微机系统与接口
——计算机中的数据表示

王江峰 副研究员
电气工程学院

1 计算机中的数据表示

- 在计算机内部，数据都是以二进制的形式存储和运算的
  - 位

    ——位(bit)简写为b，音译为比特，是计算机存储数据的最小单位，是二进制数据中的一个位，一个二进制位只能表示0或1两种状态，要表示更多的信息，就得把多个位组合成一个整体，每增加一位，所能表示的信息量就增加一倍

  - 字节

    ——字节(Byte)简记为B，规定一个字节为8位，即1Byte = 8bit。字节是计算机数据处理的基本单位，并主要以字节为单位解释信息。每个字节由8个二进制位组成。通常，一个字节可存放一个ASCII码，两个字节存放一个汉字国际码

  - 字

    ——字(Word)是计算机进行数据处理时，一次存取、加工和传送的数据长度。一个字通常由一个或若干个字节组成，由于字长是计算机一次所能处理信息的实际位数，所以，它决定了计算机数据处理的速度，是衡量计算机性能的一个重要标识，字长越长，性能越好

    ——计算机型号不同，其字长是不同的，常用的字长有8位、16位、32位和64位

1 计算机中的数据表示

- 在计算机内部，数据都是以二进制的形式存储和运算的
  - 机器数

    ——在计算机内部，任何信息都以二进制代码表示(即0与1的组合来表示)。一个数在计算机中的表示形式，称为机器数

  - 真值

    ——机器数所对应的原来的数值称为真值

1 计算机中的数据表示

- 原码、反码、补码
  - 原码

    ——原码表示法即用机器数的最高位代表符号(若为0，则代表正数，若为1，则代表负数)，数值部分为真值的绝对值的一种表示方法

  - 反码

    ——正数的反码不变，负数的反码是对应正数的原码取反

  - 补码（计算机中使用的编码）

    ——正数的补码不变，负数的补码是该数的反码+1

1 计算机中的数据表示

- 原码、反码、补码

十进制	+73	–73	+127	–127	+0	–0
二进制(真值)	+1001001B	–1001001B	+1111111B	–1111111B	+0000000B	–0000000B
原码	01001001B	11001001B	01111111B	11111111B	0000000B	10000000B
反码	01001001B	10110110B	01111111B	10000000B	00000000B	11111111B
补码	01001001B	10110111B	01111111B	10000001B	0000000B	00000000B

1 计算机中的数据表示

- 原码、反码、补码

[+1] = [00000001]原 = [00000001]反 = [00000001]补
[-1] = [10000001]原 = [11111110]反 = [11111111]补

1 - 1 = 1 + (-1) = [00000001]原 + [10000001]原 = [10000010]原 = -2

1 - 1 = 1 + (-1) = [0000 0001]原 + [1000 0001]原= [0000 0001]反
+ [1111 1110]反 = [1111 1111]反 = [1000 0000]原 = -0

1-1 = 1 + (-1) = [0000 0001]原 + [1000 0001]原 = [0000 0001]补
+ [1111 1111]补 = [0000 0000]补=[0000 0000]原

2 进制转换

- 进制转换
  - 十进制与二进制之间的转换

例如把52换算成二进制数，计算结果如图：

```
2| 52 ....0
 2| 26 ....0
  2| 13 ....1
   2| 6 ....0
    2| 3 ....1
     1 ....1
```

例如要把-52换算成二进制：

1.先取得52的二进制：00110100

2.对所得到的二进制数取反：11001011

3.将取反后的数值加一即可：11001100

  - 二进制与十六进制之间的转换

  - 十进制与十六进制转换

3 编码

- 其他编码
  - BCD码——常用8421 BCD码

常用BCD码

十进制数	8421BCD码	5421码	2421码	余3码	余3循环码
0	0000	0000	0000	0011	0010
1	0001	0001	0001	0100	0110
2	0010	0010	0010	0101	0111
3	0011	0011	0011	0110	0101
4	0100	0100	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1010	1101	1010	1111
8	1000	1011	1110	1011	1110
9	1001	1100	1111	1100	1010
10	0001 0000	0001 0000	0001 0000	0100 0011	0110 0010
11	0001 0001	0001 0001	0001 0001	0100 0100	0110 0110
12	0001 0010	0001 0010	0001 0010	0100 0101	0110 0111
…	…	…	…	…	…
19	0001 1001	0001 1100	0001 1111	0100 1100	0110 1010

3 编码

- 其他编码
  - 格雷码（Gray）——相邻的2个数只有一位不同

自然二进制码与格雷码的对照表：

十进制数	自然二进制数	格雷码	十进制数	自然二进制数	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

3 编码

- 其他编码
  - ASCII

——7位二进制编码

——表示字母、数字字符和控制字符

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

3 编码

- 其他编码
  - Unicode（统一码、万国码、单一码）

——16位编码

——对世界上所有语言大部分字符进行编码

——https://unicode-table.com/en/#control-character

4 数据类型

- 32位ARM C语言编程中所用的数据类型

数据类型	位数	范围（有符号）	范围（无符号）
char,int8_t,uint8_t	8	-128 – 127	0 – 255
short, int16_t, uint16_t	16	-32768 – 32767	0 – 65535
int, int32_t, uint32_t	32	-2^31 – 2^31-1	0 – 2^32 -1
long	32	-2^31 – 2^31-1	0 – 2^32 -1
Long long, int64_t, uint64_t	64	-2^63 – 2^63-1	0 – 2^64 -1
float	32	-3.403*10^38 – 3.403*10^38	
Double, long double	64	-1.798*10^308 – 1.798*10^308	

4 小数的表示方法

- 小数的表示方法
  - 定点数

    ——约定数值的小数点固定在某一位置，称为定点表示法，简称为定点数

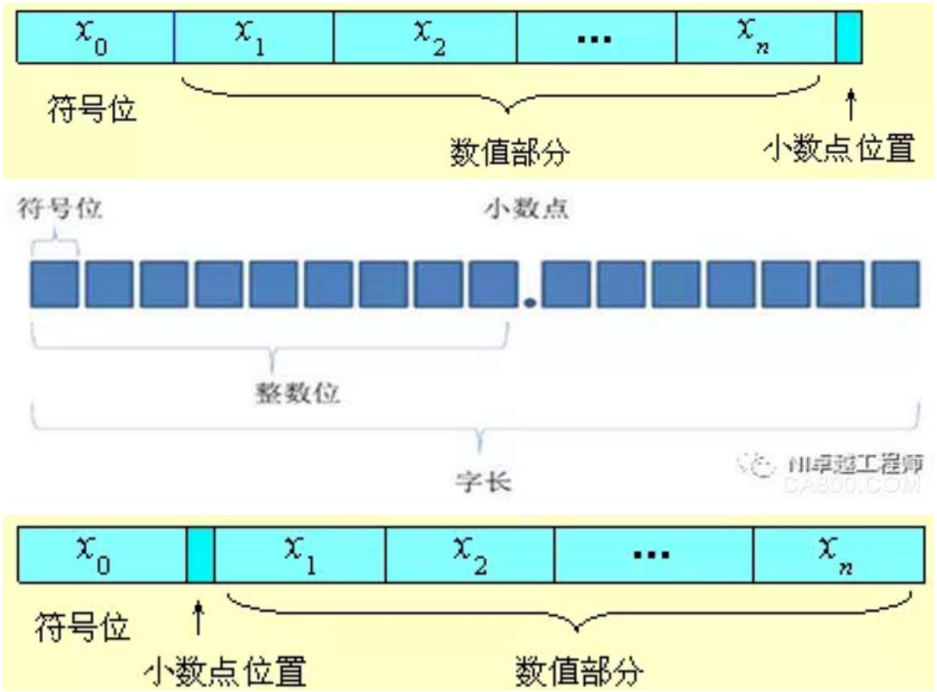
  - 浮点数

    ——小数点位置可以任意浮动，称为浮点表示法，简称为浮点数

4 小数的表示方法

- 小数的表示方法
  - 定点数

——约定数值的小数点固定在某一位置，称为定点表示法，简称为定点数



$$
\begin{aligned}
&101.011\\
&=(1*2^0+0*2^1+1*2^2)+(0*2^{-1}+1*2^{-2}+1*2^{-3})\\
&=5+0.375\\
&=5.375
\end{aligned}
$$

十进制：3.125

为二进制：11.001

4 小数的表示方法

- 小数的表示方法
  - 浮点数

——小数点位置可以任意浮动，称为浮点表示法，简称为浮点数

float的规格化表示为：$\pm 1.f \times 2^{E-127}$，其中，f是尾数，E是指数。

31	23	0
±	8位指数E	尾数f：规格化形式的整数位为固定为1，可以省略，所以能表示24位
	指数E = 真值 + 偏移值 偏移值 = 127	尾数 = 真值（原码）

其中绿色部分是符号位（占1位），蓝色部分是阶码（占8位），黄色部分是尾数（占23位）

32位浮点数转换规则

①是正数符号位是0，是负数符号位是1

②阶码E=e+127（加127的作用是将指数转换为非负数，这样省去一个指数符号位）

③M照搬，剩下位补0

4 小数的表示方法

- 小数的表示方法
  - 浮点数

——小数点位置可以任意浮动，称为浮点表示法，简称为浮点数

float的规格化表示为：$\pm 1.f \times 2^{E-127}$，其中，f是尾数，E是指数。

例：将数$(20.59375)_{10}$转换成754标准的32位浮点数的二进制储存格式。

$20.59375 = 10100.10011$

$10100.10011 = 1.010010011 * 2^4$

$M = 010010011\,,\ e = 4$

$E = e + 127 = 131 = 10000011$

最终结果为01000001101001001100000000000000

4 小数的表示方法

- 小数的表示方法
  - 浮点数

——小数点位置可以任意浮动，称为浮点表示法，简称为浮点数

float的规格化表示为：$\pm 1.f \times 2^{E-127}$，其中，f是尾数，E是指数。

比如十进制数123.125，其二进制表示为：1111011.001，规格化表示为：1.111011001×2^6
也就是$1.111011001 \times 2^{133-127}$，$f$= 111011001，$E$ = 133 = 10000101，图示如下：

+	E = 6 + 127 = 133								尾数：111011001（后面补0到23位）															
0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

4 小数的表示方法

- 浮点数
  - 浮点数使用方便
  - 运算非常复杂
  - 需要特殊的浮点运算单元，一般只有高端CPU中才会有，成本较高，而且智能处理加法、减法和乘法
  - 其他数学计算，比如除法，开方，三角函数等，则计算更加复杂耗时
  - 定点数与浮点数之间相互转换，也消耗计算时间

5 作业

1. 冯·诺伊曼计算机的基本设计思想是什么？哈佛结构的计算机，与冯·诺伊曼计算机相比，有哪些优缺点？
2. 什么是总线，总线通常有哪3组信号？各组信号的作用是什么？
3. 1）计算机的字长是什么含义？2）简述处理器中的流水线技术。
4. 将下列十六进制无符号整数，转换为十进制真值。

1) 0FFH　　2) 0H　　3) 5EH　　4) EFH

5. 如果上题中的十六进制数为8位有符号整数，请将其转换位十进制真值
6. 将下列十进制数转换位压缩BCD码

1）12　　2）24　　3）68　　4）99

7. 将下列二级制补码表示的有符号整数转换为十进制真值

1）0000 0000b　　2）0111 1111b　　3）1000 0001b　　4）1100 0111b