

H8SX, H8S およびH8ファミリ用 C/C++コンパイラパッケージ V.6 ご使用上のお願い

H8SX, H8S およびH8ファミリ用C/C++コンパイラパッケージ V.6 の使用上の注意事項を連絡します。

- __asm文内でローカル変数をアクセスする関数がインライン展開された場合の注意事項(H8C-0044)

1. 該当製品

V.6.00 Release 00 ~ V.6.01 Release 02

製品型名:

Windows版	R0C40008XSW06R
Solaris版	R0C40008XSS06R
HP-UX版	R0C40008XSH06R

2. 内容

__asm文内でローカル変数をアクセスする関数がインライン展開された場合に、インライン展開された関数内のローカル変数ではなく、その関数の呼び出し元となる関数の変数をアクセスすることがあります。

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) 以下の(A)または(B)を満たしている。
 - (A) V.6.00 Release 00 ~ Release 03を使用している場合
CPUオプションにH8SXN, H8SXM, H8SXAまたはH8SXXを使用している。
 - (B) V.6.01 Release 00 ~ Release 02を使用している場合

CPUオプションに2000N, 2000A, 2600N, 2600A, H8SXN, H8SXM, H8SXA, H8SXXまたはAE5を使用している。

ただし出力オブジェクト互換オプション-legacy=v4を使用しているときは、2000N, 2000A, 2600N, および2600Aは該当しません。

- (2) `__inline`または `#pragma inline` を使用した関数が存在する。
またはコンパイルオプション `-speed=inline` を使用している。
- (3) (2)によりインライン展開された関数に `__asm{ }`文がある。
- (4) (3)の `__asm{ }`文内で局所変数または引数をアクセスするアセンブラ命令を記述している。
- (5) (2)の関数の呼び出し元となる関数に局所変数がある。

Cソース例:

```
-----  
__inline void sub(){           //発生条件(2)  
    volatile unsigned char tmp;  
    __asm {                   //発生条件(3)  
        stc.w exr, @(tmp, sp) //発生条件(4)  
        ldc.b #0x07, exr  
    }  
  
    tmp &= 0x07;  
}  
  
void main(void){  
    volatile unsigned char sample = 0; //発生条件(5)  
    sub();  
}
```

生成コード :

```
-----  
_main:  
subs    #4,sp  
mov.b   #0:8,@sp  
stc.w   exr,@(0:32,sp)  
;間違って、局所変数"sample"をアクセスしている。  
;正しくは、stc.w exr,@(2:32,sp)。  
ldc    #7:8,exr  
and.b   #7:8,@(2:2,sp)  
adds    #4,sp
```

4. 回避策

以下のいずれかの方法で回避してください。

- (1) インライン展開された関数に対して `__inline`, `#pragma inline` および コンパイルオプション `-speed=inline` を使用しない。

発生例の回避例：

```
-----  
void sub(){          //__inlineを使用しない  
    volatile unsigned char tmp;  
    __asm {  
        stc.w exr, @(tmp, sp)  
        ldc.b #0x07, exr  
    }  
  
    tmp &= 0x07;  
}
```

- (2) `__asm{ }`内でアクセスする局所変数および引数を、外部変数を介してアクセスする。

発生例の回避例：

```
-----  
unsigned char dummy;    // コピー用の外部変数を用意する  
__inline void sub(){  
    volatile unsigned char tmp;  
    dummy = tmp;        // 外部変数へ代入  
    __asm {  
        stc.w exr, @dummy // 外部変数から代入  
        ldc.b #0x07, exr  
    }  
  
    tmp &= 0x07;  
}
```

5. 恒久対策

本内容はV.6.01 Release 03 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.