RENESAS ツールニュース

RENESAS TOOL NEWS 2007年03月16日：070316/tn5

M16Cファミリ用Cコンパイラパッケージ ご使用上のお願い

M16Cファミリ用Cコンパイラパッケージ(M3-NC308WAおよびM3T-NC30WA)の使用上の 注意事項を連絡します。

1. 該当製品

(1) M32Cシリーズ*1用Cコンパイラパッケージ (M3T-NC308WA)
V.5.40 Release 00 ～ V.5.41 Release 01
(2) M16Cシリーズ*2用Cコンパイラパッケージ (M3T-NC30WA)
V.5.40 Release 00 ～ V.5.43 Release 00

*1. M32C/80, M16C/80, およびM16C/70シリーズの総称です。
*2. M16C/60, /30, /20, /10, /TinyおよびR8C/Tinyシリーズの総称です。

2. 内容

引数がスタック渡しとなる関数では、正しく引数が参照できない場合があります。
もしくは、同関数を含むプログラムが暴走する場合があります。

3. 発生条件

以下の条件すべてに該当する場合に発生します。
(1) コンパイルオプション-ORまたは-OR_MAX(-ORM)が使用されている。
(2) コンパイルオプション-Ono_asmopt(-ONA)が使用されていない。
(3) 一つのファイル内で、引数がスタック渡しとなる関数(例：関数A)が、
別の2つ以上の関数(例：関数Bおよび関数C)から呼び出されている。
(4) 関数Bもしくは Cの少なくともどちらか一方が、関数を終了する命令exitdが
生成されるための次のいずれかの条件に該当している。
- スタック引数をもっている。
- auto変数を確保している（コンパイラが生成するテンポラリ領域も含む）。
- コンパイル時にオプション-genterを使用している。
(5) 関数Aから復帰後exitd命令までの間に、スタックを補正するコードが無い。
(6) 関数Aの引数の退避から関数Aをjsr命令で呼び出すまでのコードが
関数BとCとで同一となっている。

(7) 関数Aは、スタック引数を持っている。

発生例：

```
-------------------------------------------------
void  func1( void )  //  関数B
{
   . . . . . .
   comm_func ( int i, int *p, int *q );    //  関数A呼び出し
   . . . . . .
}
void  func2 ( void )  //  関数C
{
   . . . . . .
   comm_func ( int i, int *p, int *q );    //  関数A呼び出し
   . . . . . .
}
-------------------------------------------------
```

上記例では、関数BおよびCから関数Aを呼び出している箇所に対して次のコードが
生成されます。

```
-------------------------------------------------
   . . . . . .
        push.w    _q
        push.w    _p
        mov.w     _i,R1          // D
        jsr       $comm_func     // E
        exitd
   . . . . . .
-------------------------------------------------
```

コンパイルオプション(-ORまたは-OR_MAX)を使用することにより、上記DとEの
コードが以下に示すような共通関数_aopt_xxxに変換されます。
この結果、以下のFおよびGで退避した引数は_aopt_xxxを呼び出す際に使用した
スタック分を加えたアドレスに格納されます。
対して、$comm_func の生成コードは、DおよびEを_aopt_xxxに変換する前に生成
されたコードのままであるため、実際に引数が格納されている場所とは異なる
アドレスを参照することになります。

```
-------------------------------------------------
   . . . . . .
        push.w    _q         // F
        push.w    _p         // G
```

```
        jsr       _aopt_xxx
        exitd
    . . . . . .

      _aopt_xxx:
        mov.w     _i,R1
        jsr       $comm_func
        exitd
```

---

4. 回避策

以下いずれかの方法で回避してください。

(1) 上記例の関数Aが間接呼び出しとなる関数の場合は、-ONAオプションを追加してコンパイルしてください。

(2) 上記例の関数Aがスペシャルページ関数の場合は、-ONAオプションを追加してコンパイルするか、回避例のように関数呼び出し直後にダミーのasm関数を挿入してください。

(3) 上記例の関数Aが上記いずれでも無い場合は、-ONAまたは-OSAオプションを追加してコンパイルするか、もしくは回避例のように関数呼び出し直後にダミーのasm関数を挿入してください。

回避例:

```
void  func1(  void  )
{
   . . . . . .
   comm_func ( int i, int *p, int *q );
   asm();     // asm関数を挿入
   . . . . . .
}
```

---

5. 恒久対策

次バージョンで修正します。