

2.3 数据处理与存储

1、移位操作

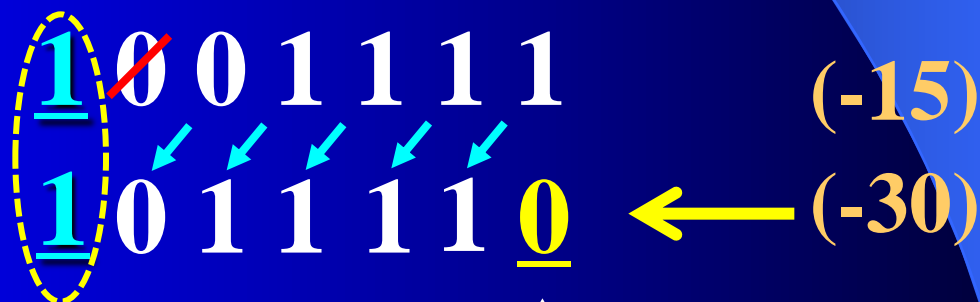
(1) 逻辑移位：数码位置变化。

循环左移：



(2) 算术移位：符号位不变、数码位置变化

算术左移：



数值发生 2^n 倍变化， n 为移动的位数

↑
空位补0

①正数补码\原码移位规则

※移位规则

数符不变（单：符号位不变；双：第1符号位不变）

空位补0（右移时第2符号位移至尾数最高位）

（1）单符号位：

	<u>0</u> 0111
左移	<u>0</u> 111 <u>0</u>
右移	<u>0</u> <u>0</u> 111
右移	<u>0</u> <u>0</u> 011

（2）双符号位：

	<u>00</u> 0111
左移	<u>00</u> 111 <u>0</u>
左移	<u>01</u> 110 <u>0</u>
右移	<u>00</u> <u>0</u> 1110
右移	<u>00</u> <u>0</u> 0111

②负数补码移位

(1) 移位规则

数符不变 (单: 符号位不变; 双: 第1符号位不变)

左移空位补0

右移空位补1 (第二符号位移至尾数最高位)。

(2) 单符号位:

	<u>1</u> 1011
左移	<u>1</u> 0110
右移	<u>1</u> 1011
右移	<u>1</u> 1101

(3) 双符号位:

	<u>11</u> 0110
左移	<u>10</u> 1100
右移	<u>11</u> 0110
右移	<u>11</u> 1011

※易出错处:

← 左 00 1110
~~00 1100~~

正确: 01 1100

← 左 11 0110
~~11 1100~~

正确: 10 1100

→ 右 01 1100
~~00 0110~~

正确: 00 1110

→ 右 10 1100
~~11 1110~~

正确: 11 0110

2、舍入方法

① 0舍1入（原码、补码）

[例] 保留4位尾数：

0 00100原 \longrightarrow 0 0010原

1 00101原 \longrightarrow 1 0011原

1 11011补 \longrightarrow 1 1110补

② 末位恒置1（原码、补码）

[例] 保留4位尾数：

0 00100原 \longrightarrow 0 0011原

1 00101原 \longrightarrow 1 0011原

1 11011补 \longrightarrow 1 1101补

3、数位扩展与压缩

(1) 符号扩展

直接把符号位 (0/1) 填充到扩展位

000A \longrightarrow 0000000A

800A \longrightarrow FFFF800A

(2) 0-扩展 高位均全补0 (针对无符号数)

002A \longrightarrow 0000002A

F12C \longrightarrow 0000F12C

(3) 位数压缩 弃高位、留低位

F12B800A \longrightarrow 800A

02A0F12C \longrightarrow F12C

4、数据存储(按字节编址)

(1) 小端模式/ Little-Endian

小地址单元存储数据的低位(即尾端)

FF FF 00 01 \Longrightarrow FFFF0001
#103 #102 #101 #100

(2) 大端模式/ Big-Endian

大地址单元存储数据的低位(即尾端)

FF FF 00 01 \Longrightarrow 0100FFFF
#103 #102 #101 #100

5、数据字的对齐(Alignment)

要求数据的地址是相应的边界地址

※按边界对齐

(假定存储字宽度为32位，按字节编址，字长32位)

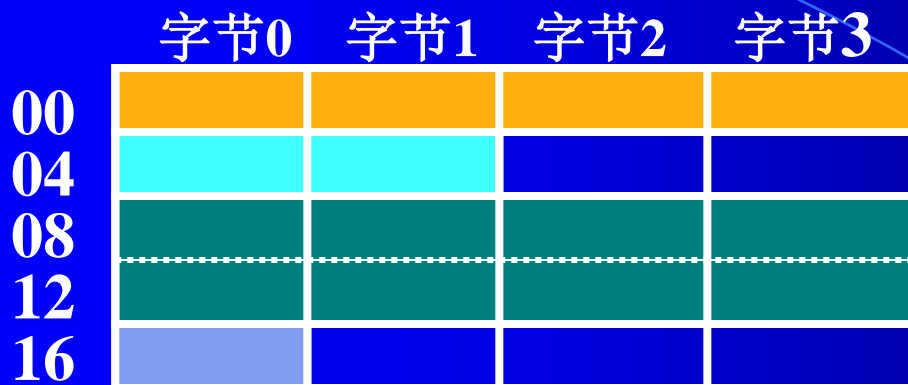
字地址：4的倍数(低两位为0)

半字地址：2的倍数(低位为0)

字节地址：任意

※不按边界对齐

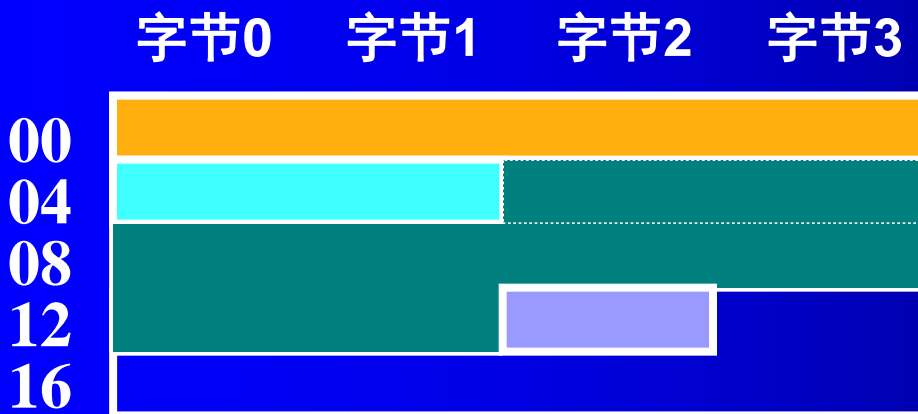
例如 int i, short k, double x, char c



i
k
x
c

按字边界对齐

则 $\&i=0$; $\&k=4$;
 $\&x=8$; $\&c=16$



i
k
x
c

不按边界对齐

则 $\&i=0$; $\&k=4$;
 $\&x=6$; $\&c=14$;

试分析访存次数的变化情况

思考下列问题:

1、在32位机器上输出si, usi, i, ui的十进制（真值）和十六进制值（补码机器数）是什么？

```
short si = -32768;
```

```
unsigned short usi = si;
```

```
int i = si;
```

```
unsigned ui = usi ;
```

已知

$32768=2^{15}$

$=1000\ 0000\ 0000\ 0000B$

2、某机存储器按字节编址,字长32位，欲存储俩数

```
short float f1=-256,f2=-128.75。
```

试分析在地址码为#1000存储单元的小端字对齐存储模式。