

M3T-NC30WAご使用上のお願い -- 4バイト整数型の変数に対する右シフトについて --

M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, およびR8C/Tinyシリーズ用 Cコンパイラパッケージ (アセンブラ・統合化開発環境付き) M3T-NC30WAの使用上の注意事項を連絡します。

- 4バイト整数型の変数に対して右シフトする場合の注意事項

1. 該当製品

M3T-NC30WA V.4.00 Release 1 ~ V.5.30 Release 1

2. 内容

4バイト整数型の変数に対する右シフトを実行した後にシフト前の値を参照している場合、不正なコードを生成する場合があります。

2.1 発生条件

以下の条件を全て満たす場合に発生します。

- (1) コンパイル時に最適化オプション -O3, -O4, -O5, -OR, -OS のいずれか1つ以上を使用している。
- (2) 4バイト整数型の変数、または4バイト整数型の変数を使用した演算の結果に対して右シフトした結果を2バイトの整数型変数に格納している。
- (3) (2)で右シフトするビット数は、11以上16以下の定数である。
- (4) (2)のシフト結果を2バイト整数型の変数に格納した後に、シフト前の4バイト整数型の変数を参照している。

2.2 発生例

- (1) 4バイト整数型の変数に対して右シフトしている場合

```
-----  
long  m, n;
```

```

int i;

void func(void)
{
    register long a = m;
    i = a >> 11; /* 発生条件(2)(3) */
    n = a; /* 発生条件(4) */
}
-----

```

(2) 4バイト整数型の変数を使用した演算の結果に対して右シフトしている場合

```

-----
long o, p;
int ii;

void func2(void)
{
    ii = (o - 1) >> 11; /* 発生条件(2)(3) */
    p = (o - 1); /* 発生条件(4) */
}
-----

```

3. 回避策

- (1) 右シフトする4バイト整数型の変数がregister宣言されており、かつコンパイルオプション"-fenable_register[-fER]"を指定している場合は次の2点を実行してください。
 - (a) 発生条件(2)の右シフトの対象になる4バイト整数型の変数に対して、register修飾子を使用しないでください。
 - (b) 4バイト整数型の変数を右シフトする式(発生条件(2)に該当する行)と4バイト整数型の変数を参照する式(発生条件(4)に該当する行)の間にダミーのasm関数を挿入してください。

```

-----
long m, n;
int i;

void func(void)
{
    long a = m; /* (a) register修飾をしない */
    i = a >> 11;
    asm(); /* (b) ダミーのasm関数を挿入 */
    n = a;
}

```

```
}
```

- (2) (1)以外の場合は、4バイト整数型の変数を右シフトする式(発生条件(2)に該当する行)と4バイト整数型の変数を参照する式(発生条件(4)に該当する行)の間にダミーのasm関数を挿入してください。

4バイト整数型の変数に対して右シフトしている場合:

```
long  m, n;
int   i;

void func(void)
{
    register long  a = m;
    i = a >> 11;
    asm(); /* ダミーのasm関数を挿入 */
    n = a;
}
```

4バイト整数型の変数を使用した演算の結果に対して右シフトしている場合:

```
long  o, p;
int   ii;

void func2(void)
{
    ii = (o - 1) >> 11;
    asm(); /* ダミーのasm関数を挿入 */
    p = (o - 1);
}
```

4. 恒久対策

M3T-NC30WA V.5.30 Release 02以降では本問題は改修されています。

[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。