

# 计算机的性能指标

## 机器字长

机器字长是指参与运算的基本位数，即CPU在同一时间内能进行一次处理的二进制数的位数

概念辨析	● 机器字长	标志着计算精度，也反应寄存器、运算部件和数据总线的位数。机器字长越多，操作位数也就越多，计算精度也就越高。
	● 存储字长	一个存储单元的二进制代码长度
	● 指令字长	一个指令所包含的二进制位数

## 数据通路带宽

数据通路是指数据总线一次能并行传送信息的位数，它影响计算机的有效处理速度

CPU内部数据同类宽带	一般等于机器字长，即内部数据线的位数
CPU外部数据同类宽带	等于系统数据总线一次所能并行传送的信息位数，即CPU与主存、输入输出设备之间进行一次数据传送的信息位数，也称为存储字长

## 主存容量

单位	位 (bit)：	表示二进制位。位是计算机内部数据储存的最小单位
	字节 (Byte)：	规定一个字节由8个二进制位构成，1B = 8bit
	字 (word)：	一个字通常由一个或多个字节构成（一半是字节的整数倍）
指一个主存储器所能存储信息的最大容量		
对于字节编址的计算机，用字节数表示主存容量		
对于字编址的计算机，用字数乘以字长表示主存容量		

## 运算速度

① 吞吐量	指系统在单位时间内处理的请求数。 主要取决于主存的存取周期
② 响应时间	指令系统对请求做出的响应时间 响应时间 = CPU时间（运行一个程序所花费的时间） + 等待时间（用于磁盘访问、主存储器访问、I/O操作、操作系统开销等时间）的总和
③ 主频	又称为时钟频率，表示在CPU内数字脉冲信号振荡的速度。 主频越高、完成指令的一个执行步骤所用的时间就 越短，指令执行速度越快
④ CPU时钟周期	是主频的倒数，简称时钟周期，是CPU中最小的时间单位。主频通常以MHz为单位，1Hz表示1次/秒
⑤ CPI	执行一条指令所需的时钟周期数
⑥ CPU执行时间	运行一个程序所花费的时间 CPU执行时间 = 时钟周期数/主频 = （指令条数*CPI）/主频
⑦ MIPS、MFLOPS、GFLOPS、TFLOPS	<div><div>MIPS</div><div>每秒执行多少百万条指令</div><div><math>MIPS = \text{指令条数} / \text{执行时间} \times 10^6 = \text{主频} / CPI</math></div></div> <div><div>MFLOPS</div><div>每秒执行百万次浮点运算</div><div><math>MFLOPS = \text{浮点操作次数} / \text{执行时间} \times 10^6</math></div></div> <div><div>GFLOPS</div><div>每秒执行十亿次浮点运算</div></div> <div><div>TFLOPS</div><div>每秒执行万亿次浮点运算</div></div>

进位计数制及其相互转化：十进制、二进制、八进制和十六进制的相互转化

真值与机器数

真值

通常用带 +、- 加上绝对值来表示数值的大小

机器数

通常约定二进制数的最高位为符号位，0表示正好，1表示符号（分为原码、补码和反码）

BCD码

用4位二进制来表示十进制，有冗余状态

六种冗余状态，超过9，需要加6（二进制0110）修正

字符与字符串

ASCII码

标准ASCII码采用7位二进制数表示控制支付、大小写字母、数字、和专用符号

汉字编码

包括输入编码、汉字内码、汉字字形码

汉字国际码 = 汉字区位码（16进制区位码） + 2020H

汉字机器内码 = 汉字国标码（16进制区位码） + 8080H

校验码

概念：校验码是指那些能够发现错误或能够自动纠正错误的数据编码

奇偶校验码

在原编码上加上一个校验位、检验整个校验码中“1”的个数是奇数还是偶数

奇偶校验

整个校验码（有效位+信息位）“1”为奇数个

偶校验

整个校验码（有效位+信息位）“1”为偶数个

只能发现奇数个错误，不能纠正错误

循环冗余校验码

在K位信息码后面再加上R位检验码、利用模2除法检验编码正确性

海明校验码

按照某种规律分成若干组、每组安排一个检验位进行奇偶性测试，产生多位检验信息

校验过程

n位信息位，K位校验位，满足 $n+k+1 \leq 2^k$

纠错能力恒小于校验能力，即校验错的位数大于纠正的错位数

1 算出k

2 写出n+k为H编号

3 分校验组

4 校验位 = 校验组异或

4 写出完整的海明码H

校验原理

$S_i = P_i$ 异或检验组，全为0不出错，不全为0，转换为十进制i，找H<sub>i</sub>取反即可

CRC循环冗余校验码

1 写出多项式，最高次数 = 信息位补0个数

2 补0后的信息位与多项式进行模2除法

3 将余数替换掉信息位后面的0

校验原理

接收方收到数据与多项式进行模2除法、余数为0，没有出错，余数不为0，将所得余数对应的信息位取反

数据的表示与运算  
(一)

数制与编码



数据的表示和运算  
(二)

定点数的表示与运算

表示

原码

★ 用机器数的最高位表示数的符号、其余各位表示数的绝对值

反码

★ 正数与原码一样，负数除符号位外，各位是原码的按位取反

补码

★ 正数与原码一样，负数为反码的末位加1

移码

★ 补码的符号位取反

注意：“0”的表达和补码多表示一位的情况

算术运算

原码

符号和数值分开处理

原码加法：先判符号，若相同，则绝对值相加，结果符号位不变；若不同，则做减法，绝对值大的数减去绝对值小的数，结果符号位与绝对值大的数相同。

原码减法：两个原码表示的数相减，首先将减数符号取反，然后将被减数与符号取反后的减数按原码加法进行运算

原码乘法

原码一位乘

原码两位乘

原码除法

恢复余数法

不恢复余数法

补码

符号和数值不用分开处理

补码加法： $[X]补 + [Y]补 = [X + Y]补$

补码减法： $[X]补 - [Y]补 = [X]补 + [-Y]补 = [X - Y]补$ ，其中 $[-Y]补$ ：将 $[Y]补$ 连同符号位一起求反，末尾加1。

补码乘法

补码一位乘（Booth）

补码两位乘

补码除法

加减交替法

先判断符号，然后绝对值相加减

溢出

概念：计算出的数超过了缓冲区的范围，导致部分数值丢失的操作

溢出判断方法

直接判断，数据是否超出表示范围

补码双符号位

根据进位情况判断（符号进位异或数据最高位进位）

浮点数的表示与运算

表示

组成：阶符、阶码、数符、尾数

规格化：规定尾数的最高位必须是一个有效值

原码：正数：0.1.....  
负数：1.1.....

补码：正数：0.1.....  
负数：1.0.....

IEEE754标准：尾数用原码表示。阶码用移码表示

规格化浮点数的范围

E = 0

M = 0

正0或负0的表示

M ≠ 0

非规格化渐进式下溢区

1 ≤ E ≤ 254

E = 255

M = 0

正或负无穷大

M ≠ 0

NaN

运算（加减运算的步骤）

① 对阶、小阶向大阶看齐

② 尾数求和

② 规格化

算数逻辑单元（ALU）

一位全加器

★ 最基本的加法单元，输入两位加数及低位的进位（三个输入）、输出和与高位的进位（两个输出）

串行加法器

★ 一位全加器的加法器简单相连、串行进位

并行加法器

★ 由若干个全加器组成，使用先行进位提高加法器的运算速度

ALU

★ 功能较强的组合逻辑电路，它能进行多种算术运算和逻辑运算

C语言

数据类型的转换

★ 同一类型但长度不同的数据间的转换

短转换成长的（扩展）

长转换成短的（截断）

★ 定点方式与浮点方式间的转换

★ 整型数中的带符号格式与无符号格式间的转换

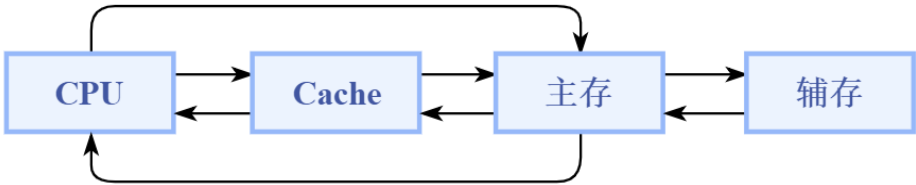
执行一个运算时间时有无符号数和带符号整数参加，那么，C编译器会隐含地将带符号整数强制类型转换为无符号数。

存储系统（一）

存储器的分类

- 按在计算机中的作用
  - 主存储器（简称主存或内存）、辅助存储器（简称辅存或外存）、高速缓冲存储器（Cache）
- 按存储介质
  - 半导体存储器、磁表面存储器、磁心存储器、光存器件
- 按存取方式
  - 随机存取存储器（RAM）、只读存储器（ROM）、顺序存取存储器（SAM）、直接存取存储器（DAM）

存储器的层次化结构

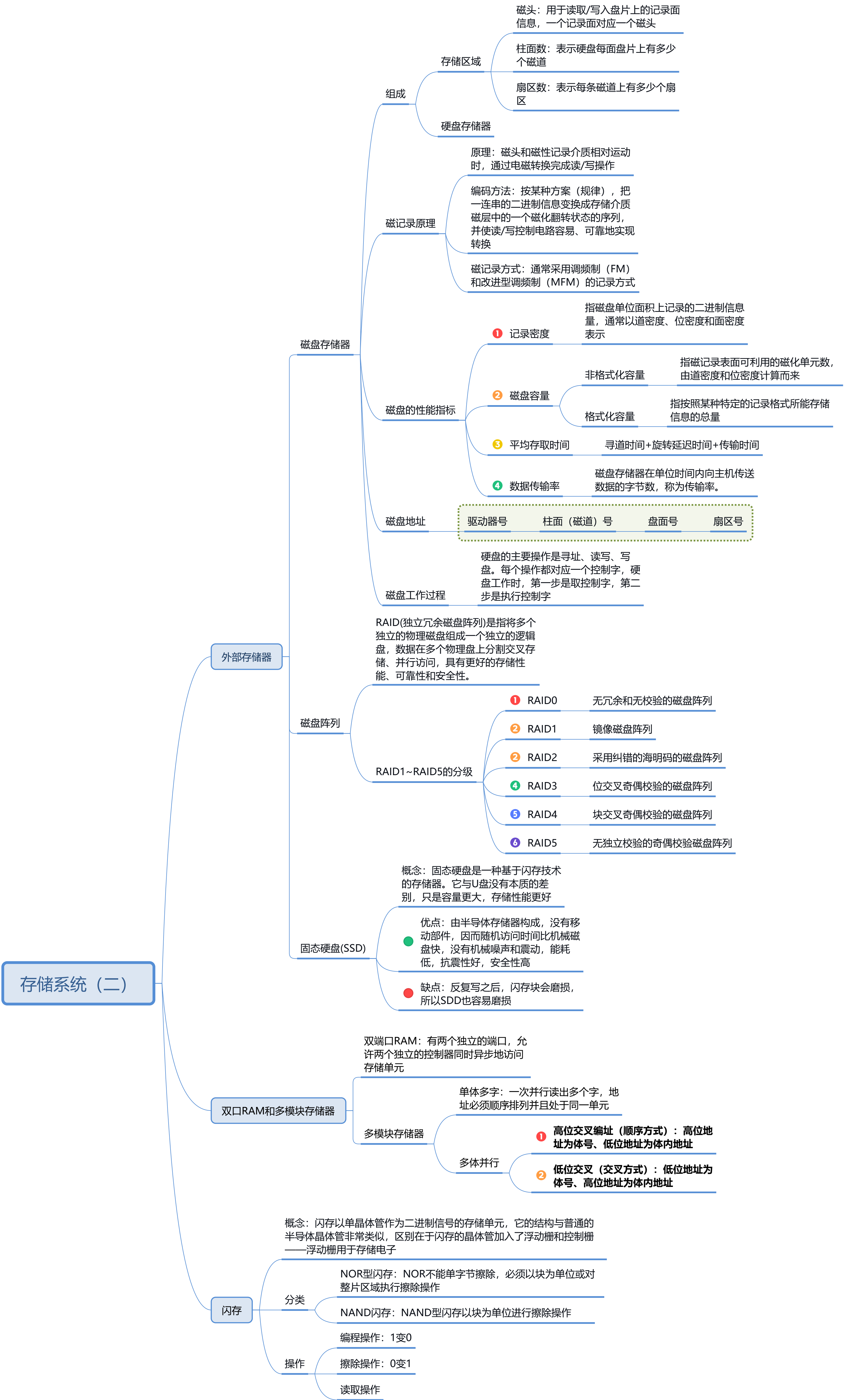


半导体存储器

- 组成
  - 数据线、地址线、存储矩阵、译码驱动、片选线、读/写电路、读/写控制线
- 分类
  - RAM
    - SRAM
      - 原理：利用双稳态触发器来记忆信息，一般用来做高速缓冲存储器
    - DRAM
      - 原理：利用存储元电路中栅极电容上的电荷来存储信息，需要定期刷新，一般用来做大容量主存系统
      - 刷新
        - 集中：用一段固定的时间依次对存储器的所有行逐一刷新
        - 分散：把对每一行刷新的时间分散到各个工作周期中去
        - 异步：把每行刷新分散到一整个刷新周期中去
  - ROM
    - MROM
      - 在芯片制造商生产过程中直接写入，以后任何人都无法改变其内容
    - PROM
      - 允许用户用专门的设备写入程序，写入后的内容无法改变
    - EPROM
      - 允许用户写入程序，用户可以对其内容进行多次改写；需要修改时，要将其全部擦除（不可局部擦除）
    - EEPROM
      - 和EPROM运行原理一样，但是既可以局部擦除，又可以全部擦除
    - Flash（闪速存储器）
      - 在不加电时仍可长期保存信息且能进行快速的擦除重写

主存与CPU的连接

- 扩展
  - 1 位扩展：将芯片地址、片选和读写控制端相应并联，数据端分别引出。增加存储字长
  - 2 字扩展：将芯片的地址、数据、读写控制线相应并联、片选译码选择相应的片/片组。增加存储单元个数
  - 3 字扩展位：既增加了存储单元个数，又增加存储字长





## 存储系统（三）

### 高速缓冲存储器Cache

引入目的：解决CPU与主存速度不匹配的矛盾

#### 映射方式

- ① 直接映射：主存数据块只能装入到Cache中唯一的位置
- ① 全相联映射：可以把主存数据块装入Cache中的任何位置
- ③ 组相联映射：将Cache分为若干组、组间直接映射、组内全相联映射

#### 替换算法

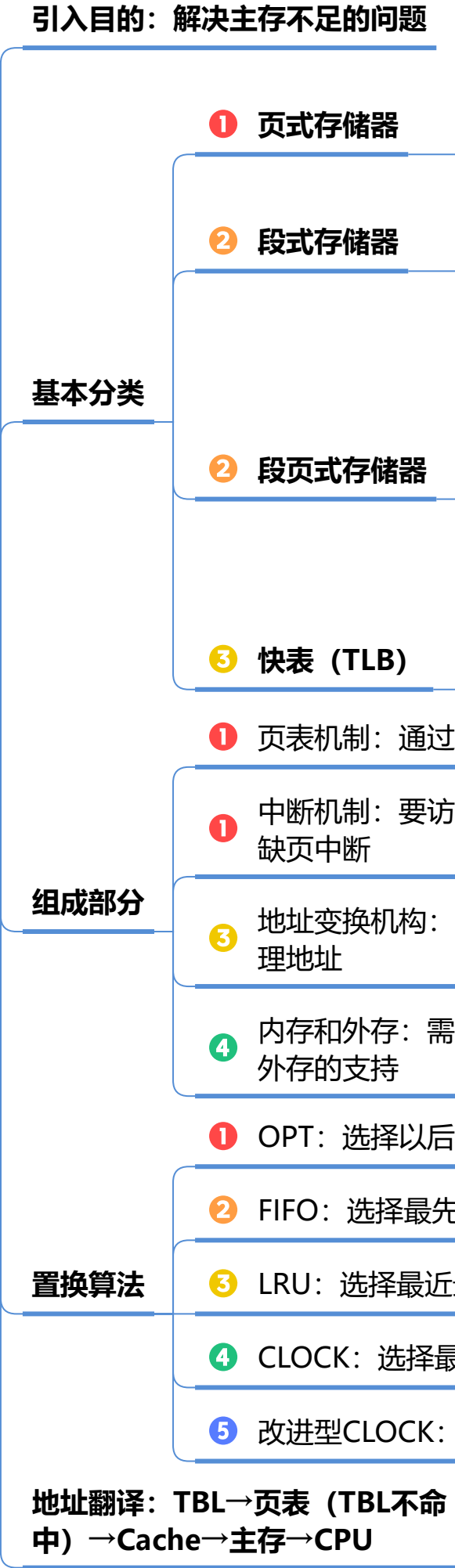
- ① 先进先出（FIFO）算法：选择最早调入的块进行替换
- ② 近期最少使用（LRU）算法：选择近期内长久未访问的块进行替换
- ③ 最不经常使用：将一段时间内被访问次数最少的存储行换出
- ③ 随机算法：随机确定被替换Cache块

#### 写策略

- ① 直写法（写直达法）：写操作时只把数据同时写入主存和Cache
- ② 写回法：写操作时只把数据写入Cache，而不写回主存，只有当Cache数据被换出时才写回主存
- 写分配：写不命中时将所写的块读入Cache
- 非写分配：写不命中时只写入主存，不调入Cache

存储系统（四）

虚拟存储器



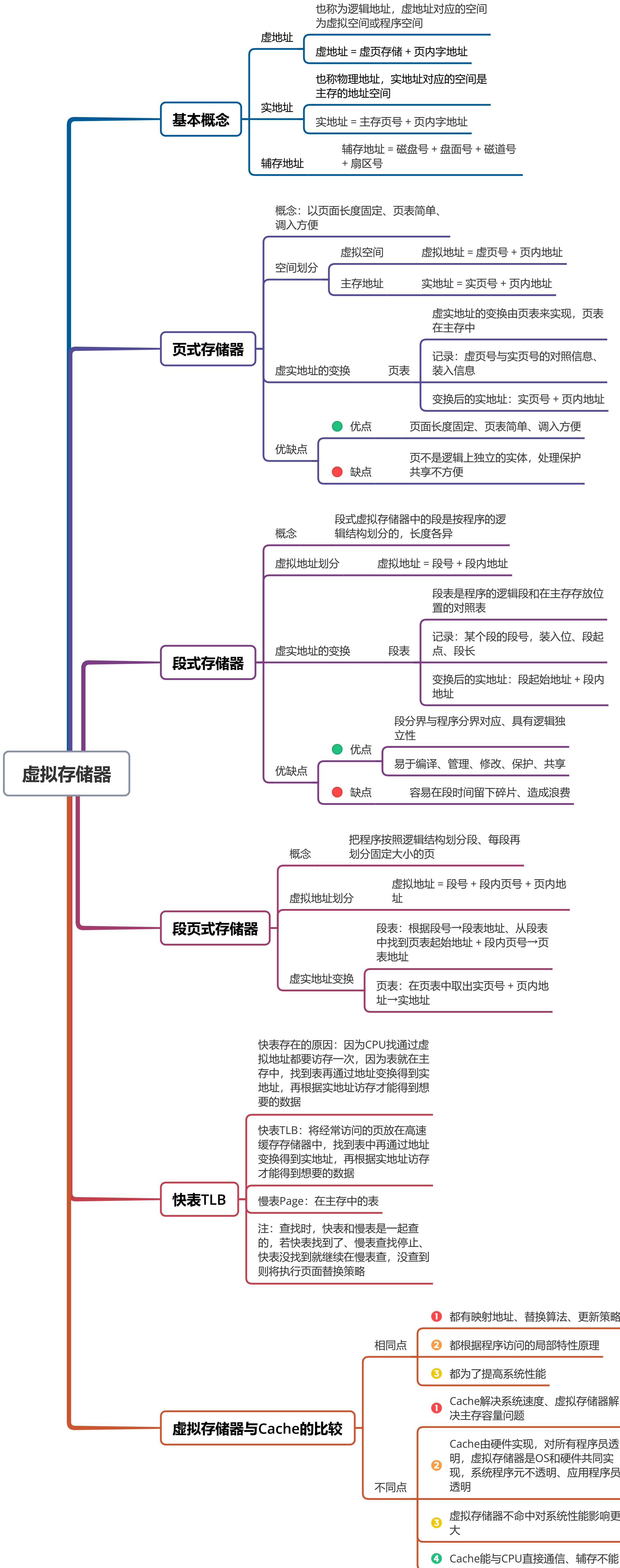
以页为基本单位。主存空间和虚拟空间都划为若干大小相等的页。

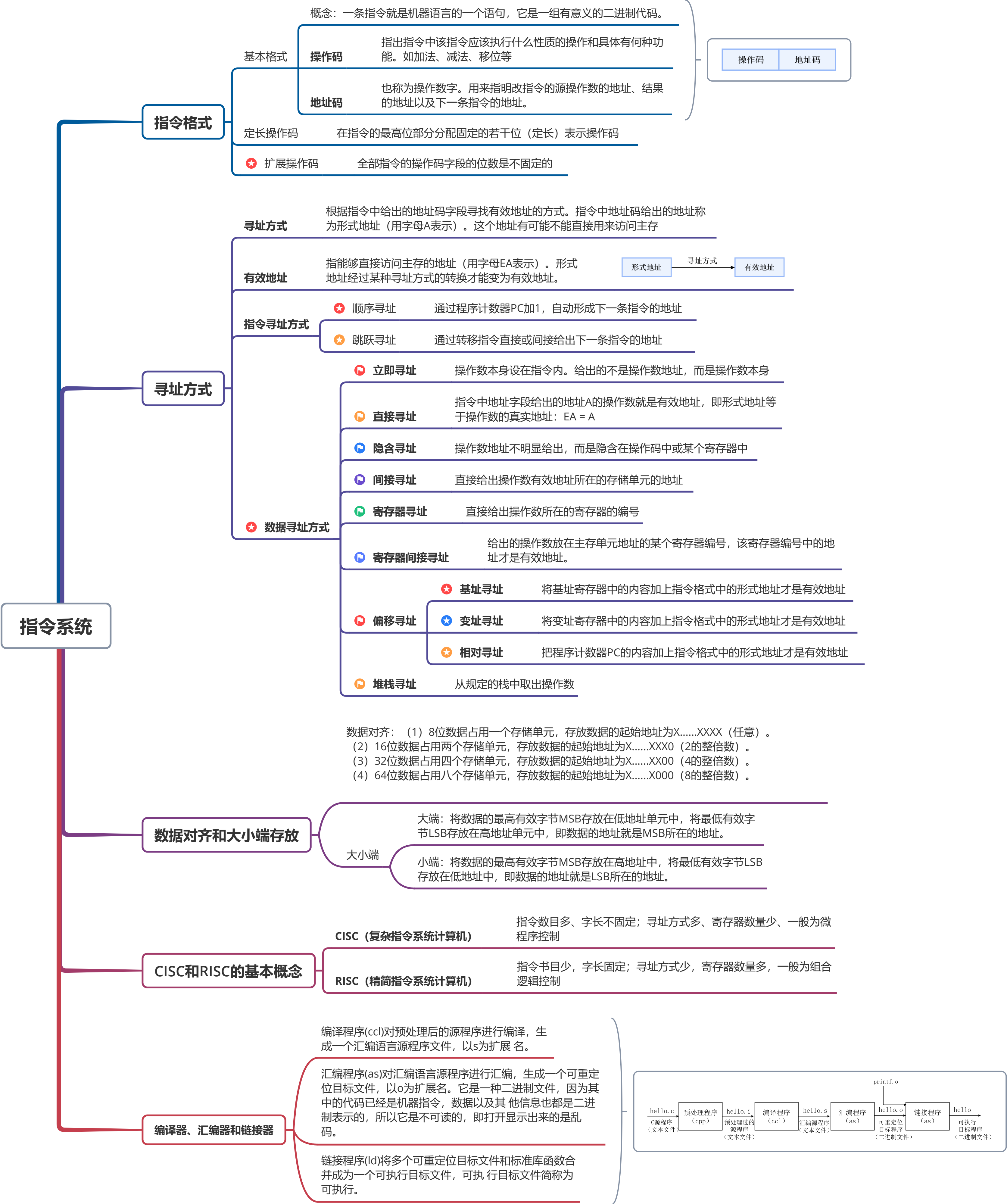
以段为基本单位。讲主存按段分配，各段的长度因程序而异。

讲程序按照其逻辑结构划分段，每段再划分为若干页；主存空间也划分为若干同样大小的页。段式和页式存储的结合。但是要经过两级查表才能完成地址转换，比较费时

讲当前最常用的信息放在一个小容器的高速存储器中，构成快表。快表扮演的角色是作为页表的Cache，对快表的查找和管理全部用硬件来实现。







中央处理器（一）

CPU的基本结构和功能

功能	指令控制	完成指令，分析指令，和执行指令的操作，即程序的顺序控制
	操作控制	取到指令后，应该产生完成每条指令所需要的控制命令
	时间控制	控制命令产生后，需要对各种控制命令加以时间上的控制
	数据加工	在执行过程中，对数据进行算术运算和逻辑运算
	中断处理	对计算机运行过程中出现的异常情况和特殊请求进行处理
基本结构	运算器	是对信息进行处理和运算的部件。它的功能是完成算数运算和逻辑运算，并将运算的中间结果暂存在运算器中
	控制器	是整个计算机的指挥中心。它的功能是控制、指挥程序和数据的输入、运行以及处理运算结果

指令执行过程

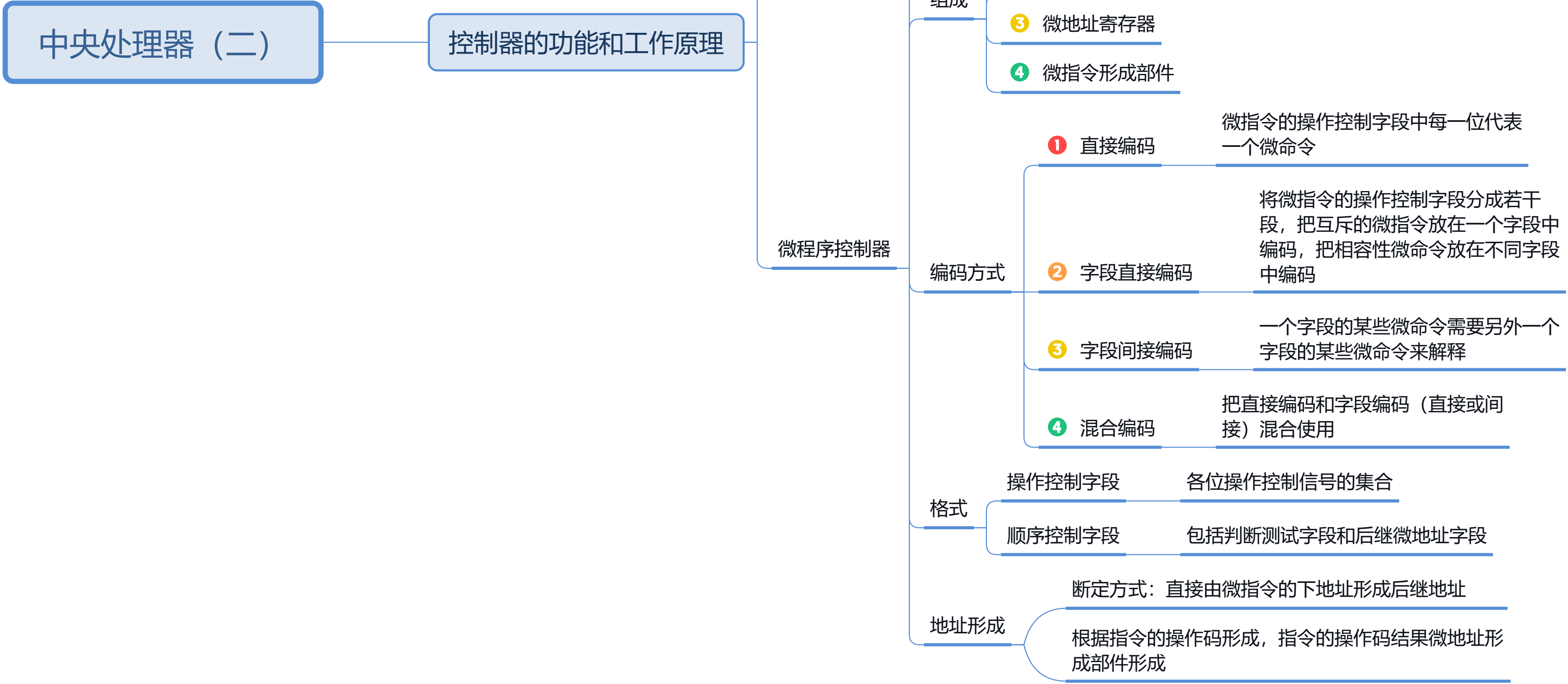
指令周期	取指周期	根据PC中的内容从指定地址读出指令代码并放在IR中
	间址周期	取操作数的有效地址（并不是所有指令的执行过程中都会有间址周期）
	执行周期	根据指令字的操作码和操作数执行相应的操作
	中断周期	处理中断请求
执行方案	单指令周期	对所有的指令都选用相同的执行时间来完成
	多指令周期	对不同类的指令选用不同的执行步骤来完成
	指令流水线	指令之间可以并执行的方案

指令并行执行

数据通路的功能和基本结构

CPU内部总线	将所有寄存器的输入端和输出端都连接到一条或多条公共的通路，将多个部件共享，可以存在一条或多条
专用数据通路总线	根据指令执行过程的数据和地址的流动安排连接的线路





中央处理器（三）

异常和中断机制

异常是由CPU内部产生的意外事件，分为硬故障中断和程序性异常

故障中断是由硬连线出现异常引起的，如存储器校验错误、总线错误等

程序性异常中断也称软件中断，指在CPU内部执行指令而引起的异常事件

异常的分类

1 故障

在引起故障的指令启动后、执行结束前被检测到的异常事件。例如，指令译码时，出现“非法操作码”

2 自陷

自陷也称陷阱或者陷入，它是预先安排的一种“异常”事件。通常是事先在程序中用一条特殊指令或设定特殊控制标志来人为设定陷阱，当执行道被设置的陷阱的指令时，CPU在执行完自陷指令后，自动根据不同的陷阱类型进行相应的处理，然后返回道自陷指令的下一条指令执行。

注意：当自陷指令是转移指令时，并不是返回到下一条执行指令，而是返回到转移目标指令执行

故障异常和自陷异常属于程序性异常（软件中断）

3 终止

在执行指令的过程中发生了使计算机无法继续执行的硬件故障，如控制器出错等，程序将无法执行，只能终止，此时调出中断服务程序来重启系统。

终止异常和外中断属于硬件中断

中断的分类

1 可屏蔽中断

指通过可屏蔽中断请求线INTR向CPU发出的中断请求。CPU 可以通过在中断控制器中设置相应的屏蔽字来屏蔽它或不屏蔽它，被屏蔽的中断请求将不被送到CPU。

2 不可屏蔽中断

指通过专门的不可屏蔽中断请求线NMI向CPU发出的中断请求，通常是非常紧急的硬件故障，如电源掉电等。这类中断请求信号不可被屏蔽，以让CPU快速处理这类紧急事件。

中断和异常的两个重要不同点

缺页或溢出等异常事件是由特定指令在执行过程中产生的，而中断不和任何指令关联，也不阻止任何指令的完成

异常的检测由CPU自身完成，不必通过外部的信号通知CPU。对于中断，CPU必须通过中断请求获取中断源的信息，才能知道哪种设备发生了何种中断

中央处理器（四）

多处理器基本概念

单指令流单数据流(SISD)结构	SISD是传统的串行计算机结构，这种计算机通常仅包含一个处理器和一个存储器，处理器在一段时间内仅执行一条指令，按指令流规定的顺序串行执行指令流中的若干条指令。
单指令流多数据流 ((SIMD)结构	SIMD是指一个指令流同时对多个数据流进行处理，一般称为数据级并行技术。这种结构的计算机通常由一个指令控制部件、多个处理单元组成。
多指令流单数据流(MISD)结构	MISD是指同时执行多条指令，处理同一个数据，实际上不存在这样的计算机。
多指令流多数据流(MIMD)结构	MMD 是指同时执行多条指令分别处理多个不同的数据，MIMD 分为多计算机系统和多处理器系统

概念：把一个重复的过程分解为若干子过程，每个子过程与其他子过程并行执行

分类	按使用级别分	部件功能级流水：将复杂的算术和逻辑运算组成流水线工作方式 指令级流水：则将对指令的整个执行过程分成多个子过程
	按完成功能分	处理机级流水：由两个或多个处理机通过存储器串行连接起来，每个处理机对同一数据流的不同部分分别进行处理 单功能流水：只能实现一种固定的功能。 多功能流水：可有多种连接方式来实现多种功能。
	按连接方式分	动态流水：则允许在同一时间内将不同的功能段连接成不同的 功能子集(前提是功能部件的使用不发生冲突)，以完成不同的功能 静态流水：在同一时间内只能以一种方式工作。它可以是单功能的， 也可以是多功能的
	按有无反馈信号分	线性级流水：从输入到输出，每个功能段只允许经过一次，不存在反馈回路，一般的流水线均属于这一类 非线性级流水：除有串行连接通路外，还有反馈回路，在流水过程中，某些功能段要反复多次使用

影响因素	结构相关	又称资源冲突，由多条指令同一时刻争用同一资源而形成的冲突
	数据相关	又称资源冒险，必须等到前一条指令执行完成后才能执行后一条指令的情况
	控制相关	又称控制冒险，遇到转移指令和改变PC情况而形成的断流

性能指标	吞吐率	指定的单位时间指令流水的完成任务量
	加速比	完成同样一批任务，顺序执行时间与流水线执行时间之比
	效率	流水线中各功能的利用率

指令流水线

结构冒险、数据冒险和控制冒险的处理	概念：在指令流水线中，会遇到使得流水线无法正确执行后续指令而引起的流水线阻塞或停顿，这种现象称为流水线冒险		
	结构冒险（资源冒险）	由于多条指令在同一时刻竞争同一资源而形成的冲突，称为资源冲突，也是由于硬件资源竞争造成的冲突	
		解决方法	前一指令访存时，使后一条相关指令（以及其后续指令）暂停一个时钟周期。
	单独设置数据存储器和指令存储器，使取数和取指令操作各自在不同的存储器中进行。事实上，现代计算机都引入了Cache 机制，而LI Cache 通常采用数据Cache和指令Cache分离的方式，因而也就避免了资源冲突的发生。		
	数据冒险（数据冲突）	在一个程序中，下一条指令会用到当前指令计算出的结果，此时这两条指令发生数据冲突。当多条指令重叠处理时就会发生冲突，数据冒险可分为三类	
		<div>写后读相关</div> <div>读后写相关</div> <div>写后写相关</div>	
控制冒险（控制冲突）	指令通常是顺序执行的，但是在遇到改变指令执行顺序的情况，例如执行转移、调用或返回等指令时，会改变PC值，会造成断流，从而引起控制冒险。		
	解决的方法	对转移指令进行分支预测	
		预取转移成功和不成功的两个控制流方向上的目标指令	
		<div>加快提前形成条件码</div> <div>提高转移方向的猜准率</div>	
多发技术	超标量技术：在每个时钟周期内可以同时并发多条独立指令		
	超流水线技术：在一个时钟周期内再分段，在一个时钟周期内一个功能部件使用多次		
	超长指令字技术 由编译程序挖掘出指令间潜在的并行性，将多条能并行操作的指令组合成一条具有多个操作码字段的超长指令		



总线

概述

概念：总线是连接多个部件的信息传输线，具有各部共享的传输介质

分类

按照数据传送方式

并行传输总线

串行传输总线

功能层次划分

● 片内总线

用来连接芯片内部的各个部件

● 系统总线

用于连接计算机系统内部各功能部件

● 通信总线

用于连接计算机系统之间或计算机系统与其他系统

性能指标

① 总线周期

一次总线操作所需要时间

② 总线频率

总线上各种操作的频率，是总线周期的倒数

③ 总线带宽

通常指总线数据的根数，也就是总线上能同时传输数据的位数

④ 总线宽度

总线的数据传输速率，及单位时间内总线上传输数据的位数。总线带宽 = 总线宽度×总线频率

仲裁

集中式

链式查询

根据线的链接顺序依次查询每个部件有无请求

计数器定时查询

对每个设备进行编号，由计数器依次查询

独立请求方式

在总线控制器中排队，按照一定规则响应某个请求

分布式

不要中央仲裁器，每个潜在的主模块都有自己的仲裁器，多个总线仲裁器竞争使用总线

总线周期



申请分配阶段

总线的主模块向总线提出使用请求



寻址阶段

通过总线发出本次要访问的从模块的地址及有关的命令



传输阶段

主模块和从模块进行数据交换，可单向或双向进行数据传送



结束阶段

主模块的有关信息从系统总线上擦除，让出总线使用权

定时

① 同步定时方式

系统采用一个统一的时钟信号来协调发送和接收双方的传递的定时关系

概念

没有统一的时钟，也没有固定的时间间隔，完全依靠相互制约的“握手”信号来实现定时通信

② 异步定时方式

分类

① 不互锁

主模块的请求信号和从模块的问答信号没有相互制约

② 半互锁

主模块的请求信号和从模块的回答信号有简单的制约关系

③ 全互锁

主模块的请求信号和从模块的问答信号有完全的制约关系

