

【注意事項】

R20TS0461JJ0100

Rev.1.00

2019.07.01 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ご連絡します。

1.ループ内の比較式に関する注意事項 (No.52)

注：注意事項の後ろの番号は、注意事項の識別番号です。

1 ループ内の比較式に関する注意事項 (No.52)

1.1 該当製品

CC-RX V2.00.00~V3.01.00

1.2 内容および発生条件

以下の発生条件をすべて満たす場合に、(4)の比較式が不正な値を用いて判定を行う場合があります。コンパイラが最適化を行った結果、内部的にこれらの条件を満たす場合も含まれます。

➤ 発生条件

- (1) -optimize=2 または -optimize=max オプションを指定している。または、-optimize オプションを省略している。
- (2) プログラム中にループがある。
- (3) (2)のループ内に、次のすべての条件を満たすループ帰納変数^(注1)が存在する。
 - (3-1) 型が signed char、unsigned char、signed short、
または unsigned short のいずれかである。^(注2)
 - (3-2) volatile 修飾されていない。
- (4) (2)のループ内に、次のいずれかのパターンの比較式が存在する^(注3)。
ここで、i は(3)を満たすループ帰納変数である。
また、X,Y は定数である (コンパイル時に定数であると分かる式も含む)。
 - (4-1) 次のいずれかの形である。
 - ・ $(i > X) \ \&\& \ (i < Y)$
 - ・ $(i > X) \ \&\& \ (i \leq Y)$
 - ・ $(i \leq X) \ \|\ (i > Y)$
 - ・ $(i \leq X) \ \|\ (i \geq Y)$

上記の X は 0 以上かつ、ループ開始時点の i の値未満の定数値である。
また Y は X より大きな定数値である。

(4-2) 次のいずれかの形である。

- ・ $(i \geq X) \ \&\& \ (i < Y)$
- ・ $(i \geq X) \ \&\& \ (i \leq Y)$
- ・ $(i < X) \ || \ (i > Y)$
- ・ $(i < X) \ || \ (i \geq Y)$

上記の X は 1 以上かつ、ループ開始時点の i の値以下の定数値である。

また Y は X より大きな定数値である。

(4-3) 次のいずれかの形である。

- ・ $(i == X) \ || \ (i == Y)$
- ・ $(i != X) \ \&\& \ (i != Y)$

上記の X は 1 以上かつ、ループ開始時点の i の値以下の定数値である。

また Y は $Y == X+1$ の関係を満たす定数値である。

(5) (2)のループの終了条件式が、(3)を満たすループ帰納変数と、ループ内で値が変化しない式との大小比較(<, <=, > または >=) である。

注 1 : ループ帰納変数とは、ループの繰り返しごとに一定値ずつ増加または減少する変数です。

注 2 : -int_to_short オプションを指定している場合、signed int/unsigned int は signed short/unsigned short とみなします。

注 3 : (4-1)~(4-3)で"&&", "||"と表現している箇所は、ビット単位の演算子("&", "|")や、if 文や三項演算子の入れ子または併記によって論理積や論理和を取る場合も含まれます。また、帰納変数を比較の右辺側に記述した場合も該当します。例えば、" $(i < X) \ \&\& \ (i < Y)$ "と" $(X > i) \ \&\& \ (Y > i)$ "は同じ比較式とみなします。

1.3 発生例 1

1:	long a;
2:	void func1(void){
3:	signed char i; // 発生条件(3-1) (3-2)
4:	for (i = 1; i <= 100; i++) { // 発生条件(2)(5)
5:	if ((i >= 1) && (i <= 10)) { // 発生条件(4-2)
6:	a++;
7:	}
8:	}
9:	}

全体:

関数 func1 には for ループが存在するため、発生条件(2)を満たします。
また、変数 i はこのループの帰納変数です。

3 行目:

変数 i は、型が signed char であるので発生条件(3-1)を満たします。
また、volatile で修飾されていないので発生条件(3-2)を満たします。

4 行目:

for ループの終了条件式は、変数 i と整数 100 の大小比較(<=)であるので発生条件(5)を満たします。

5 行目:

if 文の条件式は、不等号の向きを揃えると $(1 \leq i) \ \&\& \ (i \leq 10)$ となり、発生条件(4-2)を満たします。

この例の場合、5 行目の比較式が不正になります。

1.4 発生例 2

1:	long a;
2:	void func2(void){
3:	signed char i; // 発生条件(3-1) (3-2)
4:	for (i = 1; i <= 100; i++) { // 発生条件(2) (5)
5:	if ((i != 1) && (i != 2)) { // 発生条件(4-3)
6:	a++;
7:	}
8:	}
9:	}

全体:

関数 func2 には for ループが存在するため、発生条件(2)を満たします。
また、変数 i はこのループの帰納変数です。

3 行目:

変数 i は、型が signed char であるので発生条件(3-1)を満たします。
また、volatile で修飾されていないので発生条件(3-2)を満たします。

4 行目:

for ループの終了条件式は、変数 i と整定数 100 の大小比較(<=)であるので発生条件(5)を満たします。

5 行目:

if 文の条件式は、発生条件(4-3)を満たします。

この例の場合、5 行目の比較式が不正になります。

1.5 回避策

次のいずれかの方法で回避できます。

- (1) -optimize=0 または -optimize=1 オプションを指定する。
- (2) 発生条件(3)に該当するループ帰納変数の型を発生条件(3-1)で示す型以外にする(long 型など)。
- (3) 発生条件(3)に該当するループ帰納変数を volatile で修飾する。
- (4) 発生条件(5)の終了条件式を等価演算子("==", "!=")で記述する。

1.6 恒久対策

次版で改修する予定です。次版のリリース時期は未定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jul.01.19	-	新規発行

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ものです。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。

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。