第2章 运算方法和运算器

2.1.2 数的机器码表示

- 浮点数
  - IEEE 754浮点数标准格式

31	30 … 23	22 … 0
S	E	M

63	62 … 52	51 … 0
S	E	M

- 尾数=1.M，“1”不保存，尾数可多1位
- 32位浮点数真值=$(-1)^{s}\times(1.M)\times2^{E-127}$
- 64位浮点数真值=$(-1)^{s}\times(1.M)\times2^{E-1023}$

2.1.2 数的机器码表示

- 浮点数

【例】设浮点数x的754标准存储格式为41360000H，求其浮点数的十进制数值。

【解】

0 100 0001 0 011 0110 0000 0000 0000 0000

符号1 | 阶码8 | 数值23

指数e=E-127=1000 0010 - 0111 1111 = 3

尾数1.M=1.011 0110 0000 0000 0000 0000=1.011011

数值$x=(-1)^s \times 1.M \times 2^e=+1.011011 \times 2^3=(11.375)_{10}$

2.1.2 数的机器码表示

- 浮点数

【例】将数$(20.59375)_{10}$转换成IEEE754标准的32位浮点数的二进制存储格式。

【解】 $(20.59375)_{10}$ =10100.10011=1.010010011 $\times 2^4$

e=4，E=e+127=4+127=131=1000 0011

1.M=1.010 0100 1100 0000 0000 0000

0 100 0001 1 010 0100 1100 0000 0000 0000

41A4C000H

IEEE 754浮点数的编码表示

阶码全0和全1用作特殊值处理

单精度		双精度		表示的数
Exponent	Fraction	Exponent	Fraction	
0	000	0	000	0
0	非零	0	非零	±非规格化数
1~254	任意	1~2046	任意	±浮点数
255	0	2047	0	±无穷
255	非零	2047	非零	NaN（非数）

- **设字长32位，使用IEEE格式，则阶码采用________表示。**

A. 补码

B. 原码

C. 移码

D. 反码

[答案]C

- **IEEE754标准规定的32位浮点数中，符号位为1位，阶码为8位，则它所能表示的最大规格化正数为______。**

A. $+(2-2^{-23})\times 2^{+127}$

B. $+(1-2^{-23})\times 2^{+127}$

C. $+(2-2^{-23})\times 2^{+255}$

D. $2^{+127}+2^{27}$

$e=E-127$
$=254-127$
$=+127$

[答案]A

- **IEEE754单精度浮点格式表示的数中，最小的规格化正数是（　　）。[2018年408统考]**

A. $1.0*2^{-126}$

B. $1.0*2^{-127}$

C. $1.0*2^{-128}$

D. $1.0*2^{-149}$

$e=E-127$
$=1-127$
$=-126$

[答案]A

- **已知float型IEEE754机器数：80200000H，则x的值是______。**

A. -2^{-128}

B. -1.01×2^{-127}

C. -1.01×2^{-126}

D. 非数（NaN）

用规则弥补数值连续性问题，避免出现“数值空洞”

[答案]A

课堂练习

-0.4375的IEEE754单精度浮点数是________。

2.1.3 字符与字符串的表示方法

2.1.3 字符与字符串的表示方法

- 字符
  - 目前国际上普遍采用的字符系统是七单位的IRA（International Reference Alphabet）码，其美国版本为ASCII码（American Standard Code for Information Interchange），通常占用1个字节

高三位

低四位

	000	001	010	011	100	101	110	111
0000	NUL	DEL	SP	0	@	P		p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	”	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	DEL	ETB		7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N		n	~
1111	SI	US	/	?	O	_	o	DEL

数字0-9：　30H-39H
大写字母A-Z：　41H-5AH
小写字母a-z：　61H-7AH
空格：　20H
回车、换行：　0DH, 0AH

河海大学

2.1.3 字符与字符串的表示方法

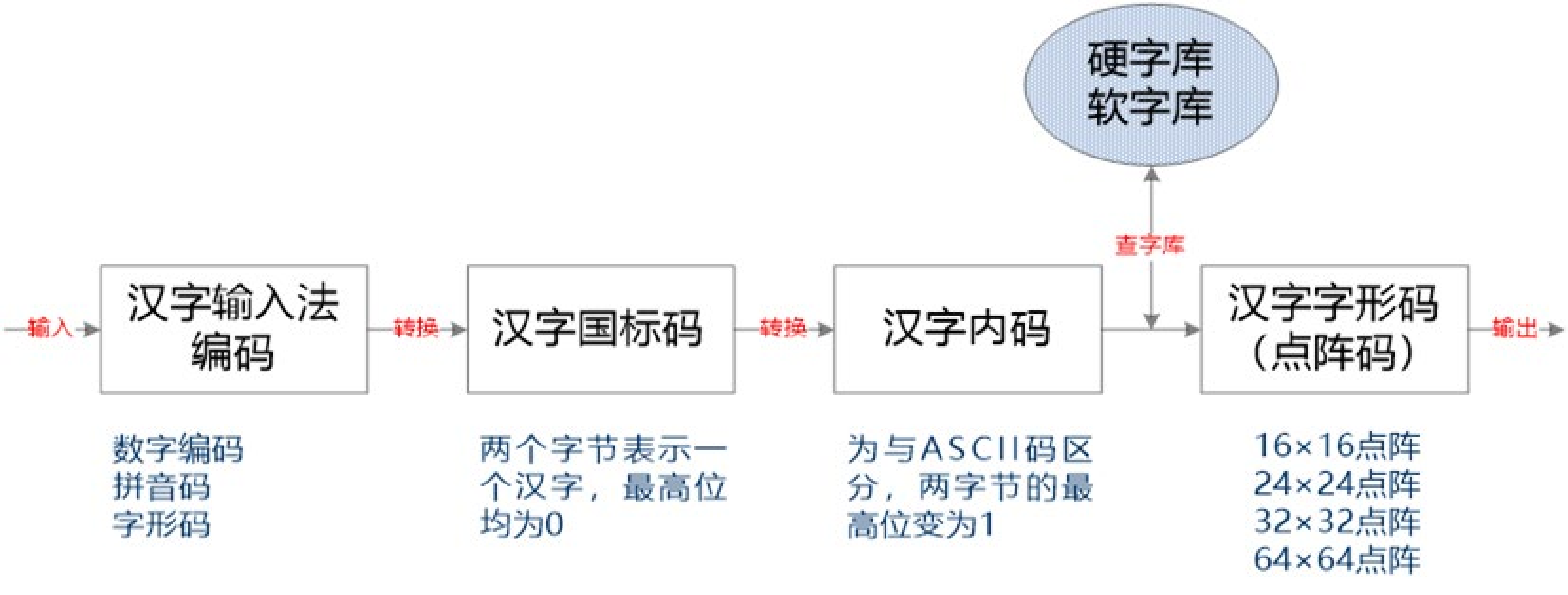
- 字符串
  - 字符串是指连续的一串字符，通常方式下，它们占用主存中连续的多个存储单元，每个字节存一个字符

【例】将字符串IF␣A>B␣THEN␣READ(C)从高位字节到低位字节依次存在主存中。源文件、字符流文件

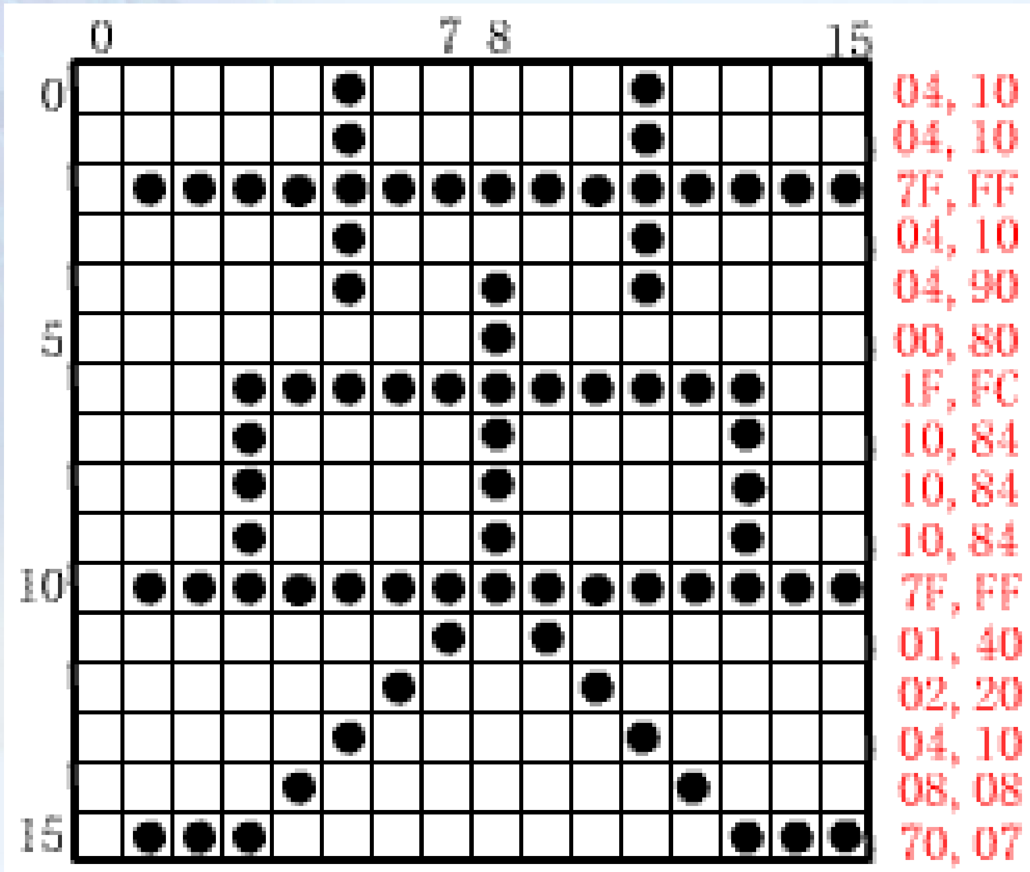
字符	ASCII码	存储单元（32位）
IF␣A	73、70、32、65	01001001 0100110 00100000 01000001
>B␣T	62、66、32、84	00111110 0100010 00100000 01010100
HEN␣	72、69、78、32	01000111 01000101 01001110 00100000
READ	82、69、65、68	01010010 01000101 01000001 01000100
(C)␣	40、67、41、32	00101000 01000011 00101001 00100000

2.1.4 汉字的表示方法

- 汉字的表示
  - 汉字国标码@1981.5：与ASCII码共存时会产生二义性
  - 汉字内码：在汉字国标码每个字节最前面加1
- 汉字的计算机处理过程

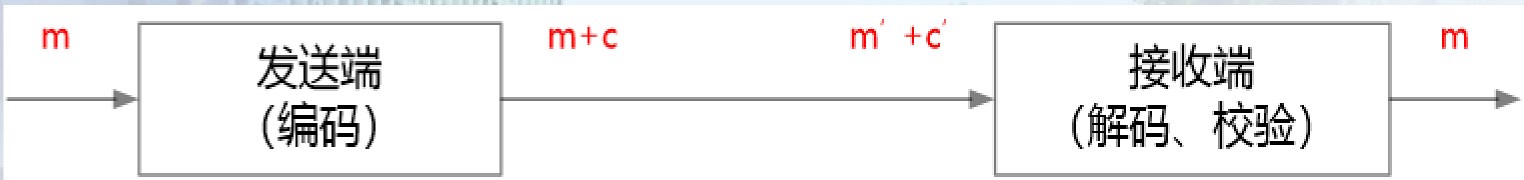


- **字模码**是用点阵表示的汉字字形代码，它是汉字的输出形式。



2.1.5 校验码

- 引入
  - 背景：元件故障、噪声干扰等各种因素常常导致计算机在信息存储、处理、传输、交换过程中出现错误
  - 措施：通常的方法是，在每个数据字/数据块上添加一些校验位，用来检错或纠错



奇偶校验码

CRC循环冗余校验码

2.1.5 校验码

- 偶校验
  - $C = x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1}$
- 发送端
  - $x_0 x_1 \ldots x_{n-1} C$
- 接收端
  - $x_0' x_1' \ldots x_{n-1}' C'$

$F = x_0' \oplus x_1' \oplus \ldots \oplus x_{n-1}' \oplus C'$

若无错，

$F = x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1} \oplus C$

$= C \oplus C = 0$

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 1$
- $1 \oplus 1 = 0$
- $C \oplus C = 0$

2.1.5 校验码

- 奇偶校验码
  - 奇校验码：m+c中有奇数个1
  - 偶校验码：m+c中有偶数个1

【例】已知下表左面一栏有5个字节数据，请分别用奇校验法、偶校验法进行编码

数　据	偶校验编码 C	奇校验编码 C
1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 0	1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 0	0 1 0 1 0 1 0 0 1	0 1 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1 1

- **1**、假定下列字符码中有奇偶校验位，但没有数据错误，采用偶校验的字符码是________。

A. 11001011

B. 11010110

C. 11000001

D. 11001001

[答案]D

- **2**、假定下列字符码中有奇偶校验位，但没有数据错误，采用奇校验的字符码有________。（四个数为 ①**10011010** ②**11010000** ③**11010111** ④**10111100**）

A. ①③

B. ①

C. ②④

D. ④

[答案]C

2.2 定点加法、减法运算

- 补码加法（2.2.1）
- 补码减法（2.2.2）
- 溢出概念与检测方法（2.2.3）
- 基本的二进制加法/减法器（2.2.4）

2.2.1 补码加法

- 运算规则

$$[X+Y]_{补} = [X]_{补} + [Y]_{补}$$

- 补码加法的特点
  - 符号位和数值位一起参加运算，即**符号位数值化**，不需要判断或特别处理符号位
  - 在有模运算时，进位丢掉即可

2.2.1 补码加法

【例】 x = +1001， y = +0101，求x + y。

【解】设机器字长为5位

$[x]_{补} = 0\ 1001$， $[y]_{补} = 0\ 0101$

$$
\begin{array}{lr}
[x]_{补} & 0\ 1\ 0\ 0\ 1 \\
+[y]_{补} & 0\ 0\ 1\ 0\ 1 \\
\hline
[x+y]_{补} & 0\ 1\ 1\ 1\ 0
\end{array}
$$

所以 $x + y = +1110$ (验证9+5=14)

2.2.1 补码加法

【例】 x = +1001， y = +0101，求x + y。

【解】设机器字长为8位

$[x]_{补} = 0\ 0001001$， $[y]_{补} = 0\ 0000101$

$$
\begin{array}{lr}
[x]_{补} & 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1 \\
+[y]_{补} & 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \\
\hline
[x+y]_{补} & 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0
\end{array}
$$

所以 $x + y = +1110$ （验证9+5=14）

2.2.1 补码加法

【例】 x = +1011， y = - 0101，求 x + y。

【解】设机器字长为5位

$[x]_{补} = 0\ 1011$， $[y]_{补} = 1\ 1011$

$$
\begin{array}{lr}
[x]_{补} & 0\ 1\ 0\ 1\ 1 \\
+[y]_{补} & 1\ \ 1\ 0\ 1\ 1 \\
\hline
[x+y]_{补} & 1\ 0\ 0\ 1\ 1\ 0
\end{array}
$$

所以 $x + y = 0110$ （验证11-5=6）

2.2.1 补码加法

【例】 x = +1011， y = - 0101，求 x + y。

【解】设机器字长为8位

$[x]_{补} = 0\ 0001011$， $[y]_{补} = 1\ 1111011$

$$
\begin{array}{lr}
[x]_{补} & 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1 \\
+[y]_{补} & 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \\
\hline
[x+y]_{补} & 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0
\end{array}
$$

所以 $x + y = 0110$ （验证11-5=6）

2.2.2 补码减法

- 运算规则

$$[X-Y]_{补} = [X+(-Y)]_{补} = [X]_{补} + [-Y]_{补}$$

$$= [X]_{补} + [[Y]_{补}]_{求补} = [X]_{补} + (\overline{[Y]_{补}} + 1)$$

- 补码减法法的特点
  - 符号位数值化
  - 减法转换为加法
  - 在有模运算时，进位丢掉即可

2.2.2 补码减法

【例】已知 $x1 = -1110$， $x2 = +1101$，求：$[x1]_{补}$，$[-x1]_{补}$，$[x2]_{补}$，$[-x2]_{补}$。

【解】设机器字长为5位　　设机器字长为8位

$[x1]_{补} = 1\ 0010$	$[x1]_{补} = 1\ 1110010$
$[-x1]_{补} = 0\ 1110$	$[-x1]_{补} = 0\ 0001110$
$[x2]_{补} = 0\ 1101$	$[x2]_{补} = 0\ 0001101$
$[-x2]_{补} = 1\ 0011$	$[-x2]_{补} = 1\ 1110011$

2.2.2 补码减法

【例】 x = +1101， y = + 0110，求x - y。

【解】设机器字长为5位

$[x]_{补} = 0\ 1101$， $[y]_{补} = 0\ 0110$, $[-y]_{补} = 1\ 1010$

$$
\begin{array}{rr}
[x]_{补} & 0\,1\,1\,0\,1 \\
+\ [-y]_{补} & 1\,1\,0\,1\,0 \\
\hline
[x-y]_{补} & 1\,0\,0\,1\,1\,1
\end{array}
$$

所以 $x - y = 0111$ (验证13-6=7)

2.2.2 补码减法

【例】 x = +1101， y = + 0110，求x - y。

【解】 设机器字长为8位

$[x]_补 = 0\ 0001101$，$[y]_补 = 0\ 0000110$，$[-y]_补 = 1\ 1111010$

$$
\begin{array}{rr}
[x]_补 & 0\,0\,0\,0\,1\,1\,0\,1 \\
+\ [-y]_补 & 1\,1\,1\,1\,1\,0\,1\,0 \\
\hline
[x-y]_补 & 1\,0\,0\,0\,0\,0\,1\,1\,1
\end{array}
$$

所以 $x - y = 0111$ (验证13-6=7)

2.2.3 溢出概念与检测方法

- **溢出概念**
  - 上溢出：计算结果绝对值超过计算机所能表示的最大值
    定点数可能回绕（如从127+1变成-128）；
    浮点数通常标记为无穷大（Infinity）或触发异常。
  - 下溢出：计算结果绝对值小于计算机所能表示的最小非零值
    定点数截断为0；
    浮点数转为非规格化数或直接归零。
- 实际意义
  - 上溢出：导致严重错误，需抛出异常
  - 下溢出：损失精度（如科学计算中微小量被忽略），需注意累积误差

2.2.3 溢出概念与检测方法

- 溢出概念

【例】 $x = +1011$， $y = +1001$，求 $x + y$。

【解】设机器字长=5， $[x]_{补} = 0\ 1011$， $[y]_{补} = 0\ 1001$

$$
\begin{array}{rr}
[x]_{补} & 0\,1\,0\,1\,1 \\
+\ [y]_{补} & 0\,1\,0\,0\,1 \\
\hline
[x+y]_{补} & 1\,0\,1\,0\,0
\end{array}
$$

11+9=20，两个正数相加，结果是负数，显然错误。实际上是超过了所能表示的最大值。机器字长=5，表示范围-16至+15。

2.2.3 溢出概念与检测方法

- 溢出概念

【例】 x = +1011， y = +1001，求x + y。

【解】设机器字长=8，$[x]_{补}=0\ 0001011$，$[y]_{补}=0\ 0001001$

$$
\begin{array}{lr}
[x]_{补} & 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1 \\
+\ [y]_{补} & 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1 \\
\hline
[x+y]_{补} & 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0
\end{array}
$$

11+9=20，结果正确，未超过所能表示的最大值。机器字长=8，表示范围-128至+127。

2.2.3 溢出概念与检测方法

- 溢出概念

【例】 x = - 1101， y = - 1011，求x + y。

【解】设机器字长=5， $[x]_{补}=1\ 0011$, $[y]_{补}=1\ 0101$

$$
\begin{array}{lr}
[x]_{补} & 1\ 0\ 0\ 1\ 1 \\
+\ [y]_{补} & 1\ 0\ 1\ 0\ 1 \\
\hline
[x+y]_{补} & 1\ 0\ 1\ 0\ 0\ 0
\end{array}
$$

-13-11=-24，两个负数相加，结果是正数，错误。实际上是超过了所能表示的最小值。机器字长=5，表示范围-16至+15。 。

河海大学

2.2.3 溢出概念与检测方法

- 溢出概念

【例】 x = - 1101， y = - 1011，求x + y。

【解】设机器字长=8，$[x]_{补}$=1 1110011，$[y]_{补}$=1 1110101

$$
\begin{array}{rr}
 [x]_{补} & 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\
+\ [y]_{补} & 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\
\hline
[x+y]_{补} & 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0
\end{array}
$$

-13-11=-24，结果正确，未超过所能表示的最小值。机器字长=8，表示范围-128至+127。。

2.2.3 溢出概念与检测方法

- 检测方法
  - 单符号位：设运算时最高数值位向符号位的进位、符号位向前的进位分别为C_{n-1}、C_f，则 $V = C_{n-1} \oplus C_f$
  - 变形补码：双符号位S_{f1} S_{f2}参加运算， $V = S_{f1} \oplus S_{f2}$
    - Sf1 Sf2 = 00 结果为正，正确
    - Sf1 Sf2 = 11 结果为负，正确
    - Sf1 Sf2 = 01 结果应该为正，运算为负，错误，上溢
    - Sf1 Sf2 = 10 结果应该为负，运算为正，错误，下溢

河海大学

2.2.3 溢出概念与检测方法

- 检测方法

【例】 x = +1100， y = +1000，求 x + y。

【解】设机器字长=5，$[x]_{补}$ = 00 1100, $[y]_{补}$ = 00 1000

$$
\begin{array}{lr}
 & [x]_{补} \quad 0\,0\,1\,1\,0\,0 \\
+ & [y]_{补} \quad 0\,0\,1\,0\,0\,0 \\
\hline
 & [x+y]_{补} \quad 0\,1\,0\,1\,0\,0
\end{array}
$$

12+8=20， “01”表示结果错误，超过所能表示的最大值。机器字长=5，表示范围-16至+15。

河海大学

2.2.3 溢出概念与检测方法

- 检测方法

【例】 x = +1100， y = +1000，求 x + y。

【解】设机器字长=8，

$[x]_{补}=00\ 0001100$, $[y]_{补}=00\ 0001000$

$[x]_{补}$	0 0 0 0 0 1 1 0 0
+ $[y]_{补}$	0 0 0 0 0 1 0 0 0
$[x+y]_{补}$	0 0 0 0 1 0 1 0 0

12+8=20， “00”表示结果正确，表示范围-128至+127。

河海大学

2.2.3 溢出概念与检测方法

- 检测方法

【例】 x =-1100， y =-1000，求x + y。

【解】设机器字长=5，$[x]_{补}$ = 11 0100, $[y]_{补}$ = 11 1000

$$
\begin{array}{lr}
[x]_{补} & 1\,1\,0\,1\,0\,0 \\
+\ [y]_{补} & 1\,1\,1\,0\,0\,0 \\
\hline
[x+y]_{补} & 1\,1\,0\,1\,1\,0\,0
\end{array}
$$

-12-8= -20， “10”表示结果错误，超过所能表示的最小值。机器字长=5，表示范围-16至+15。 。

2.2.3 溢出概念与检测方法

- 检测方法

【例】 x =-1100， y =-1000，求 x + y。

【解】设机器字长=8，

$[x]_{补}$ = 11 1110100, $[y]_{补}$ = 11 1111000

$$
\begin{array}{lr}
 & [x]_{补} \quad 1\,1\,1\,1\,1\,0\,1\,0\,0 \\
+ & [y]_{补} \quad 1\,1\,1\,1\,1\,1\,0\,0\,0 \\
\hline
 & [x+y]_{补} \quad 1\,1\,1\,1\,1\,0\,1\,1\,0\,0
\end{array}
$$

-12-8=-20， “11”表示结果正确，表示范围-128至+127。

- 1、若x=103，y=-25，则下列表示式采用8位定点补码运算实现时，会发生溢出的是（   ）。

A. x+y

B. -x+y

C. x-y

D. -x-y

- 2、某字长为8位的计算机中，已知整型变量x、y的机器数分别为$[x]_{补}$=1111 0100，$[y]_{补}$=1011 0000。若整型变量z=2*x+y/2，则z的机器数为（　　）。

A. 11000000

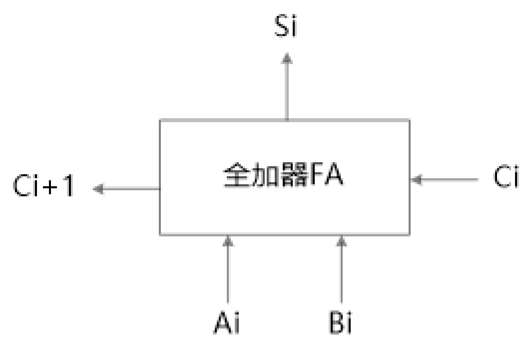
B. 00100100

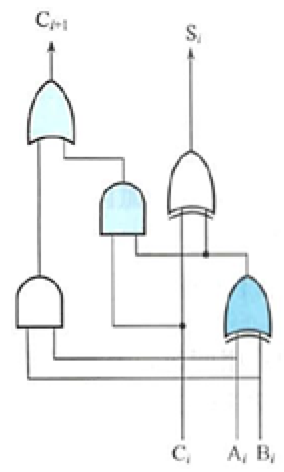
C. 10101010

D. 溢出

2.2.4 基本的二进制加法/减法器

- 基本加法单元（全加器）





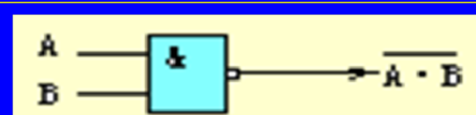
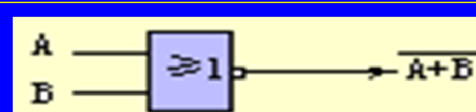
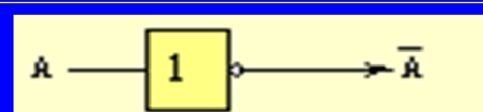
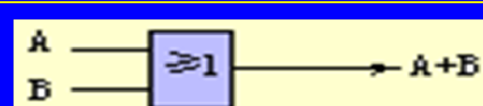
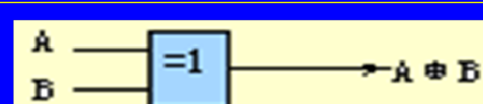
A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S_i = (A_i \oplus B_i) \oplus C_i$ 奇数个1

$C_{i+1} = A_iB_i + B_iC_i + C_iA_i$ 两个以上1

$= AiBi + (Ai \oplus Bi)\ Ci$

典型门电路的逻辑符号和时间延迟

门的名称	门的功能	逻辑符号(正逻辑)	时间延迟
与非	NAND		T
或非	NOR		T
非	NOT		T
与	AND		2T
或	OR		2T
异或	XOR		3T

2.2.4 基本的二进制加法/减法器

- 基本加法单元（全加器）——更新

◆一位全加器 (Full-Adder)

$$S_i = A_i \oplus B_i \oplus C_i = A_iB_iC_i + A_i\overline{B_i}\,\overline{C_i} + \overline{A_i}B_i\overline{C_i} + \overline{A_i}\,\overline{B_i}C_i$$

$$C_{i+1} = A_iB_i + B_iC_i + A_iC_i = A_iB_i + (A_i \oplus B_i)C_i$$

$$= A_iB_i + (A_i + B_i)C_i = \boxed{\overline{\overline{A_iB_i} \bullet \overline{(A_i \oplus B_i)C_i}}}$$

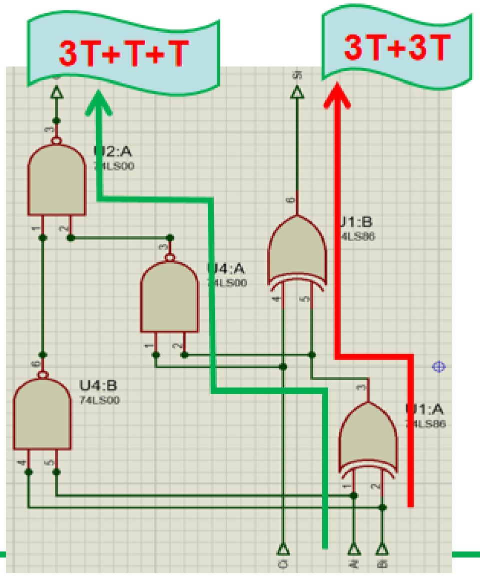
- 延迟时间：从输入数据信号有效到输出数据信号有效的时间。
- 延迟假定：

单级与非门、或非门和非门的延迟为T

单级与门、或门的延迟为2T

单级异或门的延迟为3T

- S_i的时间延迟：6T
- C_{i+1}的时间延迟：5T

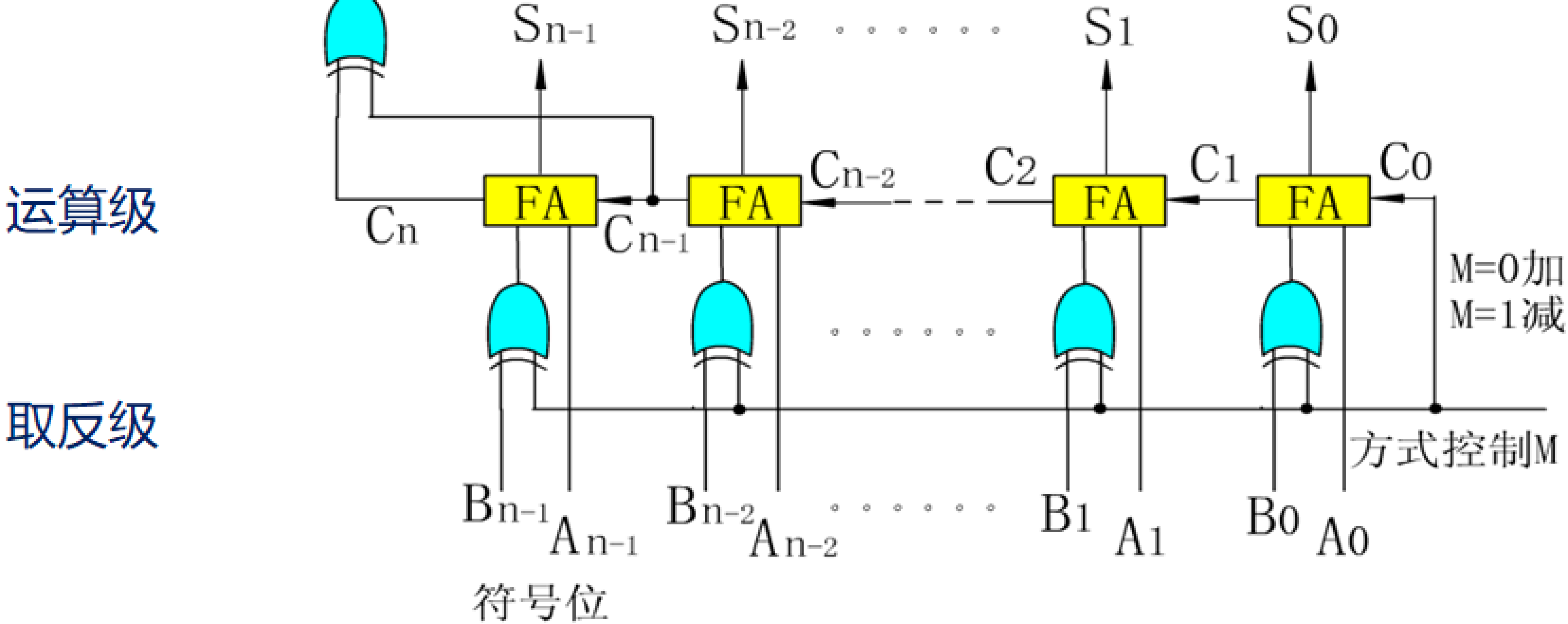


河海大学

2.2.4 基本的二进制加法/减法器

- 行波（串行）进位的加减法器

$V = C_{n-1} \oplus C_n$



M=0：A+B

M=1：$A+\overline{B}+1 = A-B$

补码加减法器

- 思考：为什么一套加法器就可以实现加法、减法运算？
- 表面上，M位的设置可以实现加减法一体运算。
- 本质上是补码的理论支持：
  - 将减法运算转换为加法运算
  - 消除原码中0有两个状态的情形
  - 可实现将符号位当做数据位执行运算

- 某加法器进位链小组信号为$C_4C_3C_2C_1$，低位来的信号为C_0，请分别按下述两种方式写出$C_4C_3C_2C_1$的逻辑表达式。

（1） 串行进位方式

（2） 并行进位方式

作业

P69习题:

5、6、11

河海大學

Q&A