

## M16Cシリーズ用Cコンパイラパッケージ M3T-NC30WAご使用上のお願い --制御式で複数のビット判定を行う場合の注意事項--

M16Cシリーズ用Cコンパイラパッケージ M3T-NC30WA の使用上の注意事項を連絡します。

- 制御式で複数のビット判定を行う場合について

### 1. 該当製品

M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, および R8C/Tiny用  
Cコンパイラパッケージ V.1.00 Release 1 ~ V.5.40 Release 00(A)

### 2. 内容

制御式で&&演算子を用いて2つまたは3つのビット判定を行うと誤ったコードを出力する場合があります。

### 3. 発生条件

以下の条件を全て満たした場合に発生します。

- (1) &&演算子を用いて2つまたは3つのビットを判定する制御式がある。
- (2) (1)のビット幅はいずれも1ビットである。
- (3) (1)のビットが含まれるビットフィールドはchar型である。
- (4) (1)のビットはいずれも、ビットフィールドの下位4ビットもしくは、上位4ビットの範囲に収まっている。
- (5) 判定するビットの位置は次のいずれかに該当する。
  - (a) (1)において2つのビットを判定する場合、各ビットの間が1もしくは、2ビット空いている。

(b) (1) において3つのビットを判定する場合、3つのうち1つが他の2つから1ビット離れている。

条件に該当するビットの組み合わせ例（下位4ビットの場合）：

ビット位置	7	6	5	4	3	2	1	0
パターン1			*		*			
パターン2		*				*		
パターン3			*		*			
パターン4		*	*	*				
パターン5		*	*		*			

(6) コンパイル時にオプション-O4を指定している。

発生例

```
-----  
struct{  
    unsigned char b0:1;    // 発生条件(2), (4) および (5)  
    unsigned char b1:1;  
    unsigned char b2:1;  
    unsigned char b3:1;    // 発生条件(2), (4) および (5)  
    unsigned char b4:1;  
    unsigned char b5:1;  
    unsigned char b6:1;  
    unsigned char b7:1;  
}bit;    // 発生条件(3)  
  
int main()  
{  
    if(bit.b0 && bit.b3){ // 発生条件(1)  
        return 1;  
    }  
}
```

-----

#### 4. 回避策

以下のいずれかの方法で回避してください。

(1) &&を用いずにifをネストして記述してください。

例：

