

计算机组成原理

---NonoTion

第 1 章 计算机系统概论

1.1 计算机系统简介

计算机系统由软件和硬件组成

计算机系统的层次结构

高级语言机器->汇编语言机器->操作系统机器->机器语言机器->微指令系统

虚拟机器M4 虚拟机器M3 虚拟机器M2 传统机器M1 微程序机器M0

1.2 计算机的基本组成

冯·诺伊曼计算机的特点

- 计算机由运算器、存储器、控制器、输入系统和输出系统五大部件组成
- 指令和数据以同等地位存放于存储器内，并可按地址寻访
- 指令由操作码和地址码组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置
- 指令在存储器内按顺序存放。通常指令是顺序执行的，在特定条件下，可根据运算结果或根据设定的条件改变执行顺序
- 机器以运算器为中心，输入输出设备与存储器之间的数据传送通过运算器完成

现代计算机以存储器为中心

各部件的功能如下

- 运算器用来完成算术运算和逻辑运算，并将运算的中间结果暂存在运算器内
- 存储器用来存放数据和程序
- 控制器用来控制、指挥程序和数据的输入、运行以及处理运算结果
- 输入设备用来将人们熟悉的信息形式转换为机器能够识别的信息形式
- 输出设备可将机器运算结果转换为人们熟悉的信息形式

现代计算机三大部分

- CPU=运算器+控制器
- 输入和输出设备简称为I/O设备
- 主存储器

1.3 计算机硬件的主要技术指标

- 机器字长——CPU一次能够处理数据的位数
- 存储容量——分为主存容量和辅存容量
- 运算速度

第2章 计算机的发展及应用

第3章 系统总线

3.1 总线的基本概念

- 总线连接——将各部件连接到一组公共信息传输线上
- 总线——连接多个部件的信息传输线，是各部件共享的传输介质

同一时间，只有一个部件可以向总线发送信息，多个部件可以从总线上接受相同的信息

3.2 总线的分类（按不同连接部件分）

3.2.1 片内总线

芯片内部的总线 连接CPU芯片内部的部件

3.2.2 系统总线

CPU、主存、I/O设备(通过I/O接口)各大部件之间的信息传输线

按系统总线传输信息的不同，可分为三类

1. 数据总线——同于传输各功能部件之间的数据信息，双向传输总线，其位数与机器字长、存储字长有关。数据总线的位数称为**数据总线宽度**
2. 地址总线——用来指出数据总线上的源数据或目的数据在主存单元的地址或I/O设备地址,CPU输出，单向传输。
3. 控制总线——用来**发出各种控制信号**的传输线，对任一控制线而言，它的传输是单向的。对于控制总线总体来说，又可以认为是双向的。还起到**监视各部件状态**的作用

3.2.3 通信总线

用于计算机系统之间或计算机系统与其它系统之间的通信

串行通信和并行通信

3.3 总线特性及性能指标

3.3.1 总线特征

- 机械特性——总线在机械连接方式上的一些性能
- 电气特性——总线的每一根传输线上信号的传递方向和有效的电平范围
- 功能特性——总线中每根传输线的功能
- 时间特性——总线的任一根线在什么时间内有效。一般用信号时序图来描述

3.3.2 总线性能指标

- 总线宽度——通常是指数据总线的根数，用bit(位)表示
- 总线带宽——总线的数据传输速率
- 时钟同步/异步——总线上的数据与时钟同步工作的总线叫做同步总线，与时钟不同步工作的总线叫做异步总线
- 总线复用——地址总线、数据总线和控制总线三种总线数的总和
- 总线控制方式——包括突发工作、自动配置、仲裁方式、逻辑方式、计数方式等
- 其他——如负载电压、电源电压等

3.3.3 总线标准

系统与各模块、模块与模块之间的一个互连的标准界面

如：ISA、EISA、VESA、PCI...

3.4 总线结构

3.4.1 单总线结构

将CPU、主存、I/O设备(通过I/O接口)挂在一组总线上，允许I/O设备之间、I/O设备与CPU之间或I/O设备与主存之间直接交换信息

优点：简单，便于扩充

缺点：影响系统工作效率的提高

3.4.2 多总线结构

双总线结构

将速度较低的I/O设备从单总线上分离出来、形成主存总线与I/O总线分开的结构

三总线结构

将速率不同的I/O设备进行分类，然后将它们连接在不同的通道上，那么计算机系统的工作效率将会更高

四总线结构

3.5 总线控制

3.5.1 总线判优控制

总线上连接的各种设备，按其总线有无控制功能可分为主设备(模块)和从设备(模块)、

多个主设备同时要使用总线时，要按一定的优先级顺序决定哪个设备能使用总线

总线判优控制可分为两种:

集中式

- 链式查询——控制总线中有三根线用于总线控制(BS总线忙, BR总线请求, BG总线同意)。
优点：几根线就能按一定优先次序实现总线控制 缺点：对电路故障很敏感
- 计数器定时查询——多了一组设备地址线，少了一根总线同意线
- 独立请求方式——每一台设备均有一对总线请求线BR和总线同意线BG

3.5.2 总线通信控制

通常完成一次总线操作的时间称作总线周期，可分为下面四个阶段

- 申请分配阶段——需要使用总线的主模块提出申请，经总线仲裁机构决定下一传输周期的总线使用授权于某一申请者
- 寻址阶段——主模块通过总线发出本次要访问的从模块的地址及有关命令
- 传数阶段——主模块和从模块进行数据交换
- 结束阶段——主模块的有关信息从系统总线上撤出，让出总线使用权

四种方式

1.同步通信

通信双方由统一时标控制数据传送称为同步通信

2.异步传输

(1)不互锁方式

主模块发出请求信号后，不必等到接到从模块的回答信号，而是经过一段时间，确定从模块已收到请求信号后，撤销其请求信号；从模块接到请求信号后，在条件允许时发出回答信号且经过一段时间确定主模块已收到回答信号后，自动撤销回答信号

(2)半互锁方式

主模块发出请求信号后，必须等到接到从模块的回答信号后，再撤销其请求信号

(3)全互锁方式

主模块发出请求信号后，必须等到接到从模块的回答信号后，再撤销其请求信号。从模块接到请求信号后，必须待获知主模块请求信号撤销后，再撤销其回答信号

异步串行通信的数据传输率用波特率来衡量。

波特率——单位时间内传送二进制位数

3.半同步通信

既保留了同步通信的基本特点，同时像异步通信意义，允许不同速度的模块进行工作，为此增加了一条等待响应线（\WAIT）

4.分离式通信

将一个传输周期(或总线周期)分解为两个子周期

第4章 存储器

4.1 概述

4.1.1 存储器分类

按存储介质分类

- 半导体存储器——易失性存储器
- 磁表面存储器
- 磁芯存储器
- 光盘存储器

按存取方式分类

- 随机存储器 RAM
- 只读存储器 ROM
- 串行访问存储器 SAM

按再计算机中的作用分类

- 主存储器
- 辅助存储器
- 缓冲存储器
- 闪速存储器

4.1.2 存储器的层次结构

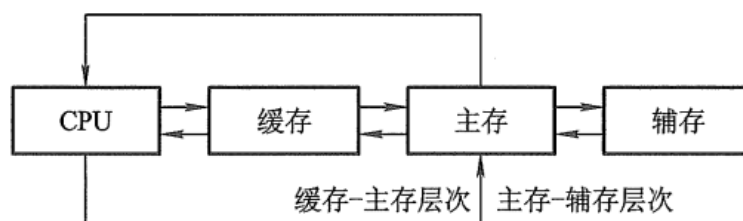


图 4.3 缓存-主存层次和主存-辅存层次

CPU和缓存、主存都能直接交换信息

缓存-主存层次 解决CPU和主存速度不匹配的问题

主存-辅存层次 解决存储系统的容量问题

4.2 主存储器

4.2.1 概述

主存中存储单元地址的分配

主存各存储单元的空间位置是由单元地址号来表示的，而地址总线是用来指出存储单元地址号的，根据该地址可以读出或写入一个存储字

通常计算机系统可以按字寻址，也可以按字节寻址

主存的技术指标

- 存储容量——主存能够存放二进制代码的总位数

$$\text{存储容量} = \text{存储单元个数} * \text{存储字长}$$

存储字长——存储单元中的二进制位数 一般取8的倍数

- 存储速度——由存取时间和存取周期来表示
 - 存取时间——启动一次存储器操作到完成该操作所需的全部时间
 - 存取周期——存储器连续进行两次独立的存储器操作所需的最小时间间隔
- 存储器带宽——单位时间内存储器存取的数据量

提高存储器带宽可采取一下措施

- 缩短存取周期
- 增加存储字长
- 怎加存储体

4.2.2 半导体存储芯片简介

1. 半导体存储芯片的基本结构

- 译码驱动——把地址线送来的地址信号翻译位对应的存储单元的选择信号
- 读/写电路——包括放大器和写入电路，用来完成读写操作
- 存储芯片——通过地址总线、数据总线、控制总线与外部连接

地址线是单向输入的，其位数与芯片容量有关

数据线是双向的，其位数与芯片可以写入或读出的数据位数有关

地址线和数据线共同反映存储芯片的容量

控制线

- 读/写控制线 决定芯片进行读/写操作

- 片选线 用来选择存储芯片

2. 半导体存储芯片的译码驱动方式

线选法

只用一根字选择线，直接选中一个存储单元的各位

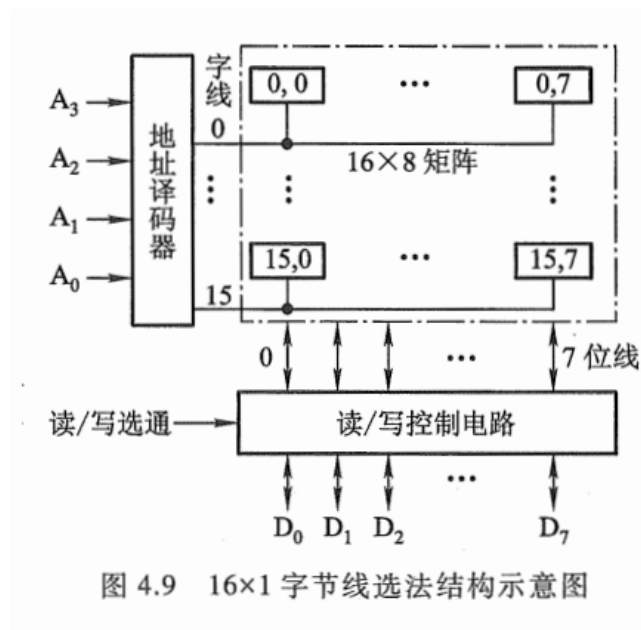


图 4.9 16x1 字节线选法结构示意图

优点：结构简单

缺点：只适用于容量不大的存储芯片

重合法

被选单元是由 X、Y 两个方向的地址线决定的

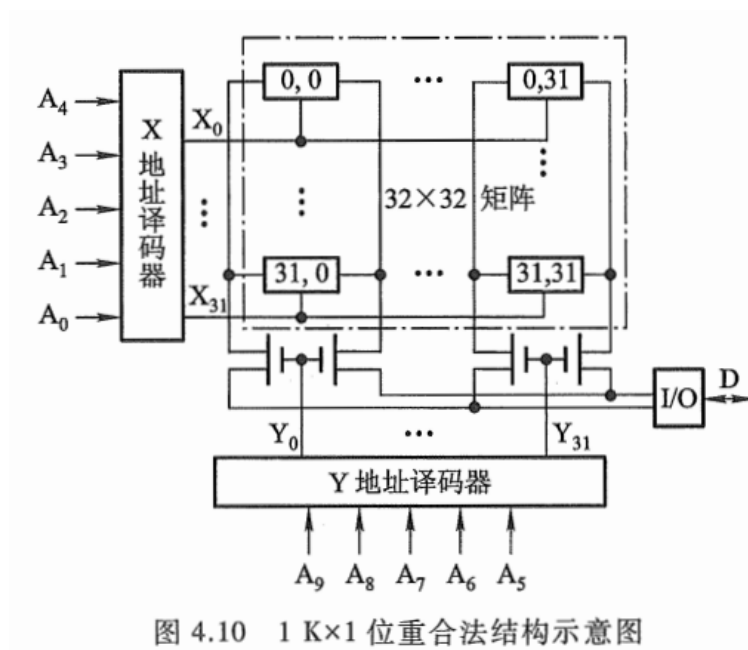


图 4.10 1Kx1 位重合法结构示意图

4.2.2 随机存取存储器

随机存取存储器按其存储信息的原理不同，可分为静态RAM和动态RAM两大类

静态RAM SRAM

触发器工作原理存储信息，因此信息被读出后，它仍保持其原状态，不需要再生。但电源掉电时，原存信息丢失，故它属易失性半导体存储器

Cache

动态RAM DRAM

有三管式和单管式两种，靠电容存储电荷的原理来寄存信息。必须进行再生或刷新

读出是破坏性读出，必须再生

比静态RAM相比，集成度更高，能耗更低

DRAM的刷新

将原存信息读出，再由刷新放大器形成原信息并重新写入的再生过程

- 集中刷新——在一个刷新周期内，对全部存储单元集中一段时间逐行进行刷新。会产生死时间或死区
- 分散刷新——对每行存储单元的刷新分散到每个存取周期内完成 存取周期变长，整个系统速度变慢

异步刷新——前两种方式的结合，缩短死时间，又充分利用最大刷新间隔2ms

动态RAM与静态RAM的比较

- 在相同大小的芯片种，动态RAM集成度远高于静态RAM
- 动态RAM的行、列地址按先后顺序输送，减少了芯片引脚和封装尺寸
- 动态RAM功耗比静态RAM小
- 动态RAM价格比静态RAM低
- 速度比SRAM低
- 动态RAM需要再生，故需配置再生电路，也需要消耗一部分功率

4.2.4 只读存储器

- 掩膜ROM 用户无法改变原始状态
- PROM 一次性编程的只读存储器
- EPROM 可擦除可编程只读存储器

4.2.5 存储器和CPU的连接

1. 存储容量的扩展

- 位扩展指的是怎加存储字长

- 字扩展指的是怎加存储器字的数量
- 字、位扩展

2. 存储器与CPU的连接

(1) 地址线的连接

CPU地址线往往比存储芯片的地址线多，通常将CPU地址线的低位与存储芯片的地址线相连。CPU地址线的高位或在存储芯片时用，或做其他用途，如片选信号等

(2) 数据线的连接

CPU的数据线与存储芯片数据线也不一定相等，所以必须对存储芯片进行位扩展

(3) 读/写控制线的连接

(4) 片选线的连接

(5) 合理选择存储芯片

通常选用ROM存放系统程序，标准子程序和各类常数等。RAM则是为用户编程而设置的

4.2.6 存储器的校验

1. 汉明码的组成

设欲检测的二进制代码位 n 位，为了使其具有纠错能力，需添加 k 位检测位，组成 $n+k$ 位的代码。为了能准确对错误定位以及指出代码没错，新增添的检测位数 k 满足

$$2^k \geq n + k + 1$$

将 k 位检测位分别安排在代码的 2^{k-1} 位上，记作 $C_i (i = 1, 2, 4, 8 \dots)$

C_1 检测的 g_1 小组包含1, 3, 5, 7, 9... 位

C_2 检测的 g_2 小组包含2, 3, 6, 7, 10... 位

以此类推

...

具体规则参照P101页

2. 汉明码的纠错过程

对传送后的汉明码形成新的检测位 $P_i (i = 1, 2, 4, 8 \dots)$

根据 P_i 的状态就可以直接指出错误的位置

$P_i = g_i$ 小组的所有数异或

将检测位 P_i 从大到小排列组成的二进制数的大小就是出错的位数（偶校验）

4.2.7 提高访存速度的措施

除了寻找高速原件和采用层次结构外，**调整主存的结构**也可提高访存速度

1.单体多字系统

由于程序和数据在存储体内是连续存放的，因此CPU访存取出的信息也是连续的，如果可以在一个存取周期内，从同一地址连续取出4条指令，然后再逐条将指令送至CPU执行，即每隔1/4存取周期，主存向CPU送一条指令，这样显然增大了存储器的带宽，提高了单体存储器的工作速度

使用这种方法的前提是：指令和数据再主存内必须是连续存放的，一旦遇到转移指令，或操作数不能连续存放，这种方法的效果就不明显

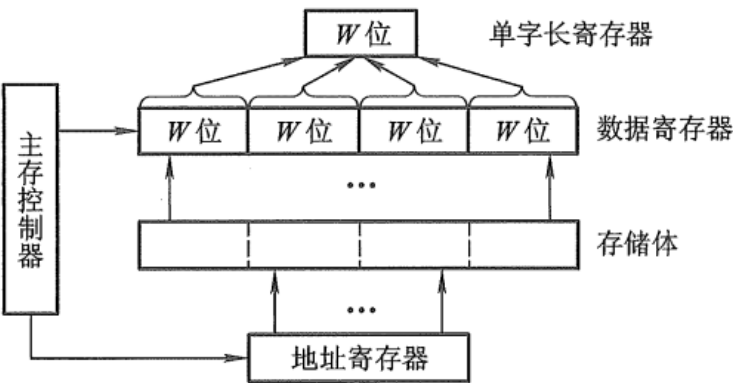


图 4.41 单体四字结构存储器

2.多体并行系统

多体并行系统就是采用多体模块组成的存储器。每个模块具有相同的容量和存取速度，各模块各自都有独立的地址寄存器(MAR)、数据寄存器(MDR)、地址译码、驱动电路和读/写电路，它们能并行工作，不能交叉工作

高位交叉编址的多体存储器

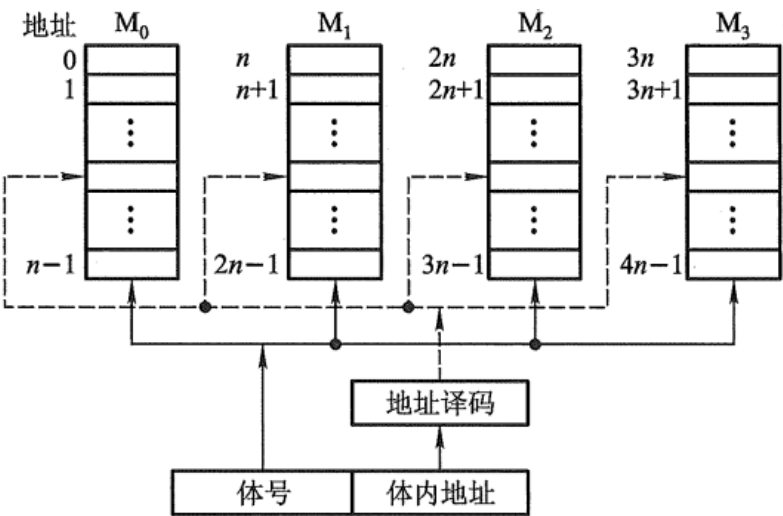


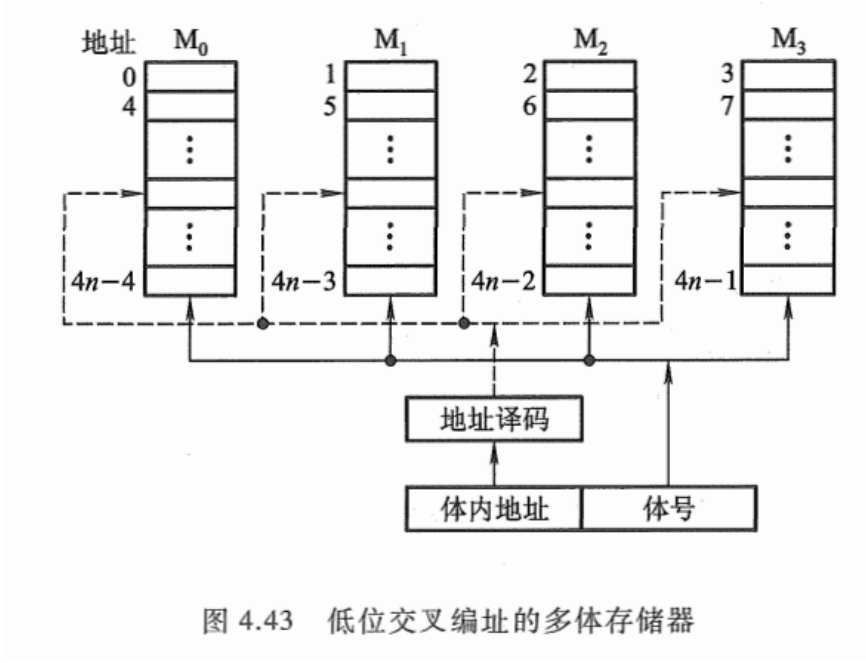
图 4.42 高位交叉编址的多体存储器

高位地址可表示体号，低位地址为体内地址

程序按照体内地址顺序存放

只要合理调动，使不同的请求源同时访问不同的体，便可实现并行工作

低位交叉编址的多体存储器



由于程序连续存放在相邻体种，故又有交叉存储之称。显然低位地址用来表示体号，高位地址为体内地址

这种编址方式又叫做模M编址（M为模块数）

多体模块结构的存储器采用交叉编址后，可以再不改变模块存取周期的情况下，提高存储器的带宽

假设每个个体的存储字长和数据总线的宽度一致，地位交叉的存储器模块数为n，存取周期为T，总线传输周期为r，采取流水线方式存取时，应满足 $T=nr$

- 采用低位交叉编址时， $t=T+(n-1)r$
- 采用高位交叉编址时， $t=nT$

多体模块存储器不仅要与CPU交换信息，也要与辅存，I/O设备等交换信息所以需要安排各部件访问的顺序

这个部件叫做存储器控制部件

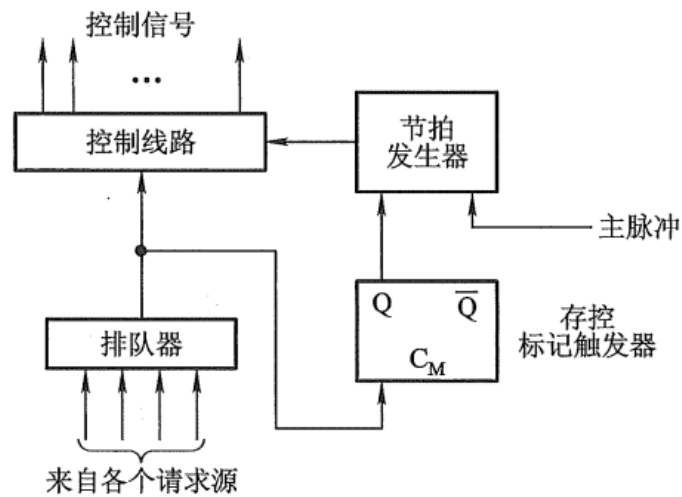


图 4.46 存控基本结构框图

有排队器、存控标记触发器、节拍发生器和控制线路组成

3.高性能存储芯片

SDRAM 同步DRAM

RDRAM

CDRAM 带Cache的DRAM

4.3 高速缓冲存储器

4.3.1 概述

1. 问题的提出

大概了解为什么要设置高速缓冲存储器P109

大体概括为两点

- 多体并行存储系统中，I/O设备访问主存的权限大于CPU，又是CPU甚至要等待几个周期才能访问主存
- 主存的速度提高始终跟不上CPU的发展

Cache的出现使CPU可以不直接访问主存，而与高速Cache交换信息

2. Cache工作原理

主存由 2^n 个可编制的字组成，每一字有唯一的 n 位地址。为了与Cache映射，将主存与缓存部分都分成若干的块，每个块内包含若干个字，并使它们的块大小相同。这就使得主存的地址分成主存块号和块内地址两部分，

缓存也分为两端：块号和块内地址

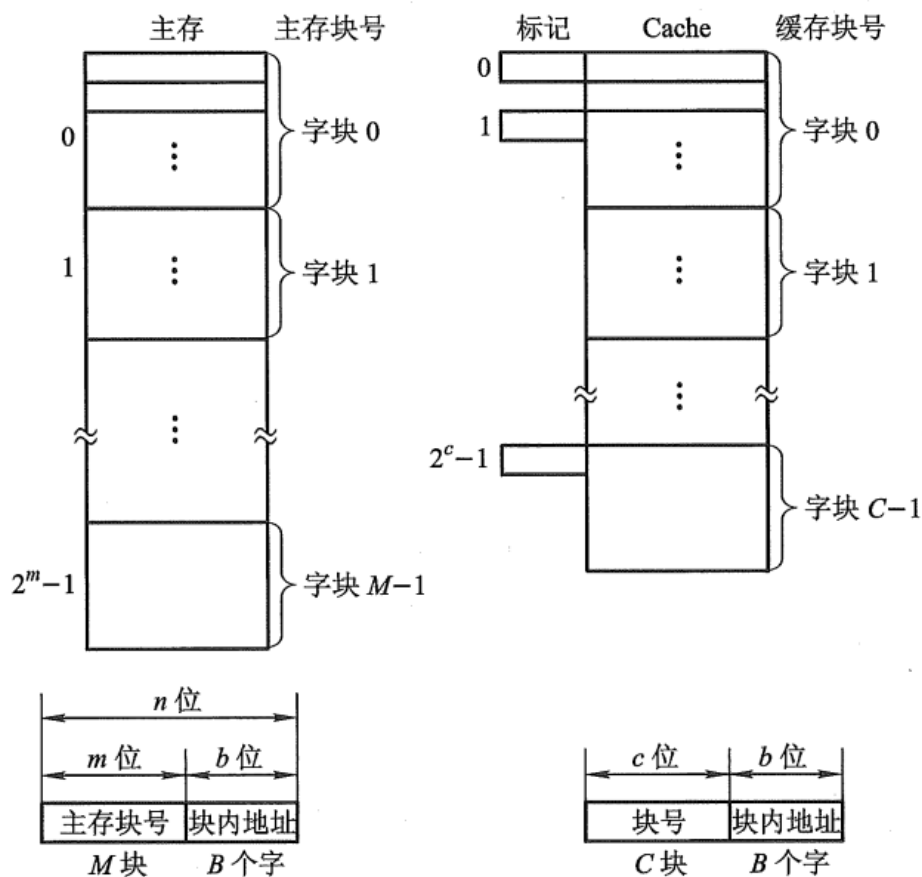


图 4.49 Cache-主存存储空间的基本结构

Cache命中与Cache不命中

- Cache命中——CPU欲读取主存某字时，所需要的字已在缓存中，即可直接访问Cache
- Cache不命中——所需的字不在缓存中，这时需要将该字所在的主存整个子块一次调入Cache中

Cache命中率

N_c 为访问Cache的命中总次数， N_m 是访问主存的总次数，则命中率 h

$$h = \frac{N_c}{N_c + N_m}$$

Cache—主存系统平均访问速度

t_c 为Cache访问时间， t_m 为未命中时的主存访问时间， $1 - h$ 表示为命中率，则平均访问时间 t_a

$$t_a = ht_c + (1 - h)t_m$$

访问效率e

$$e = \frac{t_c}{t_a}$$

3. Cache基本结构

- Cache存储体——Cache存储体以块为单位与主存交换信息，为加速Cache与主存之间的访问速度，主存大多采用多体结构，且Cache访存优先级最高

- 地址映射变换机构——将CPU送来的主存地址转换为Cache地址，主要是主存块号与Cache块号间的转换
- 替换机构——当Cache内容已满，无法接受来自主存块的信息时，就由Cache内的替换机构按一定的替换算法来确定应从Cache内溢出哪个块来返回主存，而把新的主存块调入Cache
- Cache读写操作

读操作：CPU发出主存地址后，首先判断存储字是否在Cache内，若命中，直接访问Cache，若不命中，一方面要访问主存，将该字传送给CPU，也要将该字所在的主存块装入Cache，若Cache已满，则要执行替换算法

写操作：

- 写直达法：写操作时数据既写入Cache又写入主存
- 写回法：写操作时数据只写入Cache，当Cache数据被替换出去时才写回主存

为了识别Cache中的数据是否与主存中一致，要增加一个标志位"清" "浊"

"清"表示未修改Cache，这时不必写回主存

"浊"表示修改Cache，这时要写回主存

4.Cache的改进

(1) 单一缓存和多级缓存

(2) 统一缓存和分立缓存

分立缓存：指令和数据放在两个Cache中

4.3.2 Cache——主存地址映射

1.直接映射

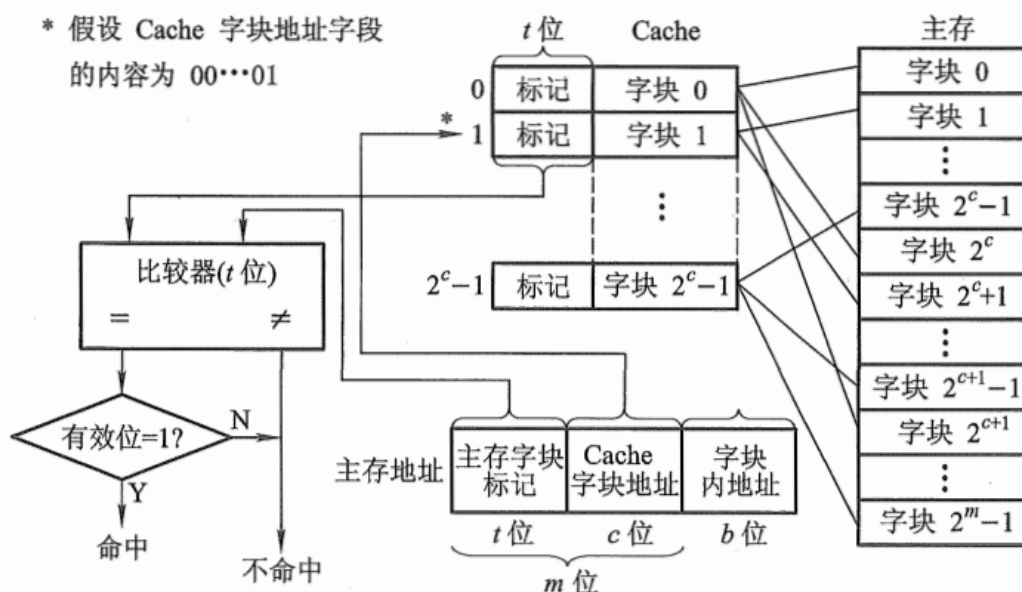


图 4.54 直接映射

每个主存块只与一个缓存块对应，映射关系式为

$$i = j \bmod C \text{ 或 } i = j \bmod 2^C$$

- i ——缓存块号
- j ——主存块号
- C ——缓存块数

优点：实现简单，只需利用主存地址的某些位置就可以直接判断所需子块是否在缓存中

工作过程：

1. 当缓存收到CPU送来的主存地址后，只需根据中间 c 位字段，找到Cache字块
2. 判断字块的“标记”是否与主存地址的高 t 位相符

若符合并且有效位为“1”（有效位用来识别Cache存储块中的数据是否有效），表示该Cache已和主存的某块建立了对应关系（即Cache命中），则可根据 b 位地址从Cache中获取信息。

若不符合，或有效位为“0”，则从主存读入新的字块来代替旧的字块，同时将信息送往CPU，并修改Cache“标记”，如果原来有效位为“0”，则还需要将有效位改为“1”

2.全相联映射

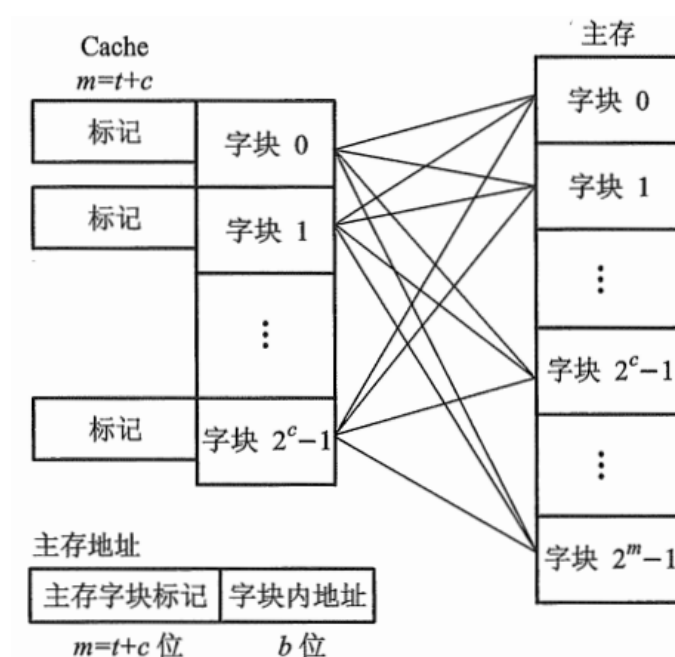


图 4.55 全相联映射

允许主存中的每一字映射到Cache中的任何一块位置上，这种映射方式可以从已被占满的Cache中替换出任一旧字块。

优点：这种方式灵活，命中率也更高，缩小了块冲突率

3.组相联映射

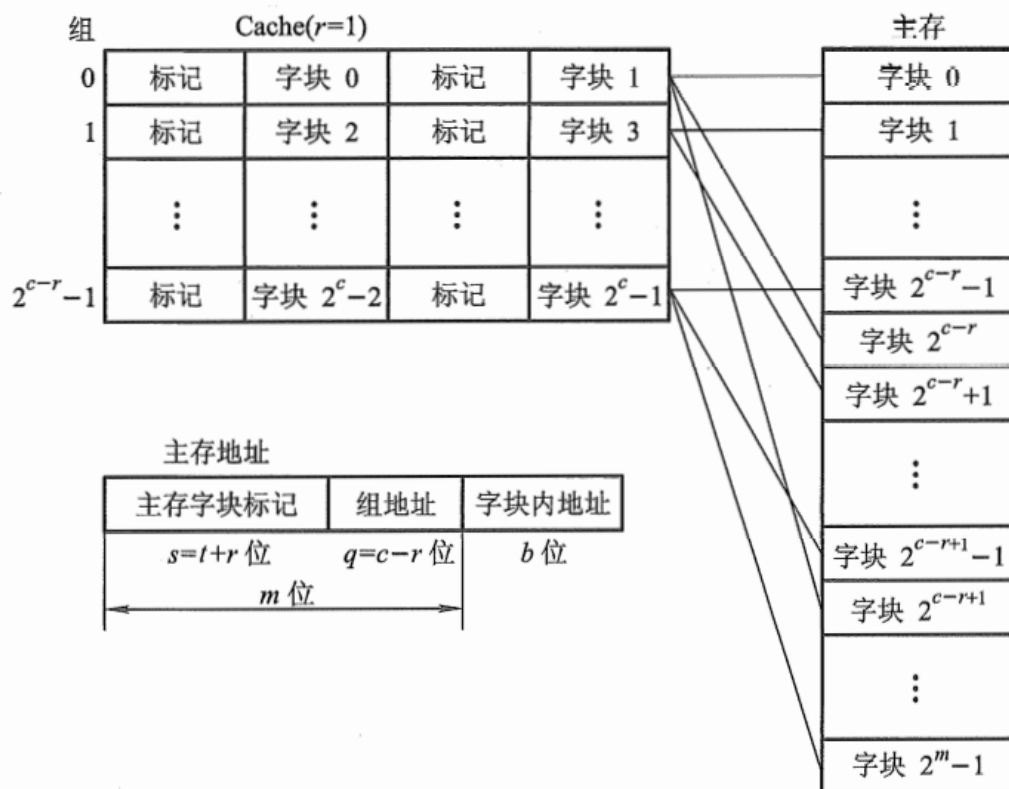


图 4.56 组相联映射