

## M16Cシリーズ,R8C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シリーズ,R8C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。

- 単精度浮動小数点型除算に関する注意事項

### 1. 該当製品

M16Cシリーズ,R8Cファミリ用Cコンパイラパッケージ (M3T-NC30WA)  
V.5.42 Release 00 ~ V.5.44 Release 00

### 2. 内容

単精度浮動小数点数の除算を行なうと、商の仮数部下位16ビットが誤った値になることがあります。この結果、仮数部は10進数で3桁程度の精度の不正確な値になります。

通常、単精度浮動小数点数の除算の精度は10進数で6桁程度です。

#### 2.1 発生条件

以下の条件をすべて満たす場合に発生します。

(1) double型またはfloat型の除算を行っている。

(2) (1)が double型の除算の場合、オプション `-fdouble_32(-fD32)`  
`-OR_MAX`, `-OS_MAX` のいずれかを選択している時に発生する。  
(1)が float型の除算の場合、オプション指定に関係なく発生する。

(3) コンパイルの結果、ランタイムルーチン `_f4div` を呼び出すコードが生成されている。

(4) `_f4div`では、被除数を上位と下位に分け、除算命令`divu`を2回実行して商を求めている。1回目の除算の剰余は、2回目の除算の被除数の上位16ビットとなる。この時、1回目の除算の剰余と2回目の除算の上位16ビットが同じ値となる場合に発生する。

## 2.2 発生例

### C言語ソース

```
-----  
#include <stdio.h>  
  
int main(void)  
{  
    float  a, b, c;  
  
    a = 60.0F;  
    b = 1.2F;  
    c = a / b;  
    printf("%f¥n", c);  
  
    a = 411999436800.0F;  
    b = 3483491.0F;  
    c = a / b;  
    printf("%f¥n", c);  
    return 0;  
}  
-----
```

### 実行結果

```
-----  
49.948486  
118130.000000  
-----
```

### 正しい結果

```
-----  
49.999996  
118272.000000  
-----
```

## 3. 回避策

被除数または除数をlong double型にキャストしてください。

```
-----  
#include <stdio.h>  
  
int main(void)  
{  
    float  a, b, c;
```

```
a = 60.0F;  
b = 1.2F;  
c = (long double)a / b;  
printf("%f¥n", c);
```

```
a = 411999436800.0F;  
b = 3483491.0F;  
c = (long double)a / b;  
printf("%f¥n", c);  
return 0;
```

```
}
```

---

#### 4. 恒久対策

V.5.45 Release 00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。