

## コンパイラパッケージ M3T-NC308WA および M3T-NC30WA 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M3T-NC308WAおよびM3T-NC30WAの使用上の注意事項を連絡します。

- for文のループ内で配列型変数をアクセスする場合の注意事項

### 1. 該当製品

- M32C/90, M32C/80シリーズ, M16C/80, および M16C/70シリーズ用  
M3T-NC308WA V.5.00 Release 1 ~ V.5.40 Release 00
- M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, および R8C/Tiny用  
M3T-NC30WA V.5.10 Release 1 ~ V.5.40 Release 00A

### 2. 内容

For文のループ内で、繰り返しの度に減じる変数を添え字に使用して配列型変数の要素をアクセスする文に対して、誤ったアドレスを参照するコードを生成することがあります。

#### 2.1 発生条件

以下の条件を全て満たす場合に、発生することがあります。

なお、以下でいう定数とは、コンパイル時に計算して定数に置き換えられたものも含まれます。

- (1) 以下の式1、式2、式3のいずれも省略されていないfor文がある。  
for (式1; 式2; 式3) 文
- (2) (1)のforループの制御を行うための変数(以下、変数cという)が、forループを含む関数内で記憶クラス指定子static、およびexternのいずれも伴わずに宣言されており、かつ、その型は

charまたはunsigned charである。

- (3) 全ての関数外で、short, unsigned short, int, および unsigned int のいずれかの型の配列型変数を宣言している。
- (4) (1)の式1は、変数cに定数を代入する式である。
- (5) (1)の式2は、変数cと定数を、演算子<, >, <=, >=, および != のいずれかで比較する式である。
- (6) (1)の式3は、変数cから1を減らす式である。
- (7) ループ内に、変数cを添字にを使って(3)の配列をアクセスする文がある。
- (8) (1)のfor文のループ内に、以下のいずれも含まない。
  - a. ラベル付き文 (caseラベルおよびdefaultラベルのみのは除く)
  - b. asm関数
  - c. #pragma ASM および #pragma ENDASM ディレクティブ
  - d. (1)のfor文の先頭へ戻るcontinue文
  - e. (1)のfor文から抜けるbreak文
  - f. (2)の変数cの値を変更する式
  - g. ポインタ変数を使って任意のメモリへの間接書き込みを行う式
- (9) コンパイル時に、最適化オプション-OSを使用している。
- (10) 以下のいずれかの条件を満たす。
  - a. 最適化オプション-OLU (繰り返し制御コードの代わりにループ内の文を繰り返し回数分コード出力する)を使用していない。
  - b. -OLUオプションを回数指定なしで使用していて、かつ、(4)、(5)、および(6)から求められる(1)のforループの繰り返し回数が5回以下である。
  - c. -OLUオプションを回数指定ありで使用していて、かつ、(4)、(5)、および(6)から求められる(1)のforループの繰り返し回数が、-OLUオプションで指定した回数以下である。
- (11) M3T-NC30の場合、(3)の配列がfar領域に配置されている。(M3T-NC308の場合は配列の配置に関わらず該当する。)

## 2.2 発生例

(ソース例)

---

```

--
extern int far arr[10];          /* 発生条件(3),(11) */

void func(void)
{
    char c;                      /* 発生条件(2) */

    for (c = 7; c > 4; c--) { /* 発生条件(1),(4),(5),(6) */
        arr[c] = 0;          /* 発生条件(7),(8) */
    }
}

```

-----

```

--
(M3T-NC30による生成コード例)

```

-----

```

--
_func:
    mov.b  #07H,R0H
    mov.w  #000eH,A0
    mov.b  R0H,R0L
L1:
    mov.w  #0000H,R1
    ste.w  R1,_arr[A0]
    add.w  #0002H,A0 ; <--- 誤り。sub.w #0002H,A0 が正しい。
    dec.b  R0L
    cmp.b  #04H,R0L
    jgtu  L1
    rts

```

-----

```

--

```

### 3. 回避策

forループ本体の文のいずれかに、ダミーのラベルを付けてください。

-----

```

extern int far arr[10];

```

```

void func(void)
{
    char c;

```

```
    for (c = 7; c > 4; c--) {  
dummy:    arr[c] = 0;    /* この行にダミーラベルを追加 */  
    }  
}
```

---

#### 4. 恒久対策

次期バージョンで改修する予定です。

---

#### [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.