



## 2. 数据的表示与存储

### 2.4 二进制数据

# 移位运算



## 2. 数据的表示与存储

### 2.4 二进制移位

- 01 逻辑移位
- 02 算术移位
- 03 循环移位
- 04 位扩展



## 2. 数据的表示与存储

### 2.4.1 移位运算

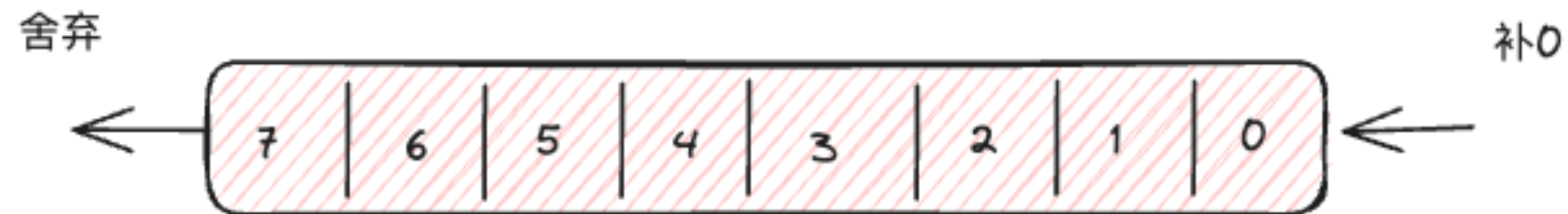
0001
0010
0100
1000



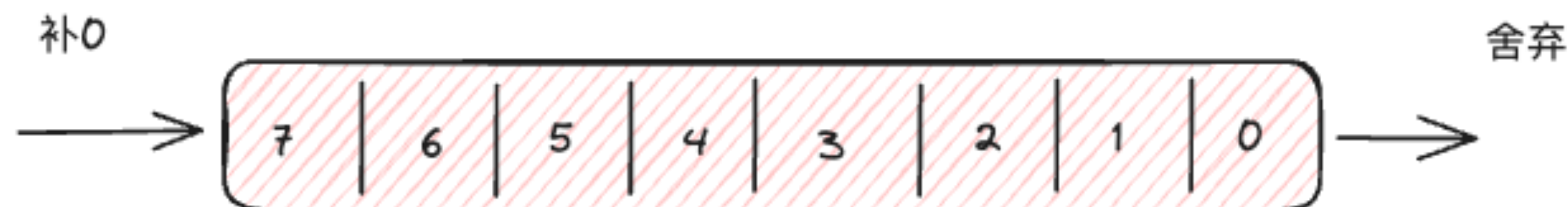
## 2. 数据的表示与存储

### 2.4.1、二进制移位运算

1、逻辑移位：在不丢失高位的情况下，左移是乘2，右移是除2



左移乘2



右移除2



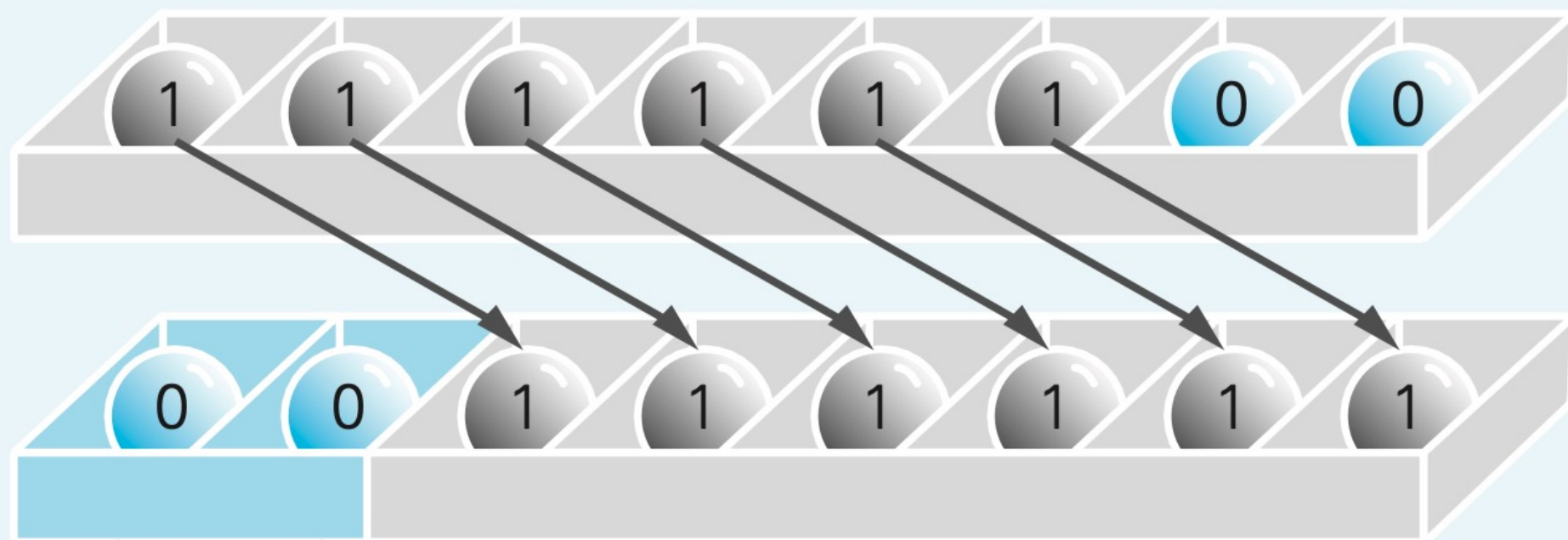


## 2. 数据的表示与存储

### 2.4.1、二进制移位运算

举例：-4右移2位

● 逻辑右移

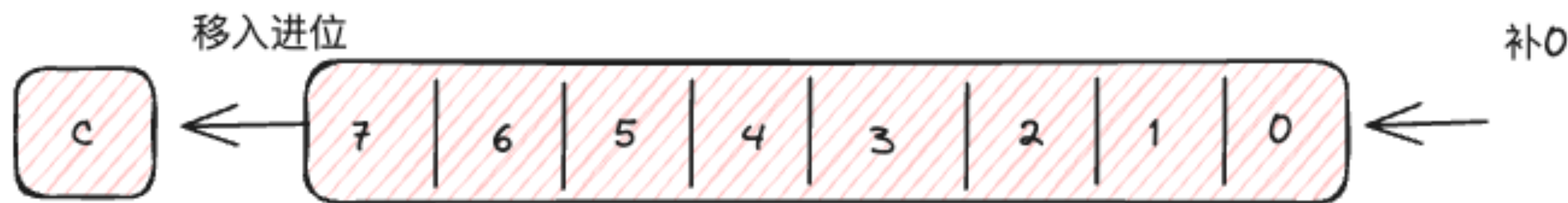




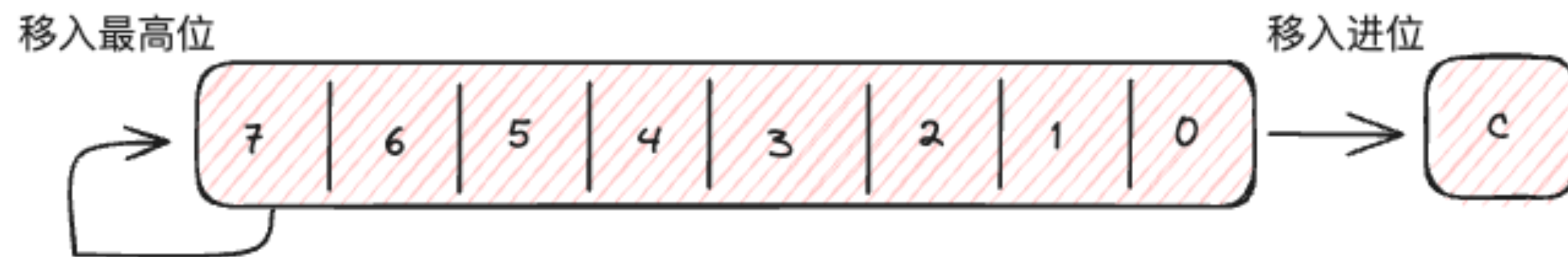
## 2. 数据的表示与存储

### 2.4.2、二进制移位运算

2、算术移位：左移时把高位移入到C位中，右移时根据符号位补充，最低位移入C位



左移乘2



右移除2

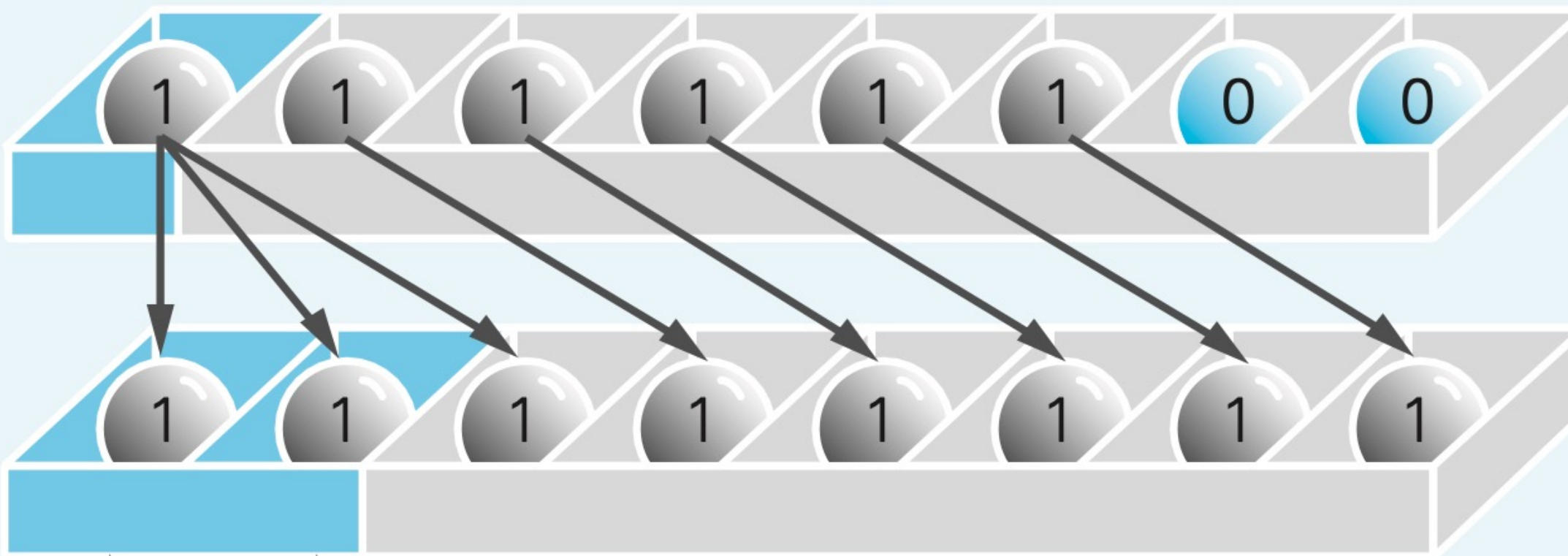


## 2. 数据的表示与存储

### 2.4.2、二进制移位运算

举例：-4右移2位

● 算数右移







## 2. 数据的表示与存储

### 2.4.2、二进制移位运算

VisUAL2

Run Reset Step Stepping 2 : 2 Hex Bin Dec uDec

Untitled.s \* +

```
1  mov  r0,#1
2  LOOP
3  lsl  r0,r0,#1
4  lsr  r0,r0,#1
```

Registers

R0	0b00000000_00000000_00000000_00000010
R1	0b00000000_00000000_00000000_00000000
R2	0b00000000_00000000_00000000_00000000
R3	0b00000000_00000000_00000000_00000000
R4	0b00000000_00000000_00000000_00000000
R5	0b00000000_00000000_00000000_00000000
R6	0b00000000_00000000_00000000_00000000
R7	0b00000000_00000000_00000000_00000000
R8	0b00000000_00000000_00000000_00000000
R9	0b00000000_00000000_00000000_00000000
R10	0b00000000_00000000_00000000_00000000
R11	0b00000000_00000000_00000000_00000000
R12	0b00000000_00000000_00000000_00000000
R13	0b11111111_00000000_00000000_00000000
R14	0b00000000_00000000_00000000_00000000
R15	0b00000000_00000000_00000000_00001000

Memory

Symbols

Shift bit out

ALU

ALU out (R0)

N 0 Z 0 C 0 V 0

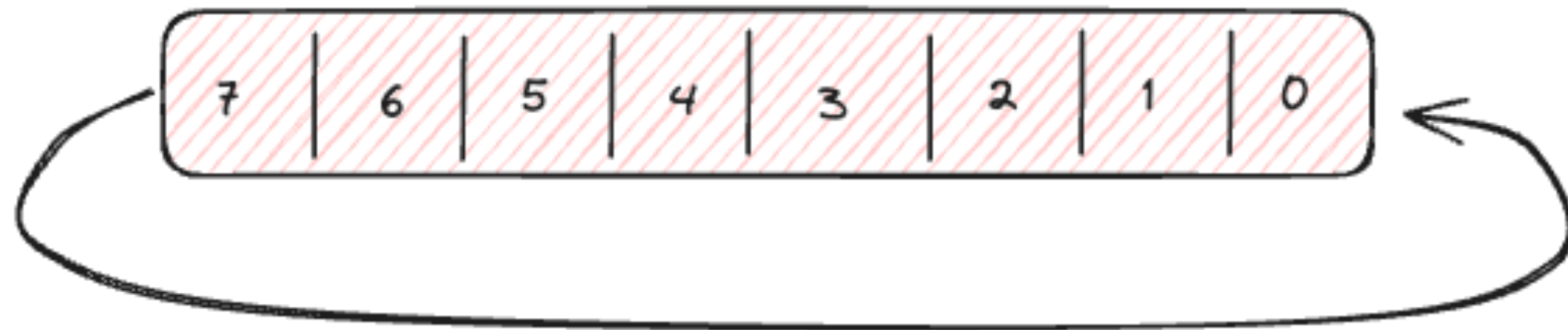




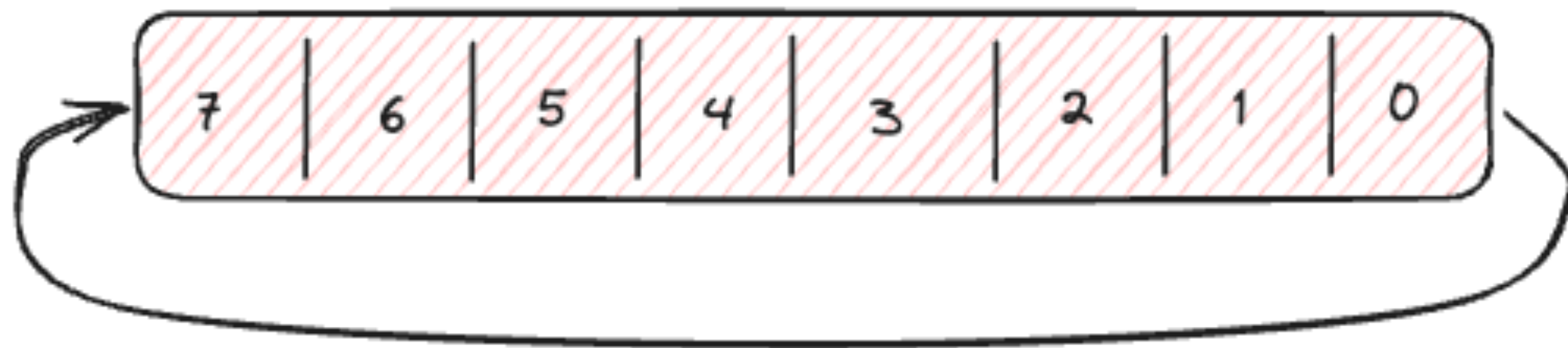
## 2. 数据的表示与存储

### 2.4.3、二进制移位运算

3、把移出的位放到另外一边，形成循环移位



循环左移



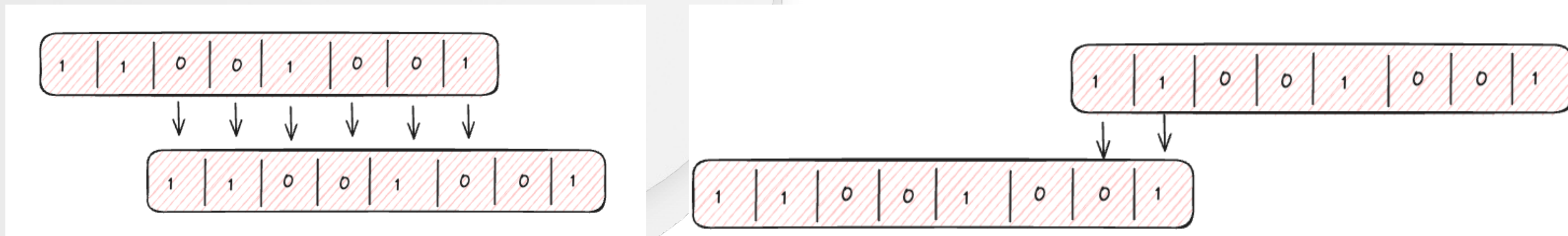
循环右移



## 2. 数据的表示与存储

### 2.4.3、二进制移位运算

示例：处理0xC9循环左移2位



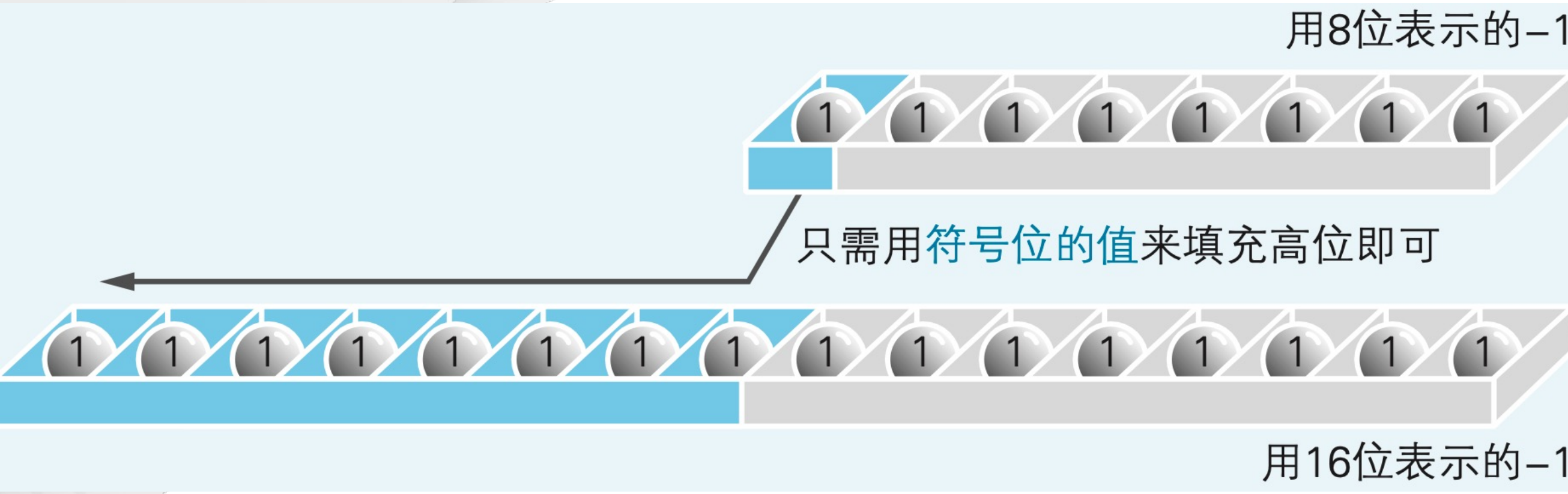
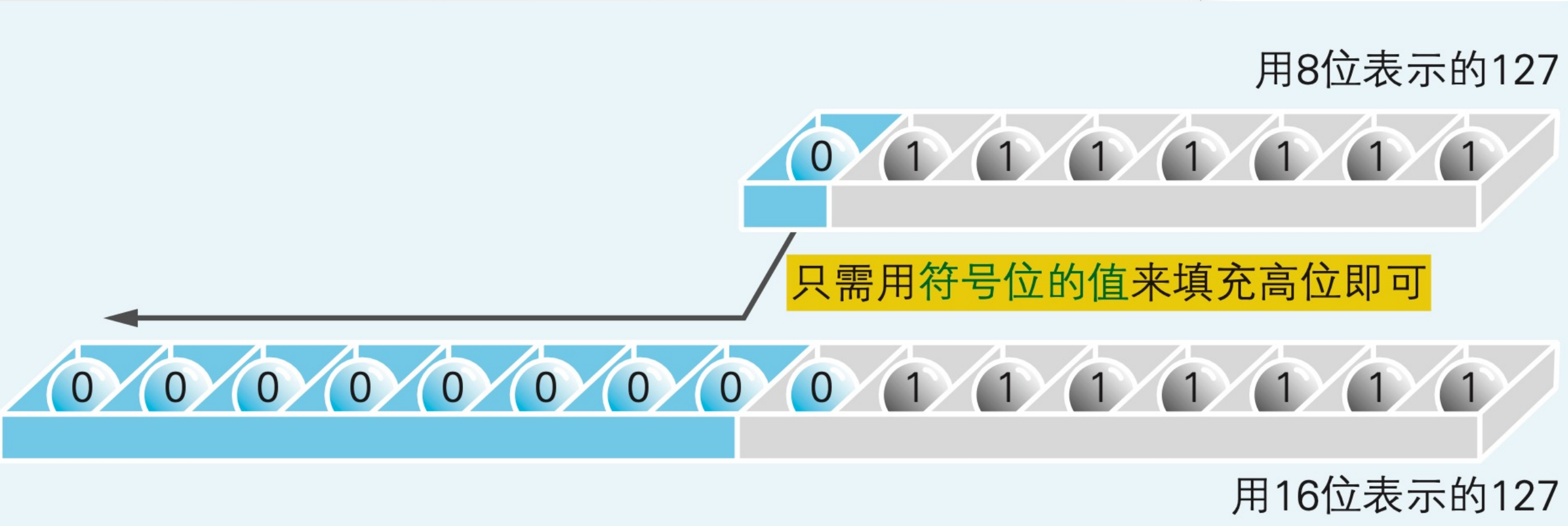
```
1 unsigned char circularLeftShift(unsigned char val, int n) {  
2     int num_bits = sizeof(val) * 8; // 计算位数, 对于字节是8位  
3     n = n % num_bits; // 确保移位数不超过位数  
4     return (val << n) | (val >> (num_bits - n));  
5 }
```





## 2. 数据的表示与存储

### 2.4.4 位扩展







## 2. 数据的表示与存储

### 2.4.4 位扩展

unsigned char

unsigned short

unsigned int

unsigned long

char

short

int

long



## 2. 数据的表示与存储

### 2.2. 本节总结

1. 二进制进行移位运算代替乘除运算速度更快
2. 移位运算也可能会引起数据丢失而影响计算结果
3. 补码进行算术移位时，移入数据有所不同
4. 不同数据类型互转，可能会发生数据丢失



# 欢迎参与学习

WELCOME FOR YOUR JOINING

船说：计算机基础