嵌入式C语言之-

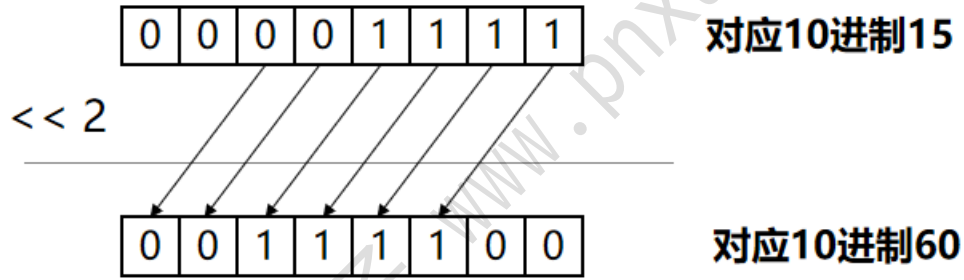
左移右移位运算符

讲师：叶大鹏

助力你成为优秀的电子工程师！

位运算符 <<

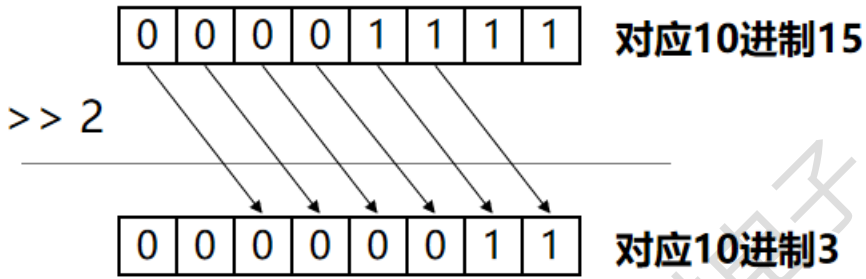
- 左移位：x << n，表示把x的每一位向左平移n位，右边空位补0

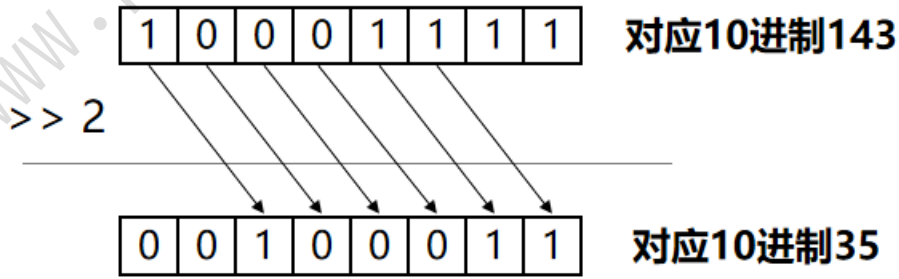


位运算符 >>

- **右移位：x >> n，表示把x的每一位向右平移n位；**

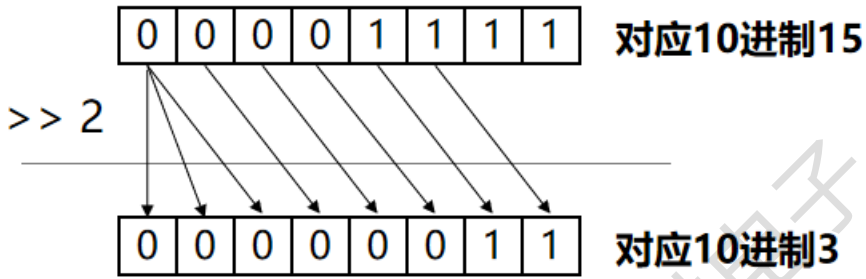
1. 当x为**无符号数**时，左边空位补0，称作**逻辑移位**：

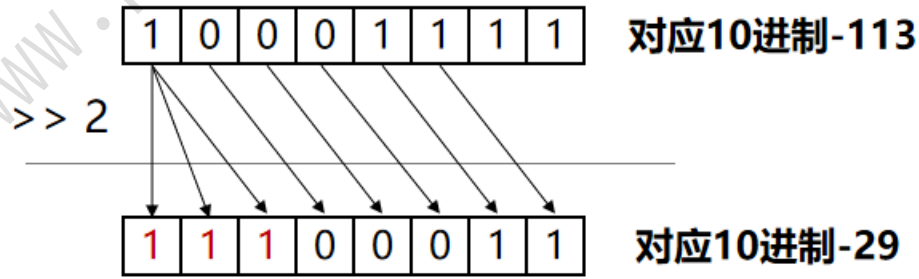




位运算符 >>

2. 当x为**有符号数**时，左边空位补最高位数值，称作**算数移位**：





位运算符 >>

```c
int main(void)
{
    volatile int8_t a = 0x8F;
    volatile uint8_t b = 0x8F;
    a >>= 2;
    b >>= 2;
    return 0;
}
```

Name	Location/Value
main	0x000004D0
a	0xE3 '?
b	0x23 '#'

位运算符 >>

在windows的计算器上有逻辑移位和算数移位的选项：

应用案例1

- 假如在蓝牙项目开发中，2个设备之间进行日期数据传输时，我们先定一个协议，用4个byte也就是uint32_t来表示一个日期，其中byte3表示年份的高位数，byte2表示年份的低位数，byte1表示月份，byte0表示日期。

  设备端现在收到另外一台设备传过来的日期数据00010100 00010011 00000110 00011101。那么我要如何解析这个数据来得到实际日期呢?

应用案例1

```
/*
*第一步，获取日期。
*日期是最后一个byte，也就是最后8位
*/
uint32_t date = 0x1413061D;   //00010100 00010011 00000110 00011101;
uint8_t  day = date;     //(计算结果是00011101，十进制表示是29，也就是日期是29)。

/*
*第二步，获取月份。
*月份是倒数第2个byte，此时需要先将最后一个byte砍掉(也就是右移8位)
*/
date = date>>8;  //(计算结果是00010100 00010011 00000110)
uint8_t  month = date;  //(计算结果是00000110，十进制表示是6，也就是月份是6月)。
```

应用案例1

```
/*
*第三步，获取年份低位。
*先将最后一个byte砍掉(也就是右移8位)
*/
date = date>>8;  //(计算结果是00010100 00010011)
uint8_t year_low = date;  //(计算结果是00010011，十进制表示是19)。

/*
*第四步，获取年份高位。
*先将最后一个byte砍掉(也就是右移8位)
*/
date = date>>8;  //(计算结果是00010100)
year_heigh = date;  //(计算结果是00010011，十进制表示是20)。
```

- **思考：学习完指针课程后，考虑还有没有更优方案?**

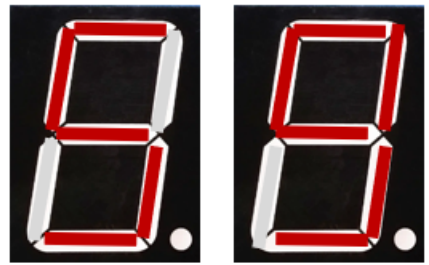
应用案例2

- 获取到时间，比如59分，59首先是一个整数，在单片机中以一个字节存储，现在需要将59显示在2个数码管上，所以需要将其分解为两个数5 和 9，如何实现?
- 通过如下数学运算得到的：

a = 59

先求5：c = a / 10 = 59 / 10 = 5

再求9：d = a % 10 = 59 % 10 = 9



应用案例2

- 扩展：5和9两个整数的范围都在0-15以内，所以可以使用4位二进制数来表示，这样可以将他们放在一个字节内：0101 1001，这就是BCD编码格式，某些时钟芯片再和单片机交互时，需要使用BCD格式。

```
static uint8_t Hex2BCD(uint8_t dat)
{
    return (((dat/10)<<4) + (dat%10));
}
```

THANK YOU!