

R32C/100シリーズ用Cコンパイラパッケージ ご使用上のお願い

R32C/100シリーズ用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。

- 配列を連続してアクセスする場合の注意

1. 該当製品

R32C/100シリーズ用Cコンパイラパッケージ
V.1.01 Release 00 および V.1.02 Release 00

2. 内容

配列への代入式が3つ以上連続している場合に、誤ったコードを生成する場合があります。

2.1 発生条件

以下の条件をすべて満たす時、発生する場合があります。

- (1) コンパイルオプション -O1 ~ -O5, -OS, -OR, -OS_MAX (-OSM) または -OR_MAX (-ORM)を選択している。
- (2) 64ビット幅を除く整数型、またはポインタ型の配列に対して、変数または定数を代入する式が3つ以上連続している。
- (3) (2)のすべての代入式は、配列の添え字が以下のいずれかに該当している。
 - (3-1) 8または16ビット幅で、整数型の外部変数である。
 - (3-2) 8または16ビット幅で、符号なし整数型の変数 (外部変数およびローカル変数)である。ただし、配列が16ビット幅の型の場合に限る。
 - (3-3) 32ビット幅で、整数型の外部変数およびローカル変数である。
- (4) (2)の代入式のうち、配列の型のビット幅が同一で、かつ、同一の添え字を使用した式が2つ以上連続している。
- (5) (4)の代入式の直後の代入式は、配列の型のビット幅か、または、添え字の変数の少なくとも一方が(4)と異なる。

2.2 発生例

コマンドライン : nc100 -O sample.c

発生ソース :

```
-----  
char c, cary1[4], cary2[4];  
short s, sary1[4], sary2[4], sary3[4];  
short index1;  
unsigned short index2;  
  
void func()  
{  
    unsigned short index3;  
  
    sary1[index1] = s; // 発生条件(2)、(3-1)および(4)  
    sary2[index1] = s; // 発生条件(2)、(3-1)および(4)  
    cary1[index2] = c; // 発生条件(2)、(3-1)、(4)および(5)  
    cary2[index2] = c; // 発生条件(2)、(3-1)および(4)  
    sary3[index3] = s; // 発生条件(2)、(3-2)および(5)  
}
```

3. 回避策

発生条件に該当するすべての箇所で、(4)の式の最後と(5)の式の間にはasm関数を挿入して下さい。

回避例 :

```
-----  
char c, cary1[4], cary2[4];  
short s, sary1[4], sary2[4], sary3[4];  
short index1;  
unsigned short index2;  
  
void func()  
{  
    unsigned short index3;  
  
    sary1[index1] = s;  
    sary2[index1] = s;  
    asm(); // ここにasm();を挿入  
    cary1[index2] = c;  
    cary2[index2] = c;  
    asm(); // ここにasm();を挿入  
    sary3[index3] = s;
```

}

4. 恒久対策

R32C/100シリーズ用Cコンパイラパッケージ V.1.02 Release 01で改修
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.