

【注意事項】

R20TS0240JJ0100

Rev.1.00

2017.12.01 号

RL78 ファミリー用 C コンパイラパッケージ

概要

RL78 ファミリー用 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. 関数が複数の引数を持ち、仮引数同士の代入または比較がある場合の注意事項 (CCRL#017)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

1. 関数が複数の引数を持ち、仮引数同士の代入または比較がある場合の注意事項 (CCRL#017)

1.1 該当製品

CC-RL V1.02.00 ~ V1.05.00

1.2 内容

関数が複数の引数を持ち、仮引数同士の代入または比較がある場合に、別の仮引数を参照するコードが不正になる場合があります。

1.3 発生条件

以下の (1) ~ (2) の条件をすべて満たす場合に、別の仮引数を参照するコードが不正になる場合があります。

- (1) 関数^(注)のインライン展開後の関数の処理において、(a) または (b) のいずれかの処理を行っている。

注: 関数内に、関数呼び出し、分岐、またはループが存在する場合、関数の先頭から最初の関数呼び出し、分岐、またはループまでの間に次のいずれかの処理が少なくとも1つあることが条件となります。なお、分岐先が実質1つしかない分岐や、最適化によりループ展開されたループの場合は含まれません。

分岐先が実質1つしかない分岐例

```
if (1) {  
    ...  
}  
else {  
    ...  
}
```

(a) レジスタ渡しの仮引数を格納する処理で、格納先のアドレスが次の(a-1) ~ (a-4) のいずれかである。

(a-1) レジスタ渡しの仮引数 (near ポインタ) が指すアドレス

例

```
*r2 = r1;
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)
仮引数 r1 と r2 が同一の仮引数である場合も含まれます。

(a-2) レジスタ渡しの仮引数 (near ポインタ) ±定数オフセットが指すアドレス

例

```
*(r2+1) = r1;
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)
仮引数 r1 と r2 が同一の仮引数である場合も含まれます。

(a-3) 定数 (near ポインタ) ±レジスタ渡しの仮引数 (オフセット) が指すアドレス

例

```
*((int *)0xF8000+r2) = r1;
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数
仮引数 r1 と r2 が同一の仮引数である場合も含まれます。

(a-4) レジスタ渡しの仮引数 (near ポインタ) ±レジスタ渡しの仮引数 (オフセット) が指すアドレス

例

```
*(r2+r3) = r1;
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)
r3 : レジスタ渡しの仮引数
仮引数 r1 と r2、または、仮引数 r1 と r3 が、同一の仮引数である場合も含まれます。

(b) レジスタ渡しの仮引数との比較処理で、比較対象が次の (b-1) ~ (b-4) のいずれかである。

(b-1) レジスタ渡しの仮引数

例

```
if (r1 == r2) {
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数

(b-2) レジスタ渡しの仮引数 (near ポインタ) が指すアドレス

例

```
if (r1 == *r2) {
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)

(b-3) レジスタ渡しの仮引数 (near ポインタ) 土定数オフセット (8bit 以内) が指すアドレス

例

```
if (r1 == *(r2+1)) {
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)

(b-4) レジスタ渡しの仮引数 (near ポインタ) 土レジスタ渡しの仮引数 (オフセット) が指すアドレス

例

```
if (r1 == *(r2+r3)) {
```

r1 : レジスタ渡しの仮引数
r2 : レジスタ渡しの仮引数 (near ポインタ)
r3 : レジスタ渡しの仮引数

(2) (1)に該当している仮引数とは別の仮引数において、レジスタ X、C、E、BC、DE のいずれかでデータを渡し、それを関数内で参照している。

例

```
i = r0;

i : 変数
r0 : レジスタ (X/C/E/BC/DE) 渡しの仮引数
```

1.4 発生例

以下に、発生例を記します。

【C ソース】

```
1 typedef struct st {
2     int s1;
3     struct st* s2;
4 } ST;
5
6 void temp(void);
7
8 int func(ST* s, int r1, int r2) {
9     s->s2 = s;           // 発生条件(1)(a-1)
10    temp();              // 最初の関数呼び出し
11    return r1 + r2;     // 発生条件(2)
12 }
```

- 8 行目：仮引数 s はレジスタ AX、仮引数 r1 はレジスタ BC、仮引数 r2 はレジスタ DE に割り当てられています。
- 9 行目：10 行目の最初の関数呼び出しまでに、レジスタ渡しの仮引数 s をレジスタ渡しの仮引数 s のメンバ s2 にストアしているため、発生条件 (1) (a) に合致します。
- 11 行目：仮引数 s とは別のレジスタ渡しの仮引数 r1 と r2 を参照しているため発生条件 (2) に合致しません。

【出力アセンブラコード】

```

1  _func:
2      .STACK _func = 8
3      push bc
4      push ax
5      movw hl, ax
6      movw ax, [sp+0x00]      ;仮引数 s を参照
7      xchw ax, hl
8      movw [hl+0x02], ax
9      call $_!_temp
10     movw ax, [sp+0x02]      ;仮引数 r1 を参照
11     movw bc, ax
12     movw ax, [sp+0x00]      ;誤って仮引数 s を参照
13     addw ax, bc
14     addw sp, #0x04
15     ret

```

- ・スタック[sp+0x00] には仮引数 s、スタック[sp+0x02] には仮引数 r1 が格納されています。仮引数 r2 はスタックに格納されていません。
- ・12 行目：本来仮引数 r2 を参照するところを、誤って仮引数 s を参照しています。

1.5 回避策

以下のいずれかにより回避可能です。

- (1) 最適化オプション-**Nothing** を指定する。
- (2) 関数にダミーの仮引数^(注) を先頭から 6 byte 分追加してレジスタに割り当て、元の仮引数がレジスタに割り当てられないようにする。

注：構造体または共用体の場合、スタック渡しになる場合があり仮引数をレジスタに割り当てられない場合があります。また、配列変数の場合は、配列変数全体のサイズではなくポインタのサイズとなります。

【変更前】

```

void func(int *r1, int r2, int r0)
{
    *r1 = r2;
    ...
}

```

【変更後】

```

void func(long dummy1, int dummy2, int *r1, int r2, int r0)
{
    *r1 = r2;
    ...
}

```

(3) 関数の先頭にダミーの `inline_asm` 指定関数の呼び出しを追加する。

なお、`inline_asm` 指定関数の内容は空でも問題ありません。

【変更前】

```
void func(int *r1, int r2, int r0)
{
    *r1 = r2;
    ...
}
```

【変更後】

```
#pragma inline_asm dummy_func
void dummy_func(void){}

void func(int *r1, int r2, int r0)
{
    dummy_func();
    *r1 = r2;
    ...
}
```

1.6 恒久対策

CC-RL V1.06.00 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	2017.12.01	-	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

<https://www.renesas.com/contact/>

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。