

第3章 新媒体的技术元素



新媒体以现代通信与计算机技术为支撑，它的不断发展也越来越依赖于这些技术的进步，使得这些技术越来越成为新媒体不可分割的基本元素。

3.1 信息编码基础

数据是记录信息的符号。用电子数字计算机进行信息处理，就是对符号形式的数据进行存储与处理。电子数字计算机以开关元件为基础。开关元件是具有两种稳定状态的电子元件，因此要用电子数字计算机进行处理的数据必须表示为用代表其电子开关两种状态的 0 和 1 表示的符号。将其他形式的数据符号转换为 0、1 表示的符号的过程，称为 0、1 编码。

3.1.1 数值数据的 0、1 编码

1. 二进制的基本概念

人们最习惯的计数方法是使用十进制。十进制具有如下 3 个特点。

(1) 用 0、1、2、3、4、5、6、7、8、9 十个符号表示数。

(2) “逢十进一”，即在某一位上，当够 10 时，就要向左边的位进一。因为每个位上最大值只能是 9。

(3) 十进制数从小数点往左，依次为十位、百位、千位、万位……从小数点向右，依次为十分位、百分位、千分位、万分位……即一个数字 m ，在位置 i 上时，其值为 $m \times 10^i$ 。 10^i 称为 i 位权。例如，个位的位权为 10^0 ，十位的位权为 10^1 ，百位的位权为 10^2 …

相对于十进制，二进制具有如下特点。

(1) 只能用两个符号表示数。在中国的八卦中，这两个符号用“—”和“--”表示；在数字电路中，这两个符号可以用高电位和低电位、有脉冲和无脉冲、开和关等表示。这里，统一用 0 和 1 表示。

(2) “逢二进一”，即在某一位上，当够 2 时，就要向左边的位进一。因为每个位上的最大值只能是 1。

(3) 十进制数从小数点往左，各位的位权依次为 2^0 、 2^1 、 2^2 …，即 1、2、4、8、16、32、64、128、256、512、1024、2048 等；小数点往右各位的位权依次为 2^{-1} 、 2^{-2} …，即 0.5、0.25、0.125、0.0625、0.03125 等。

2. 十进制数与二进制数之间的转换

表 3.1 为几个十进制数与二进制数之间的对应关系。

表 3.1 几个十进制数与二进制数之间的对应关系

十进制数	0	1	2	3	4	5	6	7	8	9	10	16	32
二进制数	0	1	10	11	100	101	110	111	1000	1001	1010	10000	100000

下面介绍一般数的转换方法。

1) 二一十 (B→D) 进制转换

规则：将 1 所在的位上的位权值相加。

例 3.1 101.11101B= ?D。

解：

位 权： 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5}

二进制数：1 0 1 . 1 1 1 0 1

计 算： $4 + 0 + 1 + 0.5 + 0.25 + 0.125 + 0 + 0.03125 = 5.90625D$

2) 整数十一二转换

规则：从足够的位权值开始，连续减去各个位权值：够则减该位取 1，不够则不减该位取 0；直到减完最后一个位权值 1。

例 3.2 158D=?B。

解：考虑 158 在 128 与 256 之间，则先减 128，该位取 1，差为 30；减 64，不足，该位取 0；再减 32，不足，取 0；再减 16，够减，取 1，差为 14；减 8。够减取 1，差为 6；减 4，够取 1，差为 2；减 2，够取 1，差为 0；减 1，不够取 0，结束。

上述过程如图 3.1 所示。

二进制位权	计算过程	二进制数各位值
256	158	
128	-128 够	1
64	30 -64 不够	0
32	30 -32 不够	0
16	30 -16 够	1
8	14 -8 够	1
4	6 -4 够	1
2	2 -2 够	1
1	0 -1 不够	0
	结束	

结果：10011110

图 3.1 将 158 转换为二进制数

3) 小数十—二进制转换

规则：从小数点开始，连续减各小数位的位权值，够则减该位取 1，不够则不减该位取 0，直到减为 0 或到要求的位数。

例 3.3 $0.24D = ?B$ ，到小数点后 4 位。

解：计算过程如图 3.2 所示。

二进制位权	计算过程	二进制数各位值
0.5	$\begin{array}{r} 0.24 \\ - 0.5 \end{array}$ 不够	0
0.25	$\begin{array}{r} 0.24 \\ - 0.25 \end{array}$ 不够	0
0.125	$\begin{array}{r} 0.24 \\ - 0.125 \end{array}$ 够	1
0.0625	$\begin{array}{r} 0.115 \\ - 0.0625 \end{array}$ 够	1
0.03125	$\begin{array}{r} 0.0525 \\ - 0.03125 \end{array}$ 够	1
	0.02125 结束，不再计算	

结果：0.00111，进位取 4 位得：0.0100

图 3.2 将 0.24 转换为二进制数

注意：第一个 0 与小数点要照写。

有时，小数十—二进制转换，会出现转换不完的情况。这时可按“舍 0 取 1”（相当于四舍五入）的原则，取到所需的位数。

3. 十六进制编码

二进制数书写太长，难认、难记。为了给程序员提供速记形式，使用中常用十六进制（Hexadecimal）作为二进制的助记符。

十六进制记数符为 0、1、2、3、4、5、6、7、8、9、A(a)、B(b)、C(c)、D(d)、E(e)、F(f)。

将二进制数由小数点起，向两侧分别以每 4 位划一组（最高位与最低位不足 4 位以 0 补），每一组便为一个十六进制数。

例 3.4 $10110\ 1110.1111B = ?H$ 。

解：

0001 0110 1110 . 1111


1 6 E F

所以 $10110\ 1110.1111B = 16E.FH$ 。

4. 符号的 0、1 编码

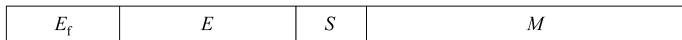
在计算机中不仅要用 0、1 编码的形式表示一个数的数值部分，而且正、负号也要用 0、

MSB=1 表示负数，如-1011 表示为 11011。

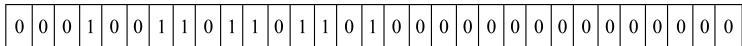
当然，不考虑数的正、负时，不需要用 1 位来表示符号。这种没有符号位的数，称为无符号数。由于符号位要占用 1 位，所以用同样字长，无符号数的最大值比有符号数要大一倍。如字长为 4 位时，能表示的无符号数的最大值为 1111，即 15，而表示的有符号数的最大值为 111，即 7。

$$N_1 = 3.14159 = 0.314159 \times 10^1 = 0.0314159 \times 10^2$$
$$N_2 = 0.011\text{B} = 0.110\text{B} \times 2^{-1} = 0.0011\text{B} \times 2^1$$
$$N=2E \times M$$

E 为数 N 的阶码; M 为数 N 的有效数字, 称为尾数。为了用二进制表示这样形式的实数, 可以把一个字长分为如图 3.3 所示的 4 部分。其中, E_F 、 S 分别称为阶码 E 和尾数 M 的符号位, 它们一般占 1b。使用这种格式的关键是规定好总的字长和每部分的长度。



例如，用 4B 字长，阶码 E 占 4b，-1011.01101 可以表示为图 3.4 所示的样子。其中的尾数 M 部分为纯小数 0.1011 0110 1000 0000 0000 0000。



西文字符中最基本、应用最多的字符有以下一些。

- (1) 26 个小写拉丁字母。
- (2) 26 个大写拉丁字母。
- (3) 约 25 个特殊字符，如[、+、-、@、|、# 等。
- (4) 10 个数码：0、1、2、3、4、5、6、7、8、9。

共计 87 个字符。由于位于 64~128 之间，所以需要 7b 的长度，再用最高位 b_7 作为奇偶校验位，正好是一个字节。

1967 年美国国家标准学会（American National Standards Institute, ANSI）提出一套编码方案作为数据传输的标准码，称为美国信息交换标准码（America Standard Code for Information Interchange, ASCII）。ASCII 后来被国际标准化组织（International Organization for Standardization, ISO）接纳，批准作为国际标准 ISO646。1981 年我国参照 ASCII 码颁布了国家标准《信息处理交换用七单位字符编码》，与 ASCII 码基本相同。ASCII 是一个如表 3.2 所示的 7b 组成的字符集。

表 3.2 ASCII 码（7 位码）字符表

高四位	ASCII非打印控制字符										ASCII 打印字符										
	0000					0001					0010	0011	0100	0101	0110	0111					
	0		1			2		3			4	5	6	7							
低四位	+进制	字符	ctrl	代码	字符解释	+进制	字符	ctrl	代码	字符解释	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL 空	16	▶	^P	DLE 数据链路转意	32		48	0	64	@	80	P	96	`	112	p
0001	1	1	☺	^A	SOH 头标开始	17	◀	^Q	DC1 设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q
0010	2	2	☺	^B	STX 正文开始	18	↕	^R	DC2 设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r
0011	3	3	♥	^C	ETX 正文结束	19	!!	^S	DC3 设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s
0100	4	4	♦	^D	EOF 传输结束	20	¶	^T	DC4 设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t
0101	5	5	♣	^E	ENQ 查询	21	§	^U	NAK 反确认	37	%	53	5	69	E	85	U	101	e	117	u
0110	6	6	♠	^F	ACK 确认	22	■	^V	SYN 同步空闲	38	&	54	6	70	F	86	V	102	f	118	v
0111	7	7	●	^G	BEL 震铃	23	↑	^W	ETB 传输块结束	39	'	55	7	71	G	87	w	103	g	119	w
1000	8	8	◼	^H	BS 退格	24	↑	^X	CAN 取消	40	(	56	8	72	H	88	X	104	h	120	x
1001	9	9	○	^I	TAB 水平制表符	25	↓	^Y	EM 媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y
1010	A	10	◼	^J	LF 换行/新行	26	→	^Z	SUB 替换	42	*	58	:	74	J	90	Z	106	j	122	z
1011	B	11	♂	^K	VT 垂直制表符	27	←	^[	ESC 转意	43	+	59	;	75	K	91	[	107	k	123	{
1100	C	12	♀	^L	FF 换页/新页	28	└	^_	FS 文件分隔符	44	,	60	<	76	L	92	\	108	l	124	
1101	D	13	♪	^M	CR 回车	29	↔	^J	GS 组分隔符	45	-	61	=	77	M	93	]	109	m	125	}
1110	E	14	♪	^N	SO 移出	30	▲	^G	RS 记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~
1111	F	15	☼	^O	SI 移入	31	▼	^-	US 单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ Back space

注：表中的 ASCII 字符可以用：ALT + “小键盘上的数字键” 输入。

在码表中查找一个字符所对应的 ASCII 码的方法是：向上找 $b_6b_5b_4$ ，向左找 $b_3b_2b_1b_0$ 。例如，字母 J 的 ASCII 码中的 $b_6b_5b_4$ 为 100B(4H)， $b_3b_2b_1b_0$ 为 1010B(AH)。因此，J 的 ASCII 码为 1001010B(4AH)。

字母和数字的 ASCII 码的记忆是非常简单的：只要记住了一个字母或数字的 ASCII 码（例如，记住 A 为十进制的 65，0 的 ASCII 码为十进制的 48），知道相应的大小写字母之间差 32，就可以推算出其余字母、数字的 ASCII 码。

由于标准 ASCII 字符集字符数目有限，在实际应用中往往无法满足要求。为此，国际

标准化组织又制定了 ISO 2022 标准，在保持与 ISO 646 兼容的前提下，将 ASCII 字符集扩充为 8b 的统一方法。多出的 128 个 0、1 编码称为扩展 ASCII 码（Extended ASCII），以适用于不同国家和地区使用。这样，就形成两个 ASCII 字符集：基本 ASCII 码（最高位为 0）和扩展 ASCII 码（最高位为 1）。

2. Unicode 字符集

扩展 ASCII 码字符集最多可以提供 256 个字符，虽然足够表示英语，但也仅局限于英语的表示，对于世界大多数文字，还是无能为力。为此，国际标准组织 ISO 于 1984 年 4 月成立 ISO/IEC JTC1/SC2/WG2 工作组，针对各国文字、符号进行统一编码。Unicode（Universal Multiple Octet Coded Character Set）于 1992 年 6 月通过 DIS（Draft International Standard）。Unicode 字符集采用 16b 表示字符，能够表示的字符数为 65 534 个，其中包含符号 6811 个、汉字 20 902 个、韩文拼音 11 172 个、造字区 6400 个，保留 20 249 个。

为了与 ASCII 兼容，Unicode 设计成 ASCII 的超集，即 Unicode 字符集中的前 256 个字符的编码与扩展 ASCII 字符集完全一样。

3. 字符的输出

西文是拼音文字，计算机的键盘就是按照这一特点设计的。在采用 ASCII 的系统中，从键盘上输入一个字符，在计算机内部存储的是一个 ASCII 码值——机内码（在机器内部存储字符的码）。但是，如何再将这个字符输出呢？并且如何由一个 ASCII 码得到某种字体的该字符呢？

实际上，不同字体的字符都是一个图形。存储这个图形的方法是在一个栅格中用方块组成的点阵画出一个字符的图形——称为字模。图 3.5 为字符 A 的字模。它是一个 8×8 的点阵。字模的点阵数越多，字形就越好看，但占用的存储空间越大。如一个 8×8 点阵字模占用的存储空间 8B，一个 16×16 点阵字模占用的存储空间 32B。一般的点阵类型有 16×16、24×24、32×32、48×48 等。除了点阵组成的字模外，还可以采用矢量或曲线画出的字模。

一种字体的所有字符的字模，构成一个字模库。要进行输出某种字体的一个字符，就须驱动该字模库中需要调用的字模的存储地址（或者干脆把某字符对应的 ASCII 码值当作字库的地址），然后控制打印机的针头或显示器的像素（发光点），打印或显示出要求字体的要求字符。

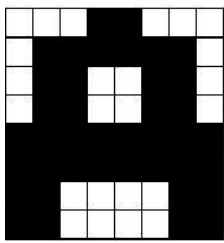


图 3.5 8×8 点阵汉字库组成的字符“A”

3.1.3 汉字的 0、1 编码

1. 几种常用的汉字机内编码方案

汉字是世界上符号最多的文字，历史上流传下来的汉字总数有七八万之多。为了解决汉字的编码问题，人们推出了多种汉字编码方案。下面介绍常用的几种常用的汉字编码方案。

1) GB 2312—1980 和 GB 2312—1990

GB 2312—1980 是 1980 年发布的中文信息处理的中国国家标准, 全称是 GB 2312—1980《信息交换用汉字编码字符集 基本集》, 共收录 6763 个简体汉字、682 个符号, 其中汉字分为两级: 一级字 3755, 以拼音排序; 二级字 3008, 以偏旁排序。

GB 2312—1980 为基本汉字编码, 在中国大陆及海外使用简体中文的地区(如新加坡等)是强制使用的唯一中文编码。1990 年又制定了繁体字的编码标准 GB 12345—1990《信息交换用汉字编码字符集 第一辅助集》, 目的在于规范必须使用繁体字的各种场合, 以及古籍整理等。该标准共收录 6866 个汉字(比 GB 2312 多 103 个字, 纯繁体字大概有 2200 余个)。

GB 2312—1980 和 GB 2312—1980 都是采用双字节编码。为了与系统中基本的 ASCII 字符集区分开, 所有汉字编码的每个字节的第一位都是 1。例如, “啊”字的编码为 0xB0A1。

2) Big5 编码

Big5 编码是目前中国台湾、香港地区普遍使用的一种繁体汉字的编码标准, 包括 440 个符号, 一级汉字 5401 个、二级汉字 7652 个, 共计 13 053 个汉字。Big 5 是一个双字节编码方案, 其第一字节的值在十六进制的 A0~FE 之间, 第二字节在 40~7E 和 A1~FE 之间。因此, 第一字节的最高位是 1, 第二字节的最高位则可能是 1, 也可能是 0。

3) GBK 编码

GBK 全称“汉字内码扩展规范”(俗称大字符集), 是中国大陆地区制定的。GBK 工作小组于 1995 年 12 月完成 GBK 规范。该编码标准兼容 GB 2312, 共收录汉字 21 003 个、符号 883 个, 并提供 1894 个造字码位, 简、繁体字融于一库。

4) Unicode 编码

如前所述, Unicode 字符集包含汉字 20 902 个。

5) GB 18030—2000

2000 年 3 月, 信息产业部和质量技术监督局在北京联合发布了《信息技术和信息交换用汉字编码字符集、基本集的扩充》, 国家标准号为 GB 18030—2000, 收录了 27 484 个汉字, 还收录了藏、蒙、维等主要少数民族的文字, 总编码空间超过 150 万个码位, 为解决人名、地名用字问题提供了方案, 为汉字研究、古籍整理等领域提供了统一的信息平台基础。该标准于 2000 年 12 月 31 日强制执行。GB 18030—2000 作为 GBK for Unicode 3.0 的更新而诞生, 并且作为 GB 2312—1980《信息交换用汉字编码字符集基本集》的扩展, 向下兼容 GBK 和 GB 2312—1980 标准。

GB 18030 是 1-2-4 字节变长编码。

单字节, 其值从 0 到 0x7F。

双字节, 第一个字节的值从 0x81 到 0xFE, 第二个字节的值从 0x40 到 0xFE (不包括 0x7F)。

四字节，第一个字节的值从 0x81 到 0xFE，第二个字节的值从 0x30 到 0x39，第三个字节的值从 0x81 到 0xFE，第四个字节的值从 0x30 到 0x39。
这几种编码都是机内码。

2. 汉字的输入——汉字外码

现在使用的计算机键盘是根据英文设计的，而汉字形状复杂，没有确切的读音信息，且一字多音、一音多字，要像输入西文字符那样在现有键盘上利用机内码进行输入非常困难。为此，不得不设计专门用来进行输入的汉字编码——汉字外码。常见的输入法有以下几类。

- (1) 按汉字的排列顺序形成的编码（流水码），如区位码。
- (2) 按汉字的读音形成的编码（音码），如全拼、简拼、双拼等。
- (3) 按汉字的字形形成的编码（形码），如五笔字型、郑码等。
- (4) 按汉字的音、形结合形成的编码（音形码），如自然码、智能 ABC。

输入码在计算机中必须转换成机内码，才能进行存储和处理。

3. 汉字的字模

汉字字模的原理与西文字符的字模基本相同，只是由于汉字笔画复杂，为了表达清晰，要比西文字符需要更多的点阵。图 3.6 为一个 24×24 点阵的汉字字模示例。

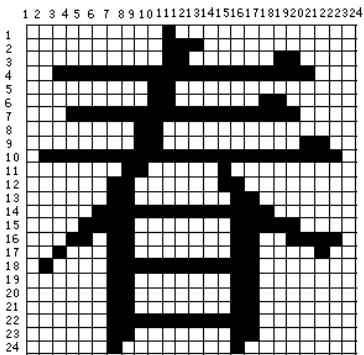


图 3.6 一个 24×24 点阵的汉字

4. 汉字系统的工作过程

图 3.7 描画了一个计算机上从汉字的输入到输出（显示）的过程。

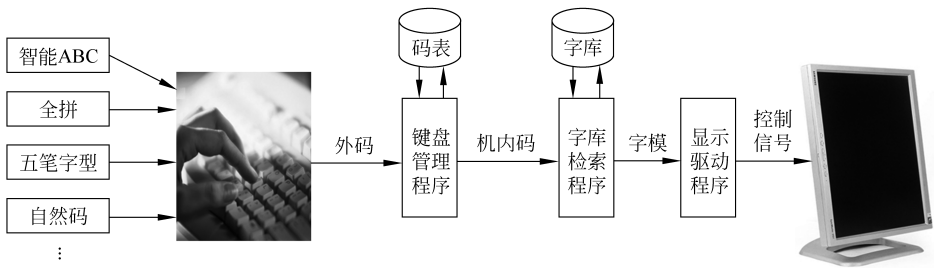


图 3.7 汉字系统工作过程

- (1) 用一种输入方法从键盘输入汉字。
- (2) 键盘管理程序按照码表将外码变换成机内码。
- (3) 机内码经字库模检索程序查对应的点阵信息在字模库的地址，从字库取出字模。
- (4) 字模送显示驱动程序，产生显示控制信号。

(5) 显示器按照字模点阵将汉字字形在屏幕上显示出来。

3.1.4 文本压缩

在数据处理中，存储与传输是两个重要的环节。而存储与传输的效率除了与技术有关外，还与数据量有关。那么，能不能把数据加以压缩呢？经过分析研究，人们找到了一些数据压缩的方法。下面介绍最容易理解的 3 种文本压缩方法。

1. 关键字编码

一种语言中的单词，在文本中出现的频度是不同的。例如，在英文中，出现频度高的单词有 the、and、which、that、what、must、these 等。关键字编码（Keyword Encoding）方法的基本思想如下。

(1) 把频率高的单词用简单的字符表示，这些字符很少出现在一般文本中。表 3.3 为几个单词的替代示例。

表 3.3 几个单词的替代示例

单 词	as	the	and	that	must	well	these
替代符	^	~	+	\$	&	%	#

(2) 不对单个字符的单词进行替代，如不对 a、I 等进行替代。

(3) 对大小写意义不同的单词，如 the 和 The 分别进行编码。

在这个基础上，还可以考虑对单词的一部分进行替代，例如，对常用前缀和后缀 ex、ing、tion 等进行替代。

关键字编码比较简单，但压缩率不高，并且容易产生二义性，如一个\$符号是单词 that，还是真正的美元符号等。

2. 行长度编码

行长度编码（RLE）是将重复字符序列用“一个标志+字符+重复次数”表示，例如，将 The aaaaaabbbbbbbKKKKKKKK OK 表示为 The *a6*b8*K9 OK。

3. 哈夫曼编码

哈夫曼编码是以其创建者 David Huffman 的名字命名的编码，它有两个重要性质。

(1) 对于字符集中的字符采用不等长编码，使用频度高的字符编码短，使用频度低的字符编码长。表 3.4 为一些字符的哈夫曼编码。

表 3.4 一些字符的哈夫曼编码

字 符	A	E	L	O	R	B	D
哈夫曼编码	00	01	100	110	111	1010	1011

例如，单词 BOARD 可以压缩为

1010110001111011



(2) 一个字符的位串不会是另一个字符的位串的前缀。这样,在解码时只需从左到右进行扫描,当发现一个位串对应于一个字符时,就一定是这个字符,而不会是另外一个字符的前几位。

哈夫曼编码的压缩率比较高。

3.2 电子数字计算机技术基础

电子数字计算机是当前广泛使用的计算机工具。“电子”指其基本元件是电子器件,“数字”指这些电子元件工作在二元——0 或 1 状态下。

3.2.1 Neumann 计算机体系组成

1. 计算工具的进步轨迹

计算始于人类社会的管理需要,也随着人类社会的发展而优先进步。从穴石记事、结绳记事,到算筹、算盘,再到 Babbage 分析机,为了提高计算的效率和精度,放大自己的计算能力和减轻自己的劳动强度方面,人类在开发计算工具上进行了不懈的努力。这些努力主要表现在如下两个方面。

1) 程序控制方式的不断进步

公元前五世纪前中华先祖发明的算盘和算筹,开启了用程序扩展硬件能力的先河。古代著名科学家祖冲之曾用此方法将圆周率计算到了小数点后 7 位。

西汉年间中国纺织工匠陈宝光及其妻子发明的提花机,开启了内程序控制的先例。受此影响,法国机械师 Jacquard 发明了自动提花机,并启发了 Babbage 制作分析机的灵感。1822 年他制作成了第一台用卡片存储程序的分析机,并提出计算机应当由存储(仓库——存储数据,穿孔卡片——存储程序)、计算(作坊)、输入(穿孔卡片)和输出(印刷厂)等几部分组成。

2) 计算工具动力引擎的进步

1642 年,法国数学家 Pascal 在他 19 岁那年发明了用发条作为动力的加法器,开启了计算机的内动力时代。

1937—1938 年间,美国贝尔实验室工程师 Stibitz 和德国工程师 Zuse 分别制造出了继电器式计算机 Model-K 和 Z-2,开启了用电作为动力的电气计算机时代。

1942 年,美国爱荷华州立大学的 Atanasoff 教授制作成功了用真空管作为元件的计算机 ABC,开启了计算机的电子纪元。

1955 年,贝尔实验室研制成功了第一台全晶体管计算机 TRADIC,开启了计算机的微电子时代。

1970 年,美国 IBM 公司采用大规模集成电路的大型计算机 IBM 370 投入市场,标志着计算机进入了大规模集成电路时代。

从上述计算机的发展过程中可以看出,它是沿着两条轨迹前进的:一条是程序外部控制进入了内部控制,另一条是合适的动力引擎的进步——从人力驱动进步到电气驱动。

2. Neumann 体系结构的确立

1) Neumann 体系结构

1943 年,著名的美籍匈牙利科学家 Neumann 总结了电子计算机发展的规律和理论,提出了如下思想。

(1) 电子数字计算机要采用二进制,不但数据采用二进制,指令也采用二进制。这样,指令和数据便可一起存储,以简化计算机的结构,提高计算机的速度。所以 IAS 计算机虽然总共采用了 2300 个电子管,运算速度却比拥有 18 000 个电子管的 ENIAC (采用十进制)提高了 10 倍。

(2) 自动计算机要采用内程序存储控制工作方式,即计算机要能将程序存储起来,并能够用所存储的程序控制运算过程,才能自动工作。为此,自动计算机应由计算器、逻辑控制装置、存储器、输入和输出 5 大部分组成。

Neumann 1945 年 6 月提出的 EDVAC 方案到 1952 年才研制成功。在这个期间内,远在大西洋对岸的英国人莫里斯·威尔克斯 (Maurice V. Wilkes) 捷足先登,在 EDVIC 方案的基础上,于 1949 年 5 月 6 日研制成功了世界第一台存储程序计算机——EDSAC (Electronic Data Storage Automatic Computer)。

2) Neumann 体系计算机的基本工作过程

直到今天,商业化的计算机还基本上采用的是 Neumann 体系。这种计算机的工作过程大致如下。

(1) 为了让计算机进行某种运算,首先要为它设计相应的程序再进行计算。程序由一条一条的指令组成。每条指令都是计算机可以执行的基本操作。

(2) 用输入设备,将程序存入到存储器中。

(3) 运算开始,机器自动把程序的第一条指令送到控制器中进行分析,再根据分析的结果,向有关部件发送控制信号,完成该指令规定的操作。

(4) 一条指令取出后,计算机就会准备取出程序中的下一条指令,如此重复,直到执行完程序中的最后一条指令,一个程序规定的任务就完成。

(5) 计算过程中,需要输出时,要安排相应的指令进行输出。

3.2.2 电子数字计算机存储原理

计算机的存储器是用来存储数据和程序的部件,它有如下一些特点。

1. 基于地址的数据存取

计算机的存储器就像中药铺中的药盒——称为存储单元，密密麻麻地排在一起，要往里放数据或指令或从中取出数据或指令的方法是预先编号，按照号码进行。这些号码就称为存储单元的地址，如图 3.8 所示。

2. 所存储的内容“取之不尽，新来旧去”

现在的存储器都是由电或磁元件做成。其存储原理就像磁带一样，存进之后，无论怎样使用（读），都不会消失；但只要存进（写入）新的内容，旧的内容就不复存在。

3. 分级存储

存储器的主要指标是速度和容量。当然人们都希望在一台计算机中配置很大（单元多）、速度很快的存储器。但是，速度快的存储器价格就大，对经济实力的要求就高。为此，计算机的存储器实行分级存储的方式：把要反复使用的内容放在速度最高、容量较小的存储器中，把马上不用但不久就要使用的内容放在速度次高、容量次大的存储器中……把最不常用的内容放在容量极大但速度不高的存储器中。合理地在不同级别的存储器之间进行存储内容的调换，就解决了速度、容量和成本之间的矛盾。就像学生在书包中装的是马上上课要用的书籍，家里书架上放的是一个阶段要用的书籍，还要用书店或图书馆作为后盾一样。

目前的计算机存储器一般分为 3 级：辅助存储器（也称为外存，如光盘、磁盘、U 盘等）、主存储器（也称为内存）和高速缓冲寄存器（Cache，简称缓存）。它们之间的关系如图 3.9 所示。辅助存储器作为主存储器的后援；主存储器可以与运算器和控制器（合起来称为 CPU）通信，也可以作为 Cache 的后援；Cache 存储 CPU 最常使用的信息。

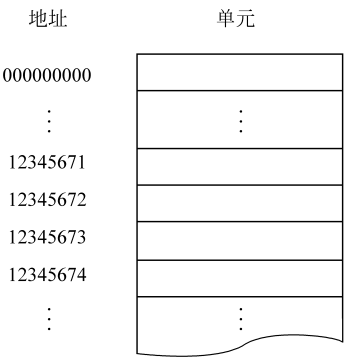


图 3.8 存储单元及其地址

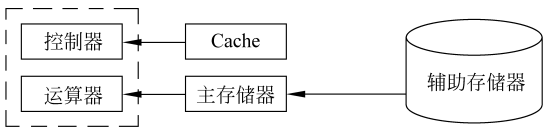


图 3.9 主存储器、辅助存储器和 Cache 及其它它们之间的关系

3.2.3 计算机的运算器和寄存器

运算器是计算机中负责对数据进行运算的部件。其核心部件是算术逻辑单元（ALU）。ALU 能进行的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作。除了 ALU 之外，运算器中包含累加器、状态寄存器、通用寄

寄存器组等，如图 3.10 所示。

从存储器中取出的数据，先存放到寄存器中，要等操作信号到来才开始交 ALU 运算。随着计算机运行速度的提高和功能的增强，寄存器的数量也在增加。

累加器是一种特殊的、也是非常重要的一个寄存器，用于存放运算的中间结果，以节省将中间结果存入内存，便立即取出的时间开锁。

状态寄存器主要反映某次运算是否产生了进位或借位。

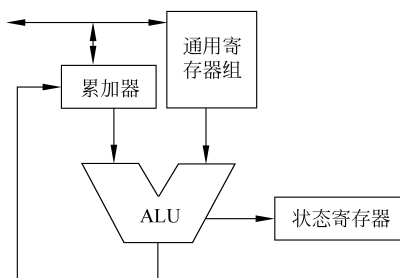


图 3.10 运算器的结构

3.2.4 计算机控制器的工作原理

1. 控制器的功能结构

如前所述，控制器是计算机的中枢神经，可以对整个计算机的工作过程进行控制。这里，其控制功能主要表现在如下 3 个方面。

(1) 定序。组成程序的指令必须按照一定的顺序被执行，不能乱套。就像做菜、开会一样，必须按照一定顺序进行，否则做的菜就无法吃，会议开不好。

(2) 定时。电子计算机是一种复杂的机器，由众多的元件、部件组成，不同的信号经过的路径也不同。为了让这些元件、部件能协调工作，系统必须有一个统一的时间标准——时钟和节拍，就像乐队的每位演奏者都必须按照指挥节拍演奏一样。计算机中的时钟和节拍是由一种振荡器提供的。振荡器的工作频率称为时钟频率。显然，时钟频率越高，计算机工作节拍越快。

(3) 产生操作控制信号。控制器应能按指令规定的内容，在规定的节拍向有关部件发出操作控制信号。

控制器的上述功能分别由指令部件、时序部件和操作控制部件承担，如图 3.11 所示。

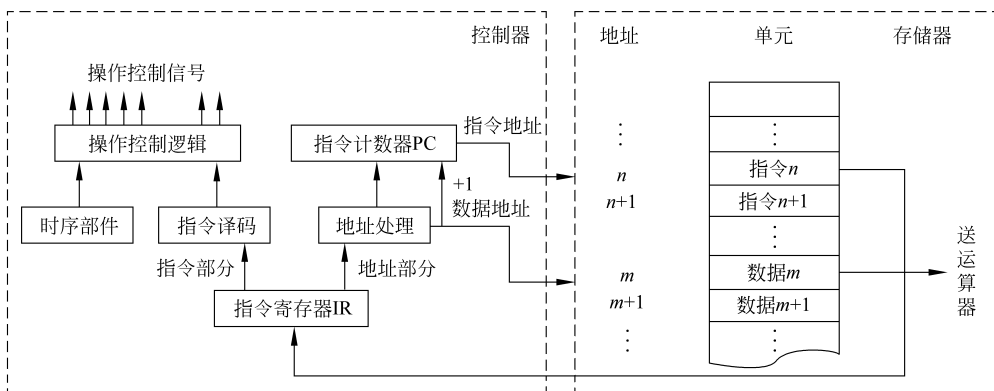


图 3.11 控制器的工作原理图

2. 控制器的工作过程

简单地说，控制器的工作过程就是反复地“取指令→分析指令→发送操作控制信号”的过程。

(1) 取指令。指令是指示计算机完成一种操作的代码，一个程序由一系列的代码组成。当一个程序要被执行时，这个程序中的指令就被存放到计算机内存的一个空间中。这个空间的开始地址，就会放入到控制器的程序计数器 PC (Program Counter) 中。程序执行开始，就将这个地址送到存储器的地址译码器中，驱动一个存储单元，取出其中的指令，送到控制器的指令寄存器 IR (Instruction Register) 中。同时，PC 会自动加 1，准备取下一条指令。

(2) 分析指令。一条指令由两部分组成：一部分称为操作码 OP (Operation Code)，指出该指令要进行什么操作；另一部分称为数据地址码，用于指出要对存放在哪个地址中的数据进行操作（如 *m*）。在分析指令阶段，要将数据地址码送到存储器中取出需要的操作数到运算器，同时把 OP 送到指令译码部件，翻译成要对哪些部件、进行哪些操作的信号。

(3) 发送操作控制信号。将有关操作控制信号按照时序安排发送到相关部件，使有关部件在规定的节拍中完成规定的操作。

3. 一个程序的执行过程

下面介绍一个假想程序的执行过程。这是一个求 $x+|y|$ 的程序。

1) 为程序分配存储单元

根据存储器的使用情况，给程序和数据分配合适的存储单元。本例假定程序从 2000 单元开始存储，数据存于 2010 单元和 2011 单元，结果存于 2012 单元，具体安排如下：

单元地址	单元内容	注 释
2000	MOV A, (2010)	;取 2010 单元中的数据到 A
2001	MOV B, (2011)	;取 2011 单元中的数据到 B
2002	JP B < 0, 2005	;若 B < 0, 转 2005 单元，否则执行下条指令
2003	ADD A, B	;将 A 与 B 中内容相加后送入 A
2004	JP + 2	;转 2006
2005	SUB A, B	;将 A 与 B 中内容相减后送入 A
2006	MOV(2012), A	;存 A 中内容到 2012 单元
2007	OUT(2012)	;输出 2012 单元内容
2008	HALT	;停机
2010	x	
2011	y	

2) 执行程序的过程

(1) 向程序计数器中送入程序首地址 2000，然后启动计算机，控制器即从 2000 单元开始执行程序。

(2) PC 内容 (2000) 送入存储驱动部件，从 2000 单元取出第一条指令“MOV A, (2010)”，送至指令寄存器 IR，同时 PC 加 1 (得 2001)，准备取下条指令。

指令“MOV A, (2010)”把 2010 单元中的数据 (x) 送到寄存器 A (CPU 内部暂存数据的元件)。

(3) PC 内容 (2001) 送入存储驱动部件, 从 2001 单元取出第一条指令“MOV B, (2011)”, 送至指令寄存器 IR, 同时 PC 加 1 (得 2002), 准备取下条指令。

指令“MOV A, (2011)”把 2011 单元中的数据 (y) 送到寄存器 B (CPU 内部的另一个暂存数据的元件)。

(4) PC 内容 (2002) 送入存储驱动部件, 从 2002 单元取出第一条指令“JP B < 0, 2005”, 送至指令寄存器 IR, 同时 PC 加 1 (得 2003), 准备取下条指令。

指令“JP B < 0, 2005”首先进行判断: B 中的数据小于 0 时, 就跳到 2005, 即往 PC 中送入地址 2005。

2005 单元中的指令为“SUB A, B”, 执行 $x-y$ 的操作, 即加了 B 中数据的负值 (绝对值): $x+(-y)$, 并把结果存在 A 中。同时 PC 加 1 (得 2006), 准备取下条指令。

(5) 若 B 中的值为正, 就不跳转, 执行 2003 中的内容“ADD A, B”, 即把 A 和 B 中的数据相加, 得 $x+y$, 送入 A 中。同时 PC 加 1 (得到 2004), 准备取下条指令。

所以, A 中存放的是 A 与 B 的绝对值的和。

(6) 不管 B 中数据为正, 还是为负, 运算后接着执行 2004 中的指令: JP + 2 是跳过两个单元。这个操作是把 PC 中的内容再加 2, 得到 2006。于是再取出 2006 中的指令。

(7) 2006 中的指令“MOV(2012), A”是把 A 中的内容 (保存着 $x+y$) 送到 2012 单元。同时 PC 加 1 (得 2007), 准备取下条指令。

(8) 2007 中的指令“OUT (2012)”是把 2012 中保存着的数据 ($x+y$) 输出。同时 PC 加 1 (得 2008), 准备取下条指令。

(9) 2008 中的指令 HALT, 要求停机。到此, 一个程序执行结束。

3.2.5 电子数字计算机的标准输入输出

现代电子数字计算机的标准输入输出设备是键盘、鼠标、显示器和触摸屏。

1. 键盘

1) 键盘及其类型

字符输入设备的实质是将要输入的字符转换成相应的 0、1 码。目前, 键盘是最重要的字符输入设备。

按照工作的物理性质键盘一般可分为如下 4 种。

(1) 触点式键盘。借助于金属把两个触点接通或断开以输入信号。

(2) 无触点式键盘。借助于霍尔效应开关 (利用磁场变化) 和电容开关 (利用电流和电压变化) 产生输入信号。

(3) 雷射式 (镭射激光) 虚拟键盘。在任意平面上投影出全尺寸的计算机键盘, 当手指头按下投影片键盘时, 会阻断该位置的红外线, 造成反射通过感知器就接收到反射的坐标, 由此得知按下的是什么键。

(4) 软键盘与虚拟键盘。稍后介绍。

2) 键盘的工作原理

键盘的基本组成元件是按键开关。如图 3.12 所示, 这些开关在线路板上排成行、列矩阵格式, 用硬件或软件对行、列分别扫描, 就可以确定被按下的键的位置。

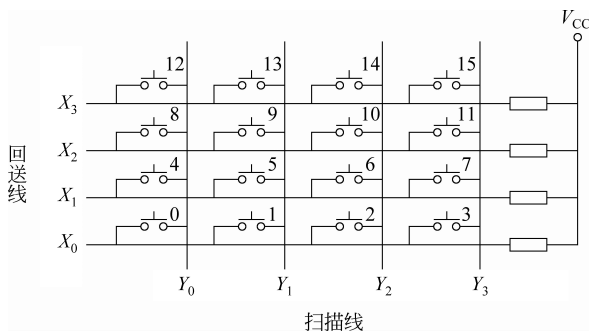


图 3.12 键盘组成的开关矩阵

在对键盘位置进行扫描的过程中所产生的用于确定按键位置的码, 称为键盘的扫描码。有了键盘扫描码后, 键盘处理器用其与只读存储器 (ROM) 内的字符映射表进行比对, 就可以得到相应的内码。例如, 字符映射表会告诉处理器单独按下 **a** 键得到的扫描码对应于小写字母 **a**, 而同时按下 **Shift** 键和 **a** 键得到的扫描码对应于大写字母 **A**。可以使用不同的字符映射表取代键盘中原来使用的映射表。不同的语言输入法有不同的字符映射表。

当按下一个键时, 键盘内的处理器会对键矩阵进行分析, 并将确定的字符保存在自己的缓冲区内, 然后才发送这些数据。

因此, 一个键盘要由下列部件组成。

- ① 开关矩阵。
- ② 键盘处理器。
- ③ 字符映射表 (ROM)。
- ④ 键盘缓冲区。

因按键时会使键产生机械抖动, 为防止因此造成的错误判断, 在键盘控制电路中都含有硬的或软的消除抖动影响的机制。

3) 键盘上的按键类型

键盘是在打字机 (Typewriter) 的基础上发展而来, 其按键数曾出现过 83 键、87 键、93 键、96 键、101 键、102 键、104 键、107 键等。104 键的键盘是在 101 键键盘的基础上为 Windows 9x 平台增加了 3 个快捷键 (有两个是重复的), 所以也被称为 Windows 9X 键盘。

不管键盘形式如何变化, 基本的按键排列还是保持基本不变, 可以分为主键盘区、Num 数字辅助 (Num) 键盘区、键功能 (F) 键盘区、控制键区, 对于多功能键盘还增添了快捷键区。

主键盘区包括字母表的各个字母的文字字符键。这些字符键在键盘上的排列方式, 称

为键盘布局。目前应用最广的是 QWERTY 键盘。它由打字机之父——美国人克里斯托夫·拉森·肖尔斯（Christopher Latham Sholes）于 1868 年发明。如图 3.13 所示，这种布局的特点是将最常用的几个字母安置在相反方向。这样可以以最大限度放慢敲键速度以避免卡键，因为当时的打字机是全机械结构，打字过快就常出现卡键现象。

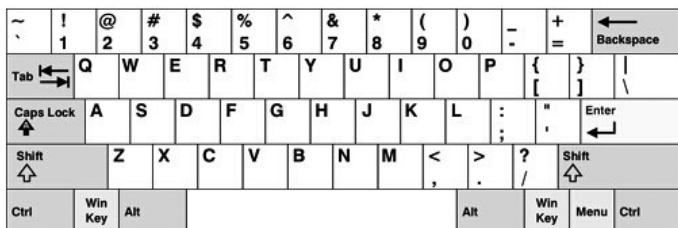


图 3.13 QWERTY 键盘

20 世纪 30 年代，由于盲打的需要，华盛顿一个叫德沃拉克（Dvorak）的人为使左右手能交替击打更多的单词又发明了一种新的排列方法——DVORAK 键盘。后来理连·莫尔特（Lillian Malt）又发明了比 DVORAK 键盘更加合理、高效的 MALT 键盘。但是，由于 QWERTY 键盘先入为主，还一直牢牢地占据着键盘市场。

打字机上的数字键原来是布置在键盘最上方的。计算机键盘最初也是这样一种布局。后来，由于随着计算机在商务环境中的应用日益增加，为了能快速录入数据并进行简单计算，开始将这些数字键组织成一个相对独立区间或制成一个独立小键盘。现在，数字小键盘上的键数一般为图 3.14 所示的 17 个键，是在 10 个数字键上添加了四则运算符以及 Enter、Delz 和 NumLock 而成，并采用加法机和计算器上的布置。

1986 年，IBM 公司对基本键盘进行了扩展，增加了功能键和控制键。应用程序和操作系统可以向功能键指定特定的命令，控制键还可以提供光标和屏幕控制。4 个箭头键呈倒 T 形分布在输入键和数字小键盘的中间，用来在屏幕上小幅移动光标。

常规键盘还具有 CapsLock（字母大小写锁定）、NumLock（数字小键盘锁定）、ScrollLock（滚动锁定键），还有 3 个指示灯（部分无线键盘已经省略这 3 个指示灯），标志键盘的当前状态。

其他常用控制键包括 Home、End、Insert、Delete、PageUp、PageDown、Ctrl、Alt、Esc。

4) 软键盘

软键盘（Soft Keyboard）是通过软件显示在“屏幕”上的模拟键盘。这种键盘只能用鼠标点击输入字符。

软键盘盘面有固定和随机两种布局形式。固定布局一般用于便携智能设备，如手机、平板电脑，可以省去携带物理键盘。随机布局软键盘常用于银行的客户端上要求输入账号和密码的地方。如图 3.15 所示，软键盘是随机生成的，每次键盘上数字的顺序都不同，除非使用快速截取屏幕或者监听网络数据包的方法，否则很难记录输入的字符，可以防止木马记录键盘输入的密码。



图 3.14 数字小键盘



图 3.15 某银行的客户端登录软键盘

5) 虚拟激光键盘

图 3.16 (a) 是一个正在操作虚拟激光键盘 (Virtual Laser Keyboard, VLK) 键盘时的情形。虚拟激光键盘也称为虚拟投影键盘，简称激光键盘或虚拟键盘，它是由光投影所形成的影像键盘。几乎能在任意平面上投影出全尺寸的影像键盘，并且在不使用时会完全消失。

如图 3.16 (b) 所示，激光投影键盘系统主要由 3 个部分组成。

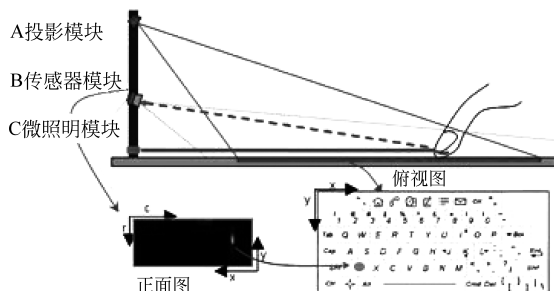
(1) 投影模块 A：该模块由经过特殊设计的高效全息光学原件照明产生，元件带有红色二极管激光器。

(2) 传感器模块 B：内含订制的硬件，能够实时确定反射光的位置。

(3) 微照明模块 C：可以产生与界面表面平行红外线光照平面。光线照在表面上几毫米处，用户是无法看到的。



(a) 正在操作的虚拟激光键盘



(b) 激光投影键盘的工作原理

图 3.16 激光投射键盘

虚拟键盘的适用性技术对用户手指运动加以研究，对键盘击打动作进行解码和记录。当手指敲击时，按键边上就会对 C 所发出的红外线产生反射，传到传感器模块 B 上。传感器芯片可以精确感知手指的动作和所敲击的位置。

2. 鼠标

鼠标器是计算机显示系统纵横坐标定位的指示器，如图 3.17 所示，因形似老鼠而得名“鼠标”（Mouse）。

1) 鼠标的接口类型

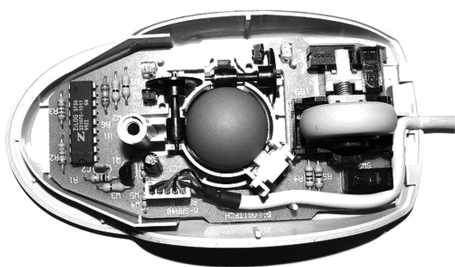
鼠标按接口类型可分为串行鼠标、PS/2 鼠标、总线鼠标、USB 鼠标（多为光电鼠标）4 种。串行鼠标是通过串行口与计算机相连，有 9 针接口和 25 针接口两种；PS/2 鼠标通过一个六针微型 DIN 接口与计算机相连，它与键盘的接口非常相似，使用时注意区分；总线鼠标的接口在总线接口卡上；USB 鼠标通过一个 USB 接口，直接插在计算机的 USB 口上。



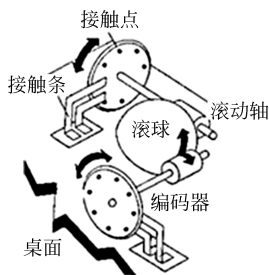
2) 鼠标的工作原理和内部结构

鼠标按其工作原理及其内部结构的不同可以分为机械鼠标、光机鼠标、光电鼠标和光学鼠标。

（1）机械鼠标。如图 3.18 所示，机械鼠标主要由滚球、辊柱和光栅信号传感器组成。当拖动鼠标时，带动滚球转动，滚球又带动辊柱转动，装在辊柱端部的光栅信号传感器产生的光电脉冲信号反映出鼠标器在垂直和水平方向的位移变化，再通过计算机程序的处理和转换来控制屏幕上光标箭头的移动。



(a) 机械鼠标内部结构



(b) 机械鼠标的结构原理

图 3.18 机械鼠标器内部结构

这种机械鼠标的底部有一个可四向滚动的胶质小球。这个小球在滚动时会带动一对转轴转动（分别为 X 转轴、 Y 转轴），在转轴的末端有一个圆形的译码轮，译码轮上附有金属导电片与电刷直接接触。当转轴转动时，这些金属导电片与电刷就会依次接触，出现“接通”或“断开”两种形态，前者对应二进制数 1、后者对应二进制数 0。接下来，这些二进制信号被送交鼠标内部的专用芯片进行解析处理并产生对应的坐标变化信号。只要鼠标在平面上移动，小球就会带动转轴转动，进而使译码轮的通断情况发生变化，产生一组组不同的坐标偏移量，反映到屏幕上就是光标可随着鼠标的移动而移动。

（2）光机鼠标。纯机械式鼠容易磨损、标精度不高。如图 3.19 所示，光机鼠标是一种

光电和机械相结合的鼠标。它在机械鼠标的基础上，将磨损最厉害的接触式电刷和译码轮改为非接触式的 LED 对射光路元件。当小球滚动时，X、Y 方向的滚轴带动码盘旋转。安装在码盘两侧有两组发光二极管和光敏三极管，LED 发出的光束有时照射到光敏三极管上，有时则被阻断，从而产生两级组相位相差 90°的脉冲序列。脉冲的个数代表鼠标的位移量，而相位表示鼠标运动的方向。由于采用了非接触部件，降低了磨损率，从而大大提高了鼠标的寿命并使鼠标的精度有所增加。光机鼠标的外形与机械鼠标没有区别，不打开鼠标的外壳很难分辨。

(3) 光电鼠标。如图 3.20 所示，这种光电鼠标没有传统的滚球、转轴等设计，其主要部件为两个发光二极管、感光芯片、控制芯片和一个带有网格的反射板（相当于专用的鼠标垫）。工作时光电鼠标必须在反射板上移动，X 发光二极管和 Y 发光二极管会分别发射出光线照射在反射板上，接着光线会被反射板反射回去，经过镜头组件传递后照射在感光芯片上。感光芯片将光信号转变为对应的数字信号后将之送到定位芯片中专门处理，进而产生 X-Y 坐标偏移数据。

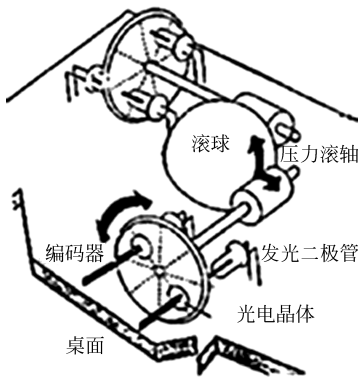


图 3.19 光机鼠标的内部结构

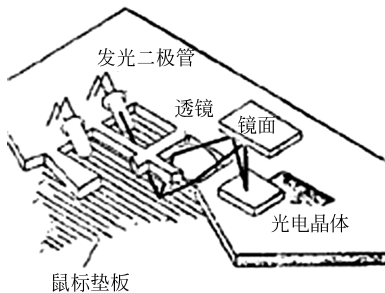


图 3.20 光电鼠标的内部结构

(4) 光学鼠标。光学鼠标的结构与上述所有产品都有很大的差异，它的底部没有滚轮，也不需要借助反射板来实现定位，其核心部件是发光二极管、微型摄像头、光学引擎和控制芯片。工作时发光二极管发射光线照亮鼠标底部的表面，同时微型摄像头以一定的时间间隔不断地进行图像拍摄。鼠标在移动过程中产生的不同图像传送给光学引擎进行数字化处理，最后再由光学引擎中的定位 DSP 芯片对所产生的图像数字矩阵进行分析。由于相邻的两幅图像总会存在相同的特征，通过对比这些特征点的位置变化信息，便可以判断出鼠标的移动方向与距离，这个分析结果最终被转换为坐标偏移量实现光标的定位。

3. 显示器

显示设备是将各种电信号变为视觉信号的一种设备，是目前计算机给人传送信息的有效设备之一。

1) 显示器的基本工作原理

计算机系统显示设备种类很多。早先显示器主要采用的阴极射线管（Cathode Ray