王道考研——组成原理

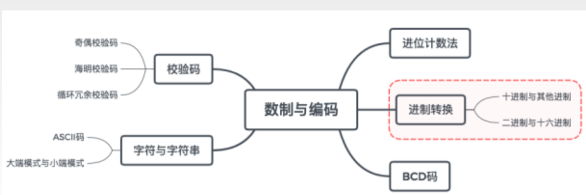
WWW.CSKAOYAN.COM

第二章 数据的表示和运算

本节内容

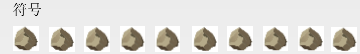
进位
计数法

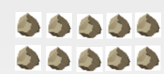
本节总览



进位计数法

符号

一进制： 



二进制： 0，1　方便对应到物理器件的状态，如高电平、低电平　1010

四进制： 0，1，2，3　22

八进制： 0，1，2，3，4，5，6，7　12

十进制： 0，1，2，3，4，5，6，7，8，9　10

十六进制： 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F　A

基数：每个数位所用到的不同符号的个数

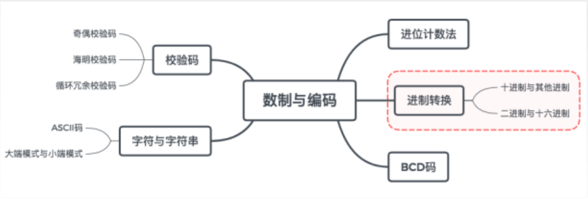
	基数大(十进制)	vs	基数小(二进制)
位数：	少		多
运算(乘法为例)：	100种情况		4种情况

本节内容

数制与编码

进制转换

本节总览



进制转换

十进制： 75.3

70 + 5 + 0.3

7×10 + 5×1 + 3×0.1

$7\times 10^1 + 5\times 10^0 + 3\times 10^{-1}$

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$ 位权

$= K_n \times \boxed{r^n} + K_{n-1} \times \boxed{r^{n-1}} + \cdots + K_2 \times \boxed{r^2} + K_1 \times \boxed{r^1} + K_0 \times \boxed{r^0} + K_{-1} \times \boxed{r^{-1}} + K_{-2} \times \boxed{r^{-2}} + \ldots + K_{-m} \times \boxed{r^{-m}}$

任意进制 —> 十进制

二进制： 101.1 —> $1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} = 5.5$

四进制： 11.2 —> $1\times 4^1 + 1\times 4^0 + 2\times 4^{-1} = 5.5$

八进制： 5.4 —> $5\times 8^0 + 4\times 8^{-1} = 5.5$

十进制： 5.5 —> $5\times 10^0 + 5\times 10^{-1} = 5.5$

十六进制： 5.8 —> $5\times 16^0 + 8\times 16^{-1} = 5.5$

进制转换

十进制 —> 任意进制

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

= 75.3

整数部分：75

$$\frac{K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0}{r} = \underbrace{K_n \times r^{n-1} + K_{n-1} \times r^{n-2} + \cdots + K_2 \times r^1 + K_1 \times r^0}_{商} \ldots \underbrace{K_0}_{余数}$$

如：十进制 —> 二进制 r = 2

$75 \div 2 = 37 \ldots \boxed{1}\ K_0$

$37 \div 2 = 18 \ldots \boxed{1}\ K_1$

$18 \div 2 = 9 \ldots \boxed{0}\ K_2$

$9 \div 2 = 4 \ldots \boxed{1}\ K_3$

$4 \div 2 = 2 \ldots \boxed{0}\ K_4$

$2 \div 2 = 1 \ldots \boxed{0}\ K_5$

$1 \div 2 = 0 \ldots \boxed{1}\ K_6$

75D = 1001011B

$(75)_{10} = (1001011)_2$

除基		取余
2	75	1
2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
2	1	1
	0	

进制转换

十进制 —> 任意进制

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

= 75.3

整数部分：75

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

75 = 64 + 11

= 64 + 8 + 3

= 64 + 8 + 2 + 1

$= 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0$

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

进制转换

十进制 —> 任意进制

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

= 75.3

小数部分：0.3

$(K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}) \times r = \underbrace{K_{-1} \times r^0}_{整数} + \underbrace{K_{-2} \times r^{-1} + \ldots + K_{-m} \times r^{-(m-1)}}_{小数}$

如：十进制 —> 二进制 r = 2

$0.3 \times 2 = 0.6 = \boxed{0} + 0.6 \quad K_{-1}$

$0.6 \times 2 = 1.2 = \boxed{1} + 0.2 \quad K_{-2}$

$0.2 \times 2 = 0.4 = \boxed{0} + 0.4 \quad K_{-3}$

$0.4 \times 2 = 0.8 = \boxed{0} + 0.8 \quad K_{-4}$

$0.8 \times 2 = 1.6 = \boxed{1} + 0.6 \quad K_{-5}$

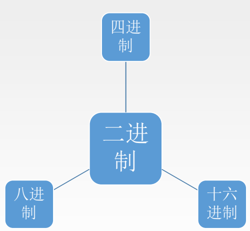
……

0.3D = 0.01001… B

乘基	取整
0.3	
× 2	
0.6	0
× 2	
1.2	1
0.2	
× 2	
0.4	0
…	

进制转换

2^n 进制之间的转换：二进制、四进制、八进制、十六进制



二进制 —> 四进制、八进制、十六进制

n 位一组，每组转换成对应进制的符号

如：1111000010.01101

11 11 00 00 10 . 01 10 10
3 3 0 0 2 . 1 2 2 四进制

001 111 000 010 . 011 010
1 7 0 2 . 3 2 八进制

0011 1100 0010 . 0110 1000
3 C 2 . 6 8 十六进制

3C2.68H = 1111000010.01101B

$(3C2.68)_{16} = (1111000010.01101)_2$

四进制、八进制、十六进制 —> 二进制

每位写成对应的二进制形式

真值和机器数

15 → 1111

8 → 1000

+15 → 0 1111

- 8 → 1 1000

真值	机器数
+1111	
- 1000	

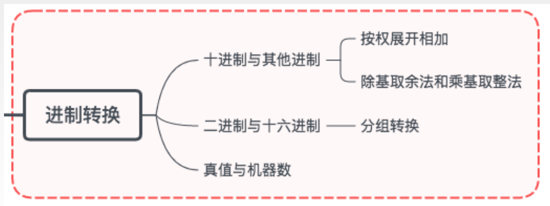
$\xrightarrow{适应运算规则}$ 原码、反码、补码、移码

$\xrightarrow{扩大表示范围}$ 浮点数

知识回顾

3C2.68H = 1111000010.01101B

$(3C2.68)_{16} = (1111000010.01101)_2$

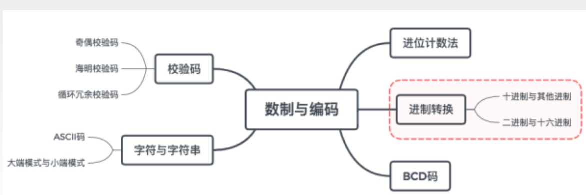


本节内容

数制与编码

BCD码

本节总览



BCD码

二进制：0，1　　方便计算机处理

十进制：0，1，2，3，4，5，6，7，8，9　　符合人类习惯

$K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0$　　转换麻烦

快速转换：一一对应　　BCD：Binary-Coded Decimal

8421码的映射关系：

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

十进制：5 + 8　　13　　1 3　　1 3

8421码：0101 + 1000　　$\boxed{1101}$ $\xrightarrow{+0110}$ 1 0011　　0001 0011

不在映射表里

8421码中 1010～1111 没有定义

BCD码

8421码的映射关系：

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

4个二进制位 —> 16种不同的状态

BCD码直使用其中10种 —> 不同的映射方案

余3码：8421码 + $(0011)_2$

0	1	2	3	4	5	6	7	8	9
0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

2421码：改变权值定义

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	1011	1100	1101	1110	1111

王道考研/CSKAOYAN.COM

知识回顾

王道考研/CSKAOYAN.COM

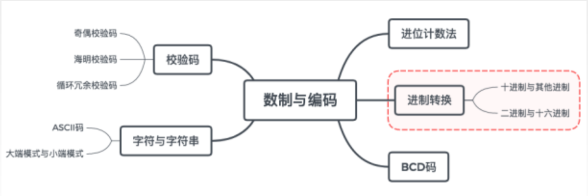
本节内容

数制与编码

字符
与
字符串

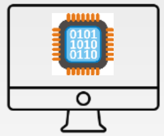
王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

ASCII码



数字
字母
符号

→ 共128个字符 —> 7位二进制编码 → ASCII码

王道考研/CSKAOYAN.COM

ASCII码

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

可印刷字符：32～126　　大写字母：65(100 0001)～90(101 1010)

数字：48(011 0000)～57(011 1001)　　小写字母：97(110 0001)～122(111 1010)

王道考研/CSKAOYAN.COM

ASCII码

例1：已知‘A’的ASCII码值为65，字符‘H’存放在某存储单元M中，求M中存放的内容。

首先明确，M中存放的是‘H’的ASCII码(二进制形式)。
再由‘A’的码值推出‘H’的码值：
思路1. A是第1个字母，H是第8个字母，则H的码值 = 65 + (8-1) = 72
　　72 对应二进制为 100 1000，故M中存放的内容为0100 1000
思路2. A的码值65写成二进制为100 0001，A是第1个字母
　　H是第8个字母，故对应100 1000，M中存放内容为0100 1000

每个存储单元存放的内容为
字节(Byte)的整数倍
即8的整数倍
这里先假设存放1B

例2：已知‘h’的ASCII码值为104，字符‘a’存放在存储单元M1中，字符‘z’存放在存储单元M2中，求M1、M2中存放的内容。

a : 104 – (8-1) = 97 —> M1中内容为0110 0001
z : 104 + (26-8) = 122 —> M2中内容为0111 1010

王道考研/CSKAOYAN.COM

字符串

每个存储单元存放4B

字符串：IF_A>B_THEN_READ(C)_
大端模式：存储单元内先存储高位字节、后存储低位字节的顺序
小端模式：存储单元内先存储低位字节、后存储高位字节的顺序

王道考研/CSKAOYAN.COM

字符串

I	F	空格	A
>	B	空格	T
H	E	N	空格
R	E	A	D
(	C	)	空格

每个存储单元存放4B

字符串：IF_A>B_THEN_READ(C)_

大端模式：存储单元内先存储高位字节、后存储低位字节的顺序

字符串

73	70	32	65
62	66	32	84
72	69	78	32
82	69	65	68
40	67	41	32

每个存储单元存放4B

字符串：IF_A>B_THEN_READ(C)_

大端模式：存储单元内先存储高位字节、后存储低位字节的顺序

字符串

I	F	空格	A
>	B	空格	T
H	E	N	空格
R	E	A	D
(	C	)	空格

大端模式：存储单元内先存储高位字节、后存储低位字节的顺序

字符串：IF_A>B_THEN_READ(C)_

小端模式：存储单元内先存储低位字节、后存储高位字节的顺序

A	空格	F	I
T	空格	B	>
空格	N	E	H
D	A	E	R
空格	)	C	(

汉字的表示和编码

GB 2312-80：汉字+各种符号共7445个

区位码：94个区，每区94个位置

啊：　16　01

$\downarrow$

10H　01H

+20H　+20H

国标码：　30H　21H

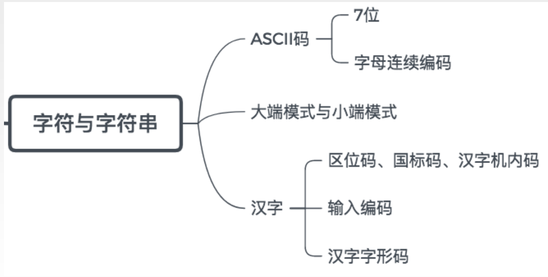
+80H　+80H

汉字内码：　B0H　A1H

输入：输入编码

输出：汉字字形码

知识回顾

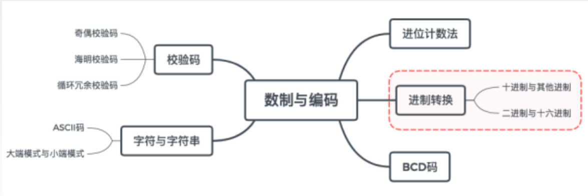


本节内容

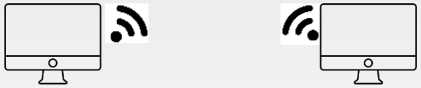
数制与编码

校验原理
奇偶校验码

本节总览



校验原理简介



信息：　A　B

码字：　0100　0001　(C、D) 1 -> 2

码字（方案二）：　00　11　2

码距：两个合法码字对应位上数字的不同位的个数

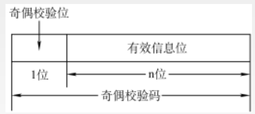
奇校验：保证一段数据中出现奇数个1

仅需1位

改变规则　->　海明码、CRC

奇偶校验码

奇校验码：整个校验码（有效信息位和校验位）中“1”的个数为奇数。
偶校验码：整个校验码（有效信息位和校验位）中“1”的个数为偶数。



【例2-3】 给出两个编码1001101和1010111的奇校验码和偶校验码。

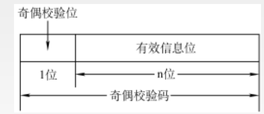
设最高位为校验位，余7位是信息位，则对应的奇偶校验码为：

奇校验：　11001101　　0 1010111

偶校验：　01001101　　1 1010111

奇偶校验码

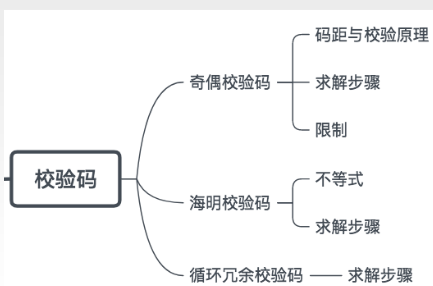
奇校验码：整个校验码（有效信息位和校验位）中“1”的个数为奇数。
偶校验码：整个校验码（有效信息位和校验位）中“1”的个数为偶数。



【例2-3】 给出两个编码1001101和1010111的奇校验码和偶校验码。

设最高位为校验位，余7位是信息位，则对应的奇偶校验码为：
1001101　　11001101（奇校验）　　01001101（偶校验）
1010111　　01010111（奇校验）　　11010111（偶校验）

知识回顾

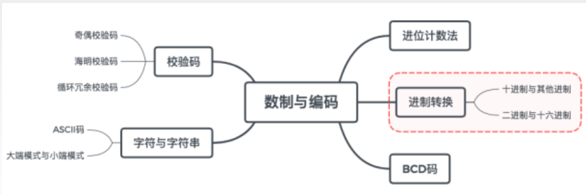


本节内容

数制与编码

海明校验码

本节总览



王道考研/CSKAOYAN.COM

海明校验码思路简介

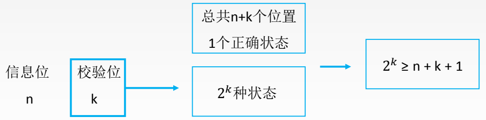
海明码设计思路：分组校验 —> 多个校验位

—>校验位标注出错位置

1010 -> 1011

校验位：001

需要多少校验位？



n	1	2-4	5-11	12-26	27-57	58-120
k	2	3	4	5	6	7

王道考研/CSKAOYAN.COM

海明码求解步骤

信息位：1010

1. 确定海明码的位数： $2^k \geq n+k+1$

n = 4 —> k = 3

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

王道考研/CSKAOYAN.COM

海明码求解步骤

信息位：1010

1. 确定海明码的位数： $2^k \geq n+k+1$

n = 4 —> k = 3

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	0	1		0		

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

3. 求校验位的值

H_3 ： 3 —> 0 1 1

H_5 ： 5 —> 1 0 1

H_6 ： 6 —> 1 1 0

H_7 ： 7 —> 1 1 1

$P_1 = H_3 \oplus H_5 \oplus H_7 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$

$P_2 = H_3 \oplus H_6 \oplus H_7 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$

$P_3 = H_5 \oplus H_6 \oplus H_7 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$

⊕：异或

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

相当于偶校验

王道考研/CSKAOYAN.COM

海明码求解步骤

信息位：1010

1. 确定海明码的位数： $2^k \geq n + k + 1$

$n = 4 \;\; \text{—>} \;\; k = 3$

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	0	1	0	0	1	0

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

3. 求校验位的值

H_3 ： 3 —> $\boxed{0}\,\boxed{1}\,\boxed{1}$

H_5 ： 5 —> $\boxed{1}\,\boxed{0}\,\boxed{1}$

H_6 ： 6 —> $\boxed{1}\,\boxed{1}\,\boxed{0}$

H_7 ： 7 —> $\boxed{1}\,\boxed{1}\,\boxed{1}$

$P_1 = H_3 \oplus H_5 \oplus H_7 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$

$P_2 = H_3 \oplus H_6 \oplus H_7 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$

$P_3 = H_5 \oplus H_6 \oplus H_7 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$

4. 纠错

$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4$
$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4$
$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4$

王道考研/CSKAOYAN.COM

海明码求解步骤

信息位：1010

1. 确定海明码的位数： $2^k \geq n + k + 1$

$n = 4 \;\; \text{—>} \;\; k = 3$

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	0	1	0	0	1	0

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

3. 求校验位的值

$P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$
$P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$
$P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$

4. 纠错

校验方程：
$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4$
$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4$
$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4$

接收到：1010010
$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$
$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

接收到：1010000
$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$
$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

王道考研/CSKAOYAN.COM

海明码求解步骤-格式变化

信息位：1010

1. 确定海明码的位数： $2^k \geq n + k + 1$

$n = 4 \;\; \text{—>} \;\; k = 3$

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	0	1	0	0	1	0

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

3. 求校验位的值

$P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$
$P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$
$P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$

设信息位$D_1D_2D_3D_4$（1010），共4位，校验位$P_1P_2P_3$，共3位，对应的海明码为$H_1H_2H_3H_4H_5H_6H_7$。

H_1	H_2	H_3	H_4	H_5	H_6	H_7
P_1	P_2	D_1	P_3	D_2	D_3	D_4
		1		0	1	0

$P_1 = D_1 \oplus D_2 \oplus D_4 = 1 \oplus 0 \oplus 0 = 1$
$P_2 = D_1 \oplus D_3 \oplus D_4 = 1 \oplus 1 \oplus 0 = 0$
$P_3 = D_2 \oplus D_3 \oplus D_4 = 0 \oplus 1 \oplus 0 = 1$

王道考研/CSKAOYAN.COM

海明码求解步骤-格式变化

信息位：1010

1. 确定海明码的位数： $2^k \geq n + k + 1$

$n = 4 \;\; \text{—>} \;\; k = 3$

设信息位$D_4D_3D_2D_1$（1010），共4位，校验位$P_3P_2P_1$，共3位，对应的海明码为$H_7H_6H_5H_4H_3H_2H_1$。

2. 确定校验位的分布

H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	0	1	0	0	1	0

校验位P_i放在海明位号为2^{i-1}的位置上

信息位按顺序放到其余位置

3. 求校验位的值

$P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$
$P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$
$P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$

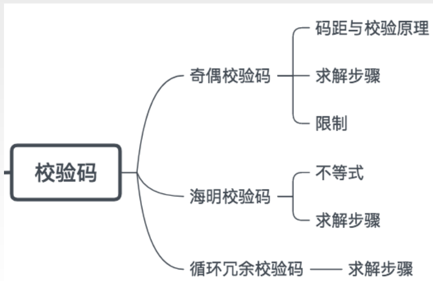
设信息位$D_1D_2D_3D_4$（1010），共4位，校验位$P_1P_2P_3$，共3位，对应的海明码为$H_1H_2H_3H_4H_5H_6H_7$。

H_1	H_2	H_3	H_4	H_5	H_6	H_7
P_1	P_2	D_1	P_3	D_2	D_3	D_4
1	0	1	1	0	1	0

$P_1 = D_1 \oplus D_2 \oplus D_4 = 1 \oplus 0 \oplus 0 = 1$
$P_2 = D_1 \oplus D_3 \oplus D_4 = 1 \oplus 1 \oplus 0 = 0$
$P_3 = D_2 \oplus D_3 \oplus D_4 = 0 \oplus 1 \oplus 0 = 1$

王道考研/CSKAOYAN.COM

知识回顾



王道考研/CSKAOYAN.COM

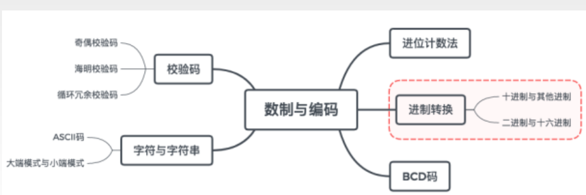
本节内容

数制与编码

循环冗余校验码

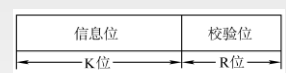
王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

循环冗余校验码



【例2-5】 设生成多项式为$G(x)=x^3+x^2+1$，信息码为101001，求对应的CRC码。

1. 确定K、R以及生成多项式对应的二进制码

K = 信息码的长度 = 6，R = 生成多项式最高次幂 = 3 → 校验码位数N = K + R = 9

生成多项式$G(x) = 1\cdot x^3 + 1\cdot x^2 + 0\cdot x^1 + 1\cdot x^0$，对应二进制码1101

2. 移位

1101 | 101001000

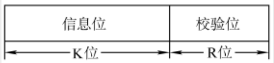
信息码左移R位，低位补0

3. 相除

对移位后的信息码，用生成多项式进行模2除法，产生余数

王道考研/CSKAOYAN.COM

循环冗余校验码



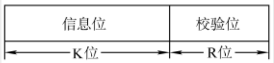
【例2-5】 设生成多项式为$G(x)=x^3+x^2+1$，信息码为101001，求对应的CRC码。

```
           110101
1101 | 101001000
       1101
        1110
        1101
         0111
         0000
          1110
          1101
           0110
           0000
            1100
            1101
             001   校验位
```

对应的CRC码：

101001 001

循环冗余校验码



【例2-5】 设生成多项式为$G(x)=x^3+x^2+1$，信息码为101001，求对应的CRC码。

1. 确定K、R以及生成多项式对应的二进制码

K = 信息码的长度 = 6，R = 生成多项式最高次幂 = 3 → 校验码位数N = K + R = 9

生成多项式$G(x) = 1\cdot x^3 + 1\cdot x^2 + 0\cdot x^1 + 1\cdot x^0$，对应二进制码1101

2. 移位

信息码左移R位，低位补0

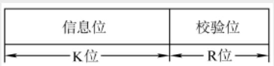
3. 相除

对移位后的信息码，用生成多项式进行模2除法，产生余数

对应的CRC码：101001 001

4. 检错和纠错

循环冗余校验码



【例2-5】 设生成多项式为$G(x)=x^3+x^2+1$，信息码为101001，求对应的CRC码。

3. 相除

对移位后的信息码，用生成多项式进行模2除法，产生余数

对应的CRC码：101001 001

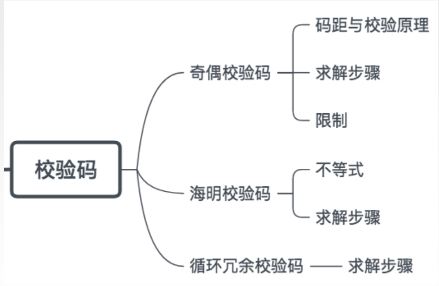
4. 检错和纠错

发送：101001001　　记为$C_9C_8C_7C_6C_5C_4C_3C_2C_1$

接收：101001001 —用1101进行模2除→ 余数为000，代表没有出错

接收：101001011 —用1101进行模2除→ 余数为010，代表C_2出错

知识回顾



王道考研——组成原理

WWW.CSKAOYAN.COM

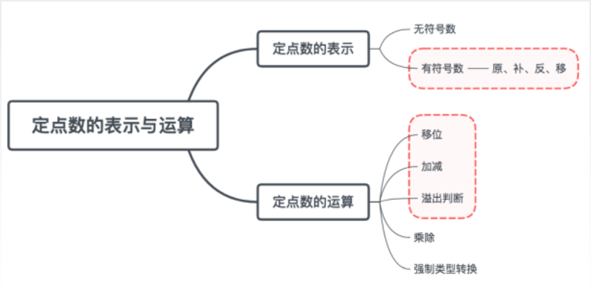
第二章 数据的表示和运算

本节内容

定点数的表示和运算

无符号数
原码

本节总览



无符号数

无符号数：整个机器字长的全部二进制位均为数值位，没有符号位，相当于数的绝对值。

1001 1100B
$=1\times 2^7+0\times 2^6+0\times 2^5+1\times 2^4+1\times 2^3+1\times 2^2+0\times 2^1+0\times 2^0$
=156D

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

表示范围

8位二进制数：2^8种不同的状态

0000 0000 ～ $\boxed{1111\ 1111}$ = 1 0000 0000 − 1

0 ～ $\boxed{255}$ = 2^8 − 1

n位的无符号数表示范围为：0 ～ 2^n-1

有符号数

+ 156 D = 0 1001 1100B
- 156 D = 1 1001 1100B

真值　　机器数

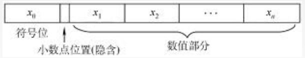
定点表示

+ 156 D = 0 1001 1100B
- 156 D = 1 1001 1100B

真值　　机器数

小数点：隐含存储(定点数：事先约定；浮点数：按规则浮动)

定点小数



+0.75D = 0.11B 存储为011
-0.75D = 1.11B 存储为111
(未考虑位数扩展)

表示范围

$K_{-1} \times 2^{-1} + K_{-2} \times 2^{-2} + \ ... + K_{-m} \times 2^{-m}$

绝对值：0.00 ~ $\boxed{0.11}$ = 1.00 - 0.01

0 ~ $1- 2^{-2}$

有n位尾数的定点小数：$-(1- 2^{-n})$ ~ $1- 2^{-n}$

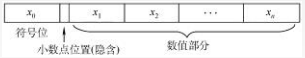
定点表示

+ 156 D = 0 1001 1100B
- 156 D = 1 1001 1100B

真值　　机器数

小数点：隐含存储(定点数：事先约定；浮点数：按规则浮动)

定点小数



+0.75D = 0.11B 存储为011
-0.75D = 1.11B 存储为111
(未考虑位数扩展)

表示范围 $-(1- 2^{-n})$ ~ $1- 2^{-n}$

定点整数

+3D = 011.B 存储为011
-3D = 111.B 存储为111
(未考虑位数扩展)

表示范围

绝对值：0 ~ $2^n - 1$

有n位尾数的定点整数：$-(2^n - 1)$ ~ $2^n - 1$

原码

+ 156 D = 0 1001 1100B
- 156 D = 1 1001 1100B

真值　　机器数

约定：用X表示真值，用$[X]_原$表示原码，$[X]_补$表示补码，$[X]_反$表示反码，$[X]_移$表示移码。
假设字长为8位(符号位+数值位)，最高位为符号位

纯小数原码

$x_1 = +0.8125$，$x_2 = -0.8125$　真值(十进制形式)

$x_1 = +0.1101$，$x_2 = -0.1101$　真值(二进制形式)

$[x_1]_原=\mathbf{0}.1101$，$[x_2]_原=\mathbf{1}.1101$　我们的做法：+换成0，-换成1

$[x_1]_原=\mathbf{0}.1101$，$[x_2]_原=0.1101 + 1.0000 =\mathbf{1}.1101$　计算机的做法：“加”

$$[x]_原 = \begin{cases} x & 1 > x \geqslant 0 \\ 1 - x = 1 + |x| & 0 \geqslant x > -1 \end{cases}$$

$[x_1]_原=\mathbf{0}.1101000$，$[x_2]_原=\mathbf{1}.1101000$

计算机中：**0**1101000，**1**1101000

若字长为n+1，则原码小数的表示范围为$-(1-2^{-n}) \le x \le 1-2^{-n}$（关于原点对称）

原码

符号、数值分开处理

[+0]原=00000...

[−0]原=10000...

+ 156 D = 0 1001 1100B

- 156 D = 1 1001 1100B

真值　　机器数

运算： 00001110 + 10001110

根据最高位调整成相应的无符号数运算

→ 00001110 - 00001110

约定：用X表示真值，用$[X]_{原}$表示原码，$[X]_{补}$表示补码，$[X]_{反}$表示反码，$[X]_{移}$表示移码。

假设字长为8位(符号位+数值位)，最高位为符号位

纯整数原码

$x_1 = +14$， $x_2 = -14$　　真值(十进制形式)

$x_1 = +1110$， $x_2 = -1110$　　真值(二进制形式)

$[x_1]_{原}=\mathbf{0},1110$， $[x_2]_{原}=\mathbf{1},1110$　　我们的做法：+换成0，-换成1

$[x_1]_{原}=\mathbf{0},1110$， $[x_2]_{原}=0,1110 + 1,0000 = \mathbf{1},1110$　　计算机的做法：“加”

$$[x]_{原} = \begin{cases} 0, x & 2^n > x \geqslant 0 \\ 2^n - x = 2^n + |x| & 0 \geqslant x > -2^n \end{cases}$$

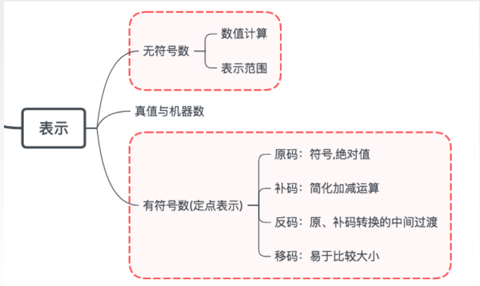
$[x_1]_{原}=\mathbf{0},0001110$， $[x_2]_{原}=\mathbf{1},0001110$

计算机中： **0**0001110， **1**0001110

若字长为$n+1$，则原码整数的表示范围为$-(2^n-1) \le x \le 2^n-1$（关于原点对称）

王道考研/CSKAOYAN.COM

知识点回顾



王道考研/CSKAOYAN.COM

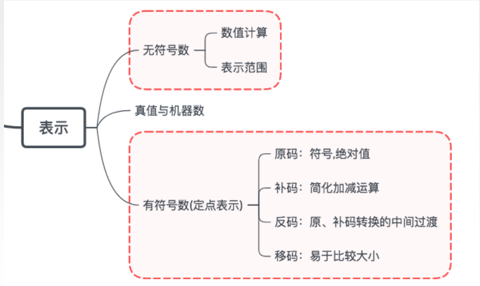
本节内容

定点数的
表示和运算

补码
反码
移码

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

加减运算

有符号数 无符号数

有符号数		无符号数
14	00001110	14
-14	+ 10001110	142
0	10011100	156
☹		☺

  00001110
+ 10001110

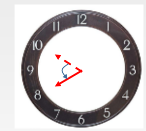
  00001110
- 00001110

  00000000

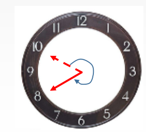
☺

王道考研/CSKAOYAN.COM

补码



10 – 2 = 8



10 + 10 = 20

$\frac{20}{12} = 1 \cdots\cdots 8$

10 + 12 – 2 = 20

20 mod $\boxed{12}$ = 8

模

王道考研/CSKAOYAN.COM

加减运算

有符号数		无符号数
14	00001110	14
-14	+ 10001110	142
0	10011100	156
☹		☺

  00001110
+ 10001110

  00001110
- 00001110 → + 1,00000000 - 00001110 → 00001110 + 11110010

  00000000 1 00000000

王道考研/CSKAOYAN.COM

加减运算

  00001110
+ 10001110

  00001110
- 00001110

  00000000

有符号数		无符号数
14	00001110	14
-14	+ 10001110	142
0	10011100	156
☹		☺

补码 正数：与原码相同
负数：符号位与原码相同，数值位由原码取反加1得到

14	00001110	14
-14	+ 11110010	242
0	1 00000000	256 mod $\boxed{1,00000000}$ = 0
☺		☺

2^8

  00001110
+ 1,00000000 - 00001110

0,00000001 + 0,11111111 - 00001110

加1 取反

$[X]_{原}$ → $[X]_{补}$：正数不变；负数符号位不变，数值位取反加1

$[X]_{补}$ → $[-X]_{补}$：连同符号位一起取反加1

王道考研/CSKAOYAN.COM

补码

对于正数，补码与原码的表示相同，$[x]_{补}=[x]_{原}$。
对于负数，原码符号位不变，数值部分按位取反，末位加1（即所谓“取反加1”）
此规则同样适用于由$[x]_{补}$求$[x]_{原}$。

$$\begin{array}{r} \mathbf{0}0001110 \\ -\ \ \mathbf{0}0001110 \\ \hline \end{array} \longrightarrow \begin{array}{r} \mathbf{0}0001110 \\ +\ \ 1,00000000 - \mathbf{0}0001110 \\ \hline \end{array}$$

纯整数补码

$$[x]_{补}=\begin{cases} 0,x & 2^n > x \geqslant 0 \\ 2^{n+1}+x=2^{n+1}-|x| & 0 \geqslant x \geqslant -2^n \end{cases} \quad (\mathrm{mod}\ \ 2^{n+1})$$

$x_1=+1010$，$x_2=-1010$，字长为8位，则其补码表示为：
$[x_1]_{补} = \mathbf{0},0001010$
$[x_2]_{补} = 2^8 - 0,0001010 = 10,0000000 - 0,0001010 = \mathbf{1},1110110 = [-x_1]_{补}$
$[x_2]_{原} = \mathbf{1},0001010$

若字长为$n+1$，则补码的表示范围为$-2^n \le x \le 2^n-1$（比原码多表示-2^n）
$[-2^n]_{补} = 10,0000 - 1,0000 = 1,0000$

王道考研/CSKAOYAN.COM

补码

对于正数，补码与原码的表示相同，$[x]_{补}=[x]_{原}$。
对于负数，原码符号位不变，数值部分按位取反，末位加1（即所谓“取反加1”）
此规则同样适用于由$[x]_{补}$求$[x]_{原}$。

$$\begin{array}{r} \mathbf{0}0001110 \\ -\ \ \mathbf{0}0001110 \\ \hline \end{array} \longrightarrow \begin{array}{r} \mathbf{0}0001110 \\ +\ \ 1,00000000 - \mathbf{0}0001110 \\ \hline \end{array}$$

纯小数补码

$$[x]_{补}=\begin{cases} x & 1 > x \geqslant 0 \\ 2+x=2-|x| & 0 > x \geqslant -1 \end{cases} \quad (\mathrm{mod}\ \ 2)$$

$x_1=+0.1001$，$x_2=-0.1001$，字长为8位，则其补码表示为：
$[x_1]_{补}=\mathbf{0}.1001000$
$[x_2]_{补} = 2 - 0.1001000 = 10.0000000 - 0.1001000 = \mathbf{1}.0111000 = [-x_1]_{补}$
$[x_1]_{原} = \mathbf{1}.1001000$

若字长为$n+1$，则补码的表示范围为$-1 \le x \le 1-2^{-n}$（比原码多表示-1）
$[-1]_{补}=10.0000-1.0000=1.0000$

王道考研/CSKAOYAN.COM

反码

对于正数，反码与原码的表示相同，$[x]_{反}=[x]_{原}$。
对于负数，原码符号位不变，数值部分按位取反，
此规则同样适用于由$[x]_{反}$求$[x]_{原}$。

表示范围：与原码一样。

纯整数反码

$$[x]_{反}=\begin{cases} 0,x & 2^n > x \geqslant 0 \\ (2^{n+1}-1)+x & 0 \geqslant x > -2^n \end{cases} \quad (\mathrm{mod}\ \ 2^{n+1}-1)$$

$x_1=+1011$，$x_2=-1011$，字长为8位，则其反码表示为：
$[x_1]_{反}= \mathbf{0},0001011$
$[x_2]_{反}= 1,1111111 - 0,0001011 = \mathbf{1},1110100$
$[x_1]_{原} = \mathbf{1},0001011$

若字长为$n+1$，则反码的表示范围为$-(2^n-1) \le x \le 2^n-1$（关于原点对称）

王道考研/CSKAOYAN.COM

反码

对于正数，反码与原码的表示相同，$[x]_{反}=[x]_{原}$。
对于负数，原码符号位不变，数值部分按位取反，
此规则同样适用于由$[x]_{反}$求$[x]_{原}$。

表示范围：与原码一样。

纯小数反码

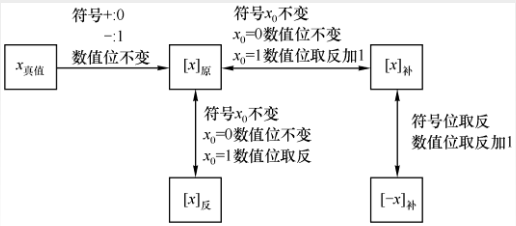
$$[x]_{反}=\begin{cases} x & 1 > x \geqslant 0 \\ (2-2^{-n})+x & 0 \geqslant x > -1 \end{cases} \quad (\mathrm{mod}\ \ 2-2^{-n})$$

$x_1=+0.0110$，$x_2=-0.0110$，字长为8位，则其反码表示为：
$[x_1]_{反}=\mathbf{0}.0110000$
$[x_2]_{反}=1.1111111 - 0.0110000 = \mathbf{1}.1001111$
$[x_1]_{原} = \mathbf{1}.0110000$

若字长为$n+1$，则反码的表示范围为$-(1-2^{-n}) \le x \le 1-2^{-n}$（关于原点对称）

王道考研/CSKAOYAN.COM

原补反相互转换



王道考研/CSKAOYAN.COM

原补反相互转换

行数	机器数	真值(十进制)			
		无符号数	原码	反码	补码
1	0000 0000	0	+0	+0	+0,-0
2	0000 0001	1	+1	+1	+1
3	0000 0010	2	+2	+2	+2
…	…	…	…	…	…
126	0111 1101	125	+125	+125	+125
127	0111 1110	126	+126	+126	+126
128	0111 1111	127	+127	+127	+127
129	1000 0000	128	-0	-127	-128
130	1000 0001	129	-1	-126	-127
131	1000 0010	130	-2	-125	-126
…	…	…	…	…	…
253	1111 1100	252	-124	-3	-4
254	1111 1101	253	-125	-2	-3
255	1111 1110	254	-126	-1	-2
256	1111 1111	255	-127	-0	-1

王道考研/CSKAOYAN.COM

移码

行数	机器数	真值(十进制)				
		无符号数	原码	反码	补码	移码
1	0000 0000	0	+0	+0	+0,-0	-128
2	0000 0001	1	+1	+1	+1	-127
3	0000 0010	2	+2	+2	+2	-126
…	…	…	…	…	…	…
126	0111 1101	125	+125	+125	+125	-3
127	0111 1110	126	+126	+126	+126	-2
128	0111 1111	127	+127	+127	+127	-1
129	1000 0000	128	-0	-127	-128	0
130	1000 0001	129	-1	-126	-127	1
131	1000 0010	130	-2	-125	-126	2
…	…	…	…	…	…	…
253	1111 1100	252	-124	-3	-4	124
254	1111 1101	253	-125	-2	-3	125
255	1111 1110	254	-126	-1	-2	126
256	1111 1111	255	-127	-0	-1	127

移码就是在真值X上加上一个常数（偏置值），通常这个常数取2^n 。

$[x]_移=2^n+x$

x_1=+10101，x_2=−10101，字长为8位，则其移码表示为：

$[x_1]_移=2^7+10101$
$=10000000+10101$
$=1,0010101$

$[x_2]_移=2^7+(-10101)$
$=10000000+(-10101)$
$=0,1101011$

王道考研/CSKAOYAN.COM

移码

行数	机器数	真值(十进制)				
		无符号数	原码	反码	补码	移码
1	0000 0000	0	+0	+0	+0,-0	-128
2	0000 0001	1	+1	+1	+1	-127
3	0000 0010	2	+2	+2	+2	-126
…	…	…	…	…	…	…
126	0111 1101	125	+125	+125	+125	-3
127	0111 1110	126	+126	+126	+126	-2
128	0111 1111	127	+127	+127	+127	-1
129	1000 0000	128	-0	-127	-128	0
130	1000 0001	129	-1	-126	-127	1
131	1000 0010	130	-2	-125	-126	2
…	…	…	…	…	…	…
253	1111 1100	252	-124	-3	-4	124
254	1111 1101	253	-125	-2	-3	125
255	1111 1110	254	-126	-1	-2	126
256	1111 1111	255	-127	-0	-1	127

移码0111 1110的真值：

1. 转换成无符号数真值：126

2. 减去偏置值1000 0000对应的无符号数真值128得到移码真值：
126 – 128 = -2

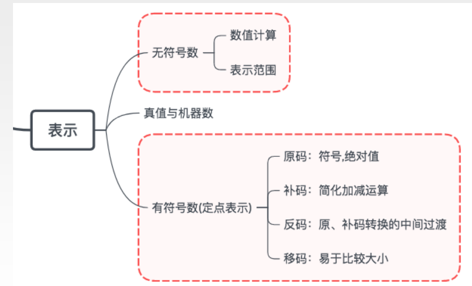
或者：
0111 1110 – 1000 0000
= 1111 1110
对应补码真值-2

王道考研/CSKAOYAN.COM

移码

真值(十进制)	补码	移码
-128	1000 0000	0000 0000
-127	1000 0001	0000 0001
-126	1000 0010	0000 0010
…	…	…
-3	1111 1101	0111 1101
-2	1111 1110	0111 1110
-1	1111 1111	0111 1111
0	0000 0000	1000 0000
1	0000 0001	1000 0001
2	0000 0010	1000 0010
3	0000 0011	1000 0011
…	…	…
124	0111 1100	1111 1100
125	0111 1101	1111 1101
126	0111 1110	1111 1110
127	0111 1111	1111 1111

知识点回顾

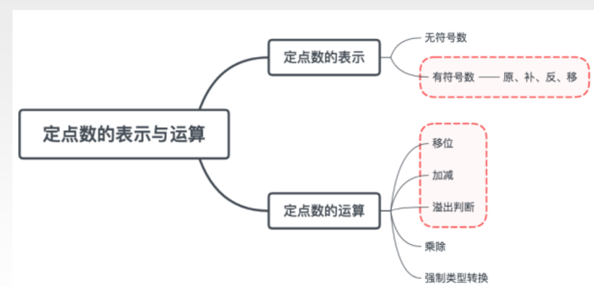


本节内容

定点数的表示和运算

移位运算

本节总览



移位运算

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以10^2

小数点右移1位：1000.，相当于乘以10，即乘以10^1

也可看作固定小数点：

数字右移2位

数字左移1位

右移n位：$\div r^n$　　左移n位：$\times r^n$

1 0 1 1 0 1 0 1

2 进制： $K_7 K_6 K_5 K_4 K_3 K_2 K_1 K_0$

$= K_7 \times 2^7 + K_6 \times 2^6 + K_5 \times 2^5 + K_4 \times 2^4 + K_3 \times 2^3 + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0$

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

1　0 1 1 0 1 0 1

移位运算

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以10^2

小数点右移1位：1000.，相当于乘以10，即乘以10^1

也可看作固定小数点：

数字右移2位

数字左移1位

右移n位：$\div r^n$　　左移n位：$\times r^n$

1 0 1 1 0 1 0 1

2 进制： $K_7 K_6 K_5 K_4 K_3 K_2 K_1 K_0$

$= K_7 \times 2^7 + K_6 \times 2^6 + K_5 \times 2^5 + K_4 \times 2^4 + K_3 \times 2^3 + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0$

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

1　0 1 1 0 1 0 1 0　　　　1 0 1 1 0 1 0　1

移位运算

r 进制： $K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

10 进制：100.0

小数点左移2位：1.000，相当于除以100，即除以10^2

小数点右移1位：1000.，相当于乘以10，即乘以10^1

也可看作固定小数点：

数字右移2位

数字左移1位

右移n位：$\div r^n$　　左移n位：$\times r^n$

1 0 1 1 0 1 0 1

2 进制： $K_7 K_6 K_5 K_4 K_3 K_2 K_1 K_0$

$= K_7 \times 2^7 + K_6 \times 2^6 + K_5 \times 2^5 + K_4 \times 2^4 + K_3 \times 2^3 + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0$

机器数采用无符号数：逻辑移位

逻辑左移时，高位移丢，低位添0；逻辑右移时，低位移丢，高位添0

0 1 1 0 1 0 1 0　　　　0 1 0 1 1 0 1 0　1

算术移位

1 0 1 1 0 1 0 1

2 进制： S $K_6 K_5 K_4 K_3 K_2 K_1 K_0$

$=(-1)^S \times (K_6 \times 2^6 + K_5 \times 2^5 + K_4 \times 2^4 + K_3 \times 2^3 + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0)$

算术移位：机器码采用有符号数

符号位不参与移位

原码：

符号位	绝对值

左移、右移都补0

1,0110101　　真值-53

左移1位(丢0)：1,1101010　　真值-106

右移1位(丢1)：1,0011010　　真值-26　　假设不丢1：1,0011010.1　　真值-26.5

再左移1位(丢1)：1,1010100　　真值-84　　假设不丢1：1, 11010100　　真值-212

再右移1位(丢0)：1,0001101　　真值-13

原码算术移位：左移丢1，运算出错；右移丢1，影响精度。

算术移位

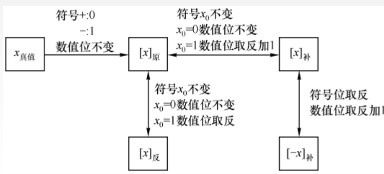
1 0 1 1 0 1 0 1

2 进制： S $K_6K_5K_4K_3K_2K_1K_0$

算术移位：机器码采用有符号数

符号位不参与移位

正数：原码、补码、反码一样 → 左移、右移都补0



负数：反码1 <—> 原码0

原码 1,0110101

反码 1,1001010

补码 1,1001011

算术移位

1 0 1 1 0 1 0 1

2 进制： S $K_6K_5K_4K_3K_2K_1K_0$

算术移位：机器码采用有符号数

符号位不参与移位

	码制	添补代码
正数	原码、补码、反码	0
负数	原码	0
	补码	左移添0
		右移添1
	反码	1

正数：原码、补码、反码一样

→ 左移、右移都补0

负数：反码1 <—> 原码0

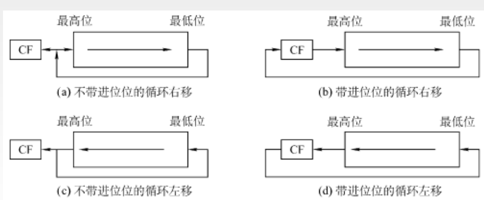
原码 1,0110101

反码 1,1001010

补码 1,1001011

循环移位

1 0 1 1 0 1 0 1



知识点回顾

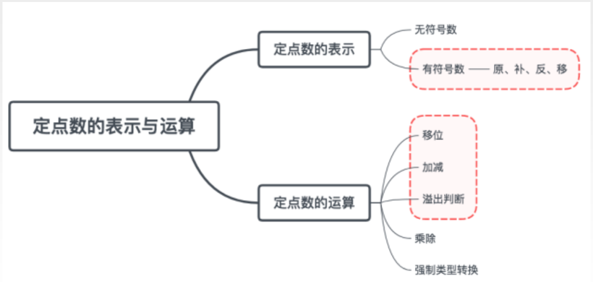
- 移位
  - 逻辑移位
  - 算术移位：左移相当于乘以基数，右移相当于除以基数
  - 空位添补规则
  - 循环移位

本节内容

定点数的表示和运算

加减运算
符号扩展
溢出判断

本节总览



符号扩展

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

原码

$A = +1111$ → $0,1111$ → $0,0001111$
$B = -11000$ → $1,11000$ → $1,0011000$

补码

$A = +1111$ → $0,1111$ → $0,0001111$
$B = -11000$ → $1,01000$ → $1,1101000$

加减运算

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

补码

$A = +1111$ → $0,1111$ → $0,0001111$
$B = -11000$ → $1,01000$ → $1,1101000$

$[A+B]_{补} = [A]_{补} + [B]_{补} = 0,0001111 + 1,1101000 = 1,1110111$
原码：1,0001001　真值-9

$[A-B]_{补} = [A]_{补} + [-B]_{补} = 0,0001111 + 0,0011000 = 0,0100111$　真值+39

$[-B]_{补}$：$[B]_{补}$连同符号位一起取反加1

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

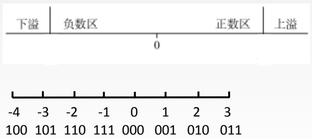
$[A+C]_{补} = 0,0001111 + 0,1111100 = 1,0001011$　真值-117
$[B-C]_{补} = 1,1101000 + 1,0000100 = 10,1101100$　真值+108

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补} = \boxed{0},0001111 + \boxed{0},1111100 = \boxed{1},0001011$　　真值-117
$[B-C]_{补} = \boxed{1},1101000 + \boxed{1},0000100 = 1\boxed{0},1101100$　　真值+108



方法一：采用一位符号位
设A的符号为A_S，B的符号为B_S，运算结果的符号为S_S，则溢出逻辑表达式为

$$V = A_S B_S \overline{S_S} + \overline{A_S}\,\overline{B_S} S_S$$

若V=0，表示无溢出；
若V=1，表示有溢出。

王道考研/CSKAOYAN.COM

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补} = \boxed{0},0001111 + \boxed{0},1111100 = \boxed{1},0001011$　　真值-117
$[B-C]_{补} = \boxed{1},1101000 + \boxed{1},0000100 = 1\boxed{0},1101100$　　真值+108

逻辑表达式
与：如ABC，表示A与B与C
仅当A、B、C均为1时，ABC为1
A、B、C中有一个或多个为0，则ABC为0

或：如A+B+C，表示A或B或C
仅当A、B、C均为0时，A+B+C为0
A、B、C中有一个或多个为1，则A+B+C为1

非：如 $\overline{A}$，表示A非
若A为1，则 $\overline{A}$ 为0
若A为0，则$\overline{A}$ 为1

方法一：采用一位符号位
设A的符号为A_S，B的符号为B_S，运算结果的符号为S_S，则溢出逻辑表达式为

$$V = \boxed{A_S B_S \overline{S_S}} + \boxed{\overline{A_S}\,\overline{B_S} S_S}$$

若V=0，表示无溢出；
若V=1，表示有溢出。

A_S为1且B_S为1且S_S为0　或　A_S为0且B_S为0且S_S为1

王道考研/CSKAOYAN.COM

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补}$ = 0,0001111 + 0,1111100 = 1,0001011　　真值-117
$[B-C]_{补}$ = 1,1101000 + 1,0000100 =10,1101100　　真值+108

方法二：采用一位符号位，根据数据位进位情况判断溢出

	符号位的进位C_S	最高数位的进位C_1
上溢	0	1
下溢	1	0

即：C_S与C_1不同时有溢出
处理“不同”的逻辑符号：异或⊕
溢出逻辑判断表达式为$V=C_S \oplus C_1$
若V=0，表示无溢出；V=1，表示有溢出。

异或逻辑：不同为1，相同为0
$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

王道考研/CSKAOYAN.COM

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补}$ = 0,0001111 + 0,1111100 = 1,0001011　　真值-117
$[B-C]_{补}$ = 1,1101000 + 1,0000100 =10,1101100　　真值+108

方法三：采用双符号位
正数符号为00，负数符号为11

$[A+C]_{补}$ = 00,0001111 + 00,1111100 = 01,0001011　　上溢
$[B-C]_{补}$ = 11,1101000 + 11,0000100 = 10,1101100　　下溢

记两个符号位为$S_{S1}S_{S2}$，则$V=S_{S1} \oplus S_{S2}$
若V=0，表示无溢出；若V=1，表示有溢出。

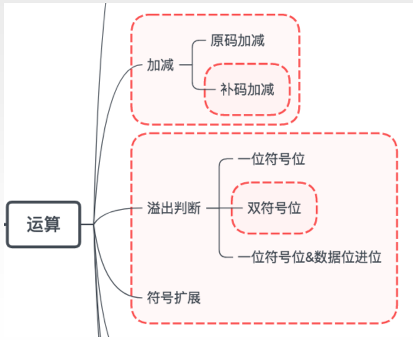
$[A+B]_{补}$ = 00,0001111 + 11,1101000 = 11,1110111
$[A-B]_{补}$ = 00,0001111 + 00,0011000 = 00,0100111

11,1110111　右移1位：11,1111011
00,0100111　左移1位：00,1001110
00,0100111　左移2位：01,0011100　上溢

采用双符号位的移位运算：低位符号位参与移位，高位符号位代表真正的符号

王道考研/CSKAOYAN.COM

知识点回顾



王道考研/CSKAOYAN.COM

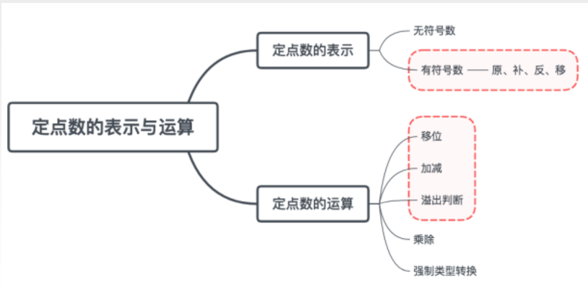
本节内容

定点数的表示和运算

乘法运算

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

原码一位乘法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），$x = -0.1101$，$y = +0.1011$，采用原码一位乘法求$x\cdot y$

符号：一正一负，结果为负，即符号位 $= x_s \oplus y_s$

$$
\begin{array}{r}
0.1101\\
\times\ 0.1011\\
\hline
\boxed{1101}\\
\boxed{1101}\ \\
\boxed{0000}\ \ \\
\boxed{1101}\ \ \ \\
\hline
\boxed{0.10001111}
\end{array}
$$

原码一位乘法

$|x|$=00.1101，$|y|$=00.1011

$$
\begin{array}{lr}
 & 00.0000\\
+|x| & 00.1101\\
\hline
 & 00.1101\\
\text{右移} & 00.0110\\
+|x| & 00.1101\\
\hline
 & 01.0011\\
\text{右移} & 00.1001
\end{array}
$$

ACC	MQ	
00000	101$\boxed{1}$	
01101	101$\boxed{1}$	
00110	110$\boxed{1}$	1
10011	110$\boxed{1}$	1
01001	111$\boxed{0}$	11

王道考研/CSKAOYAN.COM

原码一位乘法

设机器字长为5位（含1位符号位，$n=4$），$x = -0.1101$，$y = +0.1011$，采用原码一位乘法求$x\,y$

符号：一正一负，结果为负，即符号位 $= x_s \oplus y_s$

```
   0.1101
   0.1011
 ---------
     1101
    1101
   0000
  1101
 ---------
 0.10001111
```

符号位$P_s = x_s \oplus y_s = 1 \oplus 0 = 1$
$[x\,y]_{原} = 1.10001111$
即$x\,y = -0.10001111$

原码一位乘法
$|x|=00.1101$，$|y|=00.1011$

```
        00.0000
+|x|    00.1101
---------------
        00.1101
右移    00.0110
+|x|    00.1101
---------------
        01.0011
右移    00.1001
+0      00.0000
---------------
        00.1001
右移    00.0100
+|x|    00.1101
---------------
        01.0001
右移    00.1000
```

ACC	MQ	
00000	1011	
01101	1011	
00110	1101	1
10011	1101	1
01001	1110	11
01001	1110	11
00100	1111	011
10001	1111	011
01000	1111	1011

王道考研/CSKAOYAN.COM

补码一位乘法

设机器字长为5位（含1位符号位，$n=4$），$x = -0.1101$，$y = +0.1011$，采用Booth算法求$x \cdot y$

$[x]_补=11.0011$，$[-x]_补=00.1101$，$[y]_补=0.1011$

```
           00.0000
+ [-x]补   00.1101
------------------
           00.1101
右移       00.0110
+ 0        00.0000
------------------
           00.0110
右移       00.0011
+ [x]补    11.0011
------------------
           11.0110
右移       11.1011
+ [-x]补   00.1101
------------------
           00.1000
右移       00.0100
+ [x]补    11.0011
------------------
           11.0111
```

ACC	MQ	
000000	010110	
001101	010110	
000110	101011	0
000110	101011	0
000011	010101	10
110110	010101	10
111011	001010	110
001000	001010	110
000100	000101	0110
110111	000101	0110

y_n（高位）	y_{n+1}（低位）	操　作
0	0	部分积右移一位
0	1	部分积加$[X]_补$ 右移一位
1	0	部分积加$[-X]_补$ 右移一位
1	1	部分积右移一位

根据$y_{n+1} - y_n$判断：
$y_{n+1} - y_n = 0$，加0，右移一位
$y_{n+1} - y_n = 1$，加$[X]_补$，右移一位
$y_{n+1} - y_n = -1$，加$[-X]_补$，右移一位

$[x\,y]_补 = 1.01110001$
即$x\,y = -0.10001111$

王道考研/CSKAOYAN.COM

乘法运算总结回顾

乘法类型	符号位			累加次数	移位		
	参与运算	部分积	乘数		方向	次数	每位次数
原码一位乘法	否	2位	0位	n	右	n	1
补码Booth乘法	是	2位	1位	n+1	右	n	1

王道考研/CSKAOYAN.COM

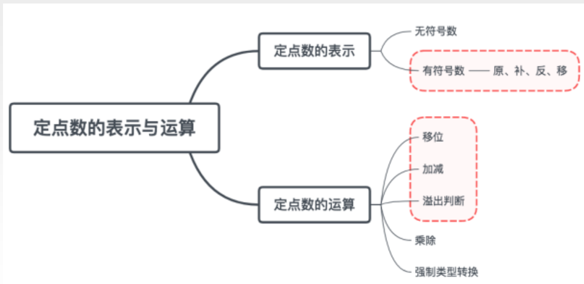
本节内容

定点数的表示和运算

除法运算

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

除法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，求x/y

```
               0.1101
       0.1101 ) 0.10110
                0.01101
                0.010010
                0.001101
                0.00010100
                0.00001101
                0.000001110
```

x/y结果为0.1101，余数为0.00000111

王道考研/CSKAOYAN.COM

原码恢复余数法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，采用原码加减交替除法求x/y

$|x|$=00.1011，$|y|$=00.1101，$[|y|]_补$=00.1101，$[-|y|]_补$=11.0011

	被除数	商
	00.1011	
$+[-\|y\|]_补$	11.0011	
	11.1110	0
$+[\|y\|]_补$	00.1101	
	00.1011	
左移	01.0110	
$+[-\|y\|]_补$	11.0011	
	00.1001	01
左移	01.0010	
$+[-\|y\|]_补$	11.0011	
	00.0101	011
	…	…

左移n次，上商n+1次

王道考研/CSKAOYAN.COM

原码恢复余数法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，采用原码加减交替除法求x/y

$|x|$=00.1011，$|y|$=00.1101，$[|y|]_补$=00.1101，$[-|y|]_补$=11.0011

	被除数		商
	00.1011		
$+[-\|y\|]_补$	11.0011		
	11.1110	a	0
$+[\|y\|]_补$	00.1101	b	
	00.1011	a+b	
左移	01.0110	(a+b)×2 = 2a + 2b	
$+[-\|y\|]_补$	11.0011	(a+b)×2 - b = 2a + 2b – b = 2a + b	
	00.1001		01
左移	01.0010		
$+[-\|y\|]_补$	11.0011		
	00.0101		011
	…		…

左移n次，上商n+1次

王道考研/CSKAOYAN.COM

本节内容

定点数的表示和运算

原码加减交替法
补码加减交替法

王道考研/CSKAOYAN.COM

原码恢复余数法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，采用原码加减交替除法求x/y

$|x|$=00.1011，$|y|$=00.1101，$[|y|]_{补}$=00.1101，$[-|y|]_{补}$=11.0011

	被除数	
	00.1011	
$+[-\|y\|]_{补}$	11.0011	
	11.1110	a
$+[\|y\|]_{补}$	00.1101	b
	00.1011	a+b
左移	01.0110	(a+b)×2 = 2a + 2b
$+[-\|y\|]_{补}$	11.0011	(a+b)×2 - b = 2a + 2b – b = 2a + b
	00.1001	
左移	01.0010	
$+[-\|y\|]_{补}$	11.0011	
	00.0101	
	…	

a左移，加b

不恢复余数法：
被除数减去除数，即$|x|+[-|y|]_{补}$，
若结果为正，商1，左移，再减去除数；
若结果为负，商0，左移，再加上除数。

… 左移n次，上商n+1次

王道考研/CSKAOYAN.COM

原码不恢复余数法

符号位	绝对值

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，采用原码加减交替除法求x/y

$|x|$=00.1011，$|y|$=00.1101，$[|y|]_{补}$=00.1101，$[-|y|]_{补}$=11.0011

	被除数
	00.1011
$+[-\|y\|]_{补}$	11.0011
	11.1110
左移	11.1100
$+[\|y\|]_{补}$	00.1101
	00.1001
左移	01.0010
$+[-\|y\|]_{补}$	11.0011
	00.0101
左移	00.1010
$+[-\|y\|]_{补}$	11.0011

ACC	MQ
01011	00000
11110	00000
11100	00000
01001	00001
10010	00010
00101	00011
01010	00110

王道考研/CSKAOYAN.COM

原码不恢复余数法

符号位与数值位分开处理

设机器字长为5位（含1位符号位，n=4），x=0.1011，y=0.1101，采用原码加减交替除法求x/y

$|x|$=00.1011，$|y|$=00.1101，$[|y|]_{补}$=00.1101，$[-|y|]_{补}$=11.0011

$Q_s=x_s \oplus y_s=0\oplus 0=0$
得x/y=+0.1101
余0.0111×2^{-4}

	被除数
	00.1011
$+[-\|y\|]_{补}$	11.0011
	11.1110
左移	11.1100
$+[\|y\|]_{补}$	00.1101
	00.1001
左移	01.0010
$+[-\|y\|]_{补}$	11.0011
	00.0101
左移	00.1010
$+[-\|y\|]_{补}$	11.0011
	11.1101
左移	11.1010
$+[\|y\|]_{补}$	00.1101
	00.0111

若余数为负，
需$+[|y|]_{补}$
得到正确余数

ACC	MQ
01011	00000
11110	00000
11100	00000
01001	00001
10010	00010
00101	00011
01010	00110
11101	00110
11010	01100
00111	01101

王道考研/CSKAOYAN.COM

补码加减交替法

设机器字长为5位（含1位符号位，$n=4$），$x=+0.1000$，$y=-0.1011$，采用补码加减交替除法求x/y

$[x]_{补}=00.1000$，$[y]_{补}=\boxed{11}.0101$，$[-y]_{补}=00.1011$　　$[x/y]_{补}=1.0101$，余0.0111×2^{-4}

	被除数
	00.1000
$+[y]_{补}$	11.0101
	$\boxed{11}$.1101
左移	11.1010
$+[-y]_{补}$	00.1011
	$\boxed{00}$.0101
左移	00.1010
$+[y]_{补}$	11.0101
	$\boxed{11}$.1111
左移	11.1110
$+[-y]_{补}$	00.1011
	$\boxed{00}$.1001
左移	01.0010
$+[y]_{补}$	11.0101
	00.0111

ACC	MQ	
01000	00000	
11101	00001	
11010	00010	
00101	00010	
01010	00100	
11111	00101	
11110	01010	
01001	01010	
10010	10100	
00111	10101	末位恒置1

被除数和除数同号，则被除数减去除数；异号则被除数加上除数。

余数和除数同号，商1，余数左移一位减去除数；余数和除数异号，商0，余数左移一位加上除数。重复n次。

王道考研/CSKAOYAN.COM

除法运算总结回顾

除法类型	符号位参与运算	加减次数	移位		说明
			方向	次数	
原码加减交替法	否	N+1或N+2	左	N	若最终余数为负，需恢复余数
补码加减交替法	是	N+1	左	N	商末位恒置1

王道考研/CSKAOYAN.COM

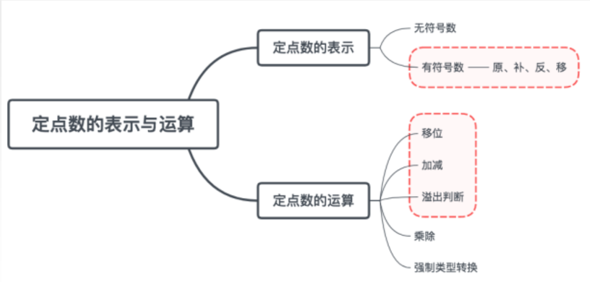
本节内容

定点数的表示和运算

强制类型转换

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

强制类型转换

```
void main(){
    short x=-4321;                    //short型占用2个字节
    unsigned short y=(unsigned short)x;

    int a=165537, b=-34991;           //int型占用4个字节
    short c=(short)a, d=(short)b;     //short型占用2个字节

    short x=-4321;
    int m=x;
    unsigned short n=(unsigned short)x;
    unsigned int p=n;
}
```

x：1110 1111 0001 1111　　y：1110 1111 0001 1111　真值61215

无符号数与有符号数：
不改变数据内容，
改变解释方式。

长整数变短整数：
高位截断，保留低位。

a：0x000286a1
c：0x86a1　　真值-31071
b：0xffff7751
d：0x7751　　真值30545

短整数变长整数：
符号扩展。

x：1110 1111 0001 1111
0xef1f
m：1111 1111 1111 1111 1110 1111 0001 1111
0xffffef1f　　真值-4321
n：1110 1111 0001 1111　　0xef1f　　真值61215
p：0000 0000 0000 0000 1110 1111 0001 1111
0x0000ef1f　　真值61215

王道考研/CSKAOYAN.COM

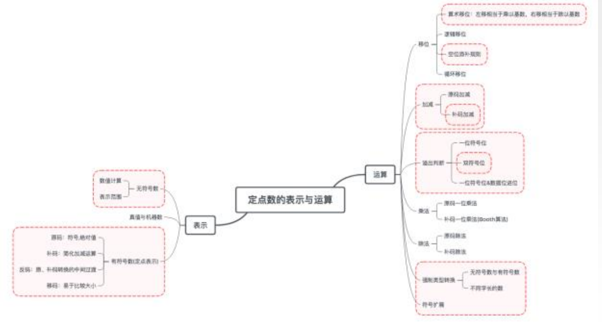
本节内容

定点数的表示和运算

本节回顾

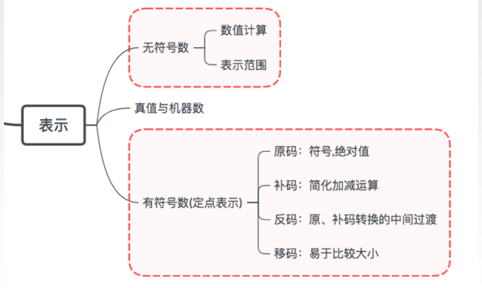
王道考研/CSKAOYAN.COM

本节回顾



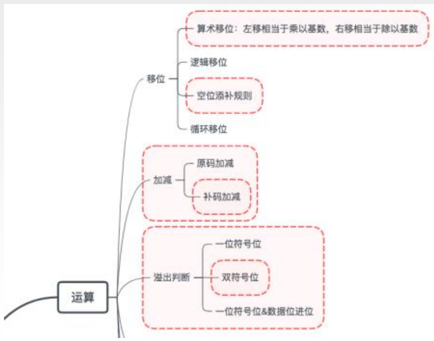
王道考研/CSKAOYAN.COM

本节回顾



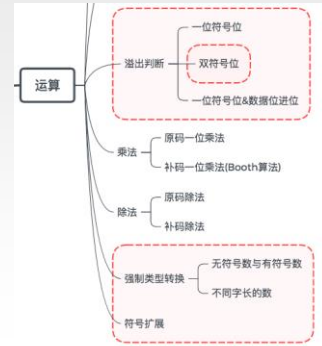
王道考研/CSKAOYAN.COM

本节回顾



王道考研/CSKAOYAN.COM

本节回顾



王道考研/CSKAOYAN.COM

王道考研——组成原理

WWW.CSKAOYAN.COM

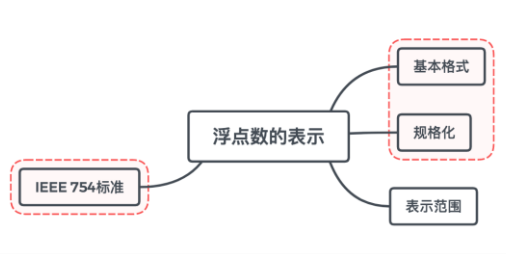
第二章 数据的表示和运算

本节内容

浮点数的表示与运算

表示

本节总览



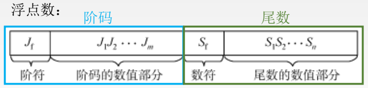
浮点数的表示

r 进制： $K_n K_{n-1} \dots K_2 K_1 K_0 K_{-1} K_{-2} \dots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \dots + K_{-m} \times r^{-m}$

定点数：如纯小数0.1011和纯整数11110

浮点数：



阶码：常用补码或移码表示

尾数：常用原码或补码表示

浮点数的真值： $N = \boxed{r}^E \times M$

阶码的底，通常为2

十进制：299792458m/s = 2.998× 10^8m/s

阶码E反映浮点数的**表示范围**及小数点的实际位置；
尾数M的数值部分的位数n反映浮点数的**精度**。

浮点数的表示

r 进制：$K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

定点数：如纯小数0.1011和纯整数11110

浮点数：

阶码		尾数	
J_f	$J_1J_2\cdots J_m$	S_f	$S_1S_2\cdots S_n$
阶符	阶码的数值部分	数符	尾数的数值部分

浮点数的真值：$N = \boxed{r}^E \times M$

阶码的底，通常为2

阶码E反映浮点数的**表示范围**及小数点的实际位置；
尾数M的数值部分的位数n反映浮点数的**精度**。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值
a = 0,01;1.1001
b = 0,01;0.01001

a：阶码0,01对应真值+1
尾数1.1001对应真值-0.0111 = - $(2^{-2}+ 2^{-3} + 2^{-4})$

或者理解为-111右移4位：$-\frac{7}{2^4} = -\frac{7}{16}$

所以$a = 2^1 \times (-0.0111) = 2^1 \times (-\frac{7}{16}) = -\frac{7}{8}$

1B的存储空间

0	0	1	1	1	0	0	1

王道考研/CSKAOYAN.COM

浮点数的表示

r 进制：$K_n K_{n-1} \ldots K_2 K_1 K_0 K_{-1} K_{-2} \ldots K_{-m}$

$= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \ldots + K_{-m} \times r^{-m}$

定点数：如纯小数0.1011和纯整数11110

浮点数：

阶码		尾数	
J_f	$J_1J_2\cdots J_m$	S_f	$S_1S_2\cdots S_n$
阶符	阶码的数值部分	数符	尾数的数值部分

浮点数的真值：$N = \boxed{r}^E \times M$

阶码的底，通常为2

阶码E反映浮点数的**表示范围**及小数点的实际位置；
尾数M的数值部分的位数n反映浮点数的**精度**。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值
a = 0,01;1.1001
b = 0,01;0.01001

b：阶码0,01对应真值+1
尾数0.01001对应真值+0.01001 = + $(2^{-2}+ 2^{-5})$

或者理解为+1001右移5位：$+\frac{9}{2^5} = +\frac{9}{32}$

所以$b = 2^1 \times (+0.01001) = 2^1 \times \frac{9}{32} = \frac{9}{16}$

1B的存储空间

0	0	1	0	0	1	0	0

王道考研/CSKAOYAN.COM

浮点数的规格化

浮点数：

阶码		尾数	
J_f	$J_1J_2\cdots J_m$	S_f	$S_1S_2\cdots S_n$
阶符	阶码的数值部分	数符	尾数的数值部分

浮点数的真值：$N = \boxed{r}^E \times M$

阶码的底，通常为2

阶码E反映浮点数的**表示范围**及小数点的实际位置；
尾数M的数值部分的位数n反映浮点数的**精度**。

阶码：常用补码或移码表示

尾数：常用原码或补码表示

例：阶码、尾数均用补码表示，求a、b的真值
a = 0,01;1.1001
b = 0,01;0.01001

b：阶码0,01对应真值+1
尾数0.01001对应真值+0.01001 = + $(2^{-2}+ 2^{-5})$

或者理解为+1001右移5位：$+\frac{9}{2^5} = +\frac{9}{32}$

所以$b = 2^1 \times (+0.01001) = 2^1 \times \frac{9}{32} = \frac{9}{16}$

1B的存储空间

0	0	1	0	0	1	0	0

$b = 2^1 \times (+0.01001)$
$= 2^2 \times (+0.10010)$

0	1	0	0	1	0	0	1

王道考研/CSKAOYAN.COM

浮点数的规格化

规格化：规定尾数的最高数位必须是一个有效值 。

左规：当浮点数运算的结果为非规格化时要进行规格化处理，
将尾数左移一位，阶码减1（基数为2时）。

右规：当浮点数运算的结果尾数出现溢出（双符号位为01或10）时，
将尾数右移一位，阶码加1（基数为2时）。

例：a = 010;00. 1100，b = 010;00. 1000，求a+b

$a = 2^2 \times 00.1100$，$b = 2^2 \times 00.1000$

$a+b = 2^2 \times 00.1100 + 2^2 \times 00.1000$
$= 2^2 \times (00.1100 + 00.1000)$
$= 2^2 \times 01.0100$
$= 2^3 \times 00.1010$

0	1	1	0	1	0	1	0

规格化浮点数的尾数M的绝对值应满足：$1/r \le |M| \le 1$

如果$r=2$，则有$1/2 \le |M| \le 1$

王道考研/CSKAOYAN.COM

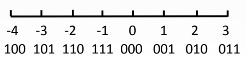
规格化浮点数的特点

规格化浮点数的尾数M的绝对值应满足：$1/r \le |M| \le 1$

如果$r=2$，则有$1/2 \le |M| \le 1$

1. 原码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。
尾数的表示范围为$-(1-2^{-n}) \le M \le -1/2$。

2. 补码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。
尾数的表示范围为$-1 \le M \le -(1/2+2^{-n})$。



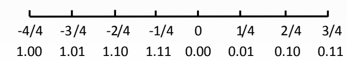
规格化浮点数的特点

规格化浮点数的尾数M的绝对值应满足：$1/r \le |M| \le 1$

如果$r=2$，则有$1/2 \le |M| \le 1$

1. 原码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。
尾数的表示范围为$-(1-2^{-n}) \le M \le -1/2$。

2. 补码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。
尾数的表示范围为$-1 \le M \le -(1/2+2^{-n})$。



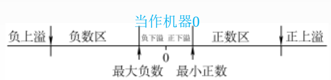
浮点数的溢出

规格化浮点数的尾数M的绝对值应满足：$1/r \le |M| \le 1$

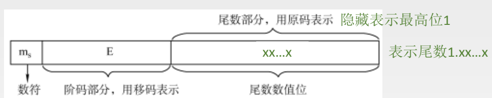
如果$r=2$，则有$1/2 \le |M| \le 1$

1. 原码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.1××…×的形式，其最大值表示为1.10…0；最小值表示为1.11…1。
尾数的表示范围为$-(1-2^{-n}) \le M \le -1/2$。

2. 补码规格化后：
正数为0.1××…×的形式，其最大值表示为0.11…1；最小值表示为0.10…0。
尾数的表示范围为$1/2 \le M \le (1-2^{-n})$。
负数为1.0××…×的形式，其最大值表示为1.01…1；最小值表示为1.00…0。
尾数的表示范围为$-1 \le M \le -(1/2+2^{-n})$。



IEEE 754标准



类　型	数　符	阶　码	尾 数 数 值	总 位 数	偏 置 值	
					十 六 进 制	十 进 制
短浮点数	1	8	23	32	7FH	127
长浮点数	1	11	52	64	3FFH	1023
临时浮点数	1	15	64	80	3FFFH	16383

1000 0001 1100 1010 0101 0000 1000 0000

1000 0001 1100 1010 0101 0000 1000 0000
0000 0000 0001 1111 0000 0000 0000 0000

规格化的短浮点数的真值为：$(-1)^s \times 1.M \times 2^{E-127}$
规格化长浮点数的真值为：$(-1)^s \times 1.M \times 2^{E-1023}$

IEEE 754标准

类型	数符	阶码	尾数数值	总位数	偏置值	
					十六进制	十进制
短浮点数	1	8	23	32	7FH	127
长浮点数	1	11	52	64	3FFH	1023
临时浮点数	1	15	64	80	3FFFH	16383

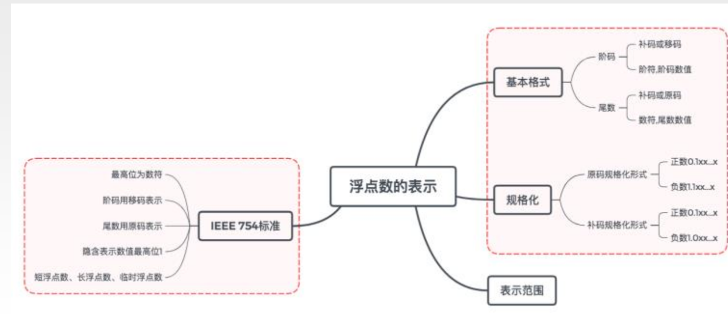
一些规定(短浮点数为例)：

1. E=0且M=0，则真值为0
2. E=0且M≠0，为非规格化数，真值 = $(-1)^s \times 0.M \times 2^{-126}$
3. 1≤E≤254时，真值 = $(-1)^s \times 1.M \times 2^{E-127}$
4. E=255且M≠0时，真值为‘NaN’(非数值)
5. E=255且M=0时，真值为正无穷或负无穷(看符号位)

格式	最小值	最大值
单精度	E=1，M=0：$1.0 \times 2^{1-127}=2^{-126}$	E=254，M=.11...1：$1.11...1 \times 2^{254-127}=2^{127} \times (2-2^{-23})$
双精度	E=1，M=0：$1.0 \times 2^{1-1023}=2^{-1022}$	E=2046，M=.11...1：$1.11...1 \times 2^{2046-1023}=2^{1023} \times (2-2^{-52})$

王道考研/CSKAOYAN.COM

本节回顾



王道考研/CSKAOYAN.COM

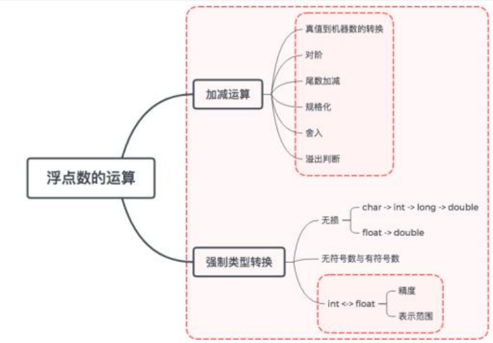
本节内容

浮点数的
表示与运算

加减运算
强制类型转换

王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM

浮点数的加减运算

浮点数加减运算步骤：
1. 对阶
2. 尾数加减
3. 规格化
4. 舍入
5. 判溢出

浮点数的加减运算

例：已知十进制数X=−5/256、Y=+59/1024，按机器补码浮点运算规则计算X−Y，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

扩展：11.011000000
双符号位补码：11.011 双符号位补码：11011
补码：1.011 补码：1011

用补码表示阶码和尾数
0. 转换格式
5D = 101B，1/256 = 2^{-8} → X = - 101 × 2^{-8} = - 0.101 × 2^{-5} = - 0.101 × 2^{-101}
59D = 111011B，1/1024 = 2^{-10} → Y = + 111011 × 2^{-10} = + 0.111011 × 2^{-4} = + 0.111011 × 2^{-100}
X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：
1. 对阶
2. 尾数加减
3. 规格化
4. 舍入
5. 判溢出

浮点数的加减运算

例：已知十进制数X=−5/256、Y=+59/1024，按机器补码浮点运算规则计算X−Y，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数
0. 转换格式
5D = 101B，1/256 = 2^{-8} → X = - 101 × 2^{-8} = - 0.101 × 2^{-5} = - 0.101 × 2^{-101}
59D = 111011B，1/1024 = 2^{-10} → Y = + 111011 × 2^{-10} = + 0.111011 × 2^{-4} = + 0.111011 × 2^{-100}
X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：
1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数每右移一位，阶码加1
   ① 求阶差：$[\Delta E]_{补}$=11011+00100=11111，知ΔE=−1
   ② 对阶：X：11011,11.011000000 → 11100,11. 101100000 X = - 0.0101 × 2^{-100}
2. 尾数加减
3. 规格化
4. 舍入
5. 判溢出

浮点数的加减运算

例：已知十进制数X=−5/256、Y=+59/1024，按机器补码浮点运算规则计算X−Y，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数
0. 转换格式
5D = 101B，1/256 = 2^{-8} → X = - 101 × 2^{-8} = - 0.101 × 2^{-5} = - 0.101 × 2^{-101}
59D = 111011B，1/1024 = 2^{-10} → Y = + 111011 × 2^{-10} = + 0.111011 × 2^{-4} = + 0.111011 × 2^{-100}
X：11011,11.011000000 Y：11100,00.111011000

浮点数加减运算步骤：
1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数每右移一位，阶码加1
   ① 求阶差：$[\Delta E]_{补}$=11011+00100=11111，知ΔE=−1
   ② 对阶：X：11011,11.011000000 → 11100,11. 101100000 X = - 0.0101 × 2^{-100}
2. 尾数加减 -Y：11100,11.000101000
   X-Y：11100, 10.110001000

   11.101100000
   \+ 11.000101000
   10.110001000

   X-Y
   = (-0.0101 × 2^{-100})- (+0.111011 × 2^{-100})
   = (-0.0101-0.111011) × 2^{-100}
   = -1.001111 × 2^{-100}
3. 规格化
4. 舍入
5. 判溢出

浮点数的加减运算

例：已知十进制数X=−5/256、Y=+59/1024，按机器补码浮点运算规则计算$X−Y$，结果用二进制表示，浮点数格式如下：阶符取2位，阶码取3位，数符取2位，尾数取9位

用补码表示阶码和尾数

0. 转换格式

5D = 101B，1/256 = 2^{-8} → X = - 101 × 2^{-8} = - 0.101 × 2^{-5} = - 0.101 × 2^{-101}

59D = 111011B，1/1024 = 2^{-10} → Y = + 111011 × 2^{-10} = + 0.111011 × 2^{-4} = + 0.111011 × 2^{-100}

X：11011,11.011000000　　Y：11100,00.111011000

浮点数加减运算步骤：

1. 对阶 使两个数的阶码相等，小阶向大阶看齐，尾数每右移一位，阶码加1
   ① 求阶差：$[\Delta E]_{补}$=11011+00100=11111，知ΔE=−1
   ② 对阶：X：11011,11.011000000 → 11100,11.101100000　　X = - 0.0101 × 2^{-100}
2. 尾数加减 -Y：11100,11.000101000
   X-Y：11100, 10.110001000

   　　11.101100000
   \+ 11.000101000
   　　10.110001000
3. 规格化
   X-Y：11100, 10.110001000 → $\boxed{1}$1101,11.011000100
4. 舍入 无舍入
5. 判溢出 常阶码，无溢出，结果真值为2^{-3}×$(−0.1001111)_2$

X-Y
= (-0.0101 × 2^{-100})- (+0.111011 × 2^{-100})
= (-0.0101-0.111011) × 2^{-100}
= -1.001111 × 2^{-100}
= -0.1001111 × 2^{-011}

王道考研/CSKAOYAN.COM

浮点数的加减运算-舍入

“0”舍“1”入法：类似于十进制数运算中的“四舍五入”法，即在尾数右移时，被移去的最高数值位为0，则舍去；被移去的最高数值位为1，则在尾数的末位加1。这样做可能会使尾数又溢出，此时需再做一次右规。

恒置“1”法：尾数右移时，不论丢掉的最高数值位是“1”还是“0”，都使右移后的尾数末位恒置“1”。这种方法同样有使尾数变大和变小的两种可能。

浮点数加减运算步骤：

1. 对阶
2. 尾数加减　如：加减结果为11100,10.110001011
3. 规格化　0舍1入：11100,10.110001011 → 11101,11.011000101 1
   　　→ 11101,11.011000110 1
   恒置1:11100,10.110001011 → 11101,11.011000101 1
   　　→ 11101,11.011000101 1
4. 舍入
5. 判溢出

王道考研/CSKAOYAN.COM

强制类型转换

类型	16位机器	32位机器	64位机器
char	8	8	8
short	16	16	16
int	16	32	32
long	32	32	64
long long	64	64	64
float	16	32	32
double	64	64	64

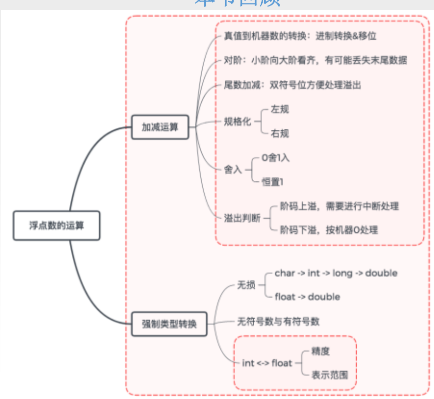
char → int → long → double
float → double
范围、精度从小到大，转换过程没有损失

int → float：可能损失精度
float → int：可能溢出及损失精度

32位
int：表示整数，范围 -2^{31} ～ 2^{31}-1 ，有效数字32位
float：表示整数及小数，范围 ±[2^{-126} ～ 2^{127}×($2-2^{-23}$)]，有效数字23+1=24位

王道考研/CSKAOYAN.COM

本节回顾

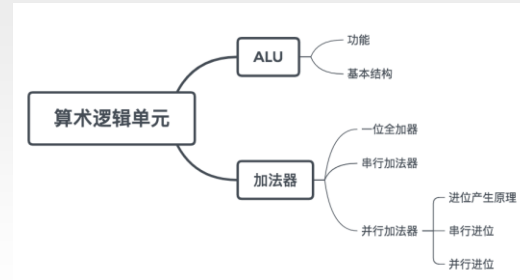


王道考研/CSKAOYAN.COM

本节内容

算术逻辑单元

本节总览



算术逻辑单元

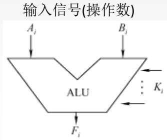
Arithmetic and Logic Unit

算术运算：加、减、乘、除等

逻辑运算：与、或、非、异或等

辅助功能：移位、求补等

输入信号(操作数)

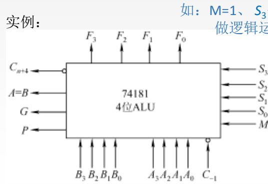


控制信号
(指令译码产生)

输出信号(运算结果)

实例：

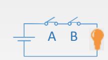
如：M=1、$S_3 \sim S_0$=1001时，做逻辑运算A⊕B

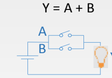


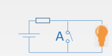
逻辑符号

	与	或	非
表达式	Y = A · B	Y = A + B	$Y = \overline{A}$

电路逻辑







真值表

与：

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

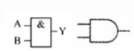
或：

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

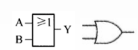
非：

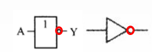
A	Y
0	0
1	1

电路符号



与门的天然逻辑：“屏蔽”





复合逻辑

反演律：
$\overline{A+B}=\overline{A}\cdot\overline{B}$
$\overline{A\cdot B}=\overline{A}+\overline{B}$

与非 半导体电路实现起来更方便 或非 异或

表达式：与非 $Y=\overline{A\cdot B}$，$\overline{A}+\overline{B}$；或非 $Y=\overline{A+B}$，$\overline{A}\cdot\overline{B}$；异或 $Y=A\oplus B$

真值表

0 1

与非：

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

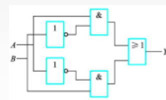
或非：

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

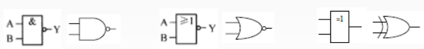
异或：

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

A和B不同
—> A=0且B=1或A=1且B=0
—> $\overline{A}\cdot B + A\cdot\overline{B}$



电路符号



与门的天然逻辑“屏蔽”

异或的天然逻辑“加法”“奇偶”

王道考研/CSKAOYAN.COM

复合逻辑

反演律：
$\overline{A+B}=\overline{A}\cdot\overline{B}$
$\overline{A\cdot B}=\overline{A}+\overline{B}$

与非 半导体电路实现起来更方便 或非 异或

表达式：与非 $Y=\overline{A\cdot B}$，$\overline{A}+\overline{B}$；或非 $Y=\overline{A+B}$，$\overline{A}\cdot\overline{B}$；异或 $Y=A\oplus B$

真值表

0 1

与非：

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

或非：

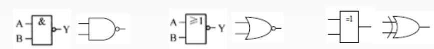
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

异或：

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

二进制加法
0 + 0 = 0 0
0 + 1 = 0 1
1 + 0 = 0 1
1 + 1 = 1 0

电路符号



与门的天然逻辑“屏蔽”

异或的天然逻辑“加法”“奇偶”

王道考研/CSKAOYAN.COM

复合逻辑

反演律：
$\overline{A+B}=\overline{A}\cdot\overline{B}$
$\overline{A\cdot B}=\overline{A}+\overline{B}$

与非 半导体电路实现起来更方便 或非 异或 同或

表达式：与非 $Y=\overline{A\cdot B}$，$\overline{A}+\overline{B}$；或非 $Y=\overline{A+B}$，$\overline{A}\cdot\overline{B}$；异或 $Y=A\oplus B$；同或 $Y=A\odot B$

真值表

0 1

与非：

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

或非：

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

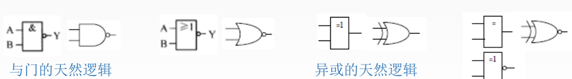
异或：

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

同或：

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

电路符号



与门的天然逻辑“屏蔽”

异或的天然逻辑“加法”“奇偶”

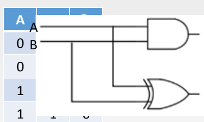
王道考研/CSKAOYAN.COM

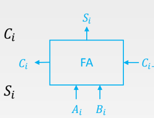
组合逻辑电路设计-一位全加器

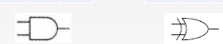
100110...0110
+ 101100...1010
******...0000

A	B	C_i
0	0	0
0	1	0
1	0	0
1	1	1

A		
0		
0		
1		
1		





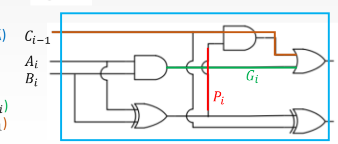


A_i
B_i
C_{i-1}

S_i： 输入中有奇数个1时为1(异或)

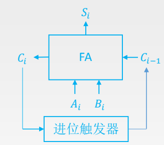
C_i： 思路1. 输入中至少2个1
思路2. 进位来源
-产生(来自本级A_i和B_i)
-传递(来自前一级C_{i-1})

$C_i = G_i + P_i\ C_{i-1}$



王道考研/CSKAOYAN.COM

串行加法器

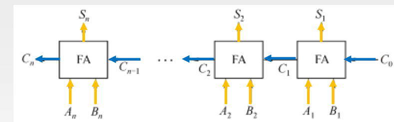


串行加法器：只有一个全加器，数据逐位串行送入加法器中进行运算。
进位触发器用来寄存进位信号，以便参与下一次运算。

如果操作数长n位，加法就要分n次进行，每次产生一位和，并且串行逐位地送回寄存器。

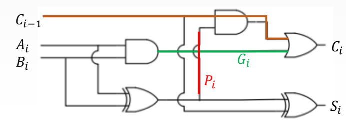
王道考研/CSKAOYAN.COM

并行加法器



串行进位的并行加法器：把n个全加器串接起来，就可进行两个n位数的相加。

串行进位又称为行波进位，每一级进位直接依赖于前一级的进位，即进位信号是逐级形成的。



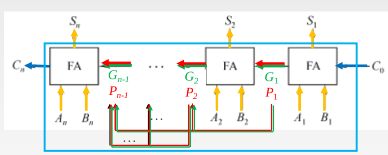
$C_i = G_i + P_i C_{i-1}$

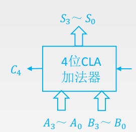
$G_i = A_iB_i$　　$P_i = A_i \oplus B_i$

$C_1=G_1+P_1C_0$
$C_2=G_2+P_2C_1=G_2+P_2G_1+P_2P_1C_0$
$C_3=G_3+P_3C_2=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1C_0$

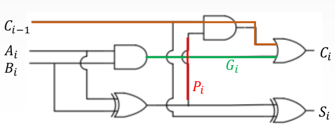
王道考研/CSKAOYAN.COM

并行加法器





并行进位的并行加法器：各级进位信号同时形成，又称为先行进位、同时进位



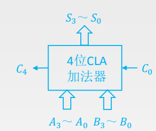
$C_i = G_i + P_i C_{i-1}$

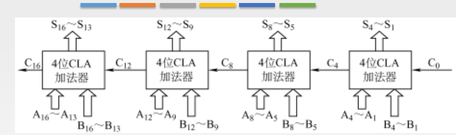
$G_i = A_iB_i$　　$P_i = A_i \oplus B_i$

$C_1=G_1+P_1C_0$
$C_2=G_2+P_2C_1=G_2+P_2G_1+P_2P_1C_0$
$C_3=G_3+P_3C_2=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1C_0$

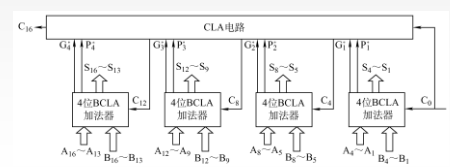
王道考研/CSKAOYAN.COM

并行加法器





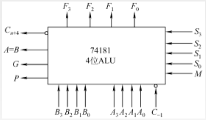
单级先行进位方式，又称为**组内并行、组间串行**进位方式。

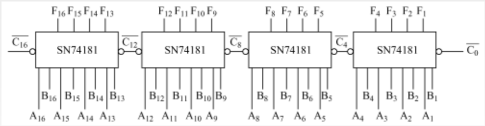


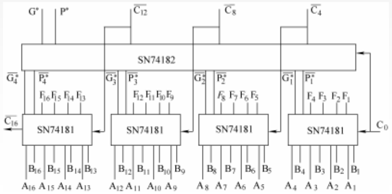
多级先行进位方式，又称为**组内并行、组间并行**进位方式

王道考研/CSKAOYAN.COM

ALU芯片的组织

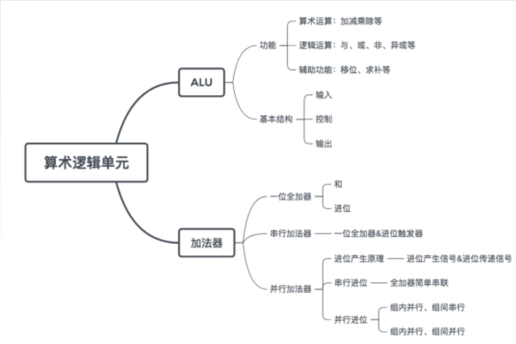






王道考研/CSKAOYAN.COM

本节总览



王道考研/CSKAOYAN.COM