

第二章：位操作

位操作：

- 位与&（每个二进制数相与、逻辑与&&是两个操作数整体相与）
- 位或|、位取反~、位异或^、左移位<<、右移位>>

嵌入式中的位操作

- ARM内存与IO统一编址，读写寄存器（按位定义）=操控硬件
- 寄存器操作要求设定特定位时不能影响其它位，读-改-写（读整体，改局部，写整体；寄存器只能整体读写）
- 特定位清零用&：构造一个数，特定位为0，其它位为1
- 特定位置1用|：构造一个数，特定位为1，其它位为0
- 特定位取反用^：构造一个数，要取反的位为1，其它位为0

位运算构建特定二进制数

- 法1：直接给出32位特定数
- 法2：移位、结合取反（0多用移位，1多用取反）
 - 使用移位获取特定位为1的二进制数
 - 例：获取bit 3~bit 7为1，同时bit 23~bit 25为1，其余为0的二进制数
 - $(0x1f << 3) | (0x7 << 23)$ //5位为1左移3位，3位为1左移23位
 - 9以内的数，十六进制和十进制一样，即：0x7和7是一样的
 - 使用取反获得特定位为0的二进制数（先构造出这个数的位相反数，再取反）
 - 例：bit 4~bit 10为0，其余位为1
 - 先构造bit 4~bit 10为1，其余位为0的数，再取反， $\sim(0x7f << 4)$

总结：与0与清零，与1或置1，特定位取反用异或

练习

- bit 3 置1: $a \mid=(1<<3)$
- bit 3~bit 7置1: $a \mid=(0b11111<<3)$ 或者 $a \mid=(0x1f<<3)$
- bit 5清除: $a \&=(\sim(1<<5))$
- bit 15~bit 23清除: $a\&=(\sim(0x1ff<<15))$
- 取出bit 3~bit 8: $a\&=(0x3f<<3)$, $a>>=3$
 - bit 3~bit 8不变, 其余清零, 然后右移三位 (先取数, 再移动)