

【注意事項】

R20TS0037JJ0100

Rev.1.00

2016.06.01 号

RL78 ファミリ用 C コンパイラパッケージ

概要

RL78 ファミリ用 C コンパイラパッケージ CC-RL の使用上の注意事項を連絡します。

1. 命令オペレーション一覧にない命令オペランドを記述した場合の注意事項 (CCRL#008)
2. 割り込みハンドラ内でレジスタを上書きするコードを出力する注意事項 (CCRL#009)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

1. 命令オペレーション一覧にない命令オペランドを記述した場合の注意事項 (CCRL#008)

1.1 該当製品

CC-RL V1.00.00 ~ V1.02.00

1.2 内容

mov 命令オペランドでオフセットを省略して命令オペレーション一覧にない命令オペランドを記述すると、オペランドを正しく補完できずにコード不正となる場合があります。

1.3 発生条件

以下のいずれかの条件を満たす時に発生します。

- (1) mov [DE],#byte または mov [HL],#byte を記述している。
- (2) mov ES:[DE],#byte または mov ES:[HL],#byte を記述し、第二オペランド#byte がラベル または ラベルを含む式である。

発生例

```
mov [DE], #3
mov [HL], #sym
mov ES:[DE], #sym
```

1.4 回避策

オフセットを省略せずに命令オペランドを記述してください。

```
mov [DE+0], #3
mov [HL+0], #sym
mov ES:[DE+0], #sym
```

1.5 恒久対策

次期バージョンで改修する予定です。

2. 割り込みハンドラ内でレジスタを上書きするコードを出力する注意事項 (CCRL#009)

2.1 該当製品

CC-RL V1.00.00 ~ V1.02.00

2.2 内容

割り込みハンドラ内で、スタックに退避した HL レジスタ または スタックに退避していないレジスタを上書きするコードを出力する場合があります。

2.3 発生条件

以下のすべての条件を満たす時に発生する場合があります。

ただし、割り込みハンドラ内の使用スタック量(.STACK)が 256 バイト未満の場合、スタックに退避していないレジスタを上書きするコードを出力することはありません。

- (1) #pragma interrupt または #pragma interrupt_brk を使用して割り込みハンドラを記述している。
- (2) (1)の割り込みハンドラでレジスタバンクを指定していない。
- (3) (1)の割り込みハンドラ内に inline_asm 関数の呼び出しがない。
- (4) (1)の割り込みハンドラ内に関数呼び出しの記述がないか、関数呼び出しが最適化によりインライン展開されアセンブラコードで CALL 命令がない。

● 発生例 1

スタックに退避した HL レジスタを上書きする場合

```
#pragma interrupt func /* 発生条件(1)(2) */

typedef struct
{
    unsigned int cnt;
    unsigned char flg;
} mem_t;
mem_t mem[3];

void func(void)
{
    int i;          /* HL レジスタ上書き */
    for ( i=0; i<3; i++ ) {
        mem[i].cnt--;
        if (0 == mem[i].cnt) {
            mem[i].flg = 1;
        }
    }
}
```

出力アセンブラコード例

```

_func:
    .STACK _func = 12
    push ax
    push bc
    push de
    push hl                ; (1)
    movw ax, #LOWW(_mem)
    movw [sp+0x00], ax    ; (2) HL レジスタ上書き

    pop hl                ; (1)
    
```

- (1) 出力アセンブラコードで、割り込みハンドラ先頭のレジスタ退避処理の最後に `push hl`、割り込みハンドラ末尾のレジスタ復帰処理の先頭に `pop hl` がある。
- (2) (1)の `push` 命令と `pop` 命令の間で、HL レジスタの内容が[SP+0] にある状態で[SP+0]への書き込みがある。

● 発生例 2

スタックに退避していないレジスタを上書きする場合

```

#pragma interrupt func2
void func2(void) {
    volatile char arr[0x300];
    arr[2] = 1;
}
    
```

出力アセンブラコード例

```

_func2:
    movw bc, ax    ; (1)BC レジスタ上書き
    movw ax, sp
    addw ax, #0xFD00
    movw sp, ax
    movw ax, bc
    
```

- (1) 出力アセンブラコードで、割り込みハンドラの先頭および末尾でスタックに退避、復帰せずに使用しているレジスタがある。上記出力アセンブラコード例では、BC レジスタが該当し、BC レジスタが上書きされる。

2.4 回避策

以下のいずれかの方法で回避してください。

- (1) #pragma 指令にレジスタバンク指定を記述する。
- (2) 空のアセンブリ記述関数(#pragma inline_asm)を用意し、割り込みハンドラ内で呼び出す。
- (3) 割り込みハンドラ内に関数呼び出しを記述し、コンパイルオプションに
-Onothing, -Oinline_level=0, または -Oinline_level=1 のいずれかを使用して、インライン展開を防ぐ。

2.5 恒久対策

次期バージョンで改修予定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.06.01	-	新規発行

ルネサスエレクトロニクス株式会社
 〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

■総合お問い合わせ先

<http://www.renesas.com/ja-jp/support/contact.html>

本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。

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。

すべての商標および登録商標は、それぞれの所有者に帰属します。