

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。

- 剰余式の除数に配列型の変数を使用する場合の注意事項

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 ~ V.1.02 Release 01A

2. 内容

剰余式の除数に配列型の変数を使用した場合に、誤ったコードが生成されます。

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) 剰余式の被除数が変数である。(注1)
- (2) 剰余式の除数が配列型の変数である。(注1)
- (3) (2)の配列の添え字が自動変数である。(注1)(注2)
- (4) (1)および(2)の変数を拡張機能#pragma EXTMEMで宣言していない。
- (5) (1)、(2)および(3)の変数の型が次のいずれかである。
 - unsigned char
 - signed char
 - unsigned short
 - signed short
 - unsigned int (注3)
 - signed int (注3)
- (6) (1)、(2)および(3)の変数のビット幅は同一である。

注1: コンパイラが定数伝播を行った結果、変数が定数に置き換わった場合を除く。

注2: 配列をポインタで表記している場合はオフセットが自動変数。

注3: コンパイルオプション -fint_16(-fI16) を選択した場合。

2.2 発生例

コマンドライン: nc100 -S sample.c

発生ソース:

```
-----  
int i;  
unsigned char Dividend;          // 発生条件(1)、(5)および(6)  
unsigned char Divisor[];        // 発生条件(2)、(5)および(6)  
  
void func(void)  
{  
    unsigned char Index = i;     // 発生条件(3)、(5)および(6)  
  
    i = Dividend % Divisor[Index]; // 発生条件(1)、(2)および(3)  
}
```

現象: 添え字の値が被除数と同じになります。

生成コード:

```
-----  
_func:  
    .line 7  
;## # C_SRC :    unsigned char Index = i;  
    mov.b  _i:16,R0L    ; Index  
    .line 9  
;## # C_SRC :    i = Dividend % Divisor[Index];  
    extz.bw _Dividend:16,R0  
    .inxlb R0L          <--- 添え字の値が被除数になる  
    edivu.b _Divisor:16,R0  
    extz.bl R0H,_i:16  
    .line 10  
;## # C_SRC :    }  
    rts
```

3. 回避策

除数を一時変数に代入し、この一時変数を除数として使用してください。

回避例:

```
-----  
int i;  
unsigned char Dividend;  
unsigned char Divisor[6];  
  
void func(void)
```

```
{  
    unsigned char Index = i;  
    unsigned char temp = Divisor[Index]; // 除数を一時変数に代入  
  
    i = Dividend % temp;                // 除数に一時変数を使用  
}
```

4. 恒久対策

改修予定はありません。

[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。