

Chip0N-IDE编译器效率实现说明

一、位取反:

```
Flagbit0 = ~Flagbit0 ;//效率低  
Flagbit0 = ! Flagbit0 ;//效率高
```

```
if(Flagbit0 ==0) //效率最高  
{ Flagbit0 = 1;}  
else  
{ Flagbit0 = 0;}
```

二、位判断:

```
if(DCIN!=1) //效率低  
if(DCIN==0) //效率高
```

三、定义

在RAM空间够的情况下,用结构体定义的位变量没有定义成字节型效率高,采用指针表达效率较低,尽量少用。

四、表达式

```
if(vot_value<(vot_temp-1)) //效率低  
  
vot_temp=vot_temp-1;  
if(vot_value<vot_temp) //效率高
```

公共表达式 提前获取结果到中间变量中。

五、逻辑运算

```
函数带返回值  
dc_value=adc(2)/2; //效率低  
  
dc_value=adc(2);  
dc_value=dc_value/2; //效率高
```

六、简单结构体、结构联合体

- 1、结构体定义不应太复杂,建议简单数据集合,不包含大数组、结构体。
- 2、位结构体实现程序标记,实现置1、清零操作。
- 3、结构联合体,整数和字节组合实现AD值的获取。

七、嵌汇编实现

```
__asm  
    BANKSEL  Var_1  
    MOV  R0, #0x32  
    MOV  Var_1, R0  
  
    CLR  PSW, 0  
    RLRC Var_1          ;:// 偏移量  
    ADD  R0, #(_Arr_Var+0) ;://操作地址  
    BANKSEL _Arr_Var  
    LD   R1, [R0]  
  
    PAGESEL _Fun_A  
    CALL _Fun_A  
    PAGESEL $           ;://函数调用  
__endasm;
```

嵌汇编实现采用如上示例的引入。可以采用 `#if 1/0 #else #endif` 的宏定义 一个条件下实现C语言，一个条件下使用嵌汇编实现。 其中涉及到的变量均使用 `BANKSEL _xx`的格式，由编译器自动优化数据区的定位，如果芯片的特殊寄存器存在多个区，也需要添加该声明，函数调用采用`PAGESEL _uuu`的格式来声明高位的处理，因指令只传递12个bit的地址，低13位及以上由编译器调用，针对4K及以下的芯片不需要改声明，但可以保留由编译器优化掉。