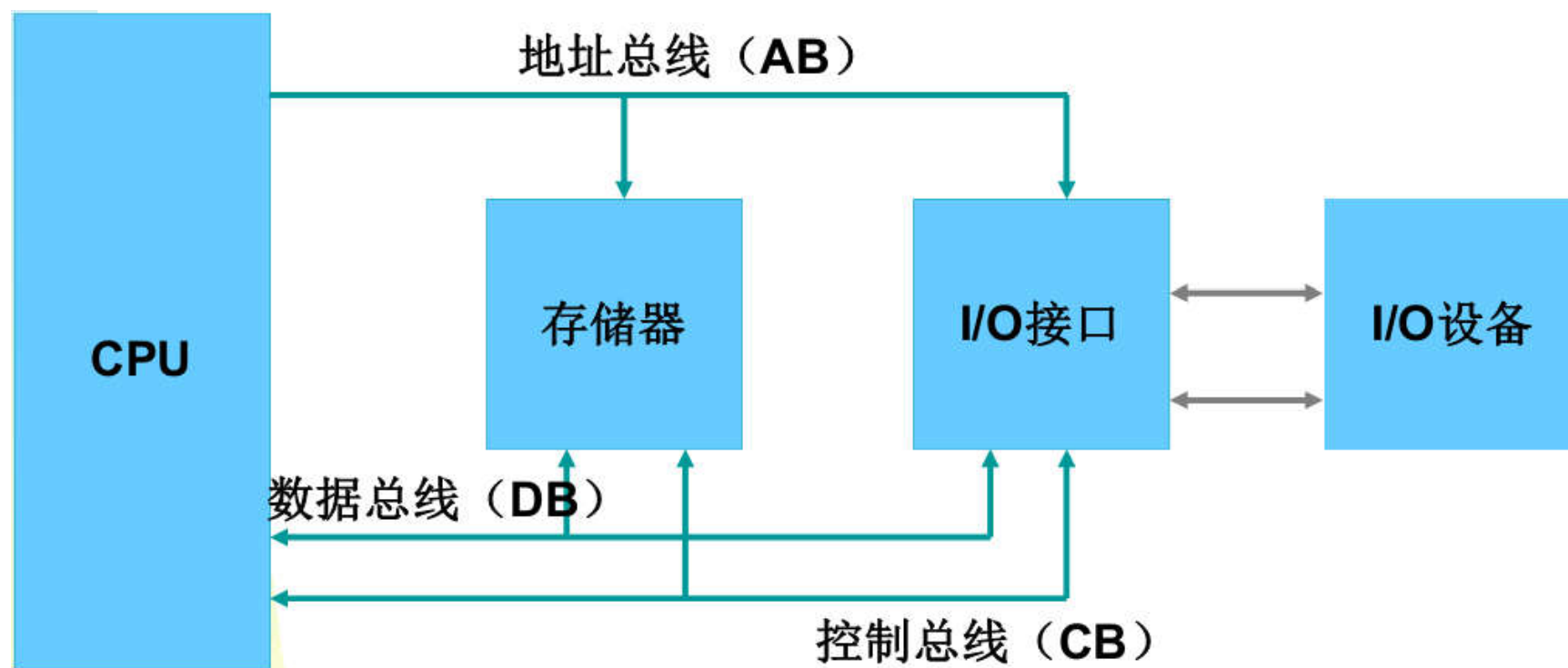


计算机组成原理

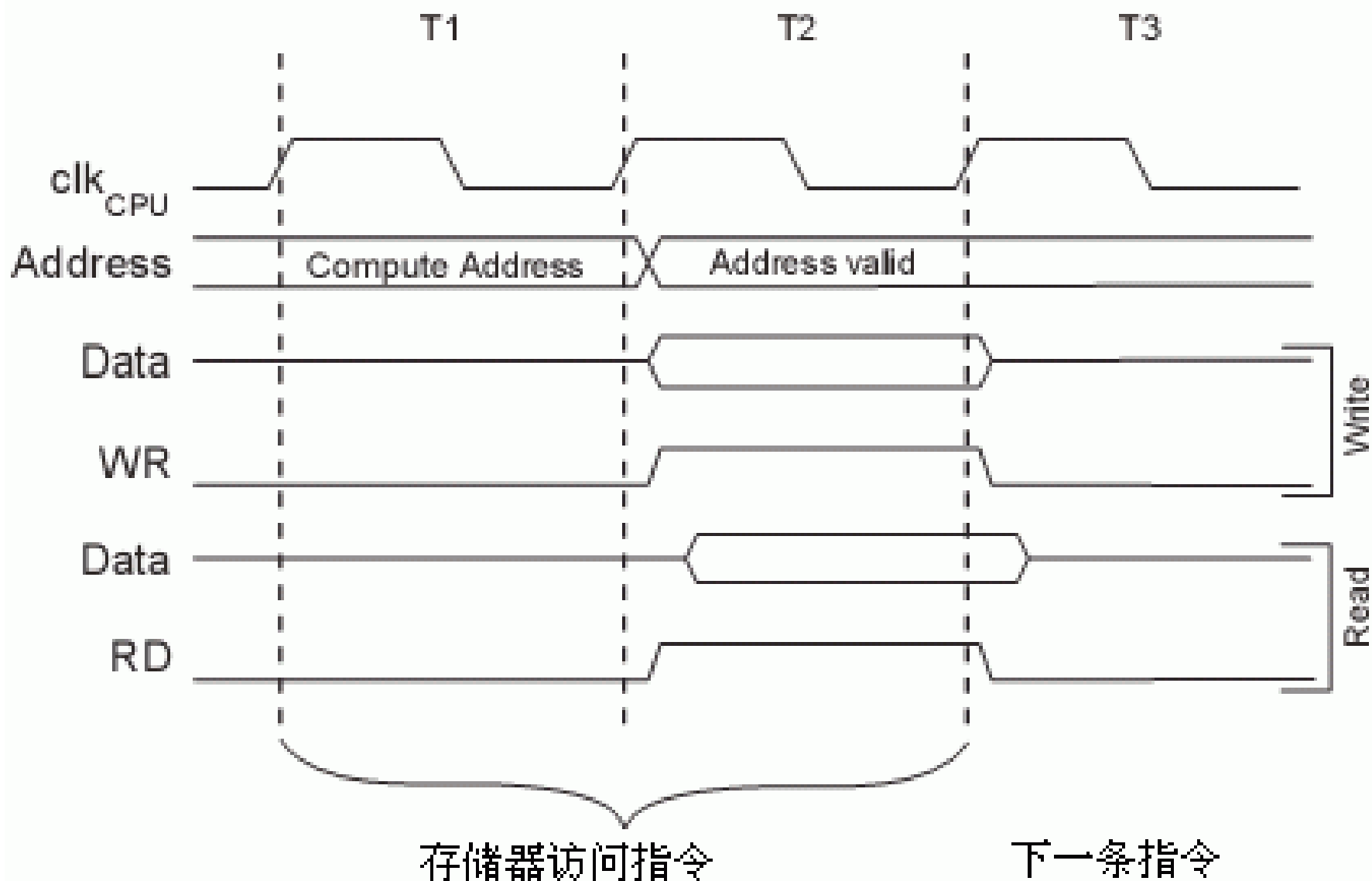
总线

共享，竞争，同步

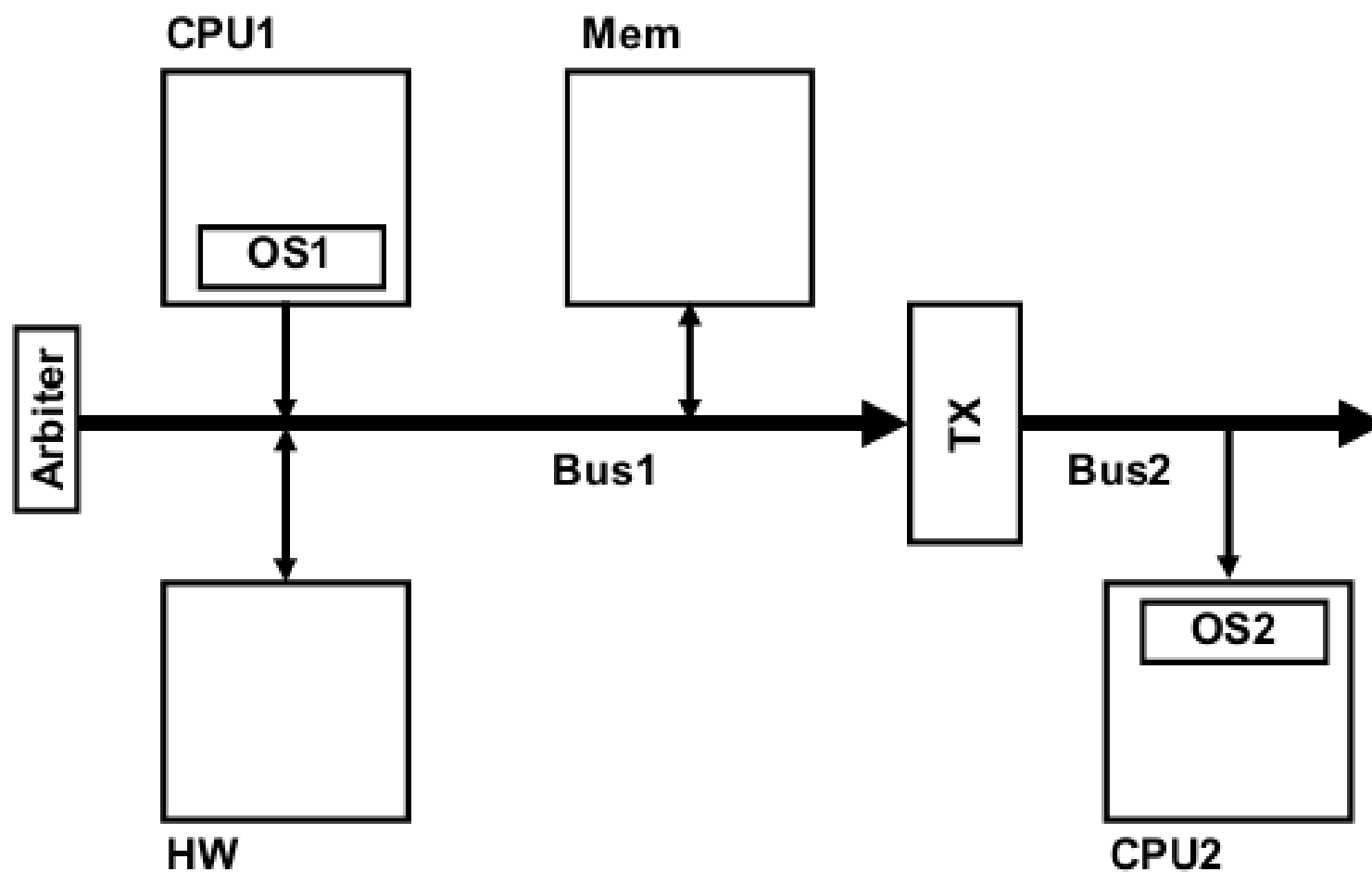
系统结构：基于总线



总线通信的同步方式



总线使用权分配





计算机组成原理

存储系统

内存，Cache，辅存
组织，地址空间，与CPU通信

PC机中的存储器

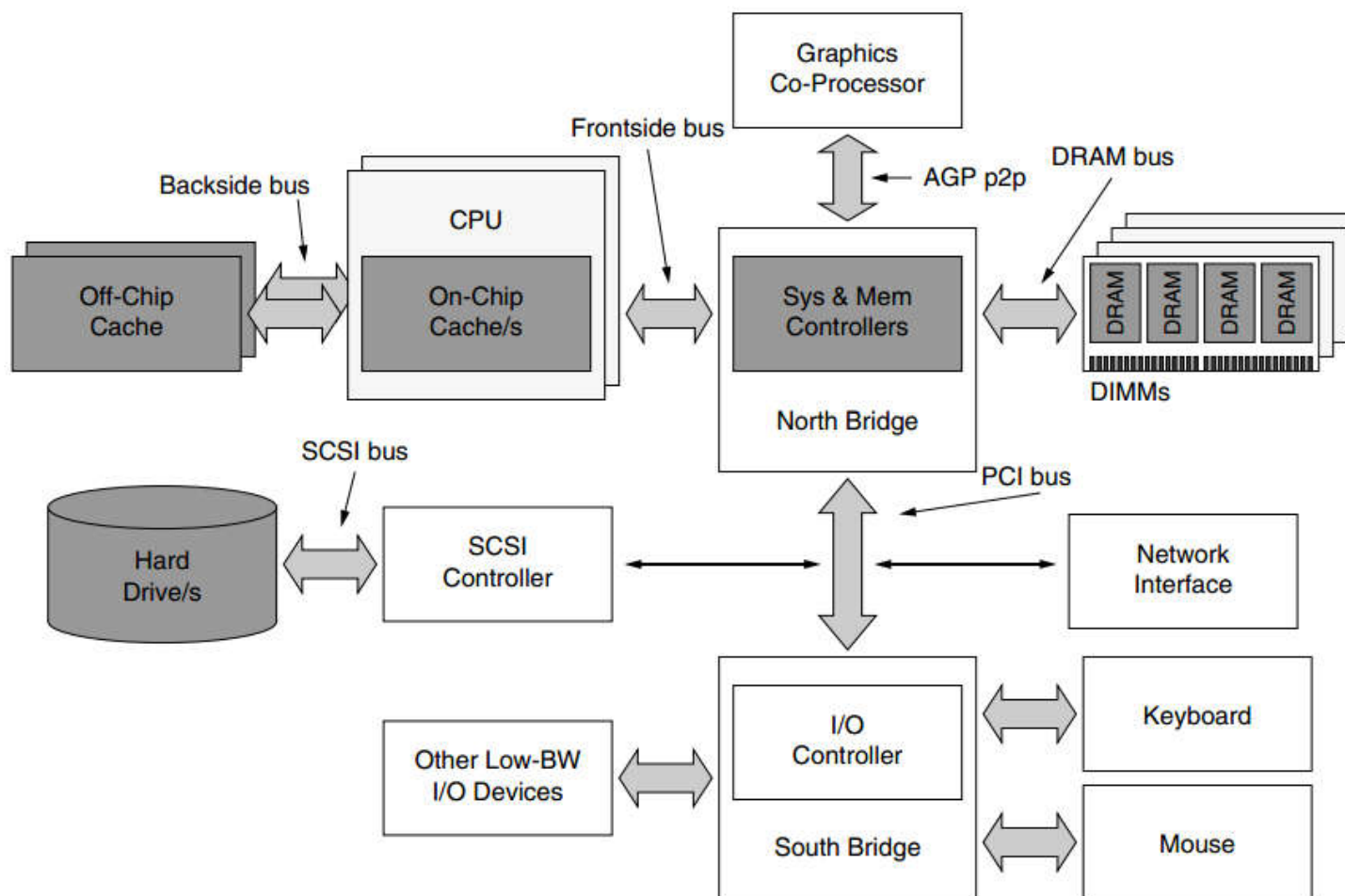
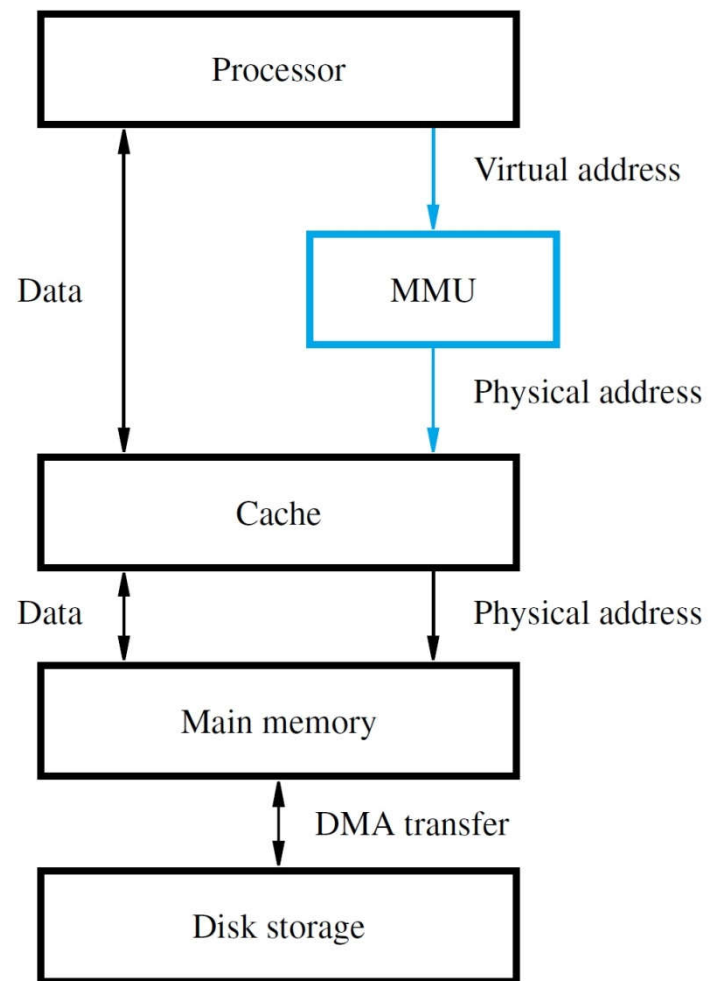
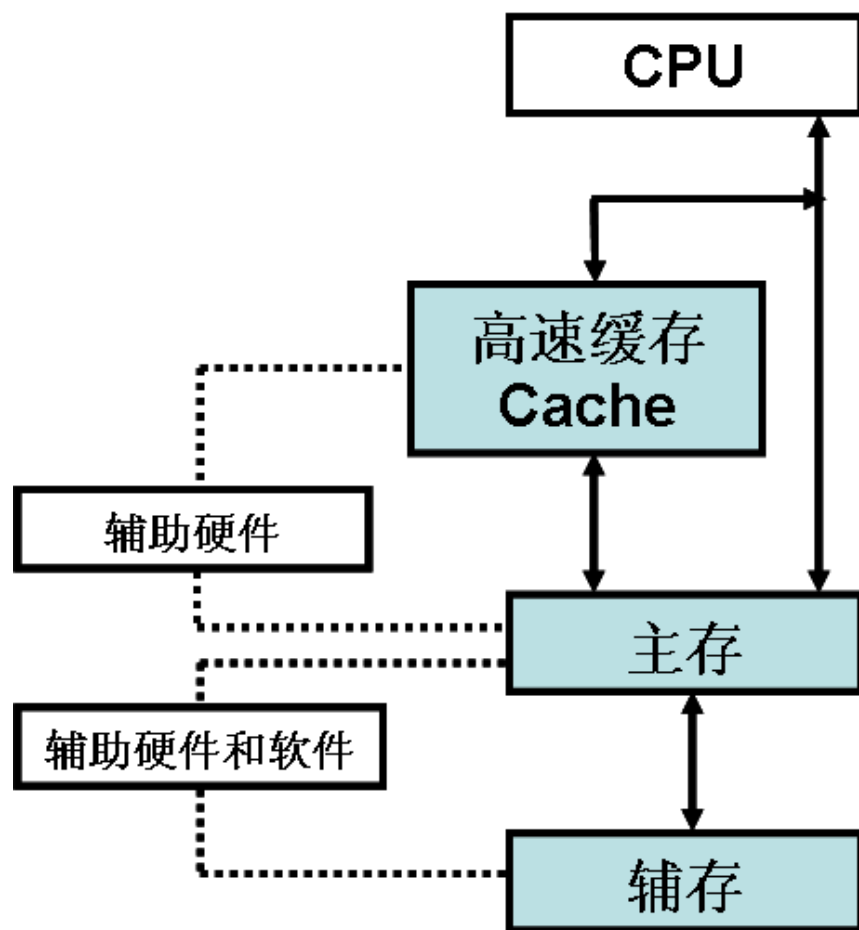
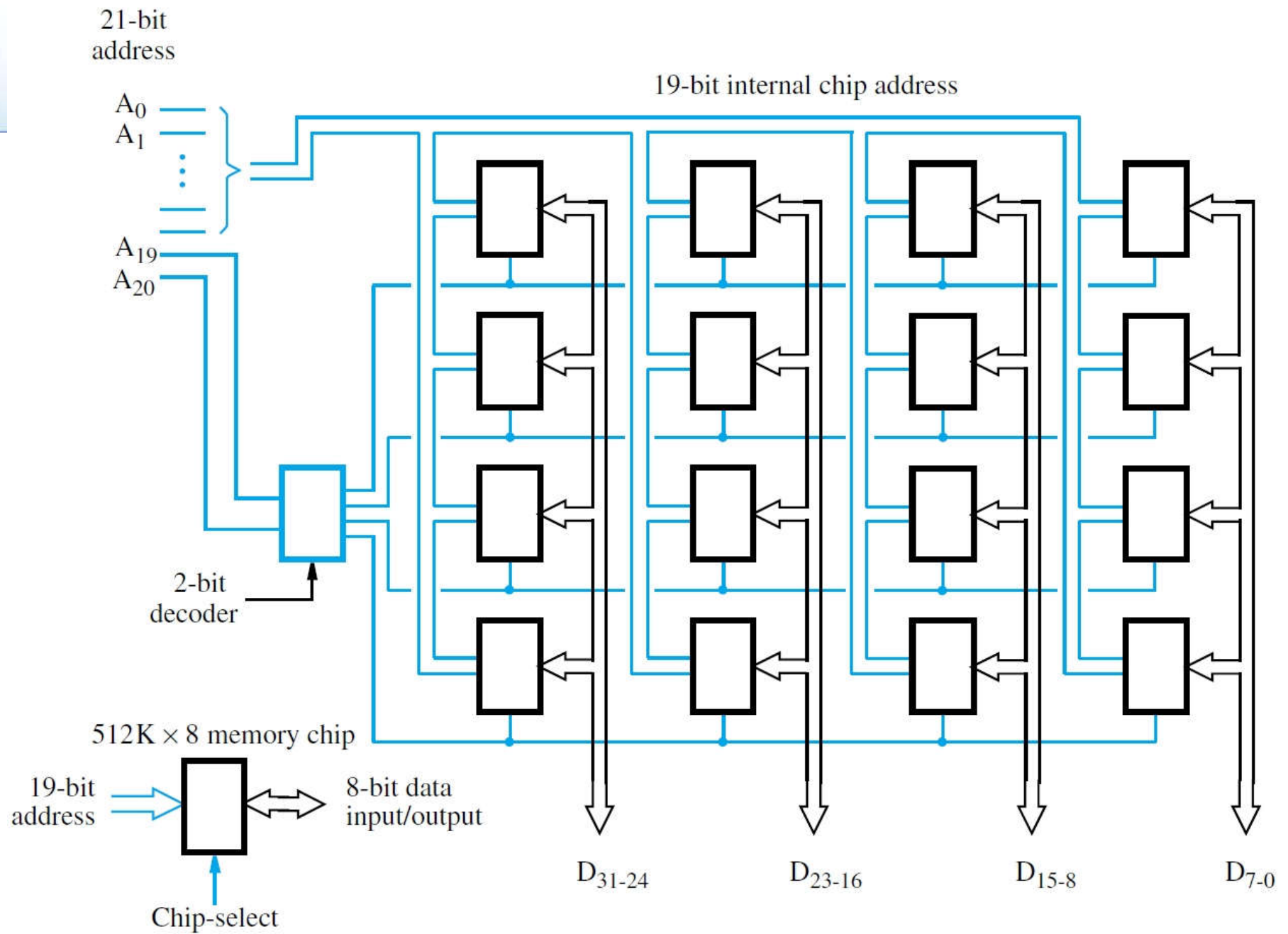


FIGURE Ov.3: Typical PC organization. The memory subsystem is one part of a relatively complex whole. This figure illustrates a two-way multiprocessor, with each processor having its own dedicated off-chip cache. The parts most relevant to this text are shaded in grey: the CPU and its cache system, the system and memory controllers, the DIMMs and their component DRAMs, and the hard drive/s.

存储系统

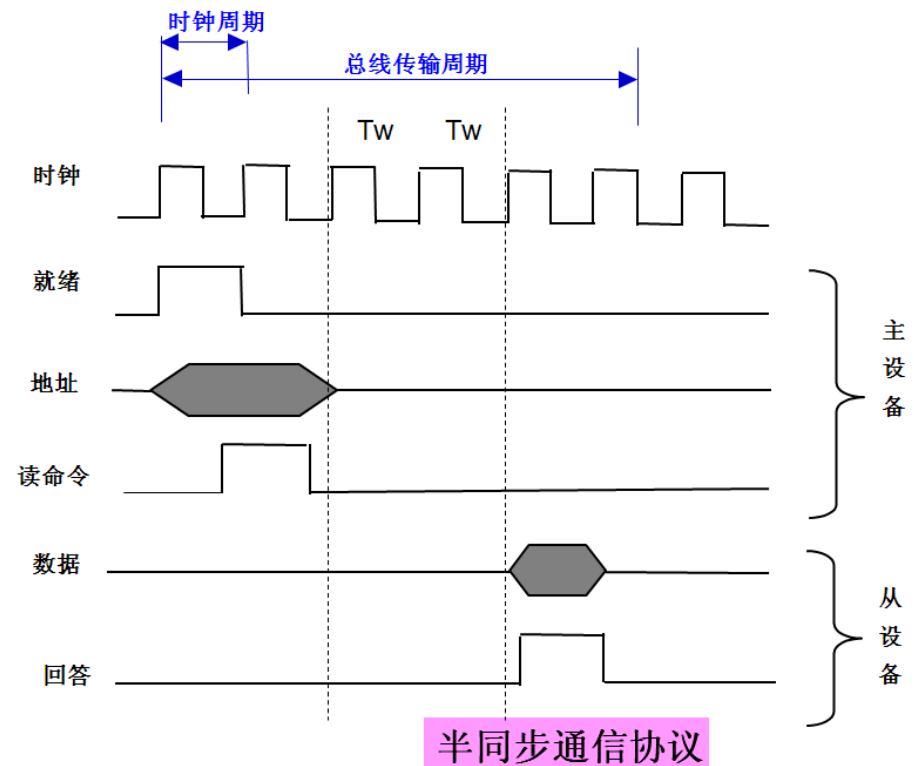
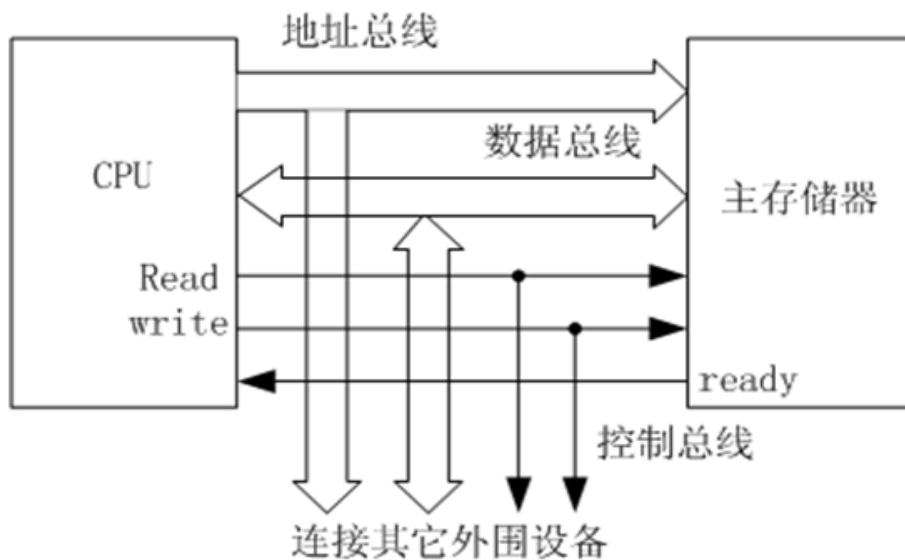




访存过程

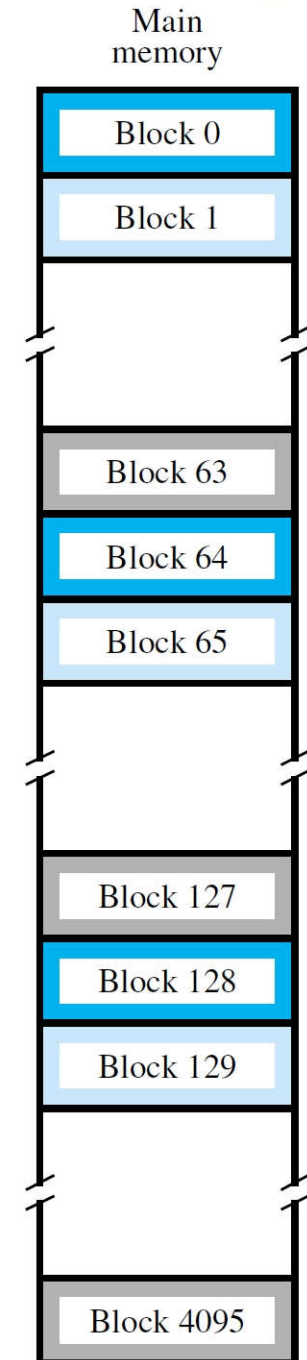
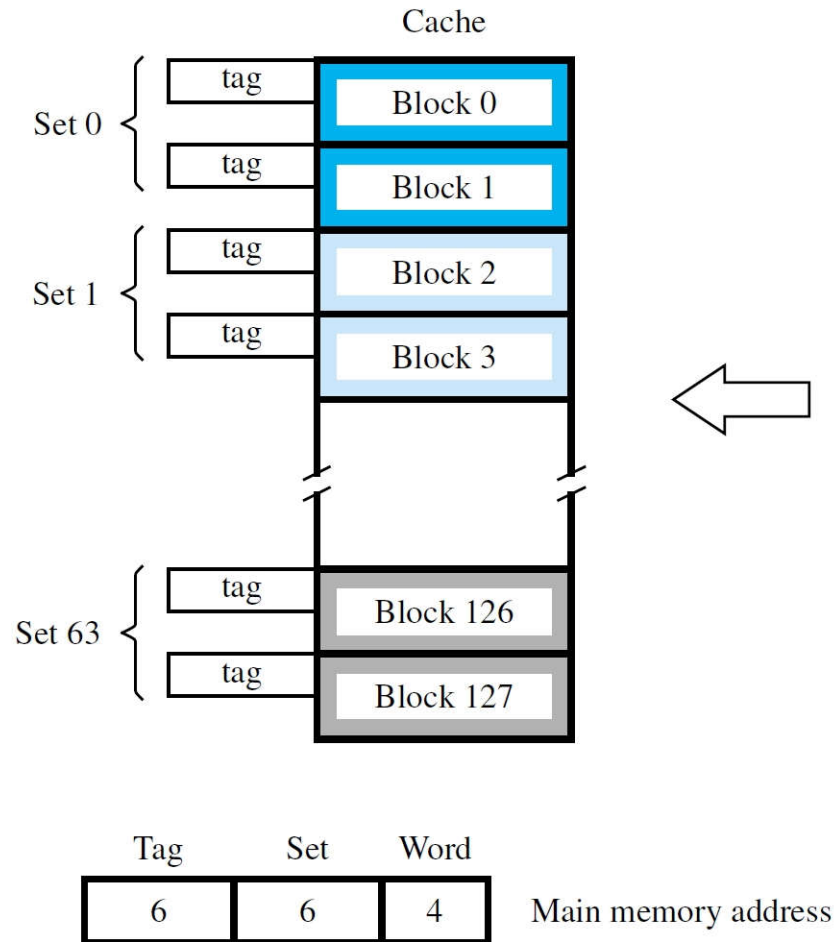


- Async SRAM/DRAM
- SSRAM/SDRAM
- 单字模式，突发模式

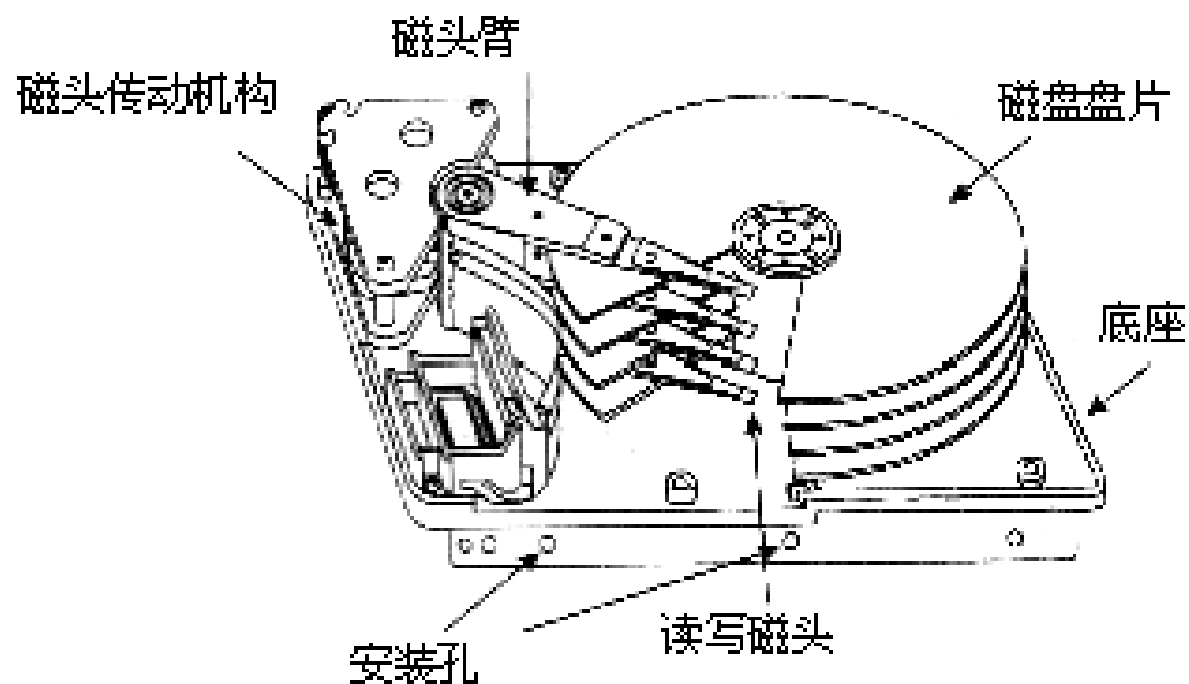


Set-Associative Mapping

- 作用
- 映射关系
- 读写过程
- 替换策略

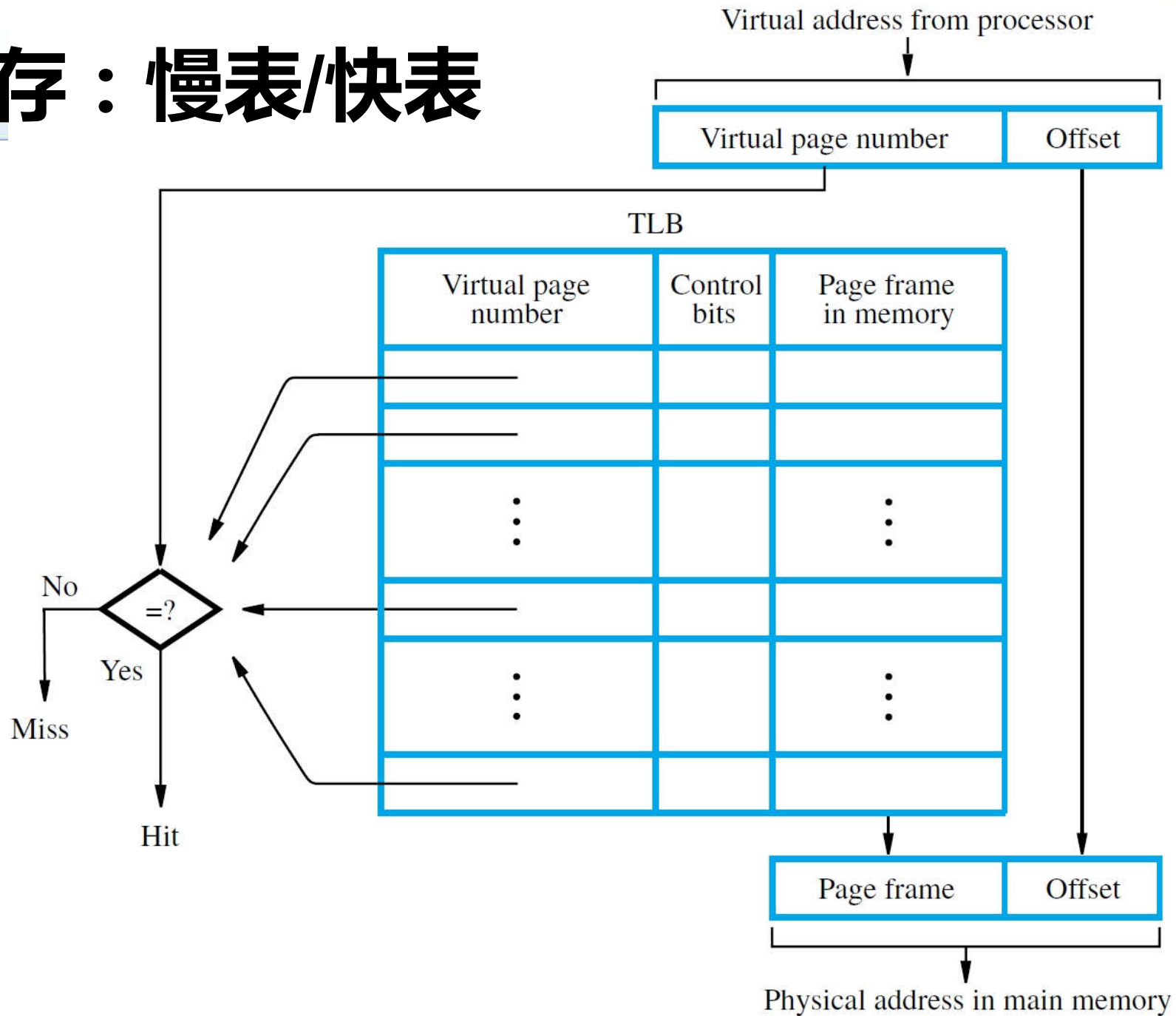


外存：硬盘



硬盘的内部结构

虚存：慢表/快表





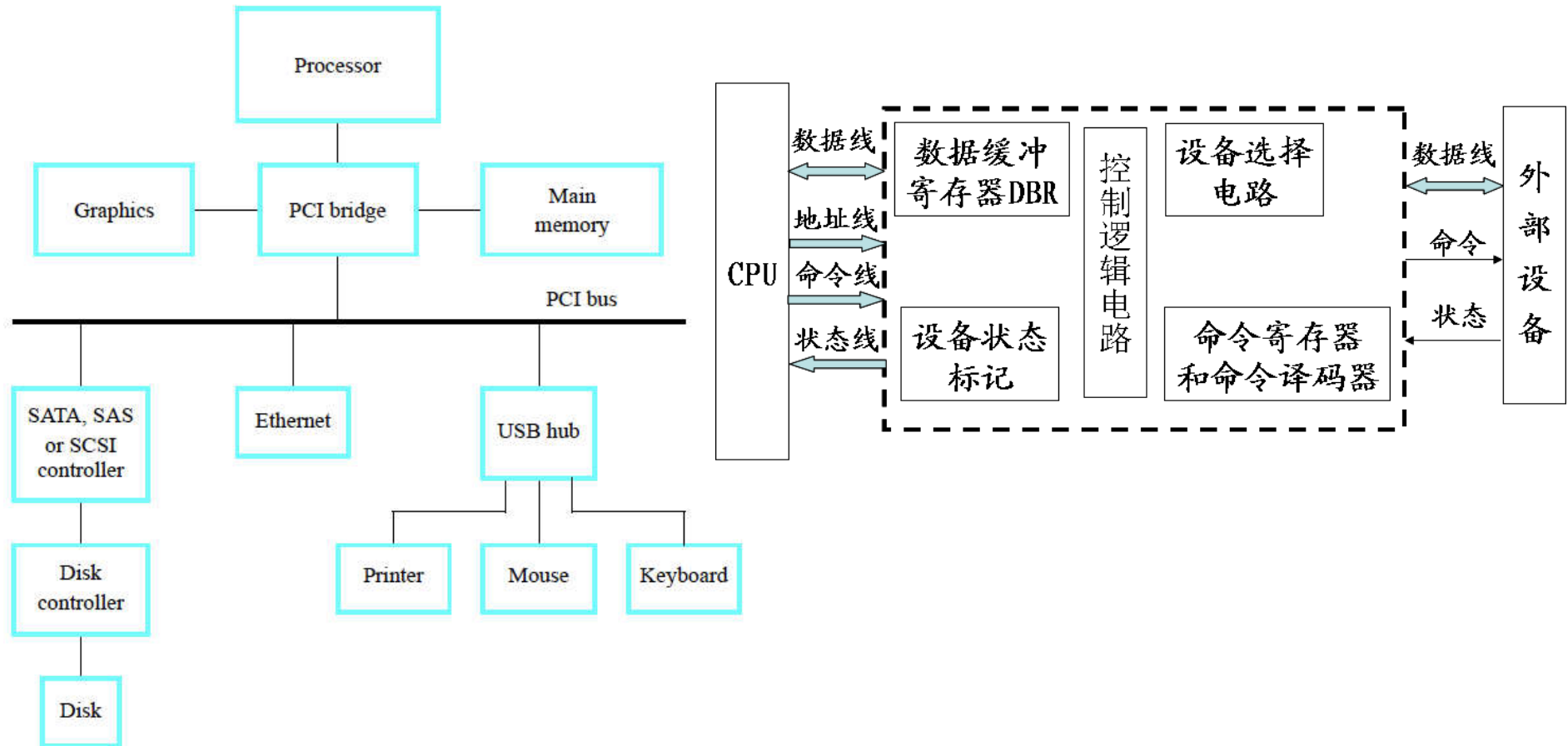
计算机组成原理

I/O

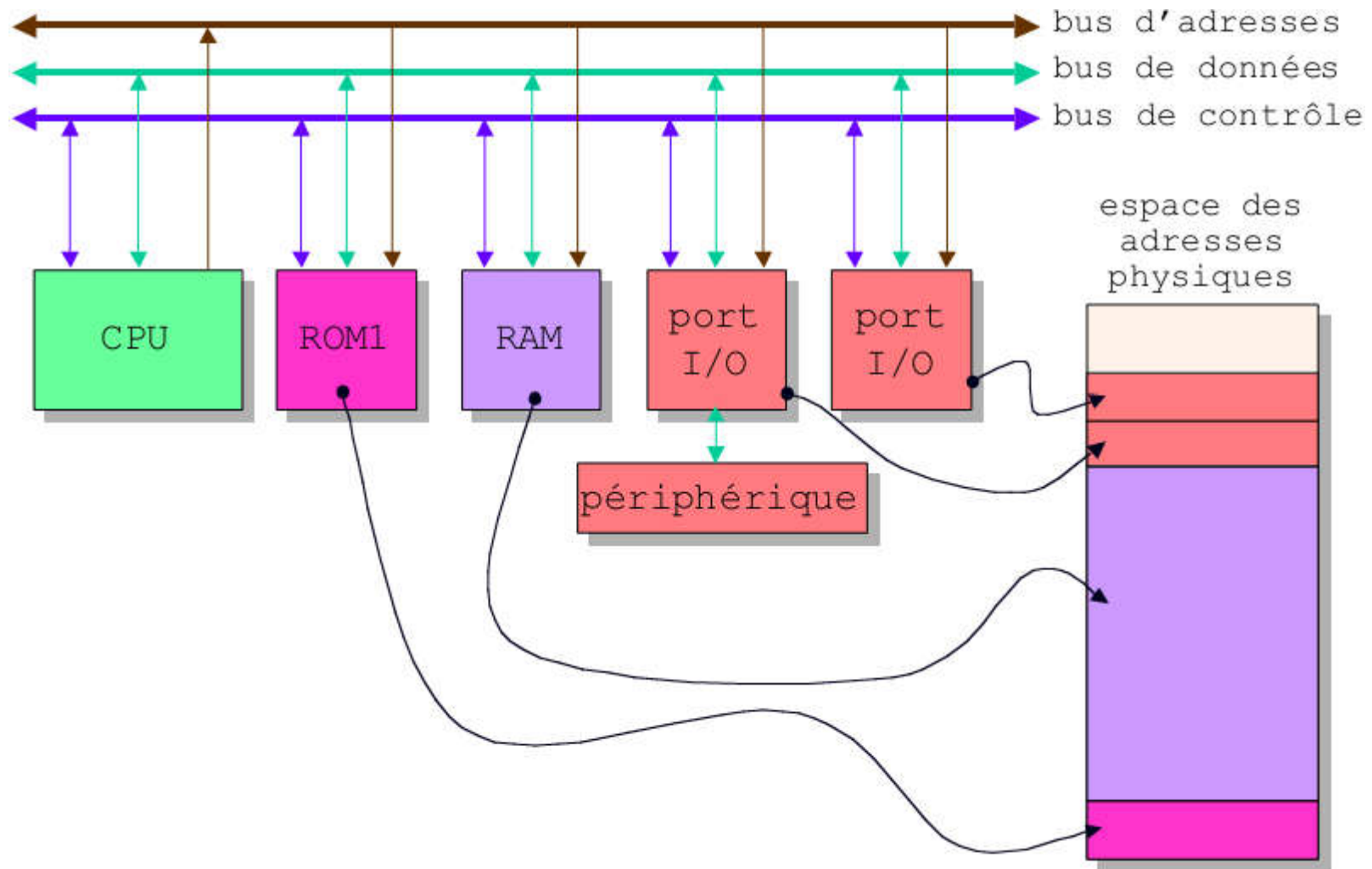
接口（端口）

程序控制，中断，DMA

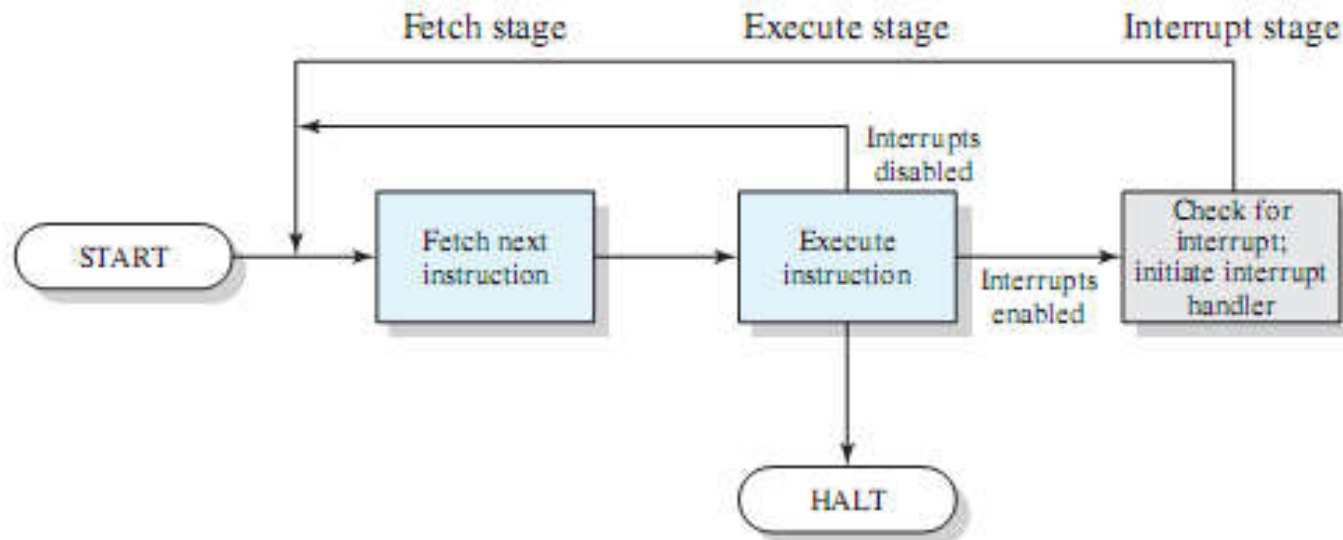
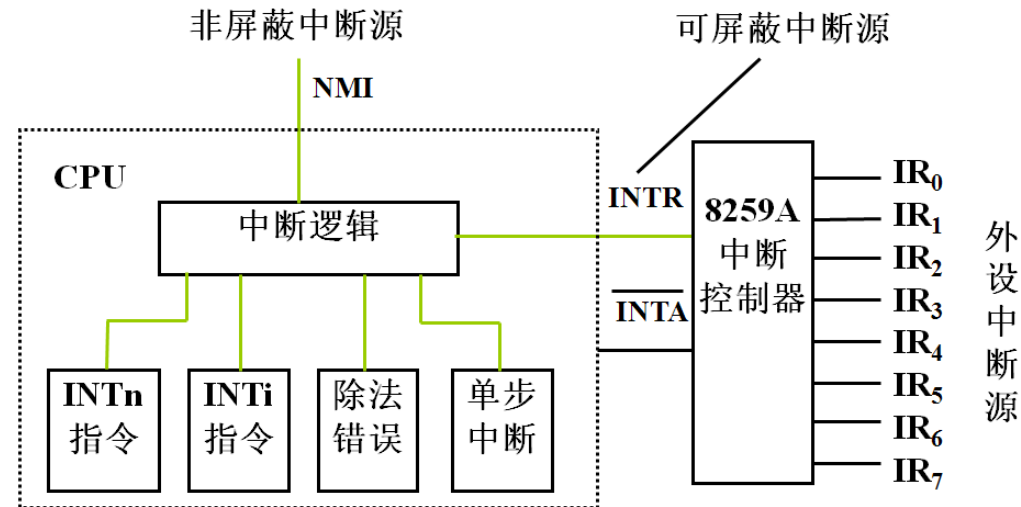
I/O Interface



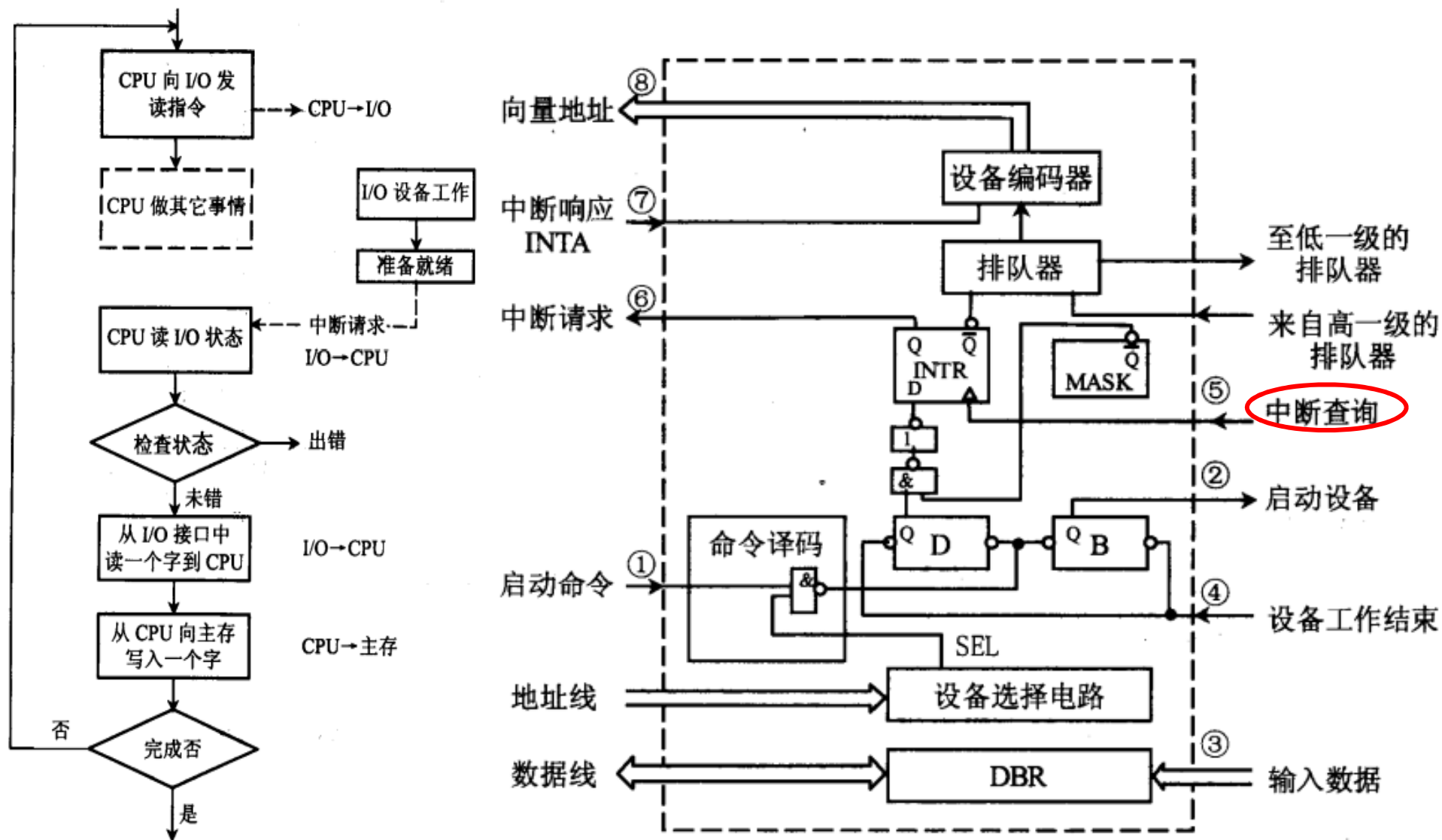
I/O Port Addressing



Instruction Cycle with Interrupts

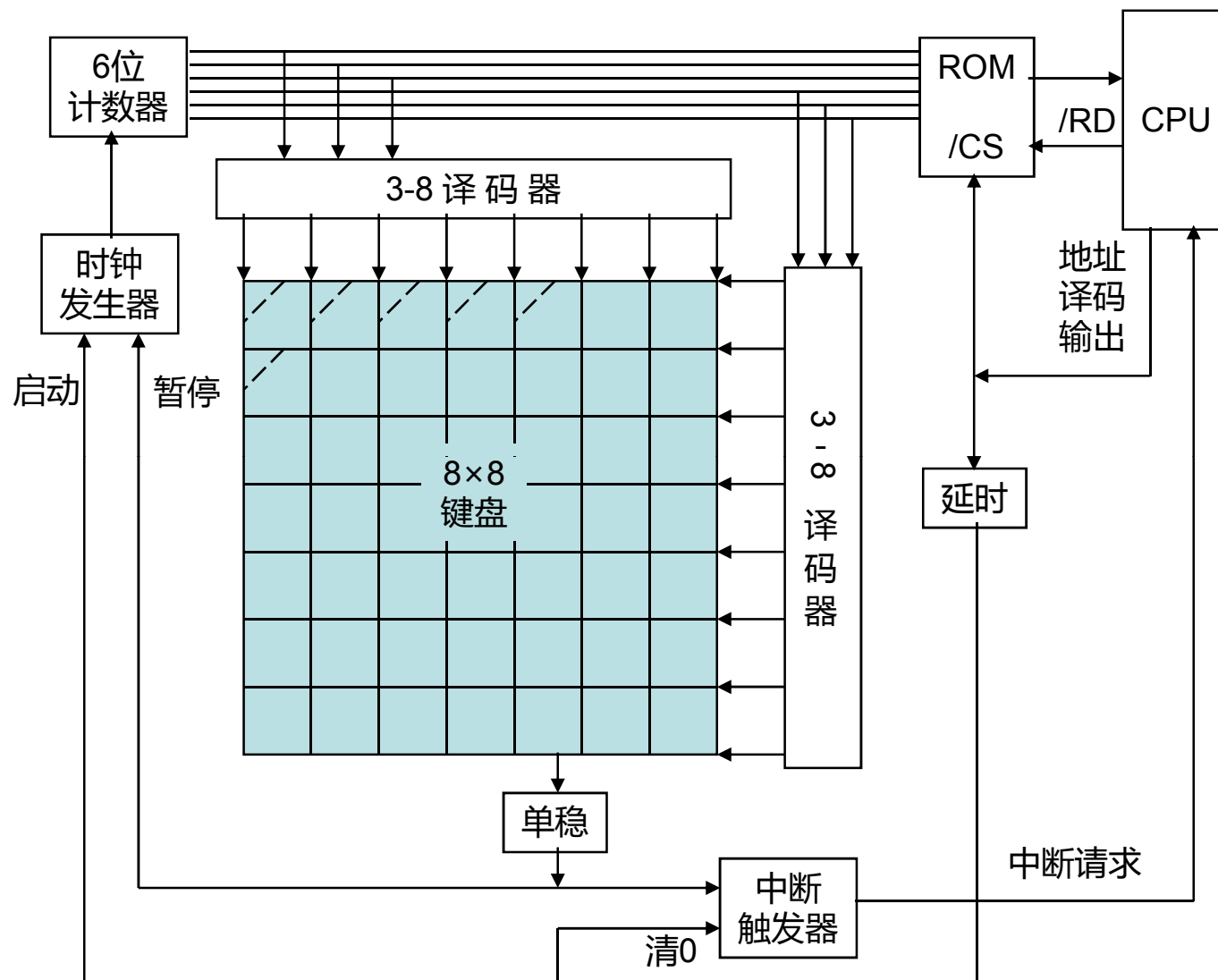


程序中中断方式接口电路的基本组成（分布式）





Ex : 编码键盘原理图



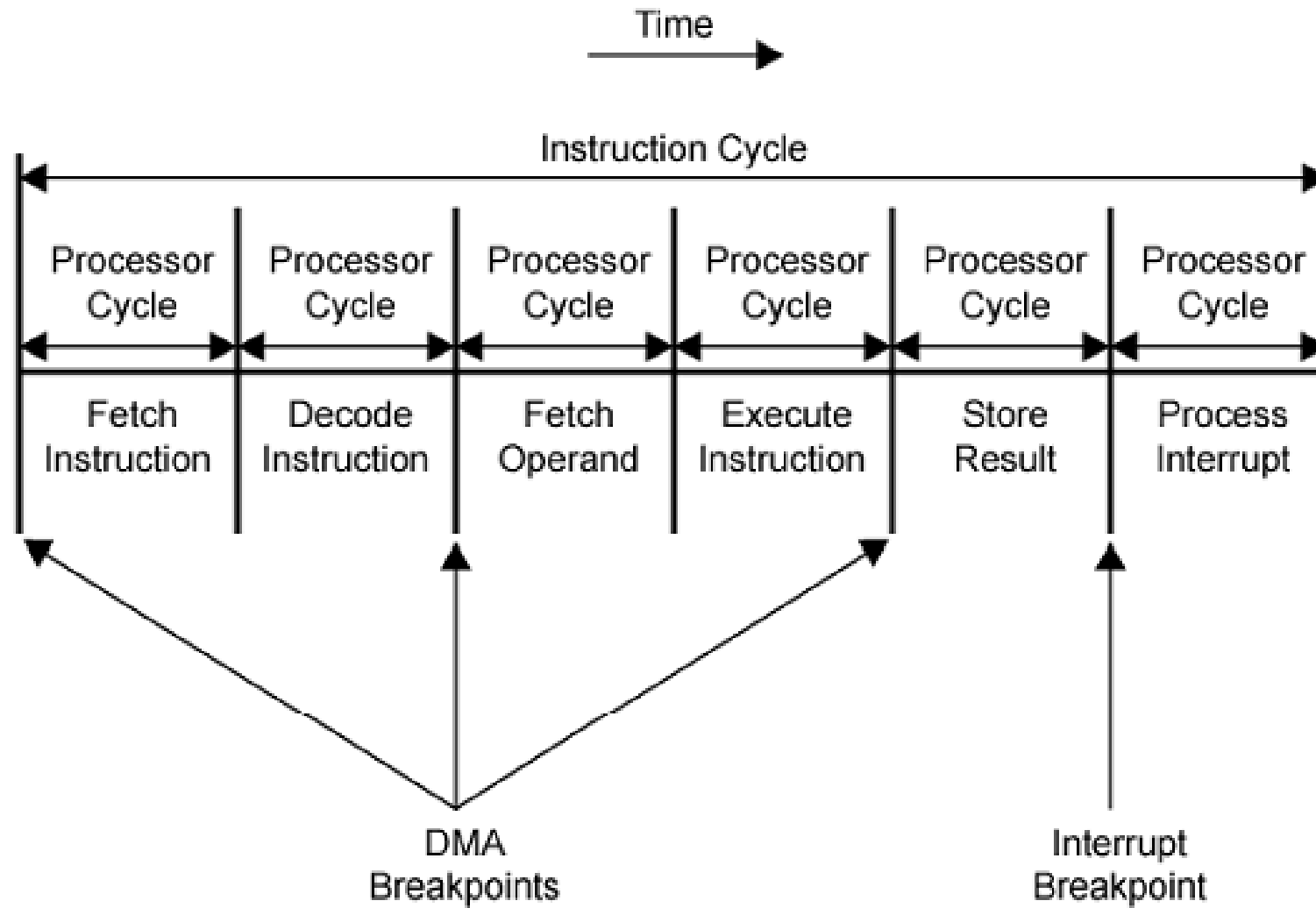
要点：

ROM中存储各个按键的**字符码** (ASCII)

计数器循环计数**扫描**键盘得到当前按键的**位置码**

按键按下时停止计数，并产生中断请求，CPU读**字符码**

DMA and Interrupt Breakpoints





计算机组成原理

指令系统ISA

程序员可见的计算机属性

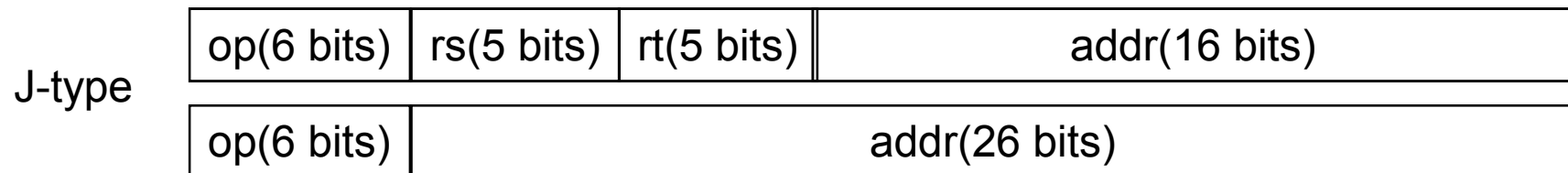
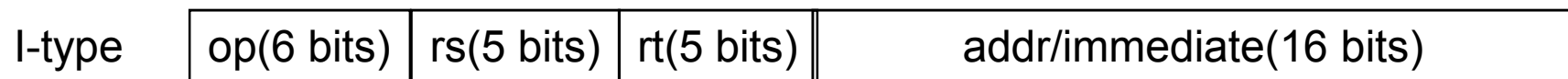
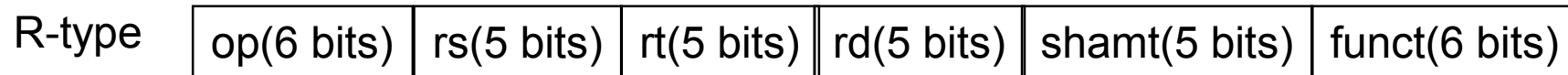
指令字格式，寻址方式，指令类型

CISC，RISC，VLIW

MIPS指令类型与格式



- 100余条指令（Hennessy中33条），共32个通用寄存器
- 指令格式：定长32位
 - R-type：arithmetic instruction
 - I-type：data transfer, arithmetic instruction（如addi）
 - J-type：branch instruction(conditional & unconditional)



MIPS寻址模式addressing mode

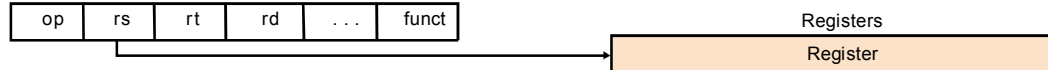
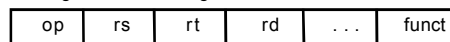


- 立即寻址
- 寄存器寻址：R-type
- 基址寻址：I-type
- PC相对寻址
- 伪直接寻址
(pseudodirect
addressing)
 - 注意：字长32位
 - 26位形式地址左移2位
(字对准)，与PC的高
4位拼接

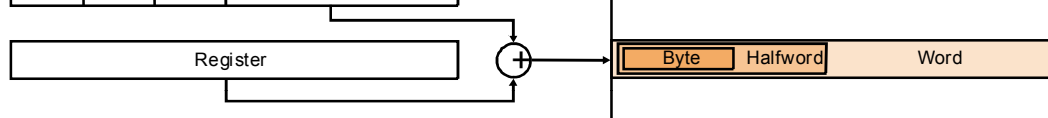
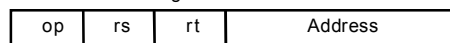
1. Immediate addressing



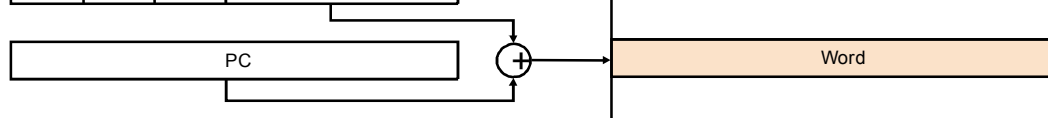
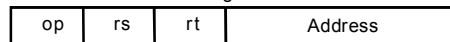
2. Register addressing



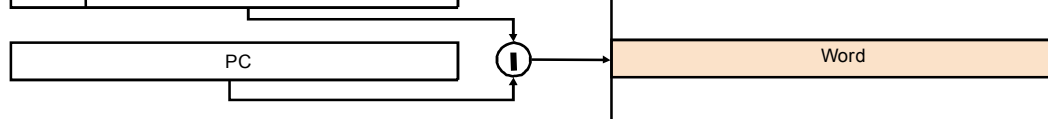
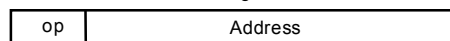
3. Base addressing



4. PC-relative addressing



5. Pseudodirect addressing





计算机组成原理

CPU

数据通路，控制器，ALU

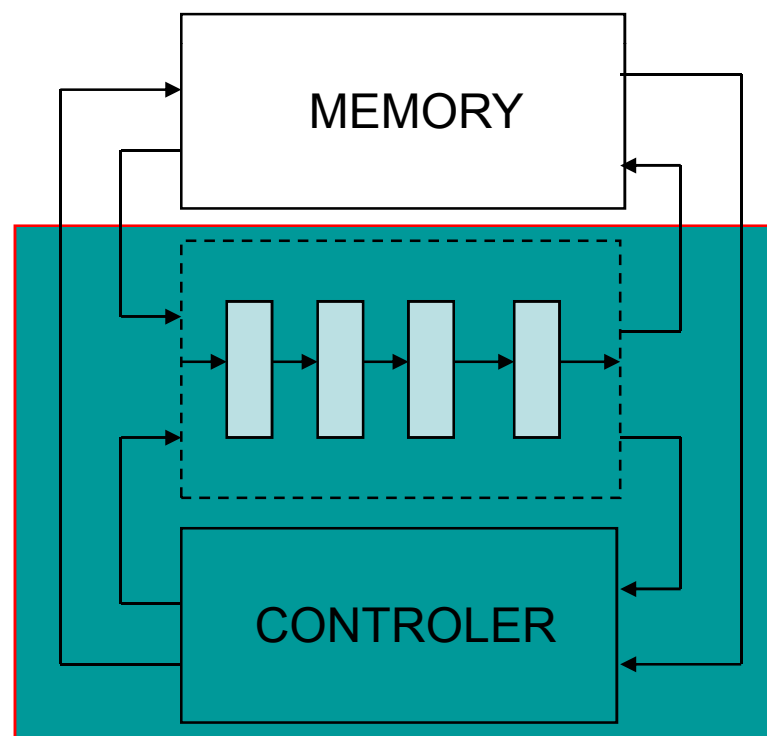
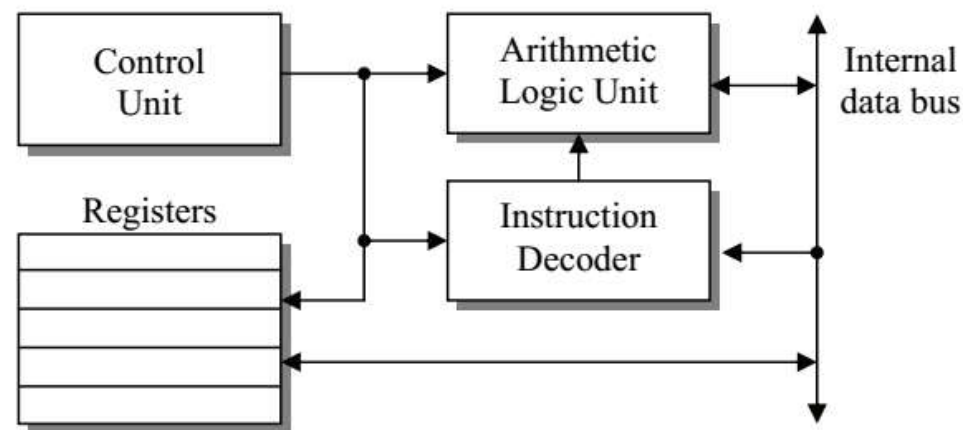
指令周期，机器周期

微操作

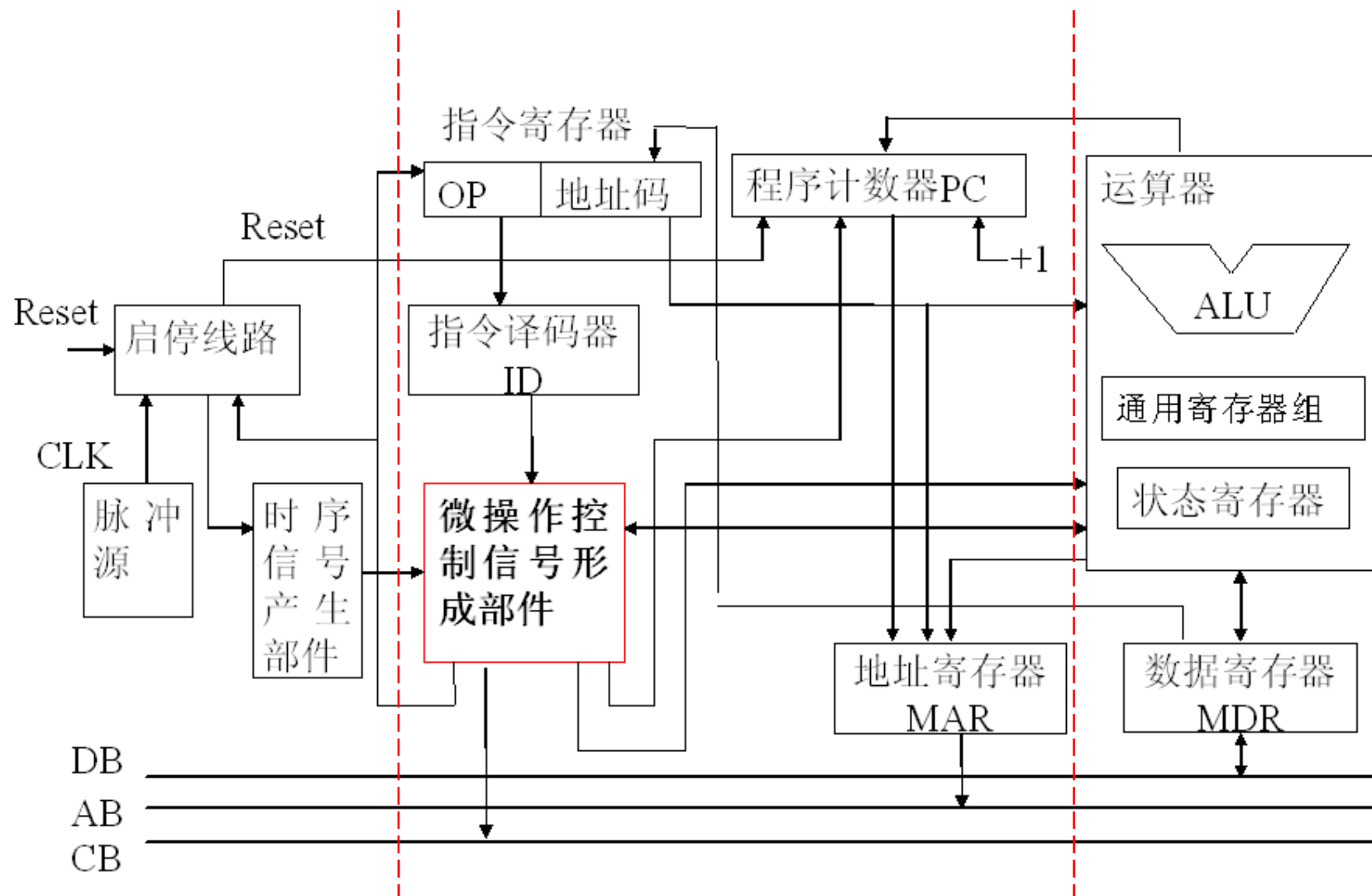
CPU的组成



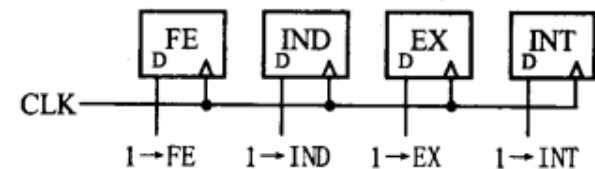
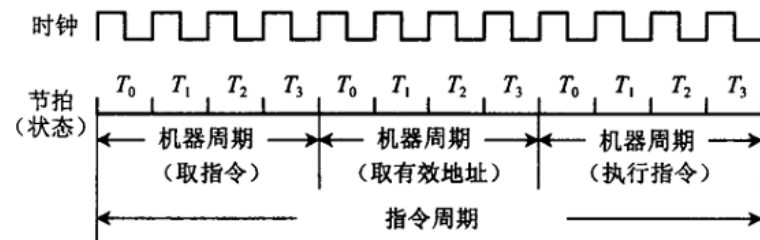
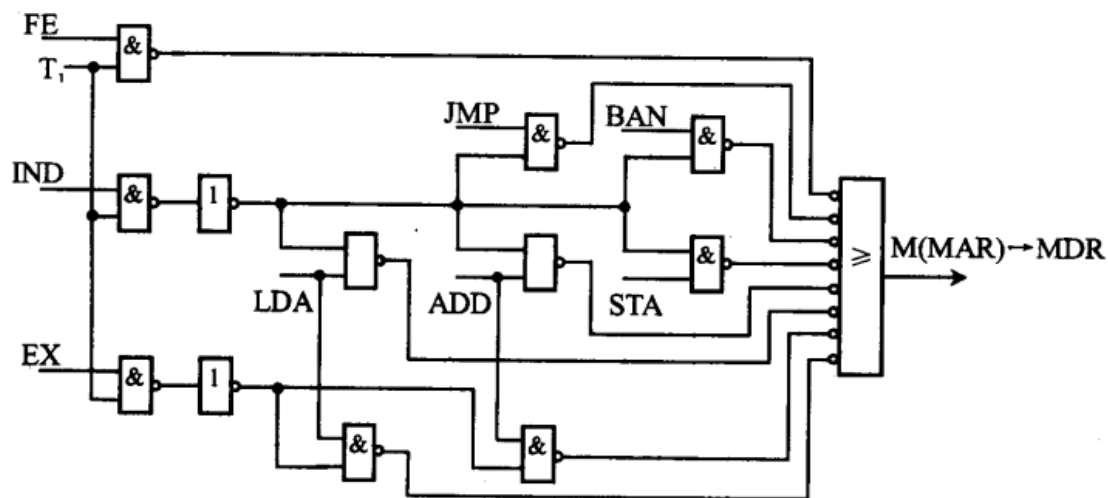
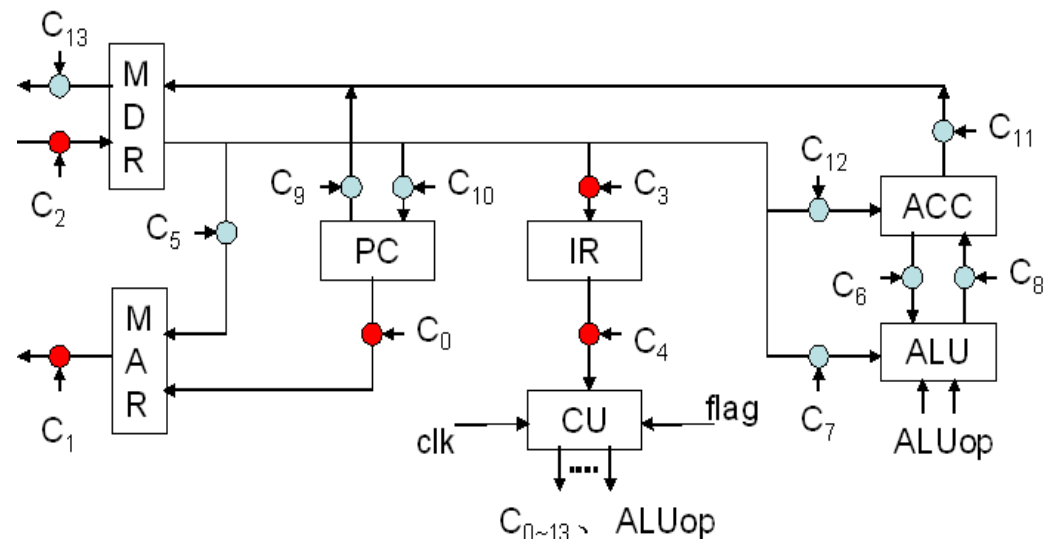
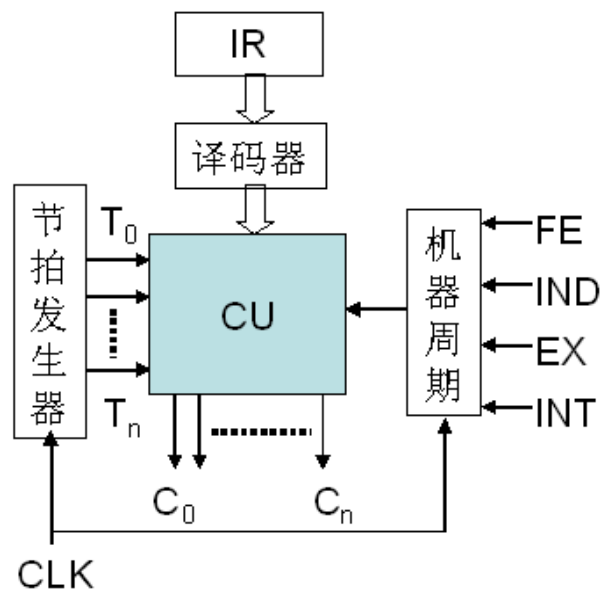
- 按功能部件划分
 - 运算器 (ALU)
 - 寄存器 (寄存器堆)
 - 控制器 (含中断控制)
 - 系统总线接口
 - MMU
 - L1 CACHE
 - 等
- 按数据流
 - 数据通路 (datapath) : 各种寄存器和运算器
 - 控制器 : 组合电路控制逻辑、微程序控制器



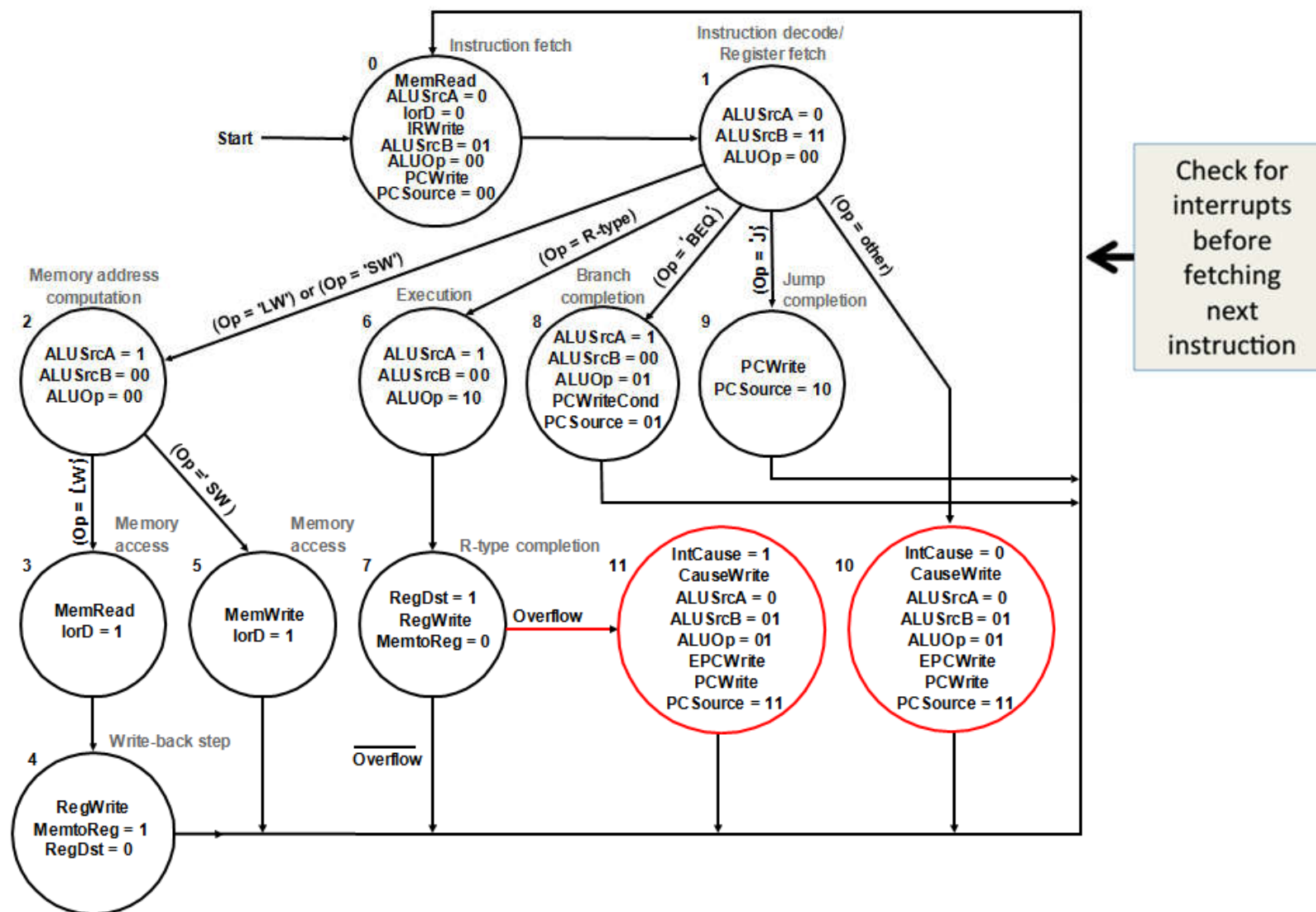
CPU = ISA/Timing/Datapath/Controller



CU设计：正确的时间，正确的控制



多周期实现的控制逻辑



DLX指令流水线



Instruction Fetch

IF

Instruction Decode
Register Fetch

ID

Execute
Address Calc.

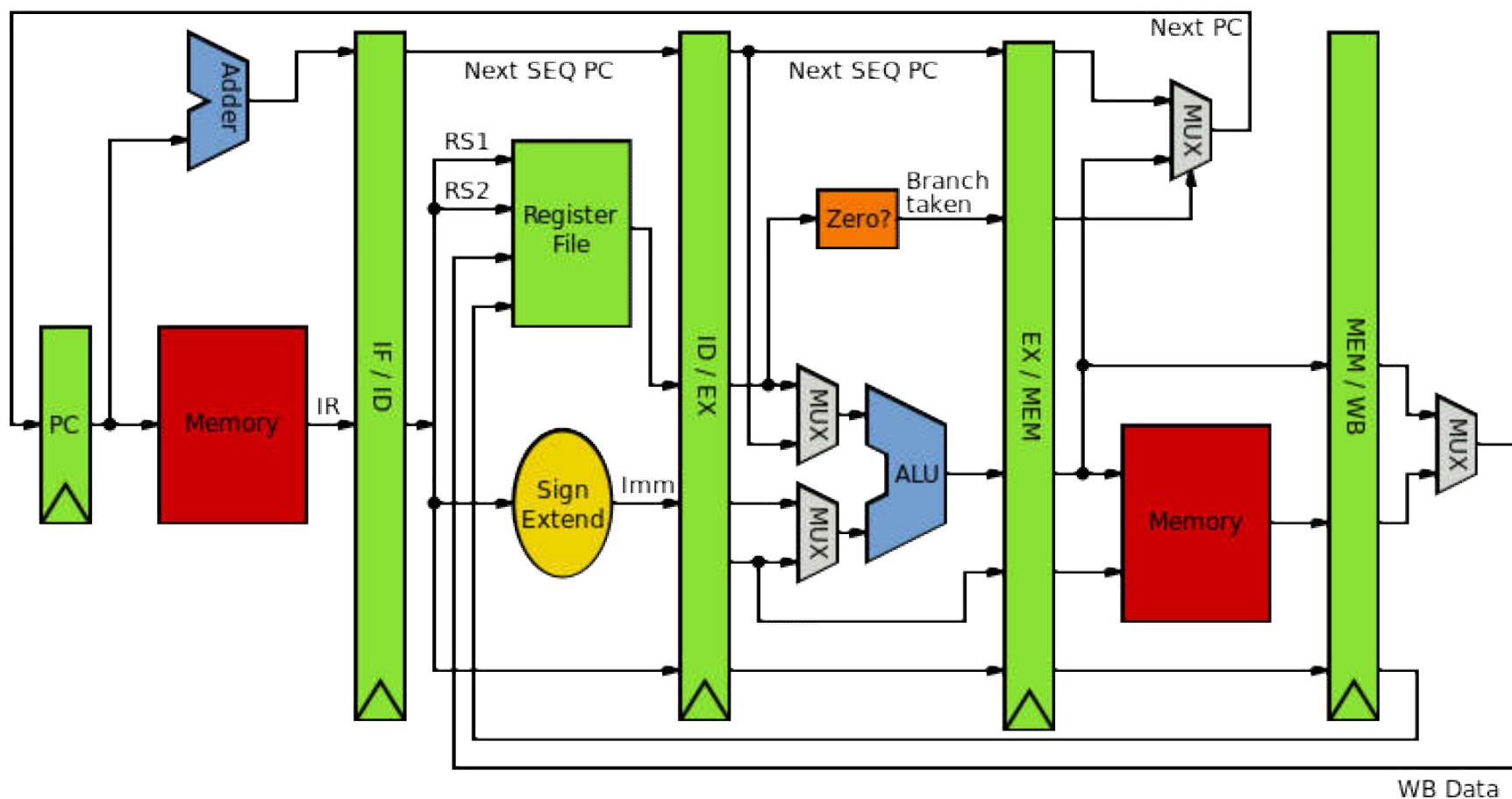
EX

Memory Access

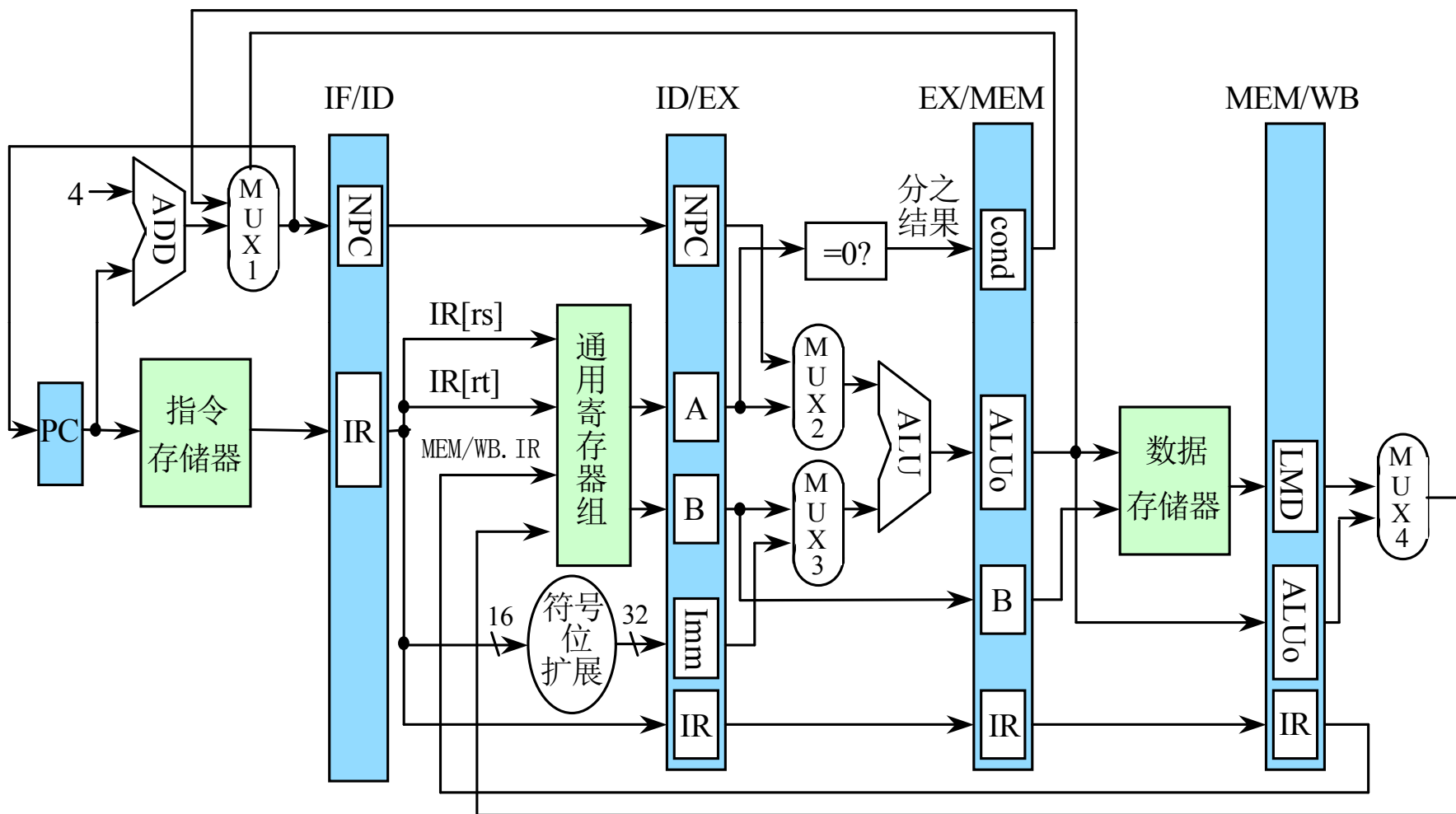
MEM

Write Back

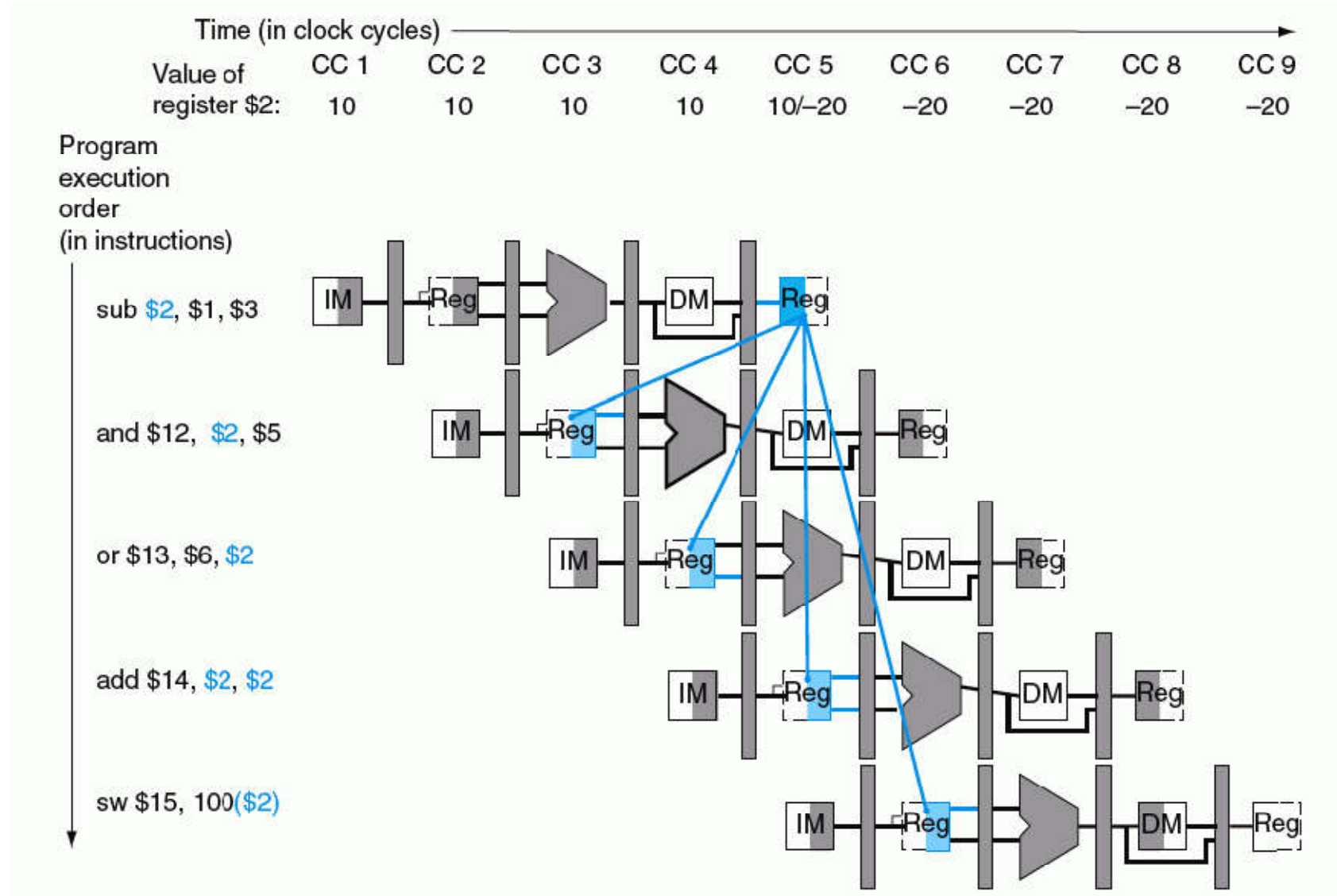
WB



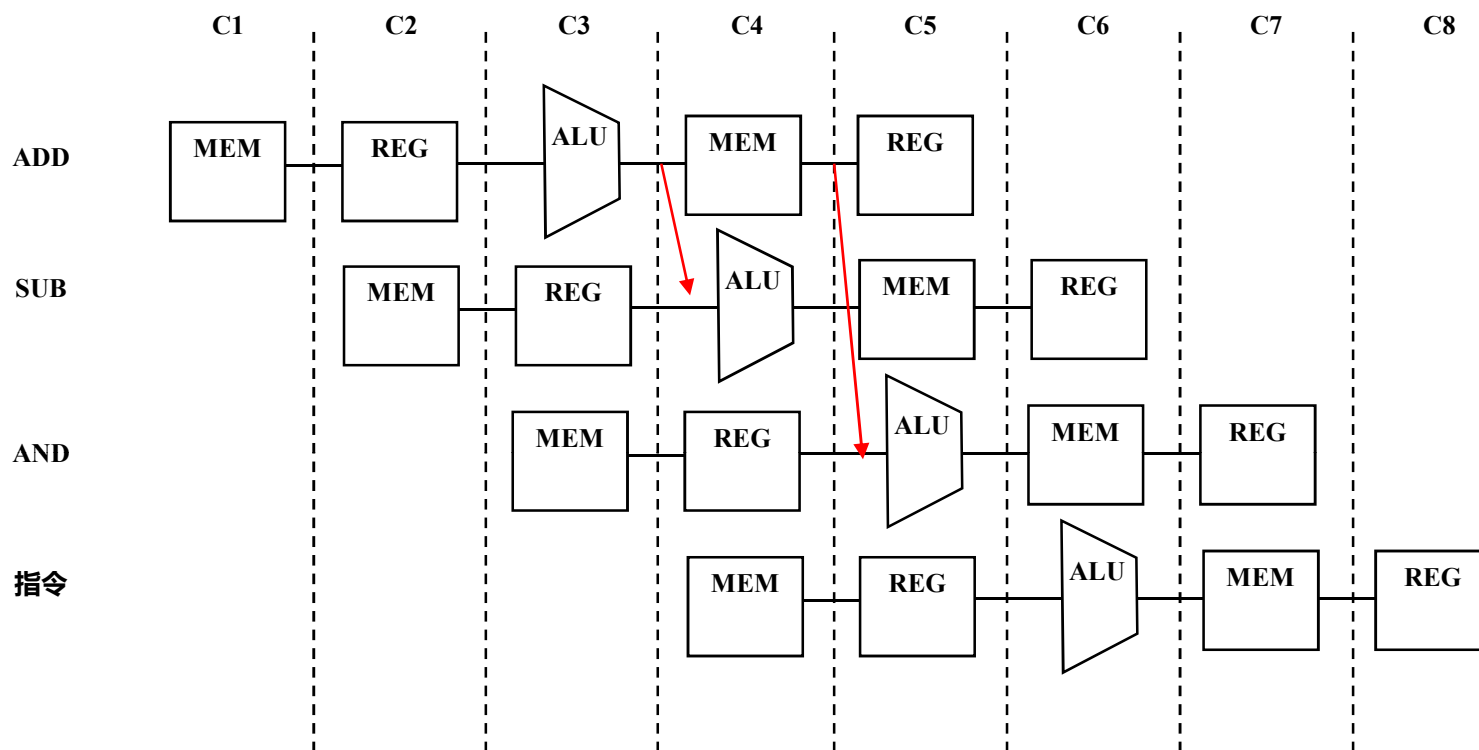
流水段寄存器



data hazard(RAW)



数据定向技术 (forwarding , bypassing)



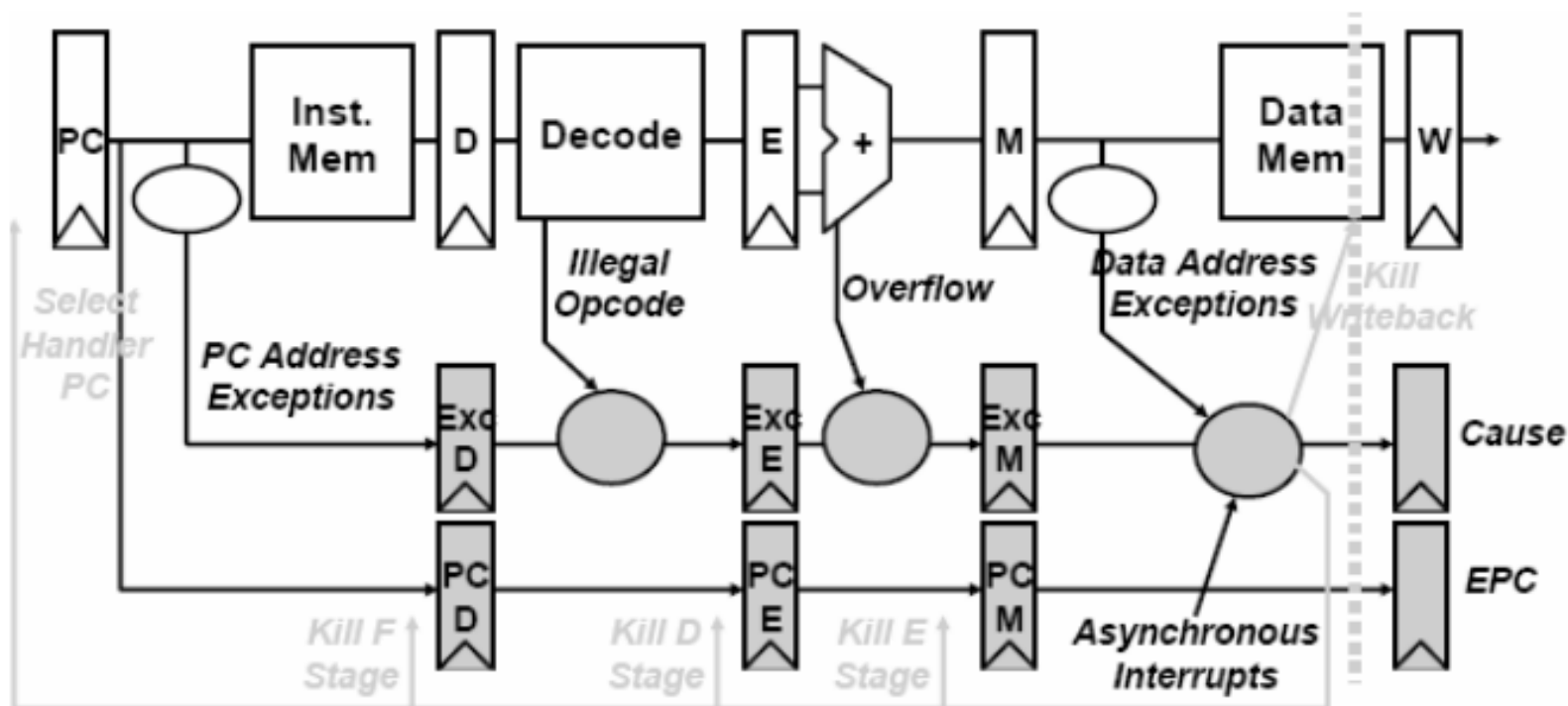
消除RAW

- 方法一：检测数据相关（互锁，INTERLOCK），前推
- 方法二：禁止生成此类指令序列
 - 编译器（静态指令调度），动态调度。

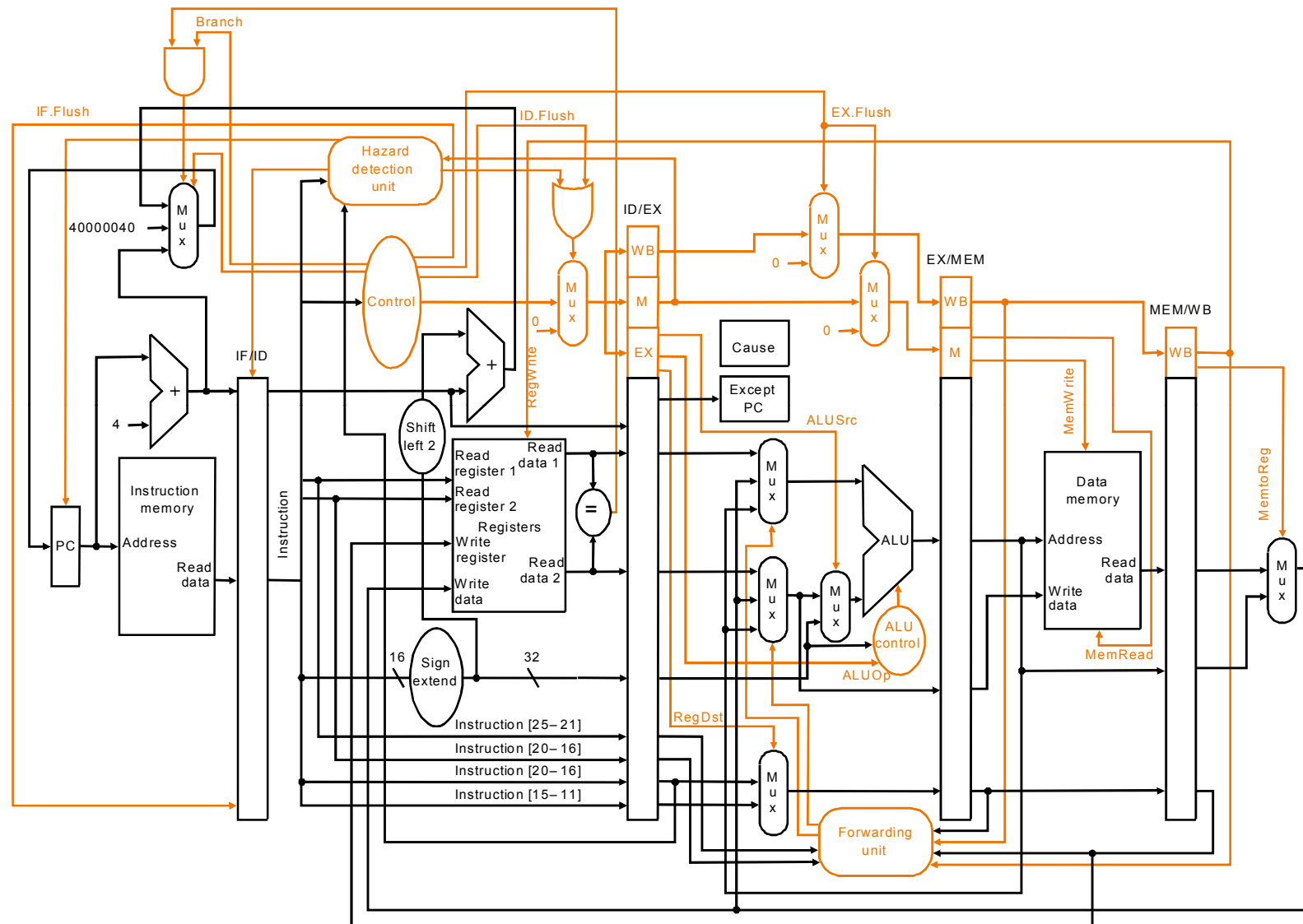


各段产生的异常及处理：MIPS的策略

- 保持流水线的异常标记直到**提交点**（M段）
- 早期流水段的异常抑制后来的异常
- 提交点处引入外部异常（抑制其他异常）
- 如果提交点有异常，则更新cause和EPC，清除所有流水段，回复PC值到fetch段



The final datapath & control

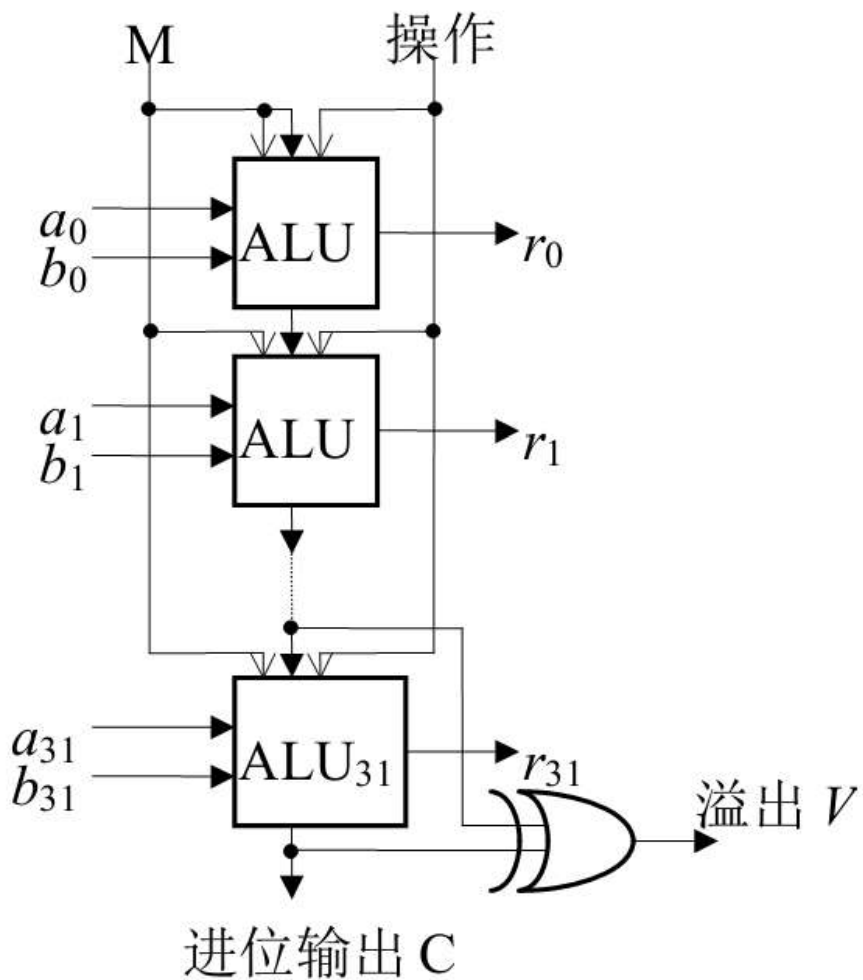
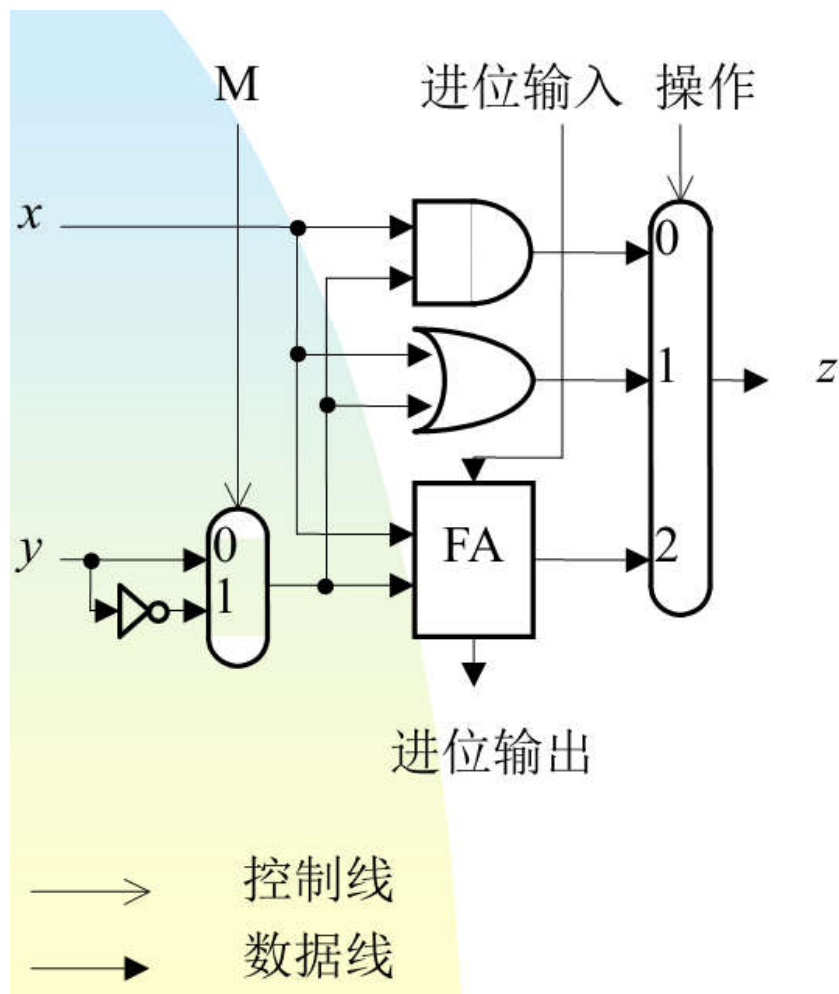




计算机组成原理

运算方法
ALU

算术逻辑运算单元ALU



非典型问题



- 最小的计算机系统？
 - 需要哪几条指令？需要哪几种寻址方式？由哪些部件构成？
- C语言的计算机模型？
 - `int i` 在哪儿？全局变量与局部变量差异？
 - `getchar()` 的执行过程？
- C语言、编译器、OS、Computer的关系？
- 函数调用过程与中断响应过程的异同？
- 计算机现在在干啥？
- 总线的副作用？
- CPU如何访问RAM？
- Cache的副作用？
- 中断的副作用？
- DMA的副作用？
- 如何确定一个程序的执行时间？



Thank you

