Xidian University

计算机导论与程序设计

1 计算机体系结构及其编码方式-2

讲授内容

- 图灵机原理-什么是计算？
- 二进制的产生和运算
- 数的表示：数据的编码的基本原理

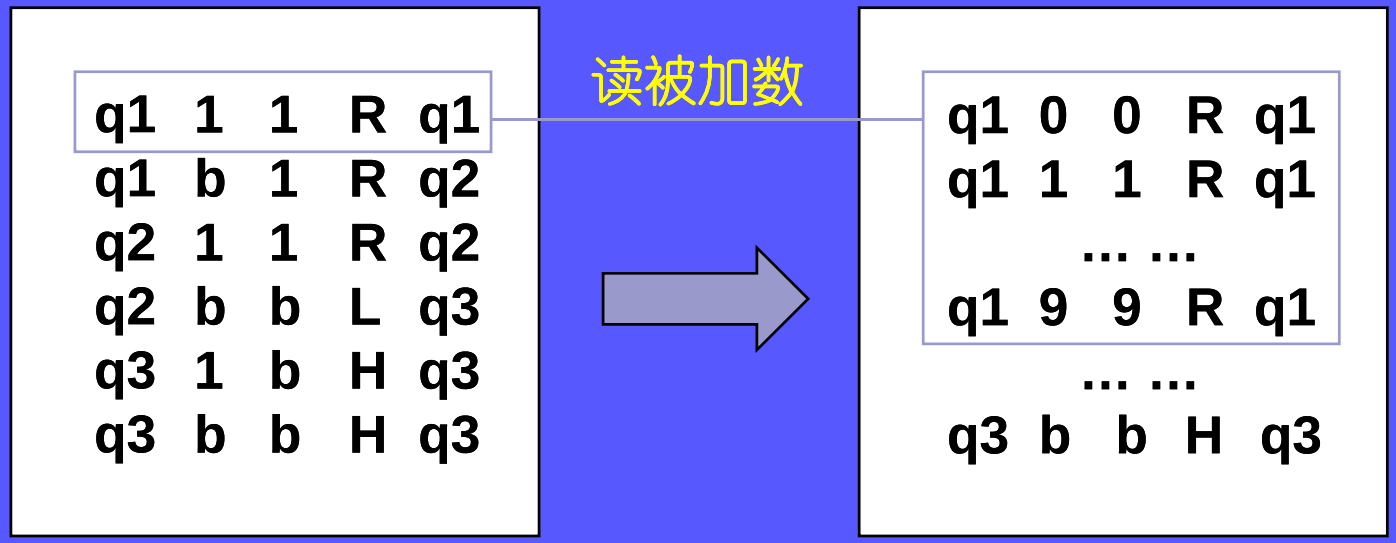
二进制(0和1)

对图灵机的进一步分析

- 在图灵机的计算例子中，字母表是 { 1, b }，其中符号 b（空白（Blank），在计算机领域中通常叫空格）的作用是什么？
  - 它是被加数、加数、和的边界符号，表示着计算对象和计算结果的边界。
- 因此，真正用来表示被加数、加数与和的数值的，只有符号 1。
  - 那么，数值 1,000,000 就应当由一百万个 1 来表示。读入这一个数，读写头就要移动一百万次。
  - 在设计真正的计算机时，这显然是不合理、不实际的。

对图灵机的进一步分析

- 如果这个字母表中一共有11个符号： { 0, 1, …, 9, b }，那么就可以用十进制来表示数值。
  - 但是，这时的程序要长得多。
  - 确定当前指令自然要花更多的时间。



怎样用机器表示带上的符号？

- 字母表中的符号越多，用电子器件表示的困难一般就越大，精确性越低。
- 一个电子器件的状态越多，可靠运行的困难通常就越大。
- 与两个状态的电子元件相比（0，1电平），有三个状态的电子元件在制造上比较困难，可靠性也比较低。

采用二进制使电子器件制造计算机成为可能！

怎样用机器表示带上的符号？

- 制作控制器的基本要求
  - 控制器必须有逻辑判断能力。例如要执行下面的指令：

    q1　1　1　R　q1
  - 控制器要作出的逻辑判断是：如果当前机器状态是q1，且读入的符号是1，则 … …

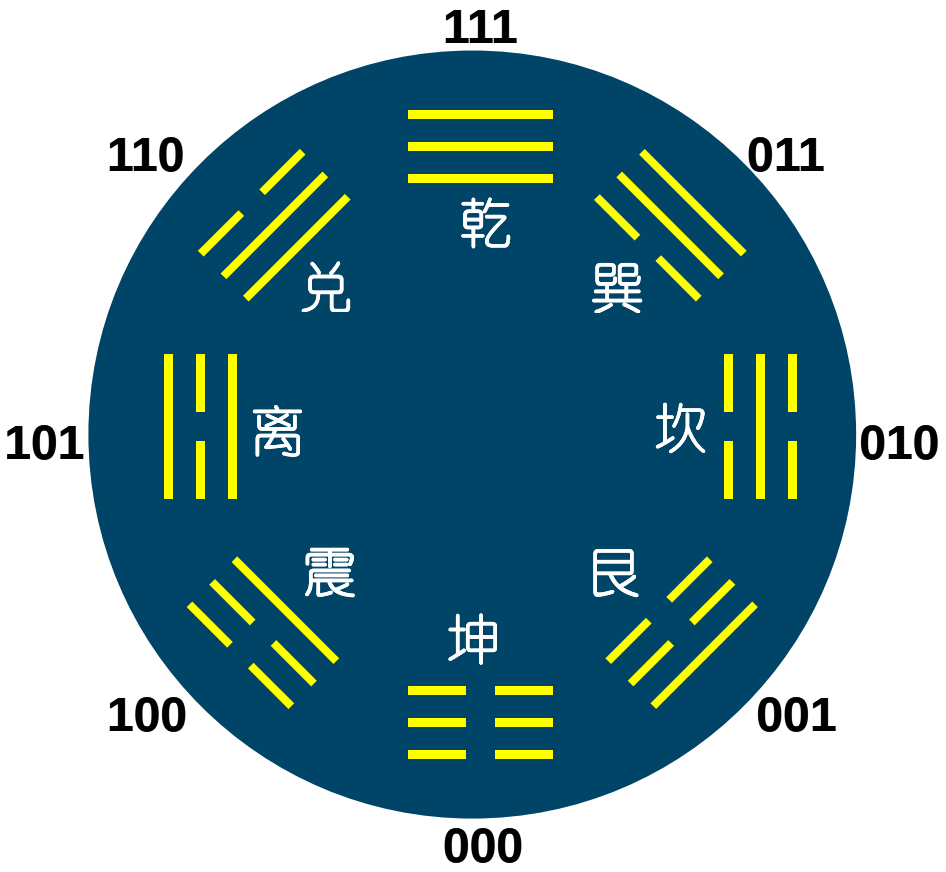
- 用机器来实现控制器，必须要考虑怎样来实现逻辑判断
  - 对于通用图灵机，程序要放在带上。还要考虑采用字母表中的符号来表示程序，而且这样的表示应当易于控制器来处理和判断（也就是：解释）。
- “真、假”（true / false）判断就是最基本的逻辑判断。

二进制的由来

- 综合以上考虑，计算机是用两个符号（0 和 1）来：
  - 表示数据；
  - 表示程序；
  - 按照以此为基础的“布尔代数”（或“二值逻辑”），来设计和制造计算机中的大多数零件（元器件）。（在以后的《数字系统设计基础》、《计算机组成原理》等课程中，同学们将学习相应的基本原理和设计方法）
- 这样的表示方法叫做二进制（Binary）表示。
  - 最早的二进制表示来自我国的《周易》。

二进制的由来

- 综合以上考虑，计算机是用两个符号（0 和 1）来：
  - 表示数据
  - 表示程序
  - 按照以此
    和制造计
    系统设计
    习相应的
- 这样的表示
  - 最早的二



），来设计
后的《数字
同学们将学

回顾：用十进制来表示数值的规则

- 一个数的十进制表示记为：

  $d_n d_{n-1} \ldots d_1 d_0$ $_{(10)}$

  其中：

  $d_n \in \{ 1, \ldots, 9 \}, n>0$ (多位数) **123, 0123**

  $d_n \in \{ 0, \ldots, 9 \}, n=0$ (一位数) **1,2, 3, 0**

  $d_{n-1}, \ldots, d_1, d_0 \in \{ 0, \ldots, 9 \}$ **123, 3040**

  则该数的值可用下式得出：

  $$d_n \times 10^n + d_{n-1} \times 10^{n-1} + \ldots + d_1 \times 10^1 + d_0 \times 10^0$$

- 例如，$123_{(10)}$ 的值是：

  $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

  $= 100+20+3$

  $= 123$

得出：用二进制来表示数值的规则

- 一个数的二进制表示记为：

  $b_n b_{n-1} \ldots b_1 b_0$ (2)

  其中：

  $b_n = 1, n>0$ 110, 0110

  $b_n \in \{0, 1\}, n=0$ 1, 0

  $b_{n-1}, \ldots, b_1, b_0 \in \{0, 1\}$ 110, 1101

  则该数的值可用下式得出：

  $b_n \times 2^n + b_{n-1} \times 2^{n-1} + \ldots + b_1 \times 2^1 + b_0 \times 2^0$

- 例如，1111011 (2) 的值是：

  $1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

  $= 64+32+16+8+0+2+1$

  $= 123$

- 一个数的十进制表示记为：

  $d_n d_{n-1} \ldots d_1 d_0$ (10)

  其中：

  $d_n \in \{1, \ldots, 9\}, n>0$

  $d_n \in \{0, \ldots, 9\}, n=0$

  $d_{n-1}, \ldots, d_1, d_0 \in \{0, \ldots, 9\}$

  则该数的值可用下式得出：

  $d_n \times 10^n + d_{n-1} \times 10^{n-1} + \ldots + d_1 \times 10^1 + d_0 \times 10^0$

- 例如，123 (10) 的值是：

  $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

  $= 100+20+3$

  $= 123$

十进制整数转换为二进制整数的规则

- 可采用辗转相除法。我们用一个例子来说明：

将123 $_{(10)}$ 转换成等值的二进制数：

除以2的商（取整）	余数
123/2 = 61	1
61/2 = 30	1
30/2 = 15	0
15/2 = 7	1
7/2 = 3	1
3/2 = 1	1
1/2 = 0	1

自下而上地依次将余数加以汇集，即得到对应的二进制数：1111011 $_{(2)}$。

八进制与十六进制

- 数据也可以用八进制和十六进制表示：
  - 八进制表示：使用 0-7 共 8 个数字。
  - 十六进制表示：使用 0-9 共 10 个数字和 A-F 共 6 个字母，后者表示 10-15 这 6 个数。
- 将二进制数转换为八进制和十六进制数：
  - 从右向左，每三位进行一次转换，即从二进制数的值转换成等值的八进制数字。
  - 从右向左，每四位进行一次转换，即从二进制数的值转换成等值的十六进制数字。
  - 例：转换 $1111011_{(2)}$
    - $1111011_{(2)} = 173_{(8)}$
    - $1111011_{(2)} = 7B_{(16)}$

1 1 1 1 1 1 (2)
32 16 8 4 2 1 (10)

1 1 1 1 (2)
8 4 2 1 =15(10)

1 0 0 1 (2)
8 4 2 1 =9(10)

采用二进制后的一个问题

- 在我们前面的例子里，b（空格）的作用很重要，它表示着一个数的边界。如果没有了它怎么办？
- 在计算机中的做法是：预先规定数据的表示单位。这样，数据就不需要用特殊的符号来表示它的边界，因为它所在的那个单位的边界，就是这个数据的边界。
- 在计算机中有这样几个基本单位：
  - 位（bit）：1 或者 0；
  - 字节（byte）：8bits（可表示256个不同的数）；
  - 字（word）：与具体的计算机有关，如：8bits、16bits、32bits、64bits等。它的长度称为对应计算机的字长，相当于在对应的图灵机中，有这么多个读写头同时读写。

计算机存储容量的常用表示单位

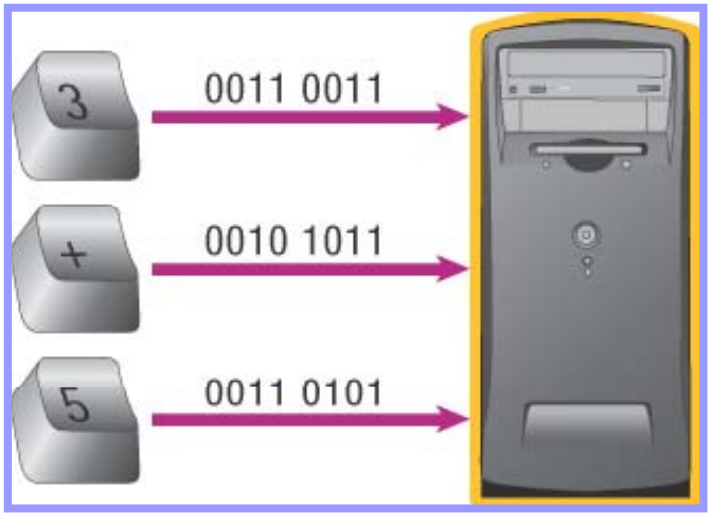
- 在图灵机中，带的长度是无限的。但是，真实的计算机不可能做到这一点，其存储容量（相当于图灵机中带的长度）总是有限的。
- 计算机的字长可能不同，为便于比较，需用与字长无关的单位来表示计算机的存储容量，即字节。常用的表示单位是：B（Bytes)、KB（Kilobytes ）、MB（Megabytes，汉语读作 “兆”）、GB（Gigabytes）、TB（Terabytes）。
  - 1KB = 1024 B = 2^{10} bytes≈10^3 bytes （千字节）
  - 1MB = 1024 KB = 2^{20} bytes≈10^6 bytes （百万字节）
  - 1GB = 1024 MB = 2^{30} bytes≈10^9 bytes （十亿字节）
  - 1TB = 1024 GB = 2^{40} bytes≈10^{12} bytes（万亿字节）

计算机中程序如何表示？—字符和符号

- 一种常用的标准表示是 ASCII（American Standard Code for Information Interchange）码。
- ASCII 码用一个字节来表示一个符号或字母。这个字节的二进制表示本身是一个数，称为对应的码值。
  - 常规 ASCII 码将这个字节的最高位（从左边数第一位）用做奇偶校验，其他 7 位可以表示 128 种符号和字母。
  - 扩展 ASCII 码将这个字节全部用来表示，因此可表示 256 种符号和字母，其中前 128 种与常规 ASCII 码相同。
  - 目前常用的是扩展 ASCII 码。

在计算机中怎样表示符号和英文字母？

- 从计算机键盘输入的字母、数字和符号，都被自动转换成对应的 ASCII 码值。例如（采用扩展 ASCII 码）：
  - 字母 R：$01010010_{(2)}$，即 $82_{(10)}$ 或 $52_{(16)}$
  - 字母 r：$01110010_{(2)}$，即 $114_{(10)}$ 或 $72_{(16)}$
  - 数字 3：$00110011_{(2)}$，即 $51_{(10)}$ 或 $33_{(16)}$
  - 符号%：$00100101_{(2)}$，即 $37_{(10)}$ 或 $25_{(16)}$



Ctrl	十进制	十六进制	字符	代码	十进制	十六进制	字符	十进制	十六进制	字符	十进制	十六进制	字符
^@	0	00		NUL	32	20		64	40	@	96	60	`
^A	1	01		SOH	33	21	!	65	41	A	97	61	a
^B	2	02		STX	34	22	"	66	42	B	98	62	b
^C	3	03		ETX	35	23	#	67	43	C	99	63	c
^D	4	04		EOT	36	24	$	68	44	D	100	64	d
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e
^F	6	06		ACK	38	26	&	70	46	F	102	66	f
^G	7	07		BEL	39	27	'	71	47	G	103	67	g
^H	8	08		BS	40	28	(	72	48	H	104	68	h
^I	9	09		HT	41	29	)	73	49	I	105	69	i
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	0	80	50	P	112	70	p
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q
^R	18	12		DC2	50	32	2	82	52	R	114	72	r
^S	19	13		DC3	51	33	3	83	53	S	115	73	s
^T	20	14		DC4	52	34	4	84	54	T	116	74	t
^U	21	15		NAK	53	35	5	85	55	U	117	75	u
^V	22	16		SYN	54	36	6	86	56	V	118	76	v
^W	23	17		ETB	55	37	7	87	57	W	119	77	w
^X	24	18		CAN	56	38	8	88	58	X	120	78	x
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
^[	27	1B		ESC	59	3B	;	91	5B	[	123	7B	{
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	|
^]	29	1D		GS	61	3D	=	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	⌂

* ASCII 代码 127 拥有代码 DEL。在 MS-DOS 下，此代码具有与 ASCII 8 (BS) 相同的效果。
DEL 代码可由 CTRL + BKSP 键生成。

在计算机中怎样表示汉字？

- 汉字的种类远比 256 种多，何况也不能占用 ASCII 码已经使用的码值。目前采用两个字节来表示一个汉字。
- 汉字的编码规则还没有统一。在我国大陆采用的标准是GB2312。 256*256= 65536个汉字
- 国内的计算机都支持汉字输入与输出，用户可用多种方法输入汉字（如拼音、五笔字形等）。
- 国际组织已经制定了一种 Unicode 标准，也是采用两个字节来表示一个数字、字母、符号或文字，并为中文、日文等都分配了相应的码段（码值连续的区间），以实现各种文字的国际交流。

计算机中二进制数的四则运算

- 仿照十进制数的运算规则，可以做二进制数的加法。

	11100	16+8+4=28
+）	110111	32+16+4+2+1=55
	1010011	64+16+2+1=83

- 用计算机做二进制数的减法
- 用计算机做二进制数的乘法：
  - 把被乘数累加乘数那么多次即可（例：2*3=2+2+2）。
- 用计算机做二进制数的除法：
  - 反复在被除数中减去除数、直到小于除数，减的次数即为商，剩下为余数（例：7/3，7-3-3=1，商为2，余数为1）。

整数在计算机中的表示

- 在计算机中，按照既定的二进制位数（称为码长），
  - 最左边的那一位（称为符号位）用来表示一个整数的正负号：0 表示正数，1 表示负数。
  - 符号位之后的那些位（称为数值位），用来表示这个整数的绝对值。
- 在计算机中，数可以有三种不同的二进制表示方法（差别在于负数之数值位的表示不同）：
  - 原码表示
  - 反码表示
  - 补码表示

00011010

+26

10011010

-26

原码表示

- 在给定码长后，根据一个整数的正负填写符号位，再将这个整数之绝对值的二进制表示，按照数值位的长度在前面补足必要的 0 后，就得到这个整数的原码表示。

若码长为 8，则 $123_{(10)}$ 的原码表示是：

01111011

$-123_{(10)}$ 的原码表示是：

11111011

若码长为 16，则 $123_{(10)}$ 的原码表示是：

0000000001111011

$-123_{(10)}$ 的原码表示是：

1000000001111011

反码表示

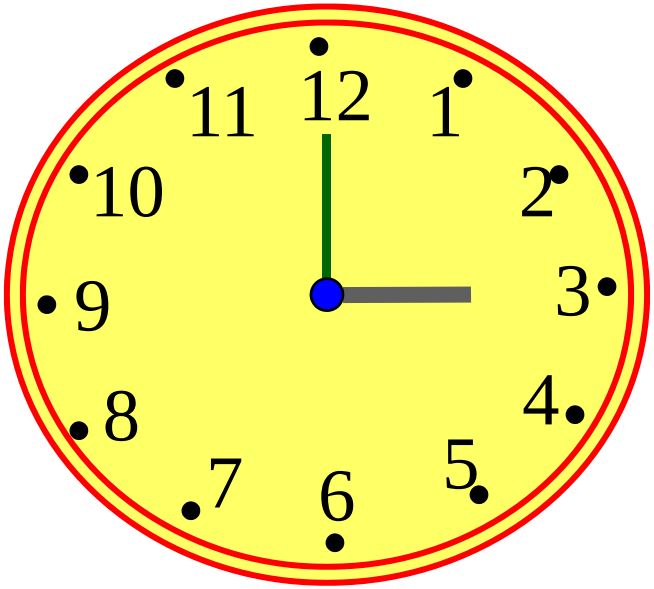
- 规定：
  - 一个正整数的反码表示与其原码表示相同；
  - 一个负整数的反码表示：对其原码表示的数值位进行按位变反的结果

    按位将 1 换成 0、将 0 换成 1
- 例如（若码长为 8）：
  - **(26)（反） = (26)（原） = 0 0011010**
  - **(-26)（反）= 11100101　　(10011010 → 11100101)**

补码表示

- 刚才留下了一个问题：
  - 用计算机怎么做二进制数的减法？
- 如果能够在负数的表示上想办法，就有可能在计算机中利用二进制加法的部件来实现二进制减法
- 由于乘法可以用加法实现、除法可以用减法实现，因而可以用统一的部件来进行二进制数的四则运算。
- 支持这种统一处理的基础，是计算机中数的补码表示。

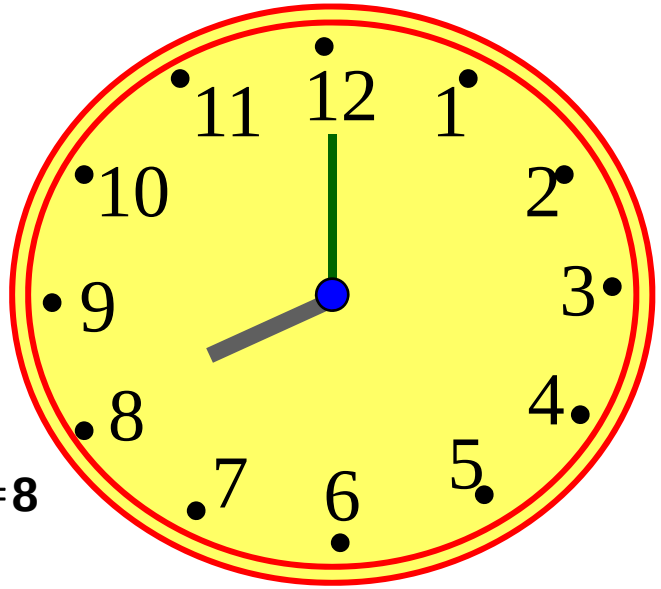
第一个例子

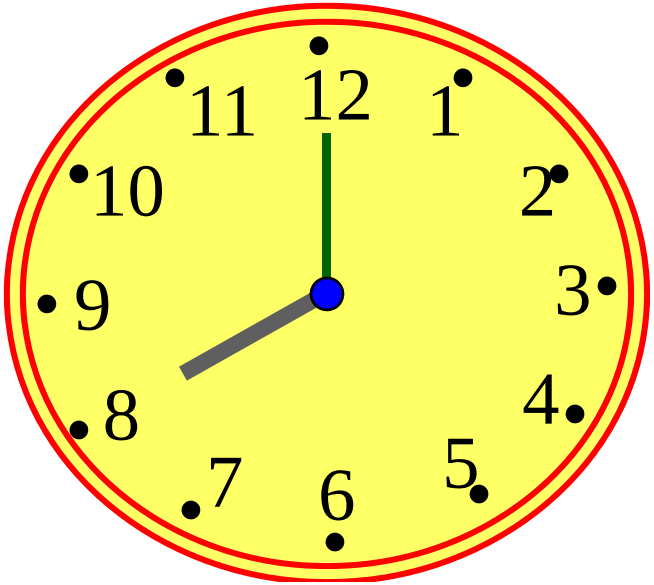


3+5=8

3 - 7=8

3 +（- 7）=8

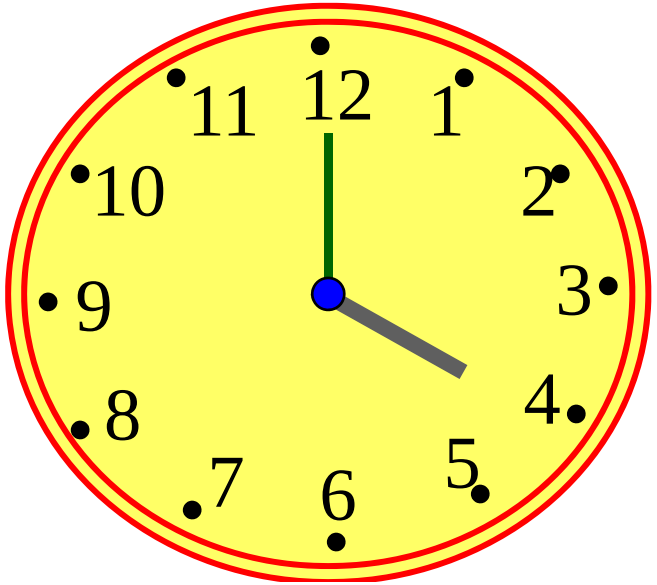




8 - 4=4

8 + 8=4

8 +(- 4)=4



第二个例子


60
100
120
140
100
80
40
160
9 9 9 9 9 9 9
60
180
120
40
200
20
20
1 3 4
220
0
140

第二个例子


60
100
120
140
100
80
40
160
60
180
40
120
200
20
20
1
3
4
220
140
0 0 0 0 0 0 3

补码表示

- 在这个例子中，当里程表上的数字是 999999时，再行进 1 公里，里程表显示的是 000000。
- 由于 999999 + 1 = 000000，（从仪表盘上看到的结果），所以从算术运算的角度看， 这里999999 的作用相当于 -1。

  即 x-1=x+(-1)=x+999999
- 这就说明，当限制了数据的表示长度时，要减去一个正整数b等价于加上一个数c。可以认为：要得到的这个数c加上这个正整数b之后等于 0。我们称之为求补。
  - 在上面的例子中，要得到 1 的负数表示 -1，就是看哪个数加上 1 后等于 0。
  - 这个数便是 999999。

数学表达

- a – b =a+(-b)= a + c (b, c都是正数)

  b + c = 0 ?

  b + c = 100000000 (前提：码长为8位)

- 提出补码概念的目的是
  - 用加法解决减法
  - 所以-b的补码等于c

补码表示

- 回到给定码长的二进制表示上来：

  例如，当码长为 8（即数值位数为 7），则

  $26_{(10)} = 00011010$

  那么，要得到 $-26_{(10)}$ 的补码，就得求一个二进制数 c：

  使得：$c + 00011010 = 100000000$

  这样相对于 $|-26_{(10)}|$ 的 c 应该为：

  11100110

  因为：

```
  11100110
+)00011010
 ----------
  00000000
```

因码长有限，
进位被丢弃

补码表示

- 规定：
  - 一个正整数的补码表示与它的原码表示相同；
  - 一个负整数的补码表示：符号位为 1，数值位是其绝对值的求补结果。
- 对于一个负整数，怎样求它的补码表示？
  - 一条简单规则：对其原码表示的数值位按位变反后加 1。
  - 例：当码长为 8，求 $-26_{(10)}$ 的补码表示（11100110）：
    - 原码表示是：10011010
    - 按位变反后：11100101　（其实是反码）
    - 加1后得到：　11100110，即得到其补码表示。

补码运算规则

- 已经证明，对于数 X 和 Y，(X+Y)（补）= X（补）+Y（补），(X-Y)（补）= X（补）+（-Y）（补）
- 两个数相加减，只需进行包括符号位在内的补码相加：

(-27)（补）= 11100101　(10011011 → 11100100 → 11100101)

(-1)（补）　= 11111111　(10000001 → 11111110 → 11111111)

(-26)（补）= 11100110　(10011010 → 11100101 → 11100110)

(-25)（补）= 11100111　(10011001 → 11100110 → 11100111)

26 - 27 = -1	26 - 26 = 0	26 - 25 = 1
00011010	00011010	00011010
+）11100101	+）11100110	+）11100111
11111111	00000000	00000001

课后作业

1、加深概念：二进制在计算机中的作用。
2、使用二进制进行数据的存储，相比其它进制，更节约存储空间。
3、熟悉原码、反码和补码的概念与计算。
4、继续熟悉C编程工具：Dev C++，VC6.0，C4droid（或其它手机编程工具）

进位计数制及其各进位制数之间的

6/159

- 计算机中常用的计数制
  - 二进制 （Binary）

R=2，各数位上只有两个数字0,1

例:有二进制数x=10110111.011

其值为:$2^7+2^5+2^4+2^2+2^1+2^0+2^{-2}+2^{-3}$ =183.375

计算机中为何要采用二进制计数?

（A）只有两种基本状态，易于硬件实现

可用任何具有两个截然不同物理状态的器件表示或存储二进制数。这些状态有:

开关:通--断　　电流:有--无

电压:高--低　　正向和反向磁化等

（B）节省存储成本

设在R进制中用n位数字能表示的最大数为N

则:$N = R^n - 1$

预备知识 二进制（Binary）

- 节省存储成本

因为每位有R种状态所以要表示一位数字的成本正比于R

则总成本正比于R*n，设它为x，即：$x=k*R*n$

设$R^n=N+1=M$ ，两边取对数：

$\ln R^n=\ln M$

$n*\ln R=\ln M$

$n=\ln M/\ln R$

则 $x=k*R*\ln M/\ln R$

为了求R为何值时，成本x最小，则令：$dx/dR=0$

即：$k*n+k*R*(\ln M/\ln R)'=0$

可以推出：$(\ln R-1)/\ln^2 R=0$

故有：$\ln R=1$，$\therefore R=e=2.71828$ 时，x取最小值。

因为R必须为整数，故可取2、3，

由于取2硬件易于实现，所以主要采用二进制计数。