

大型 IBM 计算机系统 结构与实现

上海电子计算机厂

1984.9

目 录

第一章 介绍和概述	1
1.1 CPU	4
1.2 主存	5
1.3 通道	5
1.4 控制器	6
1.5 外设	6
1.6 本书的结构	6
第二章 CPU结构	8
2.1 执行程序	8
2.2 中央处理	12
2.3 寄存器和PSW	14
2.4 CPU的状态	17
2.5 指令集	21
第三章 主存结构	29
3.1 主存	29
3.2 实存	30
3.3 虚存	30
3.4 动态地址转换	37
3.5 主存的保护、访问和更改指示符	42
第四章 中断结构	45
4.1 中断处理概述	45
4.2 I/O中断	47
4.3 外部中断	49

4.4	管理程序调用中断	1
4.5	程序中断	1
4.6	机器检查出错中断	
4.7	重启动中断	1
4.8	中断素	1
4.9	屏蔽与不屏蔽	1
4.10	中断优先级	1
第五章	通道结构	1
5.1	I/O 操作原理	1
5.2	通道 类型及其特性	1
5.3	I/O 指令	1
5.4	通道命令字	7
5.5	通道命令	7
5.6	通道程序	7
5.7	I/O 中断	1
5.8	执行 I/O 操作的阶段	8
第三部分	控制器和设备的实现	9
第十章	实现控制器和设备所使用的概念	9
10.1	共享和非共享配置	9
10.2	通道和控制器之间的接口	9
10.3	控制器操作的基本原理	9
第 11 章	用于通信网络的控制器和设备	10
11.1	IBM 数据通信基础	11
11.2	IBM 3705 通信控制器	13

11.3	IBM 3270 信息显示系统	148
第12章	IBM 磁盘存储控制器和驱动器	163
12.1	磁盘驱动器的组织	163
12.2	存贮控制功能	166
12.3	通道, 存储控制和驱动的交互作用	170
12.4	磁盘驱动器和控制器的例子	176
第13章	杂用控制器和设备(略)	179
索引		186
第6章	多处理	199
6.1	前缀	200
6.2	CPU 间的信号发送	203
6.3	串行化及同步	204
第二部分	IBM 大型处理机的实现	207
第七章	实现中信用概念	208
7.1	高速缓存 (Cache)	210
7.2	交错访问主存	211
7.3	取指部件及执行部件	212
7.4	处理机和存贮器周期	213
7.5	重叠操作	214
第八章	IBM 3033 处理机的实现	214
8.1	处理机概述	215
8.2	处理机存贮器	217
8.3	处理机存储控制功能	218
8.3.1	高速缓存	218

8.3.2 动态地址转换	220
8.3.3 转换快表	226
8.3.4 STO堆栈	231
8.4 指令预处理功能	231
8.5 执行部件	235
8.6 通道和控制器	239
8.7 维修和重执功能	241
8.8 多处理	242
8.9 选择特征	247
8.10 与 370/168 实现比较	249
第9章 IBM3081 处理机的实现	251
9.1 3081处理机概述	252
9.2 中央存储器	254
9.3 外部数据控制器	255
9.4 系统控制器	257
9.5 与 3033 实现比较	257

1、介绍和概述

一个计算机系统是由几个相互作用的部件 (Component) 组成的, 即一个或多个中央处理机、主存、数据通道和外部设备。系统的结构确定了部件的功能, 不同部件间的互相作用规则。部件又可以再由若干个成员 (member) 组成, 而他们实现起来又可以不尽相同, 但结构给出了统一的处理方法。例如: I/O 操作对于不同的外围设备, 如磁盘和终端, 从结构处理上讲是完全一致的。在结构规定的限制范围内部件可以有不同的实现方法。例如: 两个磁盘驱动器都完成结构定义的功能, 但互相的运行特性, 如: 旋转位置查找, 或多磁道操作可以是不同的。另外, 二个中央处理机可具有相同的结构, 但他们在实现细节上, 诸如: 高速缓冲存贮器的容量, 流水线操作, 以及其它一些特性可以有所不同。

本书的目的是: 1) 阐述大型 IBM 计算机系统中使用的结构 (如 370/168, 3033) 即称之为 370 结构。2) 给出实现大型处理机, 特别是 IBM 370/168 和 3033, 3081 的实例记载。3) 描述了用于 I/O 操作的控制器和设备的实现。尤其是关于通信和直接存取存贮器方面的实现。

“IBM 系统/370 的操作原理”文本详细描述了 370 结构, 它谈到以下问题:

- * 计算机系统的功能部件, 即主存、中央处理机、通道控制器 (注: 即外设控制器) 和设备。

- * 不同功能部件间相互作用的整套规程和原则。
- * CPU 可执行的指令系统。
- * 各功能部件中为处理指定集，所需要的子部件。
(如，通道中的子通道)

结构使用了包括下列各部件的计算机概念模型。

- * CPU
- * 主存
- * 通道
- * 控制器
- * 设备

这些部件间的相互联系由图 1—1 给出。当程序执行时，cpu 对主存存取，也对通道发出 I/O 指令，以便数据在设备和主存间传送。通道经控制器在设备和主存间传送数据，传送完毕，就用中断向 cpu 报告。

此结构模型与 360 中所使用的模型是类似的。360 是 370 的前辈，然而，370 在 360 的结构上增加了以下主要功能。即：

- * 程序使用虚地址，并将这虚地址转换成实存地址。
- * 使用多个 cpu 共享同一个存贮器，以提供多机处理功能。
- * 使用数组多路通道，以便对直接存取存贮器设备进行交叉 I/O 操作。

此外，为获得高效的功能，370 结构装上 MVS 那样的多道程序操作系统。为了较好地理解计算机结构和操作系统功能间的相互作用，我们编制了一个含有 MOS 操作系统中所使用概念的附

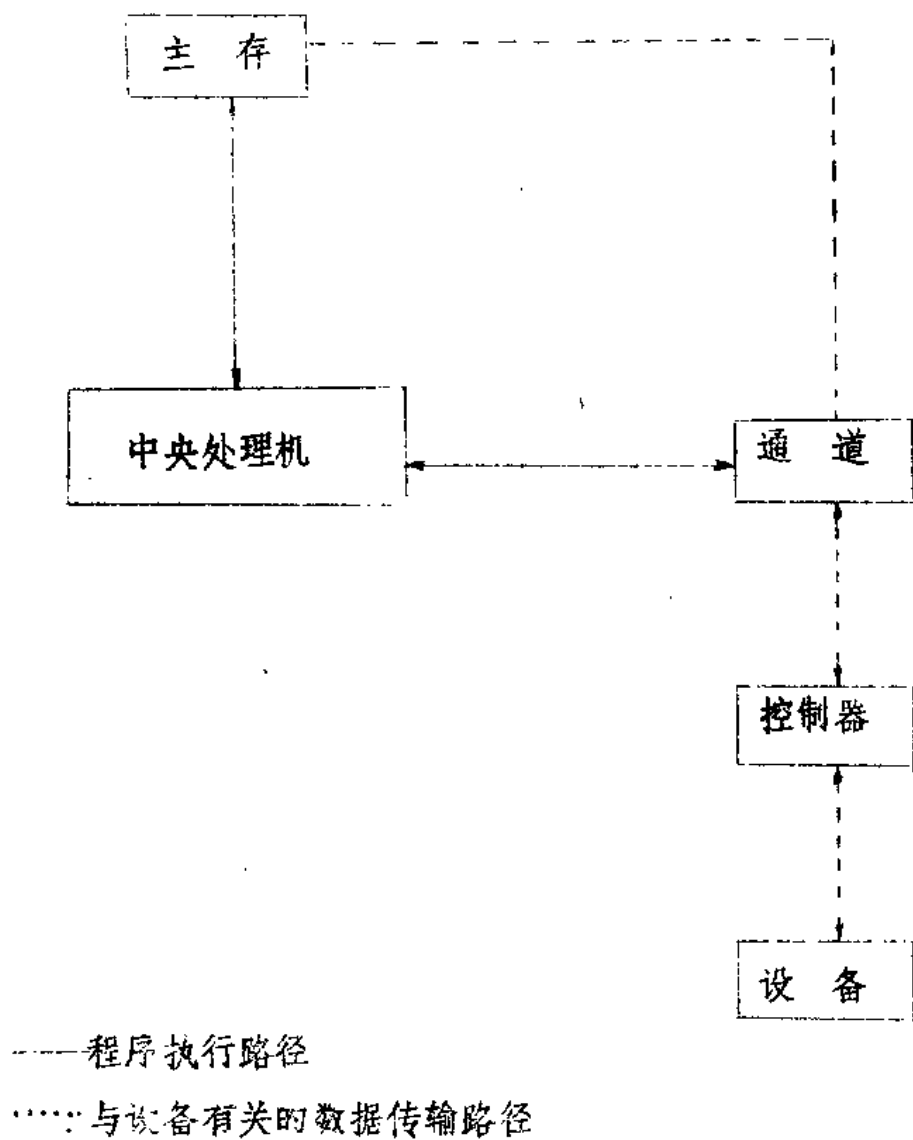


图 1—1 / IBM 大型系统的结构部件

录。读者在阅读本书时，随时参看附录将得益不浅。

下面我们将简单叙述结构部件，以及它们相互作用概况。

1.1 CPU

CPU 的主要功能是：

(1) 执行程序

(2) 处理中断

所谓程序就是一个指令序列，这些指令取自结构所规定的指令集，除非该序列由于跳转指令而修改，指令将按它们在程序中出现的先后顺序来执行，而跳转将在第二章中叙述，中断会引起 CPU 暂时挂起正被执行的程序，而去执行中断处理程序。中断可由一系列确定的事件引起，（如：机器检查出错，I/O 操作完成）结构对每个引起中断事件规定了中断码。中断处理程序在处理中断时要用到这些中断码。“处理中断”意味着：采取必要的步骤来确定中断源，并执行动作以解决中断提供的情况，（如：通知一等待程序，由它启动的 I/O 操作已经完成）。

CPU 用以下方式与主存和通道相互联系：

- * CPU 和主存的相互联系：CPU 在执行程序时从主存读取指令和数据，如果程序中指令规定要修改主存则修改主存单元。
- * CPU 和通道的相互联系：CPU 用 I/O 指令启动 I/O 操作，当通道执行这一操作时，CPU 可脱出来执行其它指令。换言之，当通道执行 I/O 操作时，CPU 并不等待。等 I/O 操作完成或执行到预定的断点时通道向 CPU 发中断（在一次 I/O 操作中可能发生好

几次中断请求)。

1.2、主 存

“主存”这个术语代表一条指令可访问的物理存贮器，它不同于CPU组成寄存器，程序状态字的局部存贮器，它也不同于微码使用的控制存贮器。

结构把主存看作一个连续的物理存贮单元的集合，它们有着顺序的地址，CPU执行程序时可以存取这些单元，通道在执行I/O操作时也可存取这些单元。

1.3、通 道

通道的功能是执行组成一个I/O操作的所有控制和数据传输操作。通道与CPU、主存、控制器相互作用，通道采用下面的物理路径来完成数据传送。

主存 $\longleftrightarrow$ 通道 $\longleftrightarrow$ 控制器 $\longrightarrow$ 设备

CPU用I/O指令启动一次I/O操作，随后，通道的执行就不再依赖于CPU。I/O操作通常包括控制操作（如：在指定磁盘的磁道上确定读/写头位置）数据传送操作（如：从磁盘上读一个记录），关于通道与控制器间相互作用，在关于通道结构、设备和控制器的实现章节中详细论述。

下面的通道操作原理将被重复叙述，因为它们是理解370结构的基础。

(1) CPU并不直接卷入外设与主存间的数据传送，也不控制一个外设是通道完成这些功能的。

(2) CPU用I/O指令启动一次I/O操作，通道执行这

一操作，通道完成 I/O 操作后向 CPU 发中断。

(3) 在执行数据传送和控制操作时，通道通过控制器与外设联系。

通道与控制器间的物理联接称为接口，它的功能在控制器和外设的实现等有关章节中有所论述。

1.4 控制器

控制器的功能是在外设和通道间充当中间人。通常几台外设联到一台控制器，几台控制器联到一个通道。通道和控制器间，控制器和设备间的相互作用根据实现情况可以有很大的差异。在控制器和外设实现的章节里给出了控制器和它们在通道与外设间相互作用的例子。

1.5 外设

外设就是一个外围设备，它是数据传送的一个端点，另一个端点是主存，外设的例子有磁盘，卡片阅读机，终端、打印机、以及类似的在数据传送中使用的外围设备，通道几个外设联到一个控制器，而它又依次联到一个通道。

1.6 本书的结构

本书分为三部分。第一部分关于 370 结构，第二部分关于 370/168, 3033 和 3081 处理机的实现情况，第三部分是关于控制器和外设的实现情况。

由于各种结构部件间的相互作用很复杂，往往读前面章节时要参考后面章节里叙述的东西，而给读者带来了不便。此外为了便于理解，我们在前面几章中简单描述了各种部件，而操作的详细情况在后面的章节中给出。在 I/O 操作和多处理机情况下，

这种情况尤为突出，在那里，前面先给出简单的描述，而在解释了各种有关题目后，再介绍详细情况。

2、CPU结构

结构为CPU确定两个主要功能。即，

- * 执行程序。
- * 处理中断

这些功能被分成几个子功能。我们将在适当的时候讨论它们。结构定义了一个包括以下内容的局部寄存器，供CPU使用。

- * 通用寄存器
- * 控制寄存器
- * 浮点寄存器
- * 程序状态字 (PSW)

这个局部寄存器是CPU的一部分，它是与主存分开的。上面罗列的项目可由适当的指令，用不同的方法处理。下面各节将详细论述寄存器和PSW。

2.1、执行程序

所谓程序是一指令序列，其中的指令都隶属于该结构定义的指令集，我们将在其它的章节中讨论指令格式和指令的分类。就本书而言，一个程序可以是一个计算机的操作系统，或是在操作系统下运行的一个应用程序。在附录1，我们讨论了一些操作系统概念，它们与硬件结构密切相关。奉劝读者查阅一个附录1，以便理解在程序执行过程中，应用程序、操作系统、硬件所起的相应作用。

结构定义了二种程序执行方式，称为基本控制 (BC) 和扩展控制 (EC)。在BC方式中，程序被看作为在主存中占据连续单

元的区域。CPU 按序执行指令，直至遇到一条开始一个新序列的转跳指令。图 2—1 描述了这种执行程序的观点。程序状态字 PSW 有一保存，下一条指令实地址的字段，在顺序执行指令的情况下，该字段以一定的字节数自动递增，其步长是现行指令的长度，在转跳情况下，转移地址存放在 PSW 中。

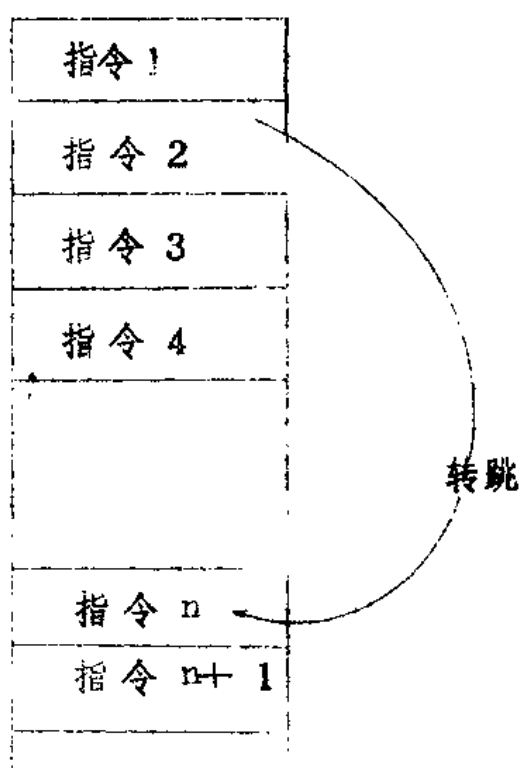
在 EC 方式中（这是 370 中普遍使用的模式），程序可用虚或实地址二种不同的方式来工作（通过设置 PSW 的第 5 位来选^模）。在虚地址方式下，程序被看作在虚存中占据连续偏址的空间，CPU 按序执行指令，直至遇到一个转跳指令，然后从转移地址开始一个新的序列。图 2—2 描述了使用虚地址的程序执行情况。注意，PSW 包含了下一条将被执行的指令的虚地址。

在程序执行时，虚地址被转换成实地址。结构定义以下用于虚到实地址的转换的硬件部件。

* 动态地址转换（DAT）。

	下一指令的主存地址
--	-----------

程序状态字 (PSW)

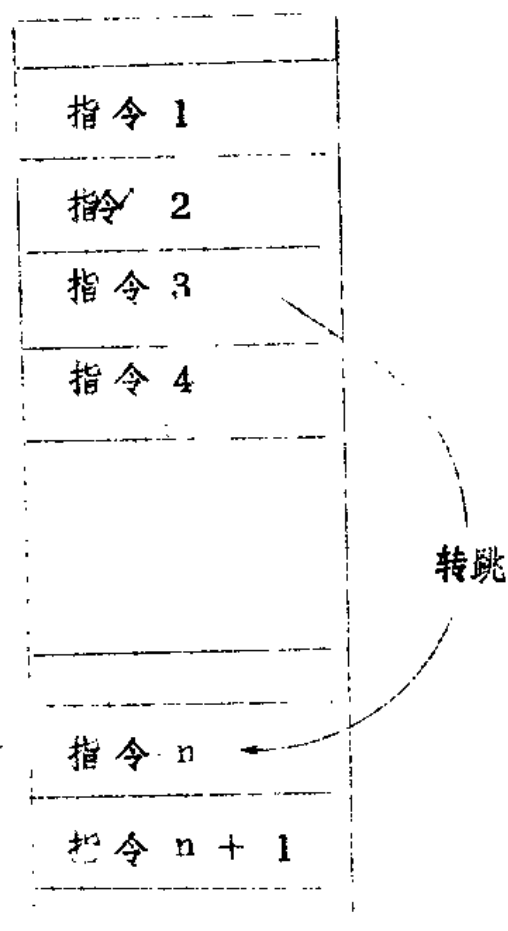


主存单元

图 2—1 / 使用实地址的程序运行记录
(BC 方式)

下一指令的虚存地址

程序状态字 (PSW)



虚存单元

图 2—2 / 使用虚地址的程序运行记录

LIB C 方式

* 翻译快表 (TLB)

在第三章关于虚存的章节里，详细说明了 DAT 和 TLB 的功能，在这，我们只指出 TLB 存贮先前转换了的虚页地址，仅当 TLB 中没有一个虚页条目时，才需实行一次动态地址转换。

CPU 在执行程序的主要功能中要完成以下子功能，

- * 从主存读取指令。
- * 译指令。
- * 从主存读取操作数。
- * 虚地址到实地址的转换。
- * 作为程序执行的一个部分对主存写入。
- * 对 CPU 局部存贮器的读写 (寄存器和 PSW) 。

结构弄不详细规定怎样读取指令，译码和执行。在有关实现的章节，我们说明了在大型处理机中，这些操作是如何完成的。结构只要求那些指令依次执行，读取和译码的次序 (order) 由具体实现自己处理。

2.2、中 断 处 理

中断是通知 CPU 发生了特殊的事件或情况。结构将引起中断的事件和情况分为 6 级。引起中断的事件和情况的例子有：完成一次 I / O 操作，硬件出错，预置的计时器时间间隔终止，浮点算术运算溢出。

后面的章节将详细记述 370 结构所定义的中断结构。目前，我们只指出下列中断特点。

(1) 正在 CPU 上执行的程序可以屏蔽某些级中断。在这种情况下，CPU 称为对那几级中断屏蔽。(当 CPU 不是屏蔽时称为