# A

Akeman hanshu

**阿克曼函数（Ackermann's function）** 一种递归而非原始递归的函数（参见**递归函数**和**原始递归函数**）。

阿克曼函数定义如下：

$$\begin{cases}\psi(0,y)=y+1\\ \psi(x+1,0)=\psi(x,1)\\ \psi(x+1,y+1)=\psi(x,\psi(x+1,y))\end{cases}$$

该函数定义中包含双重递归。当 $x>0$ 时，$\psi(x,y)$的值均可通过“较早”值 $\psi(x_1,y_1)$来定义。这里，或者 $x_1<x$ 或者 $x_1=x,y_1<y$。这一点可直观地说明该函数可计算。

$\psi$ 是递归函数的严格证明参见文献[1]，$\psi$ 为非原始递归函数的严格证明参见文献[2]。

**参考文献**

1. Cutland N. Computability, an introduction to recursive function theory. Cambridge, UK: Cambridge University Press, 1980

2. Peter R. Recursive functions. New York: Academic Press, 1967

（陈火旺 贲可荣）

anquan caozuo xitong

**安全操作系统（secure operating system）** 具有强制访问控制机制并可验证的**操作系统**。强制访问控制机制是一种用于保障信息机密性、完整性和资源隔离性等特性的信息安全机制，与自主访问控制不同，强制访问控制是完全由操作系统而不是由信息的所有者（属主）决定的访问控制；可验证指可以通过某种分析或验证手段得出所实现的安全机制与所描述的安全策略是相符的。从广义上讲，凡是拥有较强抗安全攻击能力的操作系统都可以认定为安全操作系统。

## 沿 革

安全操作系统的研究始于 20 世纪 60 年代。1967 年，计算机资源共享系统的安全控制问题引起了美国国防部的高度重视，隶属于国防科学部的计算机安全特别部队开始操作系统安全技术研究。1973 年，出现了第一个可证明的信息安全系统的数学模型 BLP 以及支持 BLP 模型的安全操作系统 Multics。1983 年，美国国防部颁布了历史上第一个计算机安全评价标准，简称橙皮书。橙皮书对安全操作系统的研究与发展具有深远影响。90 年代以来，随着网络技术的发展，操作系统的环境越来越复杂，1993 年出现了更适合互联网环境的计算机安全评价标准 Common Criteria，简称 CC 标准，1999 年成为国际标准，并在美国和西欧相继推出了符合 CC 标准的一系列用于保护重要数据和应对安全攻击风险的安全操作系统。进入 21 世纪，基于开源的安全操作系统的研究活跃，出现了一批开源的安全操作系统，著名的有美国国家安全局为主开发的基于 Linux 的 SELinux，Novel 公司推出的 AppArmor 等。

## 操作系统安全等级分类

根据安全机制及安全保障强度的不同，操作系统通常可以分为不同的安全等级。按橙皮书的经典规范，操作系统一般可以分为 D、C1、C2、B1、B2、B3 和 A1 七个安全等级。每个等级对应确定的安全特性需求和保障需求，后一个安全等级涵盖前一个安全等级的安全机制，并提供更强的安全功能和安全保障。

D 级是最小保护级。C1 级是自主安全保护级，主要包含用户标识与鉴别，自主访问控制。C2 级是受限的访问保护级，提供细粒度的自主访问控制和审计以及不允许遗留在内存或磁盘的信息（客体）被后面的进程或用户使用的“禁止客体重用”等。目前，人们广泛使用的商用操作系统多属 C1 级或 C2 级。

B1 级是安全标记保护级，引入了访问的主客体标记和强制访问控制，并作为其主要特征。符合橙皮书 B1 安全级及以上评价标准的操作系统通常才称为安全操作系统，也称可信操作系统。B2 级是结构化保护级，具有严格的安全策略及其形式化的模型，并提供可信路径以及对隐通道的分析。B3 级是安全域级，设有专门负责安全总控的引用监控机，还拥有可信恢复机制。A1 级是验证设计级，安全功能与 B3 相同，但形式化要求更高。

## 主要内容

安全操作系统包含的主要技术和安全机制通常有用户标识与鉴别、安全标记、强制访问控制、事件审计、禁止客体重用等,同时至少具有安全策略的非形式化描述,并对系统进行安全设计和验证。

用户标识与鉴别一般基于操作系统的可插拔的认证模块(PAM)实现。除了用户名与口令的认证机制外,进一步引入强化的身份认证,如基于智能卡的身份认证、基于指纹的身份认证等。安全标记是操作系统赋予访问主体(用户、进程等)和访问客体(如文件)安全属性的相应标签。用户标记由系统管理员通过操作系统设置,用户经认证后,获得用户标记,并派生进程标记;文件标记是赋予文件的安全属性,文件标记由创建文件的进程或者父目录决定。安全标记是强制访问控制的基础,为进行强制访问控制,还需要以规则形式规定主客体标记间访问关系的安全策略,强制访问控制一般采用策略实施与策略决策分离的原则实现,策略实施仅仅是嵌入到任务管理、文件系统、网络等代码中的一个控制程序,一旦进程需要访问客体,实施程序便通过调用策略决策模块进行权限的判断。基于安全策略逻辑的分析判断的实质性工作由策略决策模块完成,策略决策模块实际上是安全策略相关规则的一个通用解释程序。事件审计是对安全相关的事件进行审计,并可以在事后进行查看和分析。审计事件包括用户登录、文件访问、安全标记创建等。

## 发展趋势

随着信息安全与网络安全技术的发展,物理级、运行级、数据级和内容级等许多更有效的安全技术不断融入操作系统中。因此,只要能够为所管理的数据和资源提供相应安全级别的保护,并能有效控制硬件和软件功能的操作系统都可以认定为安全操作系统。典型的例子是 OpenBSD 操作系统,它不仅拥有很多专用的抗攻击功能,且每一行代码都经过严格细致的手工漏洞检查,被认为最安全的操作系统。可见,专门从安全的角度设计安全操作系统或者融入比强制访问控制更强和更丰富安全机制的操作系统将成为安全操作系统的发展趋势。近年来,为了适应网络应用的更高安全保障需求,广泛引入可信计算技术,从计算机体系结构入手,可信操作系统正成为构建基于信任根的计算系统信任链的重要研究方向。

**参考文献**

1. Wikipedia. http://en.wikipedia.org/wiki/Secure_operating_system

2. DoD 5200.28-STD, Department of Defense Trusted Computer System Evaluation Criteria. National Computer Security Center, Ft. Meade, MD, USA, Dec 1985

3. 刘克龙,冯登国,石文昌. 安全操作系统原理与技术. 北京:科学出版社, 2004

(何连跃)

anquan shujuku

**安全数据库(security database)** 在通用数据库管理系统基础上,提高安全性标准的**数据库管理系统**。安全数据库涉及技术、管理和物理等多个层面,主要应对数据库系统的风险和威胁,保证数据机密性、一致性、可用性等特性。和通用数据库相比,安全数据库在数据备份、访问控制、数据加密、审计、推理控制等机制方面得到加强。

数据库系统的物理损坏来源于地震、火灾、洪灾、过热、闪电等因素。数据库系统主要通过数据备份策略提高其物理安全性。

数据库访问控制机制根据系统存储的访问控制规则,确保用户只能按照其授予的权限来访问和操作数据库中的信息。按照实现方式和保护强度的不同,数据库访问控制可以分为自主访问控制、强制访问控制和基于角色的访问控制等。自主访问控制机制基于用户-对象权限矩阵,比较用户的请求与矩阵中保存的权限,判定用户的请求是否合法。强制访问控制机制为访问数据的主体和数据库中的客体设置安全性标签,通过比较安全性标签的级别判定主体是否有权对客体进行进一步的访问操作,确保信息的单向流动只能由低安全级主体向高安全级主体流动。基于角色的访问控制通过引入角色来关联权限和用户,支持灵活的权限管理。

数据库安全审计系统提供了一种事后检查的安全机制。安全审计机制将特定用户或者特定对象相关的操作记录到系统审计表中,作为后续对操作的查询分析和追踪的依据。通过审计机制,可以约束用户可能的恶意操作。

数据库加密使用已有的密码技术和算法对数据库中存储的信息进行保护。加密后数据的安全性能够进一步提高。即使攻击者获取数据源文件,攻击者也很难获取原始数据。但是,数据库加密增加了查询处理的复杂性,由于可能采用不同的加密方法,同时加密后的数据很难保证原始数据的特性,数据库管理系统一般无法直接查询加密数据,查询效率

受到较大影响。加密数据的密钥的管理和数据加密对应用程序的影响也是数据库加密过程中需要考虑的问题。

数据库推理控制机制用来避免用户利用其能够访问的数据推知更高密级的数据,即用户利用其被允许的多次查询的结果,结合相关的领域背景知识以及数据之间的约束,推导出其不能访问的数据。推理控制目前尚处于理论研究阶段。

实现安全数据库管理系统不仅和数据库系统本身的安全机制相关,还和数据库系统所在环境和安全管理密切相关。数据库系统作为基于操作系统的一种系统软件,依赖操作系统提供的安全功能和机制。同时数据库系统和客户端的信息交换都通过网络实现。因此安全的操作系统、安全的网络、完善的安全管理机制是实现安全数据库管理系统的基础和前提。

**参考文献**

1. Silberschatz A, Korth Henry F, Sudarshan S. 数据库系统概念. 6 版. 杨冬青,李红燕,唐世渭,等译. 北京:机械工业出版社,2012

2. Liu L, Ozsu M T. Encyclopedia of database systems. Springer, 2009

(杨冬青 高军)

# B

Bakesi fanshi

**巴克斯范式（Backus normal form，BNF）**

用以精确描述程序设计语言语法的一种形式体系。又称**巴克斯-诺尔形式**或**巴克斯-诺尔形式体系**。

美国 IBM 公司研究员 J. Backus 和丹麦哥本哈根大学教授 P. Naur 最早用以描述 ALGOL 60 语言的局部语法（其全程语法并不能用 BNF 描述），因而得名。

相对 BNF 所描述语法之语言说来，BNF 为**元语言**，它所描述语法之语言为对象语言。

BNF 之基本成分为如下部分。

元符号，一个是元等价符“::=”，表示“定义为”；一个是元或符“|”，表示对所指出的各款项之选择；还有一对尖括号“〈 〉”，用以连同所括之名表示非终极符。

元变量，用以表示相应对象语言之语法单位。元变量本身不属对象语言，其值为对象语言中之符号串。

以上述元符号及元变量为基础所构成的一组元公式（或称产生规则），用以刻画相应对象语言之局部语法。其形为

非终极符::=非终极符与终极符组成之符号串

其中非终极符（即上述之元变量）刻画对象语言之语法单位，终极符为对象语言之符号串。

元公式中“::=”之左方称为公式之左部，“::=”之右方称为公式之右部，具有相同左部之不同公式可以公用同一个左部。

例如，在 ALGOL 60 中，有

〈二进制数字〉::=0|1

〈十进制数字〉::=0|1|2|3|4|5|6|7|8|9

〈数字〉::=〈二进制数字〉|〈十进制数字〉

元公式之右部亦可出现正在定义的左方非终极符，此时为递归定义。如

〈标识符〉::=〈字母〉|〈标识符〉〈字母〉|〈标识符〉〈数字〉

即，标识符定义为以字母起头的字母数字串。

BNF 能严格表示一类上下文无关语言之局部语法，自从在 ALGOL 60 语言文本中首先采用以来，已在不少计算机学科中得到广泛应用和推广。

BNF 一词有其演变过程，起初代表 Backus normal form 或 Backus-Naur form，后来人们认为，既然 BNF 为一形式体系，为一元语言，称之为 form，似觉欠妥，F 以代表 formalism 为佳，从而便出现 Backus-Naur formalism，这里 formalism 一词与 formal system 同义。

**参考文献**

1. Naur P, et al. Report on the algorithmic language ALGOL 60. CACM, 1960, 3(5): 299-314

2. 陈火旺，等. 程序设计语言编译原理. 北京：国防工业出版社，1984　（陈火旺　程虎　贲可荣）

Bakesi-Nuoer xingshi tixi

**巴克斯-诺尔形式体系（Backus-Naur formalism，BNF）**　参见**巴克斯范式**。

bangong xinxi xitong

**办公信息系统（office information system）**

为政府和企事业等部门各级办公人员提供办公信息服务的**应用软件系统**。办公信息系统的作用是：以提高办公效率和办公质量为宗旨，利用部门所涉及的内外部信息，优化办公流程，改善办公信息的流转与管理，实现办公过程的信息化。在 2000 年 11 月办公自动化国际学术研讨会上，专家建议将办公自动化（OA）更名为办公信息系统。由此，一般认为办公信息系统与原名办公自动化系统是同义的。

办公自动化一词早在 1936 年就出现了，意在运用打字机和电话设备帮助办公人员处理办公业务。随着现代计算机的诞生和发展，特别是网络通信技术的发展，其外延和内涵都发生了很大变化。现代意义上的办公信息系统经历了 3 个阶段：①以简单数据处理为中心的阶段，表现为利用个人计算机和办公套件，实现数据统计和办公文档写作电子化，使办公信息载体从纸介质方式向电子方式转移。②以网络为基础、以工作流为核心的阶段，出现了文档管理、电子邮件、目录服务、公文流转和群组协同工作

等办公信息系统。③以决策管理为核心的阶段,不仅出现了网上实时交流信息的集成平台,实现公文和文档的一体化处理,而且在办公处理的每一环节上提供相关的决策知识,使办公人员从被动向主动转变,在提升办公人员决策能力的过程中,提高部门的整体办公质量、效率和应变能力。

办公信息系统由**计算机硬件**、**计算机网络**和办公专用设备,**操作系统**、**数据库系统**、**分布计算中间件**和办公套件等基础软件以及具体实现办公信息服务的应用软件组成。应用软件的开发参见**软件工程**。

办公信息系统的基本功能包括:①公文处理:实现公文的签收、登记、分发、归档等处理;②电子邮件:依据工作流程将文档传递给下一部门或人员;③签报管理:对公文进行拟稿、审核、签收、会签、拟办、签批、电子签名、交办、退稿、备查、销毁等管理;④会议与会务管理:提供会议计划、日程安排、会议资料准备、会议记录、会议查询和其他会务管理;⑤资料管理:对原始电子资料进行编辑、文摘与提要编写、刊号管理以及查询、统计和汇编;⑥案卷管理:将相关文件编辑成卷,并进行查阅和借阅管理;⑦公共信息服务:提供机构设置、职能与人员编制信息,各类业务信息以及通讯录、公告板等信息服务。此外,还有向公众提供信息发布、检索、查询、引导和专题电子论坛等功能。

在基本功能的基础上,可以按应用层次将办公信息系统分类为:①事务型,指日常办公事务处理和公文流转等行政事务处理两种;②信息管理型,指基于**管理信息系统**功能的办公信息系统;③决策支持型,指基于**决策支持系统**功能的办公信息系统。此外,还可以按应用行业特定的业务需求或特点分类为事务型、专业型、生产型、经营型、案例型和政府型等。

办公信息系统涉及软件技术、网络通信技术、行为科学和系统科学的综合应用。当前,办公信息系统已成为**电子政务系统**和**电子商务系统**的基础,在政府、企业和其他公共机构将得到更广泛更深入的应用。 (孙志挥 吴泉源)

bandaoti cunchuqi

**半导体存储器(semiconductor memory)** 用半导体大规模集成存储器芯片作为存储介质并能对数字信息进行存取的存储设备。典型半导体存储器的基本组成如图1所示,各部分的功能简述如下。

(1) 半导体存储器芯片阵列 它是信息存储单元的集合,由**半导体存储器芯片**按一定结构组成,是供**中央处理器**(CPU)或输入输出设备(I/O)存取信息的存储空间。

(2) 地址输入缓冲及驱动 由地址输入寄存器、片选译码器、地址分时电路和地址驱动器等几部分组成。地址输入寄存器的功能是接收CPU给出的地址,再将低位地址直接送给存储器芯片阵列,高位地址经过译码电路产生片选或块选信号。刷新计数器和地址分时电路为动态随机存取存储器(DRAM)所必需,它们的功能是将器件输入的行地址和列地址归并为一路,并将所产生的刷新地址送至存储器芯片的地址端口。

(3) 数据锁存及双向驱动 包含输入缓冲门、输入数据寄存器、输出锁存器和输出缓冲门等。它的功能是根据CPU的读写命令,将数据总线的内容写入被访问的存储单元,或者将被访问存储单元的内容读出到数据总线上,供CPU或I/O使用。

(4) 控制电路 包括访问信号控制和刷新电路。它的功能是接收CPU的启动、读、写、清除等命令,并对接收后的命令进行处理,产生各种时序控制脉冲和内部定时脉冲信号来控制存储器的读写操作或刷新操作。

在半导体存储器的组成中,存储器芯片阵列是核心,其余三部分是存储器的外围电路。

半导体存储器按在计算机中的工作性质可分为

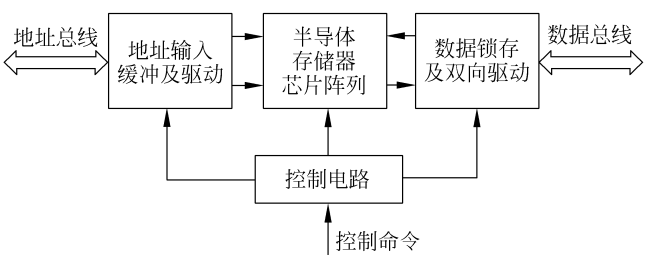


图1 典型半导体存储器的基本组成框图

**主存储器**(内存)、**辅助存储器**(外存)、**高速缓冲存储器**和控制存储器等。主存储器具有速度高、存储容量较小的特点,用来存放 CPU 当前执行的程序和数据。通常主存储器是由随机存取方式和只读方式两种半导体存储器组成,它们分享主存储器的同一地址空间。辅助存储器的特点是速度慢,存储容量大,用来存放 CPU 暂时不用的程序和数据。磁盘是辅助存储器的主流,但半导体盘(**固态盘**)已能实现磁盘的功能,并已在一些应用领域取代了磁盘。从功能上看半导体盘也可由电荷耦合器件(CCD)存储器实现,但从容量、速度和成本等因素综合考虑,则宜采用由 MOS 动态存储器(DRAM)或快可擦编程只读存储器构成。缓冲存储器用于在两个不同工作速度的部件之间交换信息过程中起缓冲作用。高速缓冲存储器是一种容量较小、速度很高的存储器,是为了克服主存储器与 CPU 在数据传输速度上的不匹配而设置的,通常采用双极型半导体存储器芯片来制作。磁盘缓冲存储器是为克服主存储器与磁盘存储器之间数据传输速度的不匹配而设置,利用磁盘缓存的方法可缩短磁盘平均取数时间,进而提高磁盘系统使用效率。磁盘缓冲存储器通常用 MOS DRAM 构成。

半导体存储器种类很多,就其制造工艺而言可分为**双极型半导体存储器**和**金属-氧化物-半导体存储器**(简称 MOS 存储器)。双极型半导体存储器以双极型触发器作为存储单元,其中采用晶体管-晶体管逻辑存储单元的称为 TTL 型存储器,采用射极耦合逻辑存储单元的称为 ECL 型存储器。MOS 存储器以金属-氧化物-半导体场效应晶体管作为存储单元。MOS 存储器的特点是集成度高,工作速度较快,适用于作容量较大的主存储器。电荷耦合器件(CCD)也是金属-氧化物-半导体存储器件,基于此器件构成的存储器称为电荷耦合器件存储器。它是一种易失性串行存储器,通常为非破坏读出,仅当写入新信息时,才清除原来存储的信息。

半导体存储器按存取方式可分为随机存取存储器(RAM)和**只读存储器**(ROM)两大类。

1. 随机存取存储器　RAM 可以随机地按指定地址从存储单元存取数据,在半导体存储器出现以前,主要是以记忆磁心为存储单元的磁心存储器。1971 年美国 Intel 公司推出 1103 型 1 kb MOS DRAM 芯片后,半导体存储器开始在一些主要厂家用作计算机的主存储器,到 1976 年 MOS RAM 主存储器的每信息位的价格已降到磁心存储器的一半,从此 MOS RAM 取代了磁心存储器,并一直占据主导地位。RAM 还可进一步分为**静态随机存取存储器** SRAM 和**动态随机存取存储器** DRAM 两类。

**静态随机存取存储器**　一种使用双稳态锁存电路存储每个信息位的半导体随机存储器。它无须像动态存储器一样周期性刷新信息位,但当没有供电情况下,其保持电荷仍然会丢失,因此在一般意义上它也还是易失性的(参见**静态随机存取存储器芯片**)。SRAM 主要用于二级高速缓存(level2 cache),或者处理器和主板存储器之间的缓存。SRAM 也有许多种,如异步 SRAM、同步 SRAM、流水式突发 SRAM(pipelined burst SRAM)等。

**动态随机存取存储器**　利用存储单元的电容内存储电荷的多寡来代表一个二进制比特(bit)是 1 还是 0 的半导体存储芯片构成的存储器。由于在现实中电容会有漏电的现象,导致电位差不足而使记忆消失,因此除非电容经常周期性地充电,否则无法确保记忆长存。由于这种需要定时刷新的特性,因此被称为"动态"存储器。DRAM 主要用于作计算机的主存储器。

基本 DRAM 存储单元结构(参见**动态随机存取存储器芯片**)多年来都没有太多的改变,但是 DRAM 间通信接口一直在变化。包括 asynchronous DRAM(异步 DRAM),视频 DRAM(VRAM),窗口 DRAM(WRAM),快页模式 DRAM(FPM DRAM),扩展数据输出 DRAM(EDO DRAM),突发 DEO DRAM(BEDO DRAM),直接 rambus DRAM (DRDRAM),同步 DRAM(SDRAM),双倍速率同步 DRAM,四倍速率同步 DRAM(DDR 2),八倍速率同步 DRAM(DDR 3)等。

2. 只读存储器　在正常运行中只能随机读取预先存入的信息而不能写入新的内容,也即信息一旦写入就不能更改。即使在断电情况下,ROM 仍能长期保存信息内容不变,所以它是一种永久存储器。

随着大规模集成电路集成技术的发展,出现了多种大规模集成电路 ROM 芯片,其中掩膜只读存储器(mask ROM)结构简单,存储信息稳定,可靠性高,能够永久性保存信息。可编程只读存储器(PROM)是由半导体厂家制作"空白"存储阵列(即所有存储单元全部为 1,或全部为 0 状态)出售,用户根据需要可以实现现场编程写入,但只能实现一次编程。可擦编程只读存储器(EPROM)、电可擦编程只读存储器(EEPROM)和快闪可擦编程只读存储器(flash EPROM)等 ROM 不仅可以现场编程,还可

以擦除原存储的信息内容,写入新的信息。目前使用的 ROM 都是大规模集成电路存储器芯片,当它们装入程序或微程序后就称为固件。

半导体存储器自 1971 年以来发展迅猛。以 MOS RAM 为例,集成度平均以每三年增加 4 倍的速度增长,存取周期缩短,逻辑功能加强。在 MOS RAM 中,DRAM 的集成度一直约为 SRAM 的 4 倍,因此 DRAM 多用于大容量存储器中。由于 DRAM 存储单元利用电容存储电荷的机理保存信息,故使用 DRAM 时必须定时周期性刷新所存储的信息,以免丢失。SRAM 不需要刷新操作,使用简便,在一些容量稍小的存储器中使用较广泛。由于 MOS SRAM 的速度不断提高,已几乎替代了双极型 RAM,促使双极型 TTL RAM 向更高速度方向发展。

半导体存储器由于存储单元阵列及其外围电路可集成在同一芯片上,因而生产过程简单,调试方便,容易实现自动化。它具有可靠性高,结构紧凑,组装密度高,体积小,工作速度快等特点;其缺点是信息易失性,所存信息因断电而消失。在不允许信息消失的应用场合,必须采取断电保护措施。但半导体存储器的综合优势是其他类型存储器不能相比的,在相当长时期内其性能与应用范围还将有较大发展。

**参考文献**

郑筠. MOS 存储系统与技术. 北京:科学出版社,1990

(郭志先　曹强)

bandaoti cunchuqi xinpian

**半导体存储器芯片(semiconductor memory chip)** 用半导体集成电路制造工艺把很多存储单元电路排列成阵列,并和外围电路集成在同一硅晶片上,形成能储存大量数据的集成电路。它广泛用于计算机、通信和家用数字电器等数字系统。

在数字系统中,指令和数据(包括声音、图像等信息)都以二进制数的形式出现。每个存储单元通常存放一位二进制数。在半导体存储器(SM)芯片中,通常把存储单元排成由行/列寻址的存储单元矩阵(阵列),存放数字系统工作过程中需暂时或长期保存的二进制数。为了存入和取出需要的数,存储阵列的外围电路包括选取存储单元的行/列地址缓冲和译码驱动电路、读出数据放大电路、写入数据驱动电路、片选和内部时序控制电路等。存储器芯片上集成了存储单元阵列及这些外围电路。这种行列结构的单元阵列给 SM 芯片带来一个共同的特点,当输入的行地址译码后,只激励一条选中的行线,其他行均为未选状态。与选中行相连的所有存储单元同时将所存的数据输出到相应的列数据缓冲寄存器,再由已输入列地址缓冲的列地址译码后将选中列的数据输出到相应的数据输出引脚;若再读出同一行地址而列地址不同的存储单元数据时,只需输入列地址,或由芯片按设定规则改变列地址,就可快速输出相应的数据。而写入时,在行地址输入后,可将仅列地址不同的数据串,根据列地址的变化写入对应的列数据缓冲中,其他数据缓冲保留行选中时读出的数据;再将数据缓冲中的数写入到选中行对应的单元中。这就是 SM 芯片发展过程中,为提高数据传送速率而采用的页面方式、同步方式的成组传送和双倍数据速率传输等技术的基础。

由于半导体存储器(SM)芯片由大量相同的存储单元组成,电路逻辑结构简单,特别易于大规模集成,成为半导体大规模集成电路产品市场非常重要且不可缺少的部分。在计算机,手持通信和娱乐设备,家用电器等数字系统中得到越来越广泛的应用。不仅有单独的 SM 芯片,而且在包括 CPU 在内的系统集成的芯片上也集成了不同类型的存储器电路。从 20 世纪 60 年代末以来,随着半导体工艺的发展,SM 芯片也经历了由双极型、P/NMOS(P/N 金属氧化物-半导体)型到互补金属氧化物-半导体(CMOS)型 SM 的发展历程。CMOS 工艺由于极低的静态功耗和大规模集成的方便,成为现今 SM 芯片的主流。

基本 SM 芯片分类如图 1 所示。根据断电后对数据的保存能力,SM 芯片可分为**易失性存储器芯片**和**非易失性存储器芯片**两大类。易失性存储器又分为随机存取存储器与非随机存取存储器。非随机存取存储器多以模块方式集成在芯片中,较少以单独芯片方式出现。而且其中的先进先出(FIFO)存储器、后进先出(LIFO)存储器的内核是静态随机存取存储器(RAM),用作高速缓存的关联存储器,也称按内容寻址存储器(CAM),其存储单元也是 SRAM 单元。所以易失性存储器芯片一般指随机存取存储器(RAM)芯片(参见**随机存取存储器芯片**)。它可从任何给定地址的存储单元快速读出或写入数据,但断电后数据消失,再加电后数据不确定。非易失性存储器芯片在断电后所存储的数据仍能保存,再加电后所存数据仍可继续使用,通常指只读存储器(ROM)芯片(参见**只读存储器芯片**)。ROM 在初期通常存储引导程序、基本操作系统和常用的不变的数据。因而将 ROM 数据的输入称为编程。

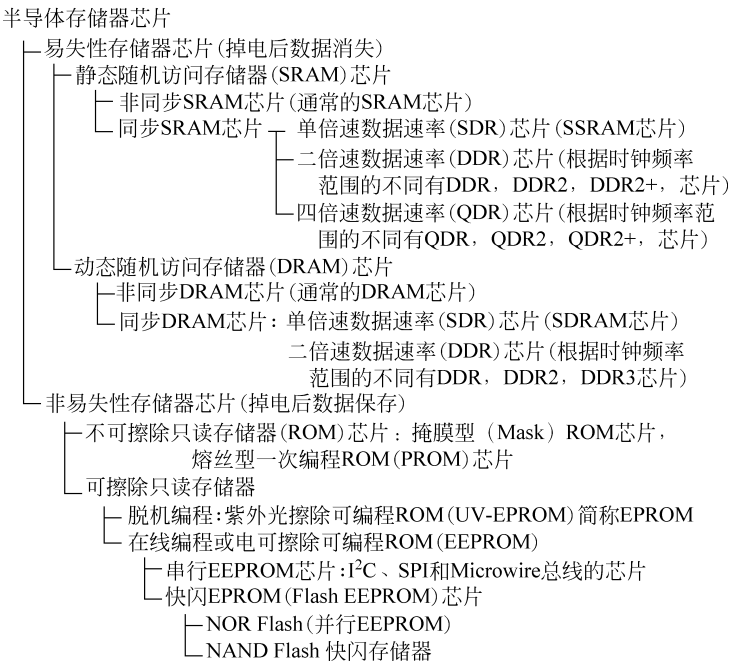


图 1　基本半导体存储器芯片分类

RAM 芯片分为静态 RAM(SRAM)芯片(参见**静态随机存取存储器芯片**)和动态 RAM(DRAM)芯片(参见**动态随机存取存储器芯片**)。SRAM 具有简单的接口、快速的地址读出时间和读出写入周期,这些优点使其作为高速缓冲存储器的应用一直延续至今。但 SRAM 偏大的存储单元面积和功耗的缺点,使 SRAM 芯片的存储容量比不上后来出现的 DRAM 芯片。DRAM 采用单管存储单元,数据存储在与单管相连的电容中,在读出后数据会被破坏,需重新写入。长时间不读时数据也会因漏电而消失,必须周期性地对所存数据进行刷新(读出/重写),以防止数据消失。这就是称为动态 RAM 的原因。而 SRAM 芯片则不需要刷新。

随着 CPU 工作频率的不断提高,对高速缓存和内存速度和容量的要求也不断提升。除工艺技术的进步外,SRAM 和 DRAM 都先后增加时钟输入,以同步地址、命令和数据接口。出现了同步 SRAM(SSRAM)和同步 DRAM(SDRAM)芯片,而内核仍是非同步的 SRAM 和 DRAM 电路。再进一步提高数据传输速率的技术是双倍数据速率(DDR),即相对于输入时钟,其上升和下降边沿都是接收数据的时钟。这样,在突发传送时,数据的传送速率是芯片输入时钟频率的二倍。此外,四倍数据速率(QDR)SSRAM 将数据输入端口和数据输出端口分开,利用流水线操作交替输入读出命令及地址和写入命令及地址。读出数据和写入数据可分别以双倍数据速率从各自的端口输入和输出。这样,芯片的最大数据输出速率可达时钟频率的四倍。

非易失性存储器中的只读存储器(ROM)芯片分为不可擦除 ROM 芯片和可擦除 ROM 芯片两种。掩膜型 ROM 芯片和熔丝型可编程 ROM(PROM)芯片(参见**可编程只读存储器芯片**)是最初使用的不可擦除 ROM 芯片。最先应用的可擦除可编程 ROM(EPROM)芯片是紫外光可擦编程 ROM(UV-EPROM)芯片,简称 EPROM 芯片。EPROM 芯片用紫外线擦除整个芯片所存数据,然后重新写入所需的数据。之后出现的电可擦除可编程 ROM(EEPROM)芯片(参见**电可擦编程只读存储器芯片**)用高电压实现擦除和写入,可实现在线擦除和编程。后来开发的技术可实现存储块或全芯片擦除,这就是快闪(快擦除)EPROM(Flash EPROM)芯片(参见**快可擦编程只读存储器芯片**),简称快闪存储器或闪存芯片。根据接口和内部组织的不同,闪存芯片又分为 NOR 和 NAND 两种类型。NOR 闪存芯片读出速度快,但存储单元面积大,不适合大容量集成。而 NAND 闪存芯片读出速度慢,但存储单元面积小,适合大容量集成。

与 RAM 不同,EEPROM(包括 NOR 和 NAND 闪存)的擦除/写入次数是有限制的,一般为 $10^5$ 或 $10^6$ 次。写入数据的保存时间可达 10 年。而且写入前要先擦除,写入时间和擦除时间都偏长,特别是擦除时间更长。因而不能作普通 RAM 使用。

为了满足客户/市场的需要,将满足客户特殊需要的外围电路与所需的核心 SM 电路集成在同一芯片上,形成了一些专门的存储器芯片。例如伪静态 RAM 芯片(或称为单管 RAM 或 Ut RAM 芯片)就是单管单元 DRAM 的内核和 SRAM 的接口,再加上 SRAM 与 DRAM 的接口转换和 DRAM 的时序控制,集成在同一芯片而成。目的在利用 SRAM 接口的方便性和 DRAM 易于大规模集成与成本低的优点。但随着技术的发展 Ut RAM 芯片现已很少单独应用。又如 OneNAND 芯片具有 NOR 闪存的接口和速度和 NAND 闪存的容量,就是在 NAND 闪存芯片上又嵌入集成了接口转换时序控制、小容量 SRAM 数据缓存和出错校验电路。可将原来存程序的 NOR 闪存芯片和存数据的 NAND 闪存芯片合为一特殊的 NAND 闪存芯片,用于要求体积小的手持/移动设备上。

用多芯片封装(MCP,参见**封装内系统**)也可形成一些专门的存储器芯片,以减小封装尺寸(例如 Ut RAM + NOR 闪存,NAND + DDR2 等),适应手持设备限制体积的要求或扩大存储容量,例如由 2/4/8 片 16 Gb/片封装成 32/64/128 Gb 的芯片,即多芯片的 DDP/QDP/ODP 方式,以满足大容量多媒体存储卡和固态硬盘的需求。

总的来说半导体存储器芯片向更大容量、更快速度、更多功能和更低功耗方向发展。为此随着工艺设计尺寸的不断缩小,MOS 管和存储单元的面积也不断减少。这为加大容量,提高速度,集成更多的功能创造了条件。同时允许的工作电压也不断降低(由初期的 5 V 降到现在的 3.3/2.5/1.8/1.5/1.35 V),功耗也可降低。为对这些基本的 SM 芯片的容量和时间参数与应用的关系有一个概念,用表 1 举例说明。

**参考文献**

1. Sharma A K. 先进半导体存储器——结构、设计与应用. 曾莹,等译. 北京:电子工业出版社,2005

2. 桑野雅彦. 存储器 IC 的应用技巧. 王庆,译. 北京:科学出版社, 2006

3. http://www.samsung.com/Products/Semiconductor

(孙祖希)

表 1 基本半导体存储器芯片的参数和应用实例

| SM 芯片名称 | QDRII SSRAM | DDR3 SDRAM | NOR 闪存 | NAND 闪存 |
|---|---|---|---|---|
| 典型容量 | 1 M×36 b | 2 Gb | 4 M×16 b | (256 M×16 b) + 16 MB ECC |
| 时钟周期 | 4.0 ns | 1.5 ns | 读周期 70 ns min | 读周期 42 ns min |
| 行地址读出时间/随机读出周期 | 4.45 ns/8 ns | 27 ns/48 ns | 70 ns/70 ns | 随机读 60 μs max |
| 突发方式数据读出时间/读出周期 | 0.45 ns/2.0 ns | 数据输出速率 1333 Mb/s | 70 ns/70 ns | 顺序访问 42 ns |
| 突发方式写入时间 | 突发方式写命令周期 8 ns/4 字 | 数据输入速率 1333 Mb/s | 字编程 14 μs<br>块擦除 0.7 s | 页面编程 930 μs<br>块擦除 16 ms max |
| 工作电源 | 1.8/1.5 V | 1.5/1.35 V | 5 ~ 1.8 V | 3.3/1.8 V |
| 主要应用 | 高速缓存 | 主内存 | 系统启动和基本操作系统 | 断电后需保存的大量数据和程序 |

ban jiandu xuexi

**半监督学习(semi-supervised learning)** 研究如何综合利用有标记样本和未标记样本的**机器学习**理论、模型与方法。机器学习算法都有输入和输出(例如分类问题输入就是数据特征,输出就是数据标签),其目的就是学习一个从输入到输出的映射函数。半监督学习研究如何在已知数据的输入和其中一部分数据的输出的情况下有效准确地学习映射函数。根据映射函数的类型,半监督学习又可分为直推(transduction)和归纳(induction),其中直推

设置下的半监督学习往往只能预测已知输入样本的输出,而不能学习具体的映射函数的形式,因而这类算法往往不具有拓展性。而归纳设置下的半监督学习能够得到映射函数的具体形式,因而具有更好的拓展性,但是具体算法也更加复杂。

传统的机器学习算法包含监督学习和非监督学习两大类。其中监督学习假设学习映射函数时已知所有输入数据的对应输出,而其目的就是学习一个具体的映射函数。非监督学习(参见**非监督学习**)假设仅仅已知输入数据,而没有任何输出的信息,其目的是从这些输入数据中学习有意义的输出。监督学习和非监督学习都有着各自的优势和局限性。监督学习由于有相应的输出监督信息,使得学习到的结果更加准确可靠,但是得到这些监督信息往往需要请相关领域专家标注,这就需要花费人力财力。相反,非监督学习不需要任何的监督信息,这也使得学习到的结果往往不可靠。而半监督学习则综合了以上两种学习算法,它往往只需要一小部分输入数据的输出,却能够得到与监督学习相当甚至更好的学习结果。

迄今为止,学者们已经提出了很多种半监督学习的算法,例如基于产生式模型的算法,低密度分离,以及基于图的算法等。半监督学习中也还有很多有争论的问题,例如没有监督信息的样本究竟什么时候会有帮助,以及算法的推广性能如何等。

**参考文献**

1. Chapelle O, Schölkopf B, Zien A. Semi-supervised learning. MIT Press. 2006

2. Zhu X, Goldberg A B. Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, 3(1): 1-130

(王飞　张长水)

ban jiegouhua shuju

**半结构化数据(semistructured data)**　介于完全结构化数据(如**关系数据库**与**面向对象数据库**中的数据)和完全无结构数据(如音像数据与文本数据)之间的数据。

和关系数据不同,半结构化数据一般是自描述的数据,它的模式不是统一的、预先给定的,而往往是不规则且经常变动的,并且它的数据和模式是混合存放的。这种特点使得半结构化数据有很大的灵活性,能够满足不同应用、不同企业之间交换信息的需要,但同时也给半结构化数据存储、查询处理带来了很大困难。目前得到广泛应用的 XML 数据就是半结构化数据。

对象交换模型(object exchange model, OEM)可以用来描述半结构化数据。它最初在斯坦福大学的异构数据集成系统 TSIMMIS 中引入。OEM 是一个简单的、自描述的、嵌套的对象模型。每个对象具有唯一的对象标识。对象分为原子对象和复合对象两种。原子对象包含一个原子类型的值,如整数、实数、字符串等。复合对象的值是有序对〈标记,子对象标识〉的集合,每一个标记是对象与子对象之间关系的描述。从结构上看,OEM 是一个带标记的有向图,节点表示对象,边表示对象间的关系。每个 OEM 图都有一个根节点,从根节点至任意节点都是可达的。

图 1 是对象交换模型的一个简单示例。它描述了某个餐饮业的结构。图中每个对象都有一个整型的对象标识,根对象包括三个子对象,两个餐馆和一个酒吧。每个餐馆都是复合对象,而酒吧是一个原子对象,字符串"酒吧 1"是该对象的值。

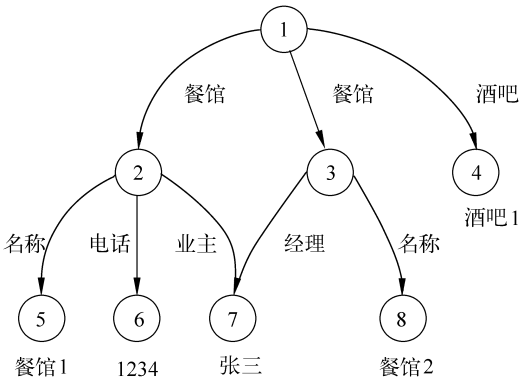


图 1　对象交换模型示例

半结构化数据管理中查询处理需要支持从现有的半结构化数据中定位特定部分,或者将一种类型的半结构化数据转换为另一种类型的半结构数据。为了提高半结构化数据查询的效率,通常需要建立面向数值或者面向结构关系的索引。

数据模式抽取和增强是半结构化数据管理中一个重要操作。为了有效地管理大量存在的无结构数据和半结构化数据,我们可以抽取或增强其模式信息,从而支持更加丰富的查询类型和精确的查询处理,实现数据优化存储,支持数据有效性验证等。

在半结构化数据存储过程中,一方面需要保持半结构化数据的结构信息,支持结构查询处理;另一