

## CubeSuite+ CXコンパイラご使用上のお願い

CubeSuite+ CXコンパイラ (V850E2Mコア用) の使用上の注意事項を連絡します。

- 複数の比較演算を含むif文についての注意事項 (No. 10)
- 命令を持たない無限ループを含む関数についての注意事項 (No. 11)

### 1. 複数の比較演算を含むif文についての注意事項 (No. 10)

#### 1.1 該当バージョン

CubeSuite+ CXコンパイラ V1.20

#### 1.2 内容

if文中に「||」演算子でつながれた複数の条件式があり、条件式の定数値が連続している場合、定数値のうち最小値以外の値との比較演算が削除されることがあります。

他の最適化などの影響によって発生しない場合もあります。

#### 1.3 発生条件

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) 最適化オプション -O、-Osize、-Ospeedのいずれかを使用している。
- (2) 以下の条件をすべて満たすif文がある。
  - (2-1) 定数と以下のいずれかの比較を行う複数の条件式を含んでいる。  
通常変数、ポインタ変数、配列要素、構造体メンバ および  
共用体メンバ
  - (2-2) (2-1)の条件式は「||」演算子でつながれている。
- (3) (2-1)の条件式の定数値が連続している。
- (4) (2-1)の定数と比較している通常変数、配列要素、構造体メンバおよび共用体メンバは、以下のいずれかをtypedefした型である。
  - (signed) short
  - unsigned short
  - (signed) char
  - unsigned char

- \_Bool
- enum

発生例 :

```
-----  
typedef short SS;  
SS ary[10];  
int sss;  
.....  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){  
    sss = 0;  
}  
else {  
    sss = 1;  
}  
-----
```

この場合、`ary[0] == 1`の条件式が誤って削除されます。

#### 1.4 回避策

以下のいずれかの方法で回避してください。

なお、チェックツールによって、誤って最適化される比較演算の行番号を特定することが可能です。

チェックツールの入手については、ルネサス営業または特約店へお問合せください。

- (1) 該当する通常変数、ポインタ変数、配列要素、構造体メンバおよび共用体メンバの型を`typedef`しない。

```
-----  
short ary[10];  
.....  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){  
-----
```

- (2) 該当する通常変数、配列要素、構造体メンバまたは共用体メンバを一旦、テンポラリ変数に格納する。

```
-----  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){  
    typedef short SS;  
    SS ary[10];  
    SS tmp = ary[0];  
    .....  
    if( tmp == 0 || tmp == 1 || tmp == 5){  
-----
```

#### 1.5 恒久対策

## 2. 命令を持たない無限ループを含む関数に関する注意事項 (No. 11)

### 2.1 該当バージョン

CubeSuite+ CXコンパイラ V1.00~V1.11

### 2.2 内容

命令を持たない無限ループを含む関数をコンパイルした場合、コンパイルが終了しない場合があります。

### 2.3 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) 最適化オプションとして -O、-Osize、-Ospeed のいずれかを使用している。
- (2) 割り込み関数、または、割り込み関数ではないが仮引数を有する関数がある。
- (3) (2)の関数の最初の文が命令を持たない無限ループである。

発生例1:

```
-----  
#pragma interrupt INTSW0 func1  
void func1(void)  
{  
    for(;;);  
    .....  
-----
```

発生例2:

```
-----  
void func2(int a)  
{  
    for(;;);  
    .....  
-----
```

### 2.4 回避策

関数の最初に `__asm("%n");` を追加してください。

発生例1の回避例 :

```
-----  
#pragma interrupt INTSW0 func1  
void func1(void)
```

```
{
__asm("¥n");    // 追加
for(;;);
.....
-----
```

## 2.5 恒久対策

CubeSuite+ CXコンパイラV1.20 で改修済みです。

CXコンパイラV1.20は、CubeSuite+パッケージ に含まれています。

---

### [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。

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