

**【注意事項】**

R20TS0193JJ0100

Rev.1.00

2017.07.16 号

**RX ファミリ用 C/C++コンパイラパッケージ****概要**

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

1. -fint\_register オプション使用に関する注意事項 (No.44)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

1. -fint\_register オプション使用に関する注意事項 (No.44)

1.1 該当製品

CC-RX V2.00.00～V2.06.00

1.2 内容

-fint\_register オプション<sup>(注)</sup>で指定したレジスタ (R10、R11、R12、R13) の値を高速割り込み関数以外で書き換える場合があります。

注: -fint\_register オプションは、高速割り込み関数でのみ使用する汎用レジスタを選択するオプションです。

### 1.3 発生条件

以下の条件をすべて満たした時に発生する場合があります。

- (1) `-fint_register={1,2,3,4}`を指定している。
- (2) 高速割り込み関数以外の関数<sup>(注)</sup>がある。

注：コンパイラ V2.00.00 を使用している場合は、`#pragma entry` を指定した関数である場合に該当します。

- (3) 以下のいずれかの条件を満たしている。

(3-1) `-optimize` を指定していないか、`-optimize={2,max}`を指定している。

(3-2) (2)に該当する関数に、次のいずれかの条件を満たすソース記述がある。

(3-2-1) 自動変数またはスタックを使用して値を渡す仮引数を使用した `memcpy()`、`memset()`および `memmove()`のいずれかがある。

(3-2-2) 構造体または共用体型の自動変数があり、その自動変数を用いた初期化または代入がある。

(3-2-3) 関数が使用するスタックサイズ<sup>(注)</sup>が 65535 を超えている。

注：関数が参照するスタック領域のサイズです。以下の2つの合計値で確認できます。

- コンパイラが出力するアセンブリソースの`STACK`制御命令の数値
- スタックを使用して値を渡す仮引数を使用するスタックサイズ

(3-2-4) `-pack` または`#pragma pack` を指定した構造体型において自動変数またはスタックを使用して値を渡す仮引数と戻り値に対してメンバ参照を行なっている。

(3-2-5) 自動変数またはスタックを使用して値を渡す仮引数を、仮引数の型のアライメントより大きいサイズの別の型のポインタを経由して間接的に参照している。

例：

```
char a[4];           // char[4] のアライメントは 1
*((int*)a) = 0;     // int のアライメントは 4
```

### 1.4 回避策

以下のいずれかにより回避可能です。

- (1) `-fint_register={1,2,3,4}`を指定しない、または`-fint_register=0` を指定する。
- (2) `-optimize={0,1}`を指定し、かつ、(3-2-1)~(3-2-5)のいずれにも該当しないソースプログラムに変更する。

### 1.5 恒久対策

CC-RX V2.07.00 で改修します。(7月20日公開予定)

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.07.16	-	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。