嵌入式C语言之-

位运算符

讲师：叶大鹏

助力你成为优秀的电子工程师！

PN学堂

应用场景

- 单片机中会集成多组IO口，每一组通常又包含16个IO口，程序里通过寄存器（32位）来控制这组IO口，比如下面的寄存器控制IO口输出高低电平：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OCTLy	端口输出控制位（y=0..15） 这些位由软件置位和清除。 0：引脚输出低电平 1：引脚输出高电平

- 假如不知道当前IO口的状态，需要在不改变其他位IO口前提下，只改变第9位IO口，从输出低电平改变为高电平如何实现？

整体介绍

- **C语言有一个重要特点就是可以直接对二进制位进行操作，运算符包括：**

  &， |， ^， ~， >>， <<

整体介绍

运算符（优先级从上往下）	运算符说明及应用场景	结合性
() [] -> .	括号（函数等），数组，结构体指针变量的成员访问，普通结构体变量的成员访问	由左向右
! ~ ++ -- + -	逻辑非，按位取反，自增1，自减1，正号，负号	由右向左
* & (类型) sizeof	间接，取地址，强制类型转换，求占用空间大小	
* / %	乘，除，取模	由左向右
+ -	加，减	由左向右
<< >>	左移，右移	由左向右

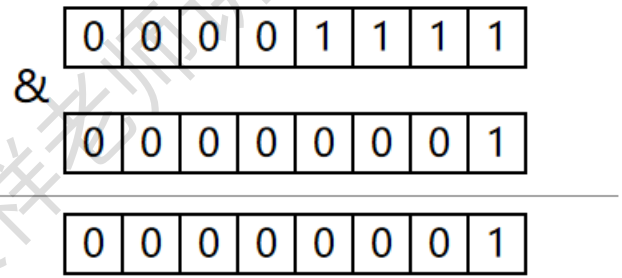
整体介绍

运算符（优先级从上往下）	运算符说明及应用场景	结合性
< <= >= >	是否小于，是否小于等于，是否大于等于，是否大于	由左向右
== !=	是否等于，是否不等于	由左向右
&	按位与	由左向右
^	按位异或	由左向右
|	按位或	由左向右
&&	逻辑与	由左向右
||	逻辑或	由左向右
? :	条件	由右向左
= += -= *= /= %= &= ^= |= <<= >>=	各种赋值运算符	由右向左
,	逗号（顺序）	由左向右

位运算符 &

- **按位与，同1与，结果为原始数值，同0与，结果为 0：**

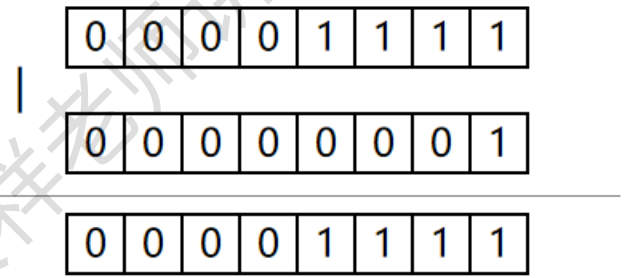
原始数值a(bit)	b(bit)	a & b
0	0	0
0	1	0
1	0	0
1	1	1



位运算符 |

- **按位或，同1或，结果为1，同0或，结果为原始数值：**

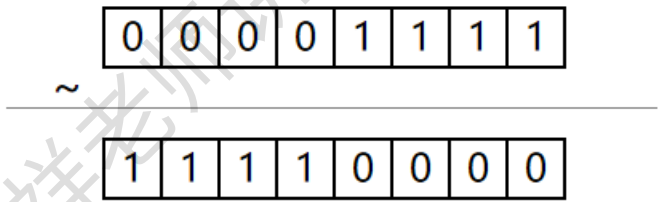
原始数值a(bit)	b(bit)	a | b
0	0	0
0	1	1
1	0	1
1	1	1



位运算符 ~

- **按位取反，是一元运算符；为0的位，结果是1，为1的位，结果是0：**

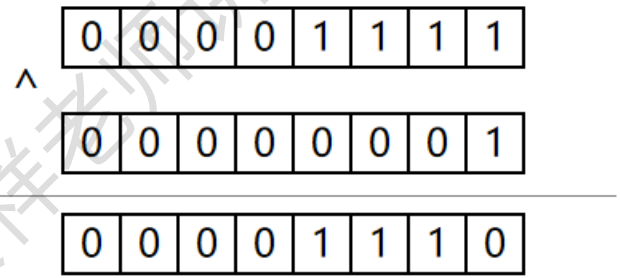
原始数值a(bit)	~a
0	1
1	0



位运算符 ^

- **按位异或，同1异或，结果取反，同0异或，结果为原始数值：**

原始数值a(bit)	b(bit)	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0



应用案例

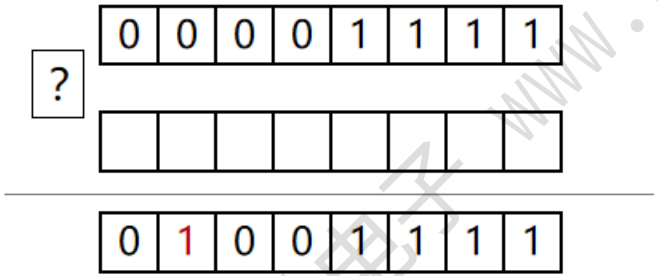
- 给定一个变量a，不改变其它位的值，只将第6位置1。

解题思路

1.我们假设a的原始值是0x0F；

2.根据需求，目标结果是0X4F；

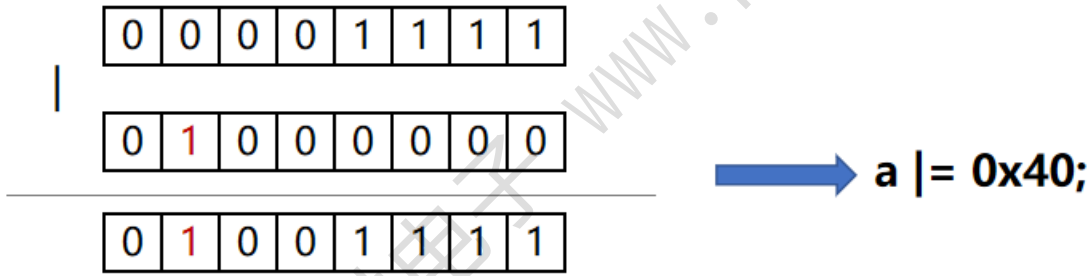
3.遍历思考各种位运算符；



解题思路

4.只有 | 满足要求；

5.构造出另一个操作数0X40：



解题思路

6.再进一步，0x40如何快速获得，它对应的二进制第6位为1；

0	1	0	0	0	0	0	0

7.其实正是因为我们的需求为将第 6 位置1；

8.将操作数再通用一些：

0	0	0	0	0	0	0	1

<< 6

0	1	0	0	0	0	0	0

9.最终获得通用表达式，其中1是固定的，6代表需求里要设置的第几位：

a |= (0x01 << 6);

应用案例

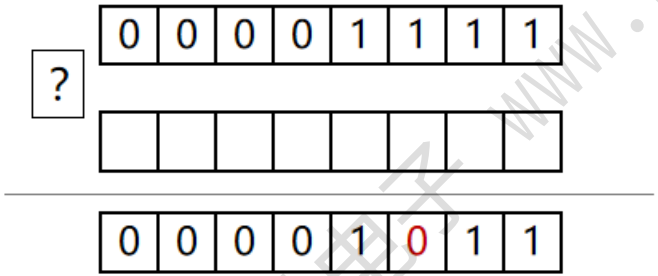
- 给定一个变量a，不改变其它位的值，只将第2位清零。

解题思路

1.我们假设a的原始值是0x0F；

2.根据需求，目标结果是0X0B；

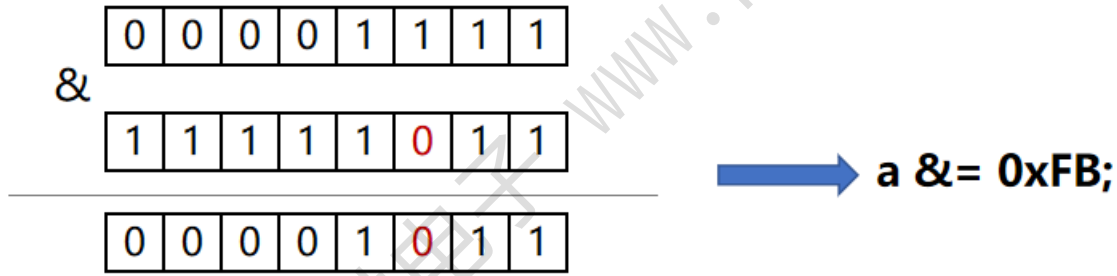
3.遍历尝试各种位运算符；



解题思路

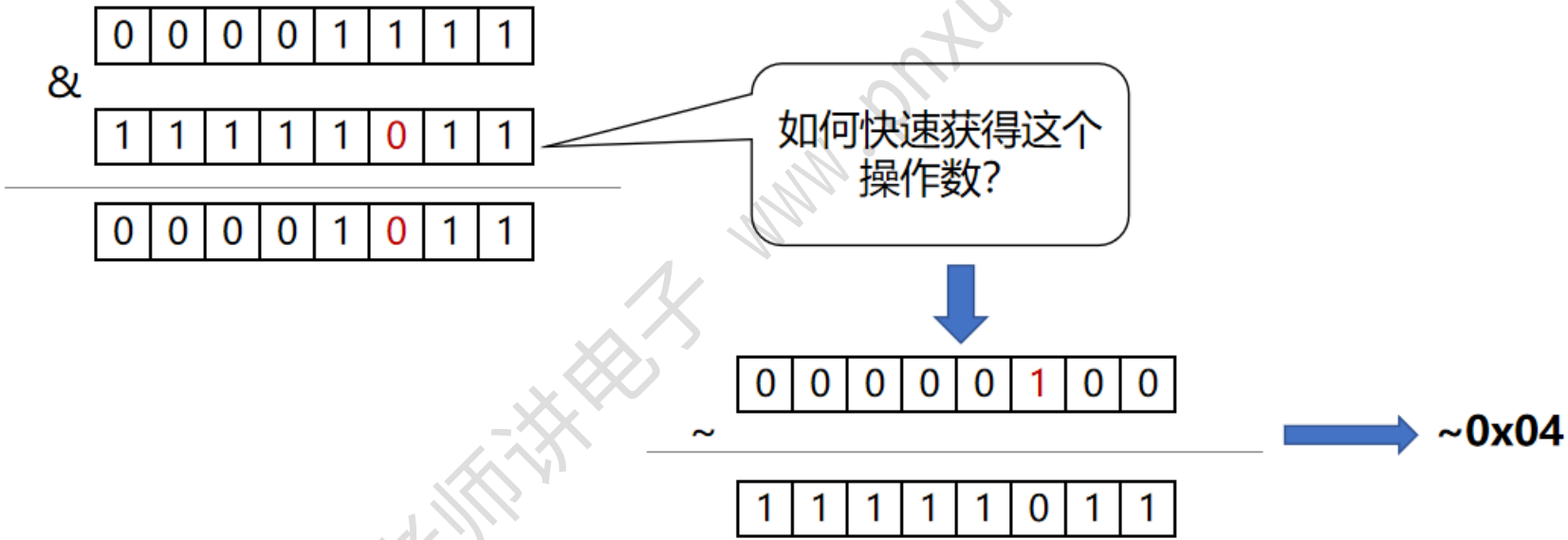
4.只有&满足要求；

5.构造出另一个操作数0XFB：



解题思路

6.如何构造通用的操作数；



7. **a &= ~0x04;**

解题思路

8.再进一步，0x04如何快速获得，它对应的二进制第2位为1；

0	0	0	0	0	1	0	0

9.其实正是因为我们的需求为将第 2 位清零；

10.将操作数再通用一些：

0	0	0	0	0	0	0	1

<< 2

0	0	0	0	0	1	0	0

11.最终获得通用表达式，其中1是固定的，2代表需求里要设置的第几位：

a &= ~(1 << 2);

应用案例

- 给定一个变量a，不改变其它位的值，只将第2位取反。

THANK YOU!