

# 计算机系统结构

胡铭曾  
编

哈尔滨工业大学出版社

23

2/1

# 计 算 机 系 统 结 构

胡 铭 曾 编

哈 尔 滨 工 业 大 学 出 版 社

## 计算机系统结构

胡铭曾 编

※

哈尔滨工业大学出版社出版  
黑龙江省新华书店发行  
哈尔滨工业大学印刷厂印刷

※

开本 787×1092 1/16 印张 9.75 字数 220,000

1985年8月第1版 1985年8月第1次印刷

印数 1—5,000

书号 15341·24 定价 1.90元

## 前 言

本书从整体的概念讲述计算机各部分的功能，与计算机原理不同之处在于计算机系统结构着重于框架、思路的描述，而对具体机器组成技术很少阐述。全书共分八章，内容包括流水工作、虚拟存贮器、通道等。容错与诊断方面的内容没有编入本书。

本书可作高等院校计算机专业的教材，也可供有关工程技术人员使用。

由于时间仓促，又限于水平，不妥之处在所难免，恳请读者给予指正。

编 者

一九八四年三月

# 目 录

|                                           |       |
|-------------------------------------------|-------|
| <b>第一章 绪论</b> .....                       | ( 1 ) |
| §1. 计算机系统的组成.....                         | ( 1 ) |
| §2. 系统结构的含义.....                          | ( 2 ) |
| §3. 计算机系统的分类.....                         | ( 2 ) |
| §4. 系统结构发展的回顾.....                        | ( 4 ) |
| §5. 器件的发展对系统结构的影响.....                    | ( 6 ) |
| §6. 程序的可移植性对系统结构的影响.....                  | ( 8 ) |
| §7. 应用对系统结构的影响.....                       | (10)  |
| 习题.....                                   | (11)  |
| <b>第二章 数据表示</b> .....                     | (12)  |
| §1. 数据表示和系统结构的关系.....                     | (12)  |
| §2. 浮点数表示基值的确定.....                       | (13)  |
| §3. 字符串的表示.....                           | (14)  |
| §4. 堆栈数据表示.....                           | (15)  |
| §5. 数组的数据表示.....                          | (16)  |
| §6. 自定义数据表示.....                          | (18)  |
| 习题.....                                   | (19)  |
| <b>第三章 指令与编址</b> .....                    | (21)  |
| §1. 指令系统简介.....                           | (21)  |
| §2. 指令格式及其优化.....                         | (23)  |
| §3. 指令系统的改进.....                          | (26)  |
| §4. 信息访问方式.....                           | (28)  |
| §5. 按堆栈访问信息.....                          | (31)  |
| §6. 堆栈机的结构特点.....                         | (34)  |
| §7. 按内容访问——相联存贮器.....                     | (39)  |
| 习题.....                                   | (42)  |
| <b>第四章 执行与控制</b> .....                    | (44)  |
| §1. 虚拟机 (Abstract machine) 的概念.....       | (44)  |
| §2. 微程序.....                              | (45)  |
| §3. 模拟 (Simulation) 与仿真 (Emulation) ..... | (46)  |
| §4. 控制流改变的处理.....                         | (48)  |
| §5. 中断系统.....                             | (49)  |
| §6. 指令的重迭工作方式 (Overlap) .....             | (51)  |

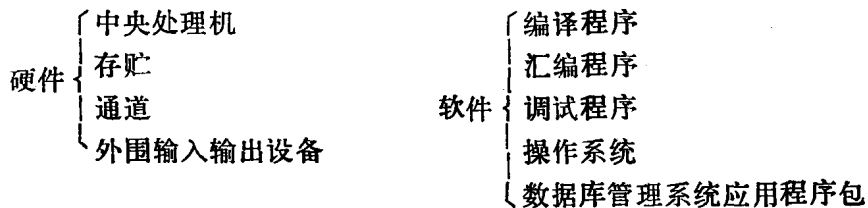
|                                          |              |
|------------------------------------------|--------------|
| §7. 流水工作方式(Pipe line).....               | (54)         |
| §8. 流水线处理机设计中的若干问题.....                  | (58)         |
| §9. 向量的流水处理.....                         | (65)         |
| 习题.....                                  | (68)         |
| <b>第五章 存贮体系 (Memory Hierarchy) .....</b> | <b>(69)</b>  |
| §1. 引言.....                              | (69)         |
| §2. 主存多体交叉存取 (Interleaving).....         | (71)         |
| §3. 程序的局部性与定位.....                       | (74)         |
| §4. 虚拟存贮器 (Virtual Memory) .....         | (77)         |
| §5. 高速缓冲存贮器 (Cache).....                 | (87)         |
| §6. 存贮器保护.....                           | (91)         |
| 习题.....                                  | (93)         |
| <b>第六章 输入输出系统.....</b>                   | <b>(94)</b>  |
| §1. 输入输出系统的发展.....                       | (94)         |
| §2. I/O系统的一般结构 .....                     | (94)         |
| §3. 总线结构.....                            | (97)         |
| §4. 输入输出过程.....                          | (101)        |
| §5. 通道程序及状态字.....                        | (104)        |
| §6. 通道结构框图.....                          | (108)        |
| §7. 外围处理机.....                           | (109)        |
| 习题.....                                  | (111)        |
| <b>第七章 并行处理与多机结构.....</b>                | <b>(112)</b> |
| §1. 并行性概念的发展.....                        | (112)        |
| §2. 并行处理机.....                           | (113)        |
| §3. 单级互连网络.....                          | (117)        |
| §4. 循环网络和多级网络.....                       | (119)        |
| §5. 多机系统的发展.....                         | (122)        |
| §6. 多机系统的硬件结构.....                       | (124)        |
| §7. 程序并行性问题.....                         | (126)        |
| §8. 数据流机器简介.....                         | (130)        |
| 习题.....                                  | (131)        |
| <b>第八章 描述与评价.....</b>                    | <b>(132)</b> |
| §1. 计算机设计语言CDL.....                      | (132)        |
| §2. 用CDL语言描述一台计算机.....                   | (135)        |
| §3. 模拟测试.....                            | (138)        |
| §4. 性能评价的作用.....                         | (142)        |
| §5. 等效指令法.....                           | (142)        |
| §6. 典型程序法.....                           | (143)        |
| §7. 评分法.....                             | (145)        |
| 习题.....                                  | (148)        |

# 第一章 绪 论

## § 1. 计算机系统的组成

早期的计算机并不构成我们今天所指的计算机系统。随着计算机的发展，中央机从只能用于科学计算而扩展到做更多的各种数据处理，为了保存大量数据形成了存贮体系，为了大量数据输入输出、联机和脱机处理又增加了外围系统，各个子系统逐渐发展与中央机共同构成了计算机系统。

计算机系统由硬件 (Hard ware) 及软件 (Soft ware) 组成。



硬件是计算机系统在实际装置。用机器语言写成的程序不需要中间插入解释程序或翻译程序就可由它直接处理。它是计算机系统的基础，是软件和人借以完成其本身功能的依据。

软件是指用户共同使用的一组程序及其有关资料。它是抽象的思想，其作用是便于计算机系统的使用，提高计算机系统的效率，并扩展硬件的功能。

软件和硬件虽然是两个不同的部分，但是在逻辑功能上它们是等效的。例如早期的机器，乘、除法是用程序即软件的方法来实现的，后来人们把这一部分“硬化”，用硬件的方法直接进行乘除运算。从理论上讲，计算机只要有“相减”“转移”两种指令就可以用来算题，硬件实现的操作都可以用软件来实现，而软件的功能如编译、操作系统中的基本操作也可以用硬件来实现。因此具有相同功能的计算机系统，它们的软、硬件之间的功能分配在很宽范围内变化，软件和硬件的交界面模糊不清，并且是动态地不断变化着。选择什么样的分配比例主要取决于现有硬件状况下的性能价格比。

最初的计算机由于硬件价格昂贵，所以往往只设置必要的硬件，而大部分功能由软件去完成，这样做的优点是造价低、灵活性和适应性好，但速度低，所需的存贮容量大。随着器件的发展以及对计算机性能要求的提高，分配给硬件的功能在不断扩大 (图 1—1)。

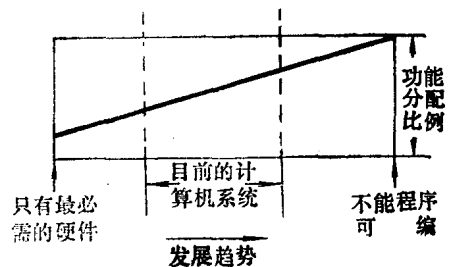


图 1—1 计算机系统的软硬件功能分配

## § 2. 系统结构的含义

我们这里是指计算机（的）系统结构（Computer Architecture）而不是计算机系统的结构（The structure of computer system）。要对系统结构下一个确切定义很困难，因为软硬件的交界面不清，并且在不断地变化。1964年美国的 Amdahe等人提出：计算机系统结构是由程序员所看到的计算机属性，即其概念性结构与功能特性。但是由不同级别的程序员看机器具有不同的属性。例如由使用高级语言（如 FORTRAN）的程序员来看 DJS-130 和 DJS-183 就几乎没有什么差别，也就是说具有相同的属性。而实际上这两台机器之间存在着差别，这种本来存在的事物或属性从某种角度看好象是不存在的现象被称为透明（Transparent）。DJS-130 和 DJS-183 在指令系统、寻址方式、输入输出设备连接方式等各方面都不相同，但是高级语言程序员却看不见，也就是说对高级语言程序员是透明的。所以计算机系统结构应该是由机器语言程序员所看到的计算机属性，它是程序员为编制出能在机器上正确运行的机器语言程序所必须了解的计算机结构。

研究计算机系统结构仅从程序员看到的机器属性是不够的，因为有很多功能对程序员来说是透明的。例如，高速缓冲存储器（Cache）、先行控制和流水等对程序员都是透明的，所以还必须从机器设计的角度，最佳地、最合理地把加法器、寄存器、控制器、存储器、磁盘机、磁带机以及各种外围设备等组合起来以实现用户的要求。最佳、最合理的组合是指机器结构的形成而不是指具体的逻辑设计及工程实现。

综合起来说，研究系统结构就是要弄清楚软硬件的交界面，研究软硬件的功能分配，提出系统必需的硬件支持以及最佳的硬件实现。要克服过去软硬件设计的脱节现象，立足于现有的器件水平，考虑到可能的各种应用的基本算法的要求，使软硬件之间密切协调配合。软硬件工作人员都应有计算机系统的整机概念，既要懂得机器的使用也要清楚机器的组成，这样才能更好地发挥作用。

## § 3. 计算机系统的分类

迄今为止，生产出来的计算机种类繁多，性能差异很大。研究计算机的分类将有助于了解各类系统的演变、属性和相互关系，有助于深化对某些特定系统的了解，有助于开发对新系统探索的途径。

### 1. 以“代”分类

根据计算机所采用的基本器件不同，可将计算机分为第一代至第四代计算机，这种分类反映了计算机发展的时间顺序，是最普遍采用的分类方法。

第一代计算机的硬件由电子管组成，这类机器只有少量的软件即汇编程序，处理方式手工方式。

从六十年代初开始，出现了由晶体管组成的第二代计算机，软件有了高级语言，作业连续处理，并有了 I/O 控制系统。

从六十年代中期到七十年代初，应用的主要是由集成电路组成的第三代计算机，它采用多道程序处理，作业可以多重处理，分时系统及数据管理也有了很大进展。

七十年代生产的称为“第三代半”的计算机的硬件中采用了中、大规模的集成电路，处理方式发展为联机处理，并行作业。

用超大规模集成电路组成的机器称为第四代计算机，软件功能也大大加强。如可进行数据库管理、文字图形语声等处理、超高级语言处理等等。

## 2. 以“型”分类

通常将计算机分为巨型、大型、中型、小型和微型机。但是依照什么分型呢？是依照速度、性能、大小，还是依照其它指标呢？一般说来巨型机的速度比大型机快，大型机的比中型的快，依此类推。但是从机器主频及每秒执行指令数来说，微型机的主频很可能超过小型机及中型机而与大型机相近。从性能上讲，一般大型机字长比小型机长，数据类型也比较多。例如大型机字长为32位，小型机字长为16位，微型机字长为8位；大型机有定点、浮点、甚至向量、数组等表示，而小型机、微型机只有定点表示等等；大型机有较完整的存贮体系（超高速缓存、主存、磁盘、磁带等），而小型机一般只有主存和辅存；大型机和小型机及微机相比配备了更多种高级语言，更完善的操作系统和更多的应用程序包。另外，大型机在提高可靠性和可用性方面还采取了很多措施，如纠错编码、复执、诊断以及其它冗余方法。

但是我们都不能以某一具体指标来划分各型机器，例如，不能规定字长超过32位的就是大型机，小于16位的就是小型机或微型机，现在字长16位的微型机已有好几种，字长32位的微型机预计不久也将问世。再如，也不能规定具有超高速缓存、虚拟存贮器的就是大型机，最近好几种型号的小型机也都有了超高速缓存和虚存。当然更不能以体积大小来划分机器类型。需要指出的是上述各指标都只是在某一段时间间隔内各型机的表征，随着时间的推移，特别是器件技术的发展，过去高档机器的性能可能还不如现在的低档机器。例如五十年代的大型机占地面积几百平方米，运算速度每秒一万次，主存容量16384字，但其功能还比不上现在的某些微型机。所以，有人认为区别各型机器的标准是长期变化不大的价格。计算机系统的性能和价格之间的关系如图1—2所示。图中的虚线为等性能线，表

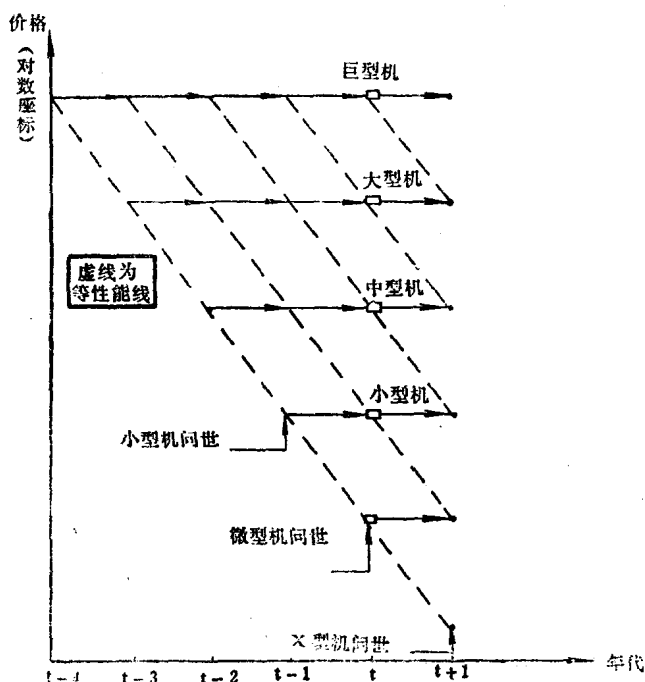


图 1—2 分型和性能、价格关系的示意图

计算机系统的性能和价格之间的关系如图 1—2 所示。图中的虚线为等性能线，表

明过去价格相当于现在巨型机价格的计算机在性能上只相当于现在的微型机，这种现象称为性能下移。现在计算机工业的发展，一是维持价格不变，充分利用器件技术的进展不断提高机器性能（沿图中实线方向）；一是在性能基本不变的情况下，不断降低机器价格（沿图中虚线方向），这两种方法对计算机的发展都具有重要意义。

### 3. 以“流”分类

根据计算机系统的信息流特征分类是反映系统组织特点的方法，也是更深入地研究计算机系统属性的方法。M.J. Flynn 提出按照计算机系统中指令流与数据流的相互关系可将系统分为四类（图 1—3）：

#### (1) 单指令流-单数据流 (SISD)

系统，即基本的单处理机。

#### (2) 单指令流-多数据流 (SIMD)

系统，即在同一时刻此系统对多个数据进行相同的处理，一般具有多个处理单元，为并行处理机。

#### (3) 多指令流-单数据流 (MISD)

系统，Flynn 把 CDC 6600 和 7600 的外围处理机以及 Texas 仪器公司的 ASC 列入这一系统，但从严格意义上讲并不存在这类系统。

#### (4) 多指令流-多数据流 (MIMD) 系统，同时有多个程序对它自己的数据进行处理，存在着多个处理机、存贮器和多个控制器。典型的 MIMD 系统就是多计算机系统。多个独立的单处理机的集合也可以说属于此类。

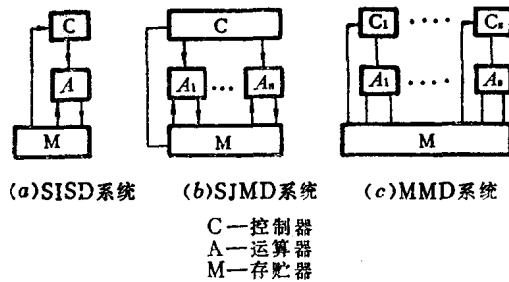


图 1—3 按“流”分类的计算机系统

## § 4. 系统结构发展的回顾

1946年普林斯顿高级研究所数学教授冯诺曼 (Von Neumann) 所提出的机器结构由运算器、控制器、存贮器及输入输出设备组成。

### Von Neumann 机器的主要结构特点

归纳如下：

(1) 存贮器是顺序线性编址的一维结构，按地址访问每个编址单元。机器速度取决于存取周期。

(2) 指令和数据一起存贮在一个同样的存贮媒体中，并且不能区分。

(3) 指令由操作码和地址码组成。

操作数的数据类型由操作码确定，操作数本身确定不了它是何种数据类型。

(4) 指令在存贮器中顺序存放，由指令计数器指明将被执行的下一条指令在内存的地址，每执行一条指令，计数器自动加 1。

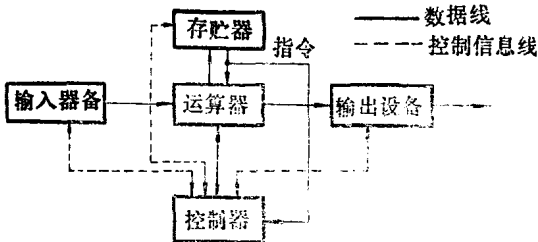


图 1—4 Von Neumann 型机器的结构

(5) 指令在执行过程中可以被修改。

(6) 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都经过运算器，各部件的操作以及它们之间的联系由控制器集中控制。

Von Neumann 提出的这种结构奠定了计算机发展的基础。虽然，当初设计时没有考虑到采用高级语言，也没有顾及操作系统及很多实际应用的各种要求，但是这类机器结构还是成功地在科学计算、数据处理和工业控制等广泛的领域得到了应用。三十多年来 Von Neumann 型机器有了很大的改进，但绝大多数计算机仍以上述结构特点为基础。

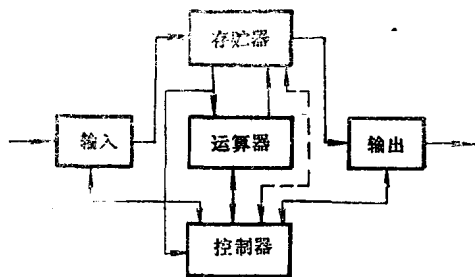


图 1—5 以存储器为中心的结构

这种以运算器为中心的结构，运算器十分繁忙，而变成以存储器为中心的结构（图 1—5）输入输出的数据不再通过运算器而直接进入存储器，形成了输入输出与存储器之间的直接通路。如把对存储器及 I/O 集中控制改变成分散于存储器和 I/O 的分布控制，就可使运算器能与输入输出同时并行工作。

I/O 控制系统又把选设备、启停、数据校验等功能也接收过来形成一个功能有限的小计算机并称为通道（Channel）。它可以执行由系统控制软件所组成的通道程序，形成了象 IBM 360 那样具有通道的结构形式（图 1—6）。

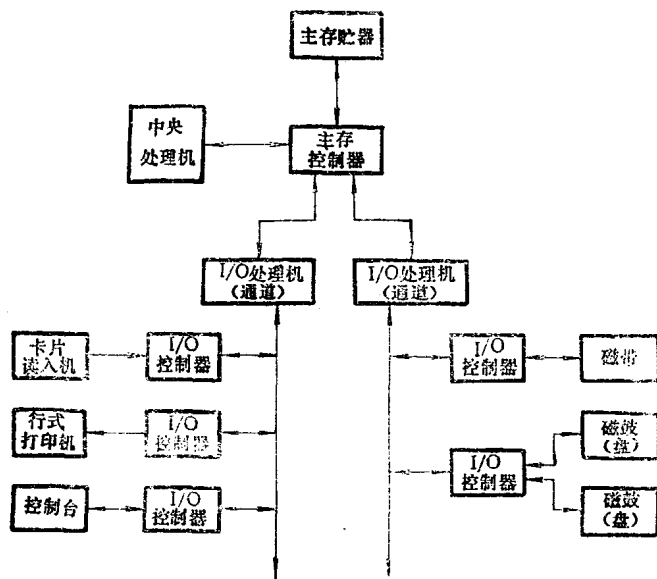


图 1—6 具有通道的机器结构

DEC 公司在设计 PDP-11 时又提出了一种总线结构，它对存储器、通道、CPU 等给予同等看待，接到统一的公共总线上。这种结构的优点是易于扩展，成本低，工作可靠。它的性能价格比可以在很宽的范围内（指各种性能的计算机系统）都取得较好的结

果。现在大多数小型机都采用这种结构。

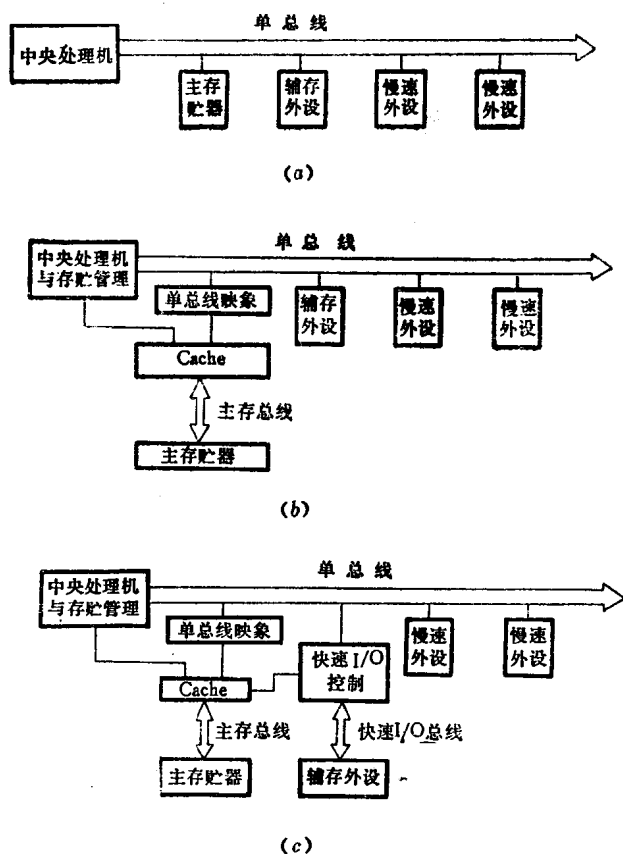


图 1—7 总线结构的计算机

上述的几种结构形式只是系统的宏观结构形式。实际上为了提高系统的速度还普遍采用了先行控制、流水线方式；存贮器应用交叉存取、超高速缓冲存贮器等技术。一方面增加了一些新的功能部件，另一方面系统中原来的部件在微观结构上也发生了很大的变化。目的就是给系统软件提供足够的硬件支持并提高系统的性能。具体内容将在以后各章节中叙述。

## § 5. 器件的发展对系统结构的影响

器件是组成计算机系统的细胞，它的发展是推动计算机发展的主要动力，并对系统结构的发展起着关键性的作用。

器件从电子管开始，经历了晶体管、集成电路、大规模集成电路和超大规模集成电路，其发展是异常迅猛的。由于器件的发展，计算机的每级门的延迟由几百毫微秒缩短到现在的几毫微秒；体积也大为缩小，同样的逻辑功能结构，80年代的只是50年代（用电子管）的1/500；存贮器变化更大，80年代的为50年代（采用磁心）的1/1600；从可靠性来看，大规模集成电路的失效率比电子管线路的失效率降低了三至四个数量

级, 现在每个元件的平均无故障时间(MTBF)已达到  $10^7$  小时以上; 器件的基本性能迅速提高而价格随时间指数下降(图1—8), 它的性能价格比的提高更为显著。正是有了这个基础, 计算机才有今天这样巨大的发展。

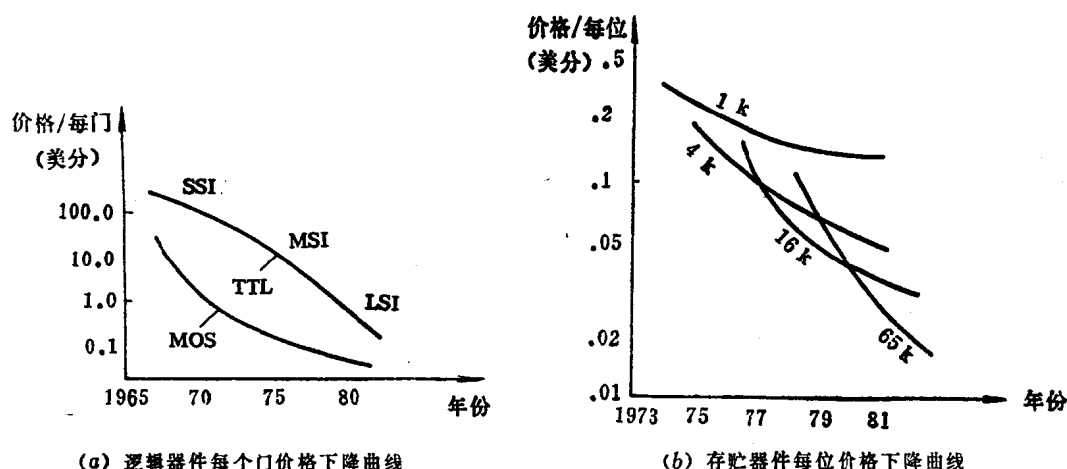


图 1—8

器件的发展改变了逻辑设计的传统方法, 现在的着眼点就不在于要以最少的门数来实现机器的构造, 而是着重于分析采用什么样的组成方法才能利用市场上大量生产的LSI和VLSI片子。过去的控制是用硬联逻辑完成的, 很不规整, 不易大量通用, 而现在改用微程序控制, 在PROM中存入微程序, 把硬联控制网络的复杂逻辑联结数变为控制存储器的各种码点。PROM这种片子的通用性很高, 可以大量生产。微程序的设计不能仅用过去那种逻辑设计方法, 要用软件的方法来实现。机器设计中的这种变化也是软、硬件相互渗透, 相互影响的一种表现。

器件的发展为系统结构的发展提供了可能性。例如, 并行处理需要用大量的器件, 只有器件的可靠性大幅度提高, 价格大为降低才有可能被采用。又如, 由于高速的价格便宜的双极性存储片组成的高速缓冲存储器Cache的出现, 大大缩短了存储器的存取时间, 使机器的解题速度得到了显著的提高。

器件的发展促使系统结构的“性能下移”, 把原先在大型机采用的, 并已被证实是有用的新结构下移到销售量比大型机多的中、小型机去, 这使厂家愿意生产新的功能更强的VLSI片子。所以小型机的设计原则是充分发挥器件技术进展快的优势去实现效能肯定的机器结构, 而对大型机或巨型机, 就不能拘束于原有的结构, 而要在器件发展的基础上, 在新的组成方法上下功夫。所以, 大型机或巨型机不可能采用集成度很高的片子, 但它们成功的组成方式可以为中小型机所承袭而采用大规模集成电路。

器件发展除了促使系统结构性能下移外, 也促进分布处理的发展。例如VLSI进入终端和I/O控制器, 改变了原来CPU与I/O控制器之间的分工, 把原先由CPU实现的部分软件功能分散给终端及I/O控制器, 使它们具有运算、存储及处理能力, 成为智能终端(如PDP 11/130)。微处理机价格的不断下降, 从硬件角度已可以构成几十甚至几百台微处理机组成的并行处理系统, 但目前还没有合适的算法, 所以又迫使人们去研究新的并

行算法。

总之，系统结构设计者必须密切注视器件的发展，并不断探索这些新器件和生产工艺给系统结构带来的新途径，创造出更好的系统结构。

## § 6. 程序的可移植性对系统结构的影响

早期由于当时的器件贵、不可靠，而软件的价格比较便宜，所以设计者尽量简化硬件结构，把更多的功能交给软件去实现。应用上的很多要求都由软件解决，就使软件日益庞大，并且积累了大量应用软件与系统软件。软件的编写虽然比较容易，但是很难排除错误，生产效率很低，所以除非有极大的好处，程序设计者和生产厂家是不愿意也不应该按新的系统结构（新的指令系统）重新设计软件。器件的迅速发展使软硬件的状况发生了很大的变化，硬件价格持续下降，可靠性日益提高，而软件的价格却一直上升，可靠性提高缓慢。庞大的不易改变的软件和计算机硬件的迅速发展产生了矛盾，而计算机系统没有完善的软件是不能工作的，新的系统结构就不得不考虑程序的可移植性（Portability）。所谓程序的可移植性指的是一个程序可以不经修改地由一台机器搬到另一台机器。这样，经过证实是可靠的软件就能长期使用，而不必随着机器的更新重新编制软件，大大节省了编制软件的工作量，使新的计算机系统能很快的发挥作用。怎样才能更好地实现程序的可移植性呢？其一是采用统一的高级语言。如果真能用一种统一的能满足各方面需要的高级语言，那么至少应用程序的可移植性问题就解决了。如果操作系统的全部或一部分可用高级语言来编制，那么系统程序中的这部分也可以移植。但目前存在着上百种高级语言，没有一种语言能满足各方面需要。即使是同一种高级语言在不同机器上也各有其特点，并不完全一致；各个生产厂家也并不全遵守统一的规定，而有自己的“方言”。再考虑到在某些高级语言编制的程序中，为了节省存贮空间提高执行速度，有些部分采用汇编语言或机器语言书写，这样就更难实现移植了。其二是采用统一的机器语言或汇编语言来达到程序的可移植性。但是机器语言和系统结构是密切相联的，所以只能在具有相同系统结构的机器上才能采用统一的机器语言或汇编语言，根据这些就提出了系列机的概念。

由于器件发展很快，所以机器性能不断提高，但要求软件也很快改变是不可能的，软件希望环境长期稳定，也只有这样，软件才能不断提高质量。系列机就是先设计好一种系统结构，软件设计者就按这种系统结构设计软件，而对硬件实现是按技术及器件水平提供不同档的机器。系列机内各档机器具有相同的属性，按这个属性编制的机器语言程序及编译程序，各档机器可以通用。属性中最主要的是指令系统、数据表示及概念性结构。

IBM 360/370系列的概念性结构如图1—9。它是按照科学计算、事务处理及实时控制这三方面的要求确定的。它具有完整的科学计算用的算术指令，还有用于事务处理的字符行运算指令及十进制运算指令以及实时控制所需的比较完善的中断系统及有关指令。系列内的各档机器速度和价格不同，它们的数据通路宽度也有8位、16位、32位和64位之分，但是数据表示的字长都是32位。程序设计员看到的定点数是统一的半字长或全字长，浮点数为单、双或四字长。IBM发展到今天这样庞大就是因为有了系列

机，它成功地解决了软件要求环境稳定与器件迅速发展之间的矛盾。二十年来，它的机器不断革新，先后推出性能价格比极好的各档机器，上下价格差几百倍，而软件的环境几乎稳定了二十年，可以应用于系列内的各档机器，也就是说这些机器是软件兼容的 (Software Compatibility)。

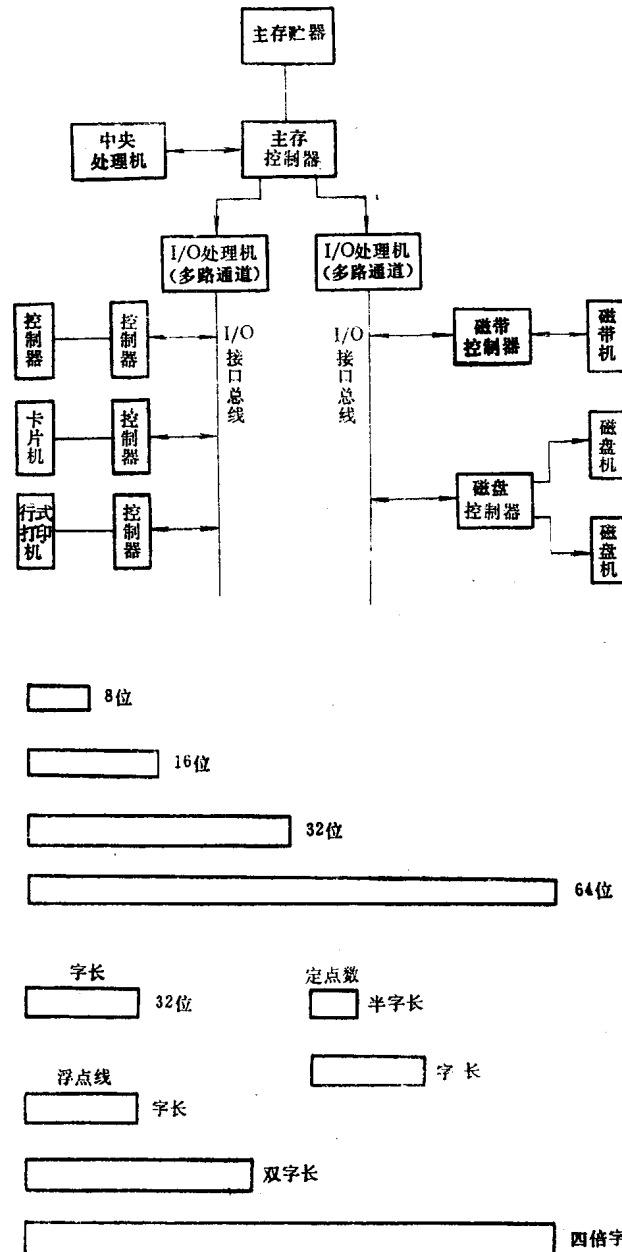


图 1—9 IBM 360 的概念结构

软件兼容是指同一软件可以不加修改地用于其它各档机器，所得结果相同。其中又

有向上兼容(即低档机上的软件可以不经修改用于高档机)、向下兼容(与向上相反)、向后兼容(先投入市场的机器用的软件可以不经修改用于后投入市场的机器)及向前兼容(与向后相反)之分。一般向上兼容易于满足,向下就不一定。而向后兼容是必须确保的,这样才能把以前在软件方面取得的大量成果用于新机器上。向上兼容不一定确保,例如 PDP 11/44 比 PDP 11/70 性能低,但在 PDP 11/44 中加了一套事务处理指令,所以它的程序不能完全用于 PDP 11/70。系列机结构设计得好坏、是否有生命力,就看是否能在向后兼容的前提下不断地改变其构成。构成的改变往往只引起系统软件的修改(主要是编译程序),而不会改变应用软件。

系列机的实现解决了软件要求环境稳定而机器构造、性能不断发展之间的矛盾,对计算机工业的发展起到了推动作用。但是在软件兼容的前提下机器结构的发展毕竟是有限的,要求现在新设计的机器和十几年甚至几十年以前的机器兼容就很难有结构上的突破。所以,从某种意义上讲,庞大的不易改变的软件已成为计算系统迅速发展的保守力量,阻碍了计算机系统的进一步发展。

## § 7. 应用对系统结构的影响

最早的计算机是为科学计算服务的,它的主要特点是二进制、浮点、固定字长、输入输出量不大。五十年代末期,计算机使用范围扩大到商业、事务处理,这类应用要求机器具有十进制运算及字符行处理的能力。它的计算量少,但需要大量的输入输出,因而发展了通道,使输入输出和中央处理机可以同时工作。当机器的应用扩大到过程控制时,这种应用就要求机器具有实时处理能力(较强的中断处理能力)及和外界的多种接口。可见对于不同的应用,机器具有不同的特点。

但对用户来说,总是希望机器所适用的范围越广越好,即科学计算用的机器可用于统计,过程控制机可以做某些工程计算。于是把科学计算、事务处理和实时控制三方面的结构特点合并到一台机器中来,从而形成多功能通用机。多功能通用机的出现减少了机器的型号,扩大了一个系列的应用,使环境稳定、软件硬件都便于发展。

随着科学的发展,巨量计算对机器的运算速度提出了更高的要求。原子能研究、飞行器设计、反导弹武器系统、气象预报等往往需要做 $10^{12}$ — $10^{17}$ 的运算。以计算量为 $3.6 \times 10^{14}$ 为例,每秒做一亿次运算的机器需要计算 1000 小时,在这些巨量计算中,蕴含着并行的因素,这就促使巨型机相继出现,如向量机 STAR-100、CRAY-1、阵列机 ILLIAC IV、相联处理机 STARAN 等。这些巨型机价格高,使用复杂,非一般用户所能应用,但是在研制和使用这些机器的实践中所积累的经验 and 取得的成果,被多功能通用机所采用以提高其性能。例如阵列运算部件 FPS-100 有独立的加法器、乘法器及流水控制,它被作为一台外设接入通用机,新部件的加入不影响通用机原有的兼容性,对用户程序来说是透明的,但是提高了通用机的性能,这一点是系统结构的发展,不仅被大型机而且也被小型机所接受。

八十年代非数值运算的应用,主要是数据管理方面的应用,并且将会有很大的发展,它要求有很大的数据库(大于  $10^{10}$  字节)。用现有的索引方法不可能实现快速查

询或修改，软件过于复杂并满足不了速度要求，因此必须借助于能提高机器管理程度的各种硬件，并依靠硬件来实现信息保护。近年来对数据库专用机的要求越来越强烈，数据库机器必将是计算机发展的一个重要方面，它的研究成果，如相联存贮器、智能磁盘控制器等也会相应地发展而被通用机所接受。

综上所述，可以看出系统结构同器件、软件等密切相关，系统结构设计者不能只从硬件本身的需要进行设计和研究，而应充分考虑程序设计者和用户的需要，怎样更好地软硬结合地提高解题能力和速度，合理地分配软硬件的功能，更好地给软件提供足够的硬件支持，不断地在原系列机中引入新的部件以提高机器性能等。由于受程序可移植性和复杂的系统软件的限制，系统结构还只能是逐步地发展，近期内难于有新的突破。

## 习 题

1. 试举例说明软件和硬件在逻辑功能上是等效的。
2. 以你熟悉的某台机器为例，指出哪些是软硬件的交界面。
3. 一个计算机系统的整机概念应包括哪几方面？
4. 画出按“流”分类的计算机系统的概念性结构。
5. 何谓软件兼容？在系列机中为什么必须确保向后兼容？
6. 试说明器件发展如何引起系统结构性能下移。