**位带操作**

参考<官方开发手册>P29总线架构

参考<stm32-Datasheet>P28存储器映像

1. **原理**
2. **位带操作的本质是：**

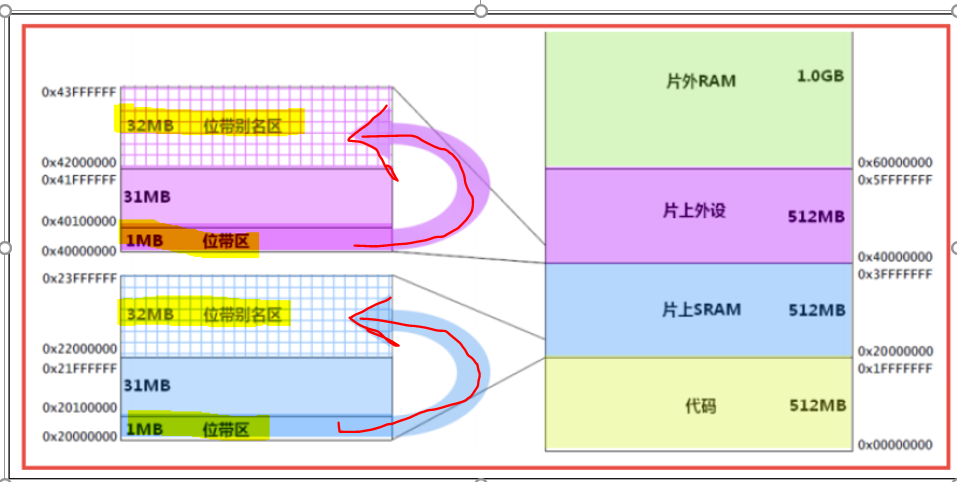
给寄存器的**每个位**赋予一个新地址，这每个位就会膨胀成**一个字**(4个字节32bit，因为32bit操作最快)。然后对这个字地址操作就相当于给这个寄存器的位进行操作。但这个字只有最低位有效.

理论上，每个寄存器都能映射到位带别名区，但实际只映射GPIO.其它的寄存器映射无意义。

1. **干的事就是：**

把**寄存器的每一位单独提出来操作**。而不是每次最少只写入32位来对整个寄存器操作

1. **片上的位带**



片上外设和SRAM均有一个1M的位带区。**位带区**里的每个位均可以通过**位带别名区**的地址来访问。位带区里的一个位对应位带别名区的一个字(4字节)。

如图位带区1M, 因为位带区里的一个位对应位带别名区的一个字(4字节)。所以**位带别名区**为32M.

* 1. 片上SRAM(存储局部/临时变量，掉电丢失)：

位带区地址： 0x2000 0000~0x2010 0000 (10 0000h/1024/1024=1M)

位带别名区地址： 0x2200 0000~0x23FF FFFF (1FF FFFFh/1024/1024=32M)

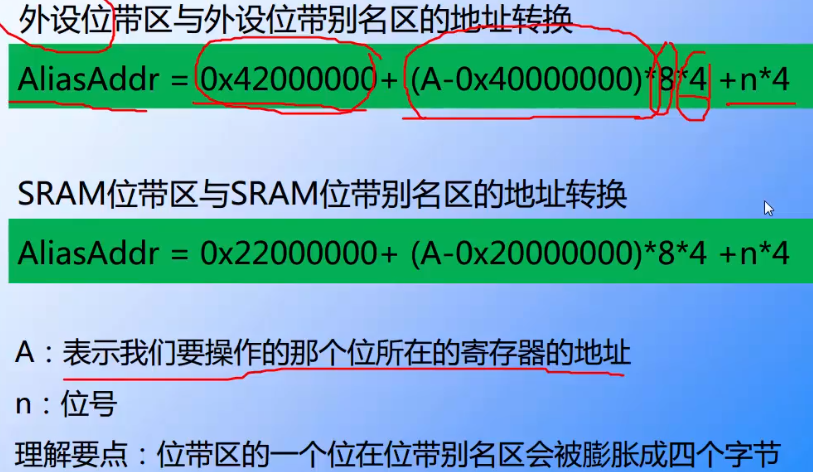
1M膨胀成32M

* 1. 片上外设(各种外设的地址)：

位带区地址： 0x4000 0000~0x4010 0000

位带别名区区地址：0x4200 0000~0x43FF FFFF

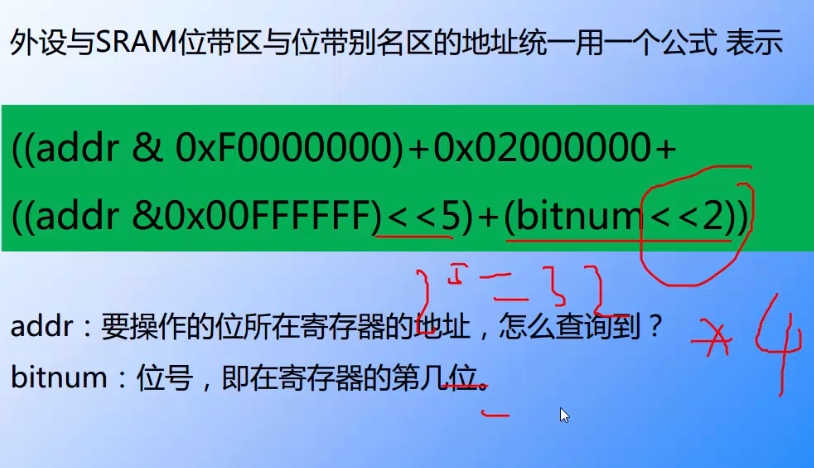
1. 位带区地址和位带别名区地址**转换公式**
   1. **核心概念**：位带区里的一个位对应位带别名区的一个字,一个地址对应一个字节
   2. **公式：**



为什么公式是这么表示的呢？？？

* 1. A是寄存器地址，A-0X4000 0000相当于A-位带区的起始地址。算出寄存器A**相对于位带区起始地址的偏移地址**
  2. 在膨胀成一个字(4个字节)，所以\*8\*4
  3. 然后根据位带别名区的起始地址：0x4200 0000，开始排列。占4个字节，所以N\*4.
  4. 最后再+位带别名区的起始地址

**4.3把上图这个公式，编写成程序如下：**



* 1. (addr&0xF000 0000)取出最高位 (二者公式就最高位不同)
  2. +0x0200 0000 加上次高位，片上外设和片上SRAM的次高位都是0x0200 0000
  3. (addr&0x00FF FFFF)取出低6位，高两位怎么都会被减掉
  4. <<5 = 8\*4
  5. bitnum= <<2N\*4