

M16Cシリーズ、R8Cファミリ用C/C++コンパイラパッケージ (M3T-NC30WA) V.6.00 Release 00ご使用上のお願い

M16Cシリーズ、R8Cファミリ用C/C++コンパイラパッケージ (M3T-NC30WA) V.6.00 Release 00 の使用上の注意事項を連絡します。

- 呼び出されないinline関数を定義している場合の注意事項

1. 内容

呼び出されないinline関数を定義している場合、リンケージエディタ (optlnk) でInternal Errorが発生する場合があります。
もしくは、呼び出されないinline関数で参照している外部変数をデバッガで正しく参照できない場合があります。

2. 発生条件

2.1 発生条件1

以下の条件をすべて満たす場合に発生することがあります。

- (1) inline関数を定義している。
- (2) (1)のinline関数内で外部変数を参照している。
- (3) (1)のinline関数の呼び出しがなく、かつ(2)の外部変数が同一ソースファイル内で参照されていない。

注：インクルードファイル内に記載されている場合も同一ファイル内とみなします。

発生例:

```
-----  
extern int val;  
inline void func( void ) // 発生条件(1)および (3)  
{
```

```
    val = 0x01;        // 発生条件(2)および (3)
}
```

2.2 発生条件2

以下の条件をすべて満たす場合に発生することがあります。

(1) -Ostatic_to_inline (-OSTI) かつ -Oforward_function_to_inline (-OFFTI) オプションを使用している。

または、-OS_MAX (-OSM)を使用している。

(これらのオプションにより、static関数がinline関数として扱われます。)

(2) static関数を定義している。

(3) (2)のstatic関数内で外部変数を参照している。

(4) (3)の外部変数が、同一ソースファイル内で参照されていない。

注： インクルードファイル内に記載されている場合も

同一ファイル内とみなします。

(5) (2)のstatic関数の呼び出しがなく、最適化により関数定義が削除される。

発生例:

```
extern int val;
static void static_func( void ) // 発生条件(2)
{
    val = 0x01;                // 発生条件(3)
}
```

3 回避策

3.1 発生条件1に対する回避策

inline関数を static関数に変更して使用してください。

または、呼び出されないinline関数は定義しないでください。

3.2 発生条件2に対する回避策

-OFFTI指定を外し最適化によるstatic関数の削除を抑止してください。

または、呼び出されないstatic関数は定義しないでください。

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.