

M16Cファミリ用Cコンパイラパッケージ ご使用上のお願い

M16Cファミリ用Cコンパイラパッケージの使用上の注意事項を連絡します。

1. 該当製品

M32Cシリーズ*1用Cコンパイラパッケージ (M3T-NC308WA)
V.5.00 Release 1 ~ V.5.40 Release 00

M16Cシリーズ*2用Cコンパイラパッケージ (M3T-NC30WA)
V.5.10 Release 1 ~ V.5.40 Release 00A

*1 M32C/80, M16C/80, およびM16C/70シリーズの総称です。

*2 M16C/60, /30, /20, /10, /TinyおよびR8C/Tinyシリーズの総称です。

2. 内容

関数呼び出しの前後で、ビットフィールドへの書き込みおよびビットフィールドからの読み出しを行うと、ビットフィールドの値が間違っている場合があります。

3. 発生条件

以下の条件すべてに該当する場合に発生する場合があります。

- (1) 最適化オプション-O[1-5],-OR,-OSのいずれかを指定している
- (2) 記憶クラスが自動記憶領域であるビットフィールドがある。
- (3) (2)のビットフィールドは、volatile修飾されていない。
- (4) (2)のビットフィールドに書き込みを行っている。
- (5) (4)の後に以下のいずれかに該当するソースがある。
 - (a) (2)のビットフィールド構造体のアドレスを引数とする関数呼び出しがある。
 - (b) (2)のビットフィールド構造体のアドレスを別の変数に代入した直後に関数呼び出しがある。
- (6) (5)の関数の中で(2)のビットフィールドの値を変更している。
- (7) (5)の関数呼び出し後に(2)のビットフィールドを参照している。

発生例

```

-----
struct tag{    // 発生条件(2), (3)
    unsigned char b0:1;
    unsigned char b1:1;
    unsigned char b2:1;
    unsigned char b3:1;
    unsigned char b4:1;
    unsigned char b5:1;
};

main()
{
    struct tag bit2;

    bit2.b1 = 1;    // 発生条件(4)
    func(&bit);    // 発生条件(5)-(a)

    if(bit2.b1==1) // 発生条件(7)
        sub();
}
-----

```

4. 回避策

ビットフィールドへの書き込みと関数呼び出しの間にダミーのasm関数を挿入してください。

```

-----
.....
main()
{
    struct tag bit2;

    bit2.b1 = 1;
    asm();    // asm関数を挿入
    func(&bit);
}
-----

```

5. 恒久対策

本問題は以下のバージョンで改修済みです。下記バージョン以降をご使用ください。

M32Cシリーズ用Cコンパイラパッケージ (M3T-NC308WA) V.5.41 Release 00

M16Cシリーズ用Cコンパイラパッケージ (M3T-NC30WA) V.5.42 Release 00

【免責事項】

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。

© 2010-2016 Renesas Electronics Corporation. All rights reserved.