RENESAS ツールニュース

RENESAS TOOL NEWS 2014年01月28日：140128/tn1

RL78ファミリおよび78K0R用Cコンパイラ CA78K0R、78K0R用Cコンパイラ CC78K0R、および78K0R用アセンブラ RA78K0Rご使用上のお願い

RL78ファミリおよび78K0R用Cコンパイラ CA78K0R、78K0R用Cコンパイラ CC78K0R、および78K0R用アセンブラ RA78K0Rの使用上の注意事項を連絡します。

- 浮動小数点定数の多重キャスト処理で誤ったコードを出力する注意事項
- 配列へのポインタのnear/far修飾処理で誤ったコードを出力する注意事項
- 乗算・除算・剰余算と間接参照式で誤ったコードを出力する注意事項
- assert関数が正常動作しない注意事項
- 条件式で1ビット幅のビットフィールドを使用したときにエラーになる注意事項
- strtol関数, strtoul関数で文字列数値変換が誤った値になる注意事項
- RL78-S1コアにおいてアセンブラで解決するシンボル参照を行うと誤ったアドレス参照になる注意事項
- CALL疑似命令で誤ったコードを出力する注意事項
- BR疑似命令, CALL疑似命令でエラーになる注意事項

---

1. 浮動小数点定数の多重キャスト処理で誤ったコードを出力する注意事項

1.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
CC78K0R V1.00 ～ V2.13 (統合開発環境 PM+)

1.2 内容

浮動小数点定数に多重キャストを用いると演算結果が誤った結果になります。

1.3 発生条件

以下のすべての条件を満たす場合に発生します。
(1) 浮動小数点定数、または浮動小数点型にキャストした定数を、浮動小数点型にキャストしている。
(2) (1)を整数型にキャストしている。

(3) (2)を下記以外の演算で使用している。
- 単純代入演算: =
- 論理演算: &&または||
- 条件演算: ? :
- 単項演算: !

1.4 発生例

```
------------------------------------------------------------------
[*.C]
#define A ((long)((double)6031.0))  //(1)および(2)
void func(void)
{
  long x;
  x=A<<1;                    //(3)
}
------------------------------------------------------------------
```

x = 12062 (= 6031 * 2) となりません。

1.5 回避策

浮動小数点定数、または浮動小数点型にキャストした定数を、浮動小数点型にキャストしないでください。

```
------------------------------------------------------------------
#define A ((long)6031.0)
void func(void)
{
  long x;
  x=A<<1;
}
------------------------------------------------------------------
```

x = 12062 (= 6031 * 2) となります。

2. 配列へのポインタのnear/far修飾処理で誤ったコードを出力する注意事項

2.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
CC78K0R V1.00 ～ V2.13 (統合開発環境 PM+)

2.2 内容

式中の型名を記述する箇所で、配列ポインタに対して near または far修飾が効かずに誤ったコードになる場合があります。また、その場合に警告メッセージを正しく出力しない場合があります。

2.3 発生条件

以下の発生条件1から3のいずれかの条件を満たす場合に発生します。

発生条件1:
以下の3つの条件を全て満たしている場合に誤ったコードを出力します。
また、その場合に警告メッセージを正しく出力しない場合があります。
(1) コンパイルオプションの -ms (スモール・モデル)、または
-mm (ミディアム・モデル)を使用している。
または、メモリ・モデルの種類を変更していない。 (注)
(2) キャスト演算子中の型名にfar配列へのポインタを使用している。
(3) farアドレスの入ったオペランドを(2)でキャストしている。

注: メモリ・モデルの種類を変更しない場合、デフォルトとして
ミディアム・モデルが選択されるため問題が起こります。

発生条件1の発生例:

```
------------------------------------------------------------------
[*.C]
typedef __far int *P;
typedef __far int (*PA)[10];

P p2;
PA pa1, pa2;
int i1;
void func(void)
{
 . . . . . .
   pa2 = (__far int (*)[10])p2;       //(2)および(3) 、下記(a)
   i1 = (*(__far int (*)[10])pa1)[0];  //(2)および(3) 、下記(b)
 . . . . . .
}
------------------------------------------------------------------
```

far 修飾が効かないため、(int (*)[10]) のキャストになります。
スモール・モデル、およびミディアム・モデルの場合、(int (*)[10]) は
(__near int (*)[10])と解釈されるため、誤ったコードを出力します。

(a) 右辺の__farへのキャストが行われないために左辺とポインタのサイズが
異なり、以下の警告を表示し、誤ったコードを出力します。
警告:

```
   ------------------------------------------------------------------
      CC78K0R warning W0416: Illegal type and size (far/near) pointer combination
   ------------------------------------------------------------------
```

コード出力例:

```
   ------------------------------------------------------------------
    ; line    X :     pa2 = (__far int (*)[10])p2;
    ;       誤ったコード                    正しいコード
           movw ax,!_p2                    ; movw ax,!_p2
```

```
        mov  _@SEGAX,#0FH ; 15      ;
        cmpw ax,#00H          ; 0      ;
        sknz                          ;
        clrb _@SEGAX                  ;
  ?L0004:                             ;
        movw !_pa2,ax                 ; movw !_pa2,ax
        mov  a,_@SEGAX                ; mov  a,!_p2+2
        mov  !_pa2+2,a                ; mov  !_pa2+2,a
  ------------------------------------------------------------------
```

(b) 右辺に間接参照が入り左辺と型が一致するため警告を表示せずに、誤ったコードを出力します。

コード出力例:

```
  ------------------------------------------------------------------
   ; line    xx :    i1 = (*(__far int (*)[10])pa1)[0];
   ;       誤ったコード              正しいコード
        movw de,!_pa1           ; movw de,!_pa1
                                ; mov  a,!_pa1+2
                                ; mov  ES,a
        movw ax,[de]            ; movw ax,ES:[de]
        movw !_i1,ax            ; movw !_i1,ax
  ------------------------------------------------------------------
```

発生条件2:

以下の2つの条件を全て満たしている場合に誤ったコードを出力します。

(1) コンパイルオプションの -ms (スモール・モデル)、または
    -mm (ミディアム・モデル)を使用している。
    または、メモリ・モデルの種類を変更していない。 (注)

(2) sizeof演算子中の型名に far配列へのポインタを使用している。

注: メモリ・モデルの種類を変更しない場合、デフォルトとして
    ミディアム・モデルが選択されるため問題が起こります。

発生条件2の発生例:

```
------------------------------------------------------------------
[*.C]
int i3;

void func(void)
{
. . . . . .
  i3 = sizeof(__far int (*)[10]); // (2)
. . . . . .
}
```

---
far 修飾が効かないため、(int (*)[10]) のサイズを求めてしまいます。
スモール・モデル、およびミディアム・モデルの場合、 (int (*)[10]) は
(__near int (*)[10]) と解釈されるため、サイズを誤って出力します。

発生条件2のコード出力例:

```
 ; line    22 :     i3 = sizeof(__far int (*)[10]);
 ;      誤ったコード                正しいコード
        movw ax,#02H    ; 2       ; movw ax,#04H    ; 4
        movw !_i3,ax                : movw !_i3,ax
```

発生条件3:
 以下の2つの条件を全て満たしている場合に誤ったコードを出力します。
 (1) コンパイルオプション-ml (ラージ・モデル)を使用している。
 (2) sizeof演算子中の型名に near配列へのポインタを使用している。

発生条件3の発生例:

```
[*.C]
int i4;

void func(void)
{
. . . . . .
   i4 = sizeof(__near int (*)[10]); // (2)
. . . . . .
}
```
---
near 修飾が効かないため、(int (*)[10]) のサイズを求めてしまいます。
ラージ・モデルの場合、 (int (*)[10]) は (__far int (*)[10]) と解釈
されるため、サイズを誤って出力します。

発生条件3のコード出力例:

```
 ; line    23 :     i4 = sizeof(__near int (*)[10]);
 ;     誤ったコード                正しいコード
       movw ax,#04H   ; 4         ; movw ax,#02H   ; 2
       mov  ES,#highw (_i4)       ; mov  ES,#highw (_i4)
       movw ES:!_i4,ax            ; movw ES:!_i4,ax
```

2.4 回避策

near または far修飾を行う配列ポインタの型定義は、typedef型定義を使って

型名を記述してください。

発生条件1の回避例:

```
------------------------------------------------------------------
typedef __far int (*PA)[10];

void func(void)
{
 . . . . . . . . . . . . . . . . . . .
   pa2 = (PA)p2;       //(a)
   i1 = (*(PA)pa1)[0];  //(b)
 . . . . . . . . . . . . . . . . . . .
}
------------------------------------------------------------------
```

発生条件2の回避例:

```
------------------------------------------------------------------
typedef __far int (*PA)[10];

void func(void)
{
 . . . . . . . . . . . . . . . . . . .
   i3 = sizeof(PA);
 . . . . . . . . . . . . . . . . . . .
}
------------------------------------------------------------------
```

発生条件3の回避例:

```
------------------------------------------------------------------
typedef __near int (*PA)[10];

void func(void)
{
 . . . . . . . . . . . . . . . . . . .
   i4 = sizeof(PA);
 . . . . . . . . . . . . . . . . . . .
}
------------------------------------------------------------------
```

3. 乗算・除算・剰余算と間接参照式で誤った式を出力する注意事項

3.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
CC78K0R V1.00 ～ V2.13 (統合開発環境 PM+)

3.2 内容

char、signed char、またはunsigned char型の乗算、除算、または剰余算の
オペランドが、添え字が定数でない配列要素、またはポインタを使った
間接参照式を含んでいると、誤ったコードとなる場合があります。

3.3 発生条件

以下の条件をすべて満たす場合に発生します。

(1) コンパイルオプションの最適化で -qc が有効である。
　-qc は以下のいずれかの設定で有効になります。
　- -qcオプションを使用している
　- -qx(数値)オプションを使用している (数値は1～3)
　- -qオプションを最適化種別を省略して使用している
　- -qオプションを省略している

(2) char、 signed char、またはunsigned char型の二項演算がある。

(3) (2)の左オペランドは char、signed char、またはunsigned char型の
　演算結果である。

(4) (2)の右オペランドは char、signed char、またはunsigned char型の乗算、
　除算、剰余算の演算結果である。

(5) (4)の左オペランドまたは右オペランドは、添え字が定数でない配列
　要素式、またはポインタを使用した間接参照式を含んでいる。

3.4 発生例

```
------------------------------------------------------------------
[*.C]
unsigned char x1, uca1[5], uc1, uc2;
unsigned int ui1;
void func(void)
{
  x1 = (uc1 + 1) & (uca1[ui1] * uc2);  //(2)(3)(4)および(5)
}
------------------------------------------------------------------
```

コード出力例:

```
------------------------------------------------------------------
; line    5 :    x1 = (uc1 + 1) & (uca1[ui1] * uc2);
  mov    a,!_uc1
  inc      a              ; Aレジスタに(uc1 + 1)の演算結果が残る
  movw   bc,!_ui1
  mov    a,_uca1[bc]     ; Aレジスタを上書きしている
  mov    x,!_uc2
  mulu   x
  mov    a,b
  and    a,x
  mov    !_x1,a
------------------------------------------------------------------
```

3.5 回避策

テンポラリ変数を設け、1バイトデータの演算結果をテンポラリ変数に
代入しながら演算してください。

回避例:

```
----------------------------------------------------------------
unsigned char x1, uca1[5], uc1, uc2;
unsigned int ui1;
void func(void)
{
   unsigned char temp; // テンポラリ変数を設ける
   temp = uc1 + 1;     // 1バイトデータの演算結果をテンポラリ変数に代入
   x1 = temp & (uca1[ui1] * uc2); // テンポラリ変数と間接参照式を演算
}
----------------------------------------------------------------
```

4. assert関数が正常動作しない注意事項

4.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
CC78K0R V1.00 ～ V2.13 (統合開発環境 PM+)

4.2 内容

メモリ・モデルの種類で、スモール・モデル、またはミディアム・モデルを
指定した場合、または、メモリ・モデルの種類を変更しない場合 (注)、
assert関数が正常に動作しません。

注: メモリ・モデルの種類を変更しない場合、デフォルトとして
　ミディアム・モデルが指定されるため問題が起こります。

4.3 回避策

assert関数の代わりに assert_f関数を使用してください。

5. 条件式で1ビット幅のビットフィールドを使用したときにエラーになる注意事項

5.1 該当製品およびバージョン

CA78K0R V1.50 ～ V1.60 (統合開発環境CubeSuite+) (注)
　注: CubeSuite+の該当バージョンは V1.03.00以降です。

5.2 内容

条件式で1ビット幅のビットフィールドを使用したとき、エラー
「C0101: Internal error」が発生する場合があります。

5.3 発生条件

以下のすべての条件を満たす場合に発生します。

(1) 条件式で、以下を使用している。
- 比較(等価・関係)演算子: ==, !=, <, >, <=, および>=

(2) (1)の左オペランドが、定数 0 または 1 である。

(3) (1)の右オペランドが、アドレス値 0FFE20H-0FFFFFH の偶数定数番地の
1ビット幅のビットフィールドである。

5.4 発生例

```
------------------------------------------------------------------
struct _st {
   unsigned int b0:1;
   unsigned int b1:1;
   unsigned int b2:1;
};
#define bitsfr(n)  (((struct _st *)0xffd0)->b ## n) // (3)アドレス値
                                                    // 0ffd0Hのビット
                                                    // フィールド
int i;

void func(void)
{
   if (0 == bitsfr(1)) i++; // (1)および(2)
}
------------------------------------------------------------------
```

5.5 回避策

条件式で、定数を右オペランドに記述する。

回避例:

```
------------------------------------------------------------------
struct _st {
   unsigned int b0:1;
   unsigned int b1:1;
   unsigned int b2:1;
};
#define bitsfr(n)  (((struct _st *)0xffd0)->b ## n)
int i;

void func(void)
{
   if (bitsfr(1) == 0) i++;  //定数を右オペランドに記述
}
------------------------------------------------------------------
```

6. strtol関数, strtoul関数で文字列数値変換が誤った値になる注意事項

6.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
CC78K0R V1.00 ～ V2.13 (統合開発環境 PM+)

6.2 内容

strtol関数およびstrtoul関数を使用して文字列数値変換を行った場合、正しく
数値変換せずに、オーバフロー値を返す場合があります。

6.3 発生条件

以下のすべての条件を満たす場合に発生します。
(1) strtol関数およびstrtoul関数を使用して文字列を数値に変換している。
(2) (1)の関数処理で、文字列の先頭から順に数値変換を行い、
変換の途中に 0x10000 ごとの桁上がりが発生する。
0x10000 ごとの桁上がりが発生する場合とは、数値変換中に下記の条件を
満たす場合です。
((N * base) / 0x10000) != ((N * base + c) / 0x10000)
N: 先頭から n文字目までを変換した値
base: 基数
c: n+1 文字目の値

発生例:
strtol("65537", &err, 10)を実行した場合、
4文字目までを変換した値 6553、基数 10、5文字目の値 7となり、
左辺 = ((6553 * 10) / 0x10000) = 0
右辺 = ((6553 * 10 + 7) / 0x10000) = 1
左辺 != 右辺 が成立して、オーバフロー値になります。

6.4 回避策

strtol関数で基数が10の場合には、atol関数に置き換えてください。
strtoul関数で基数が10の場合、かつ、変換後の結果がsigned longで表現できる
値の場合には、atol関数に置き換えてください。
上記で対応できない場合は、回避策はありません。

7. RL78-S1コアにおいてアセンブラで解決するシンボル参照を行うと誤ったアドレス参照になる注意事項

7.1 該当製品およびバージョン

CA78K0R V1.50 ～ V1.60 (統合開発環境CubeSuite+) (注)
注: CubeSuite+の該当バージョンは V1.03.00以降です。

7.2 内容

RL78-S1コアにおいて絶対番地指定のセグメントに所属するシンボルを
以下のいずれかで参照すると誤ったコードになります。
- オペランドに、#word、ES:!addr16、ES:word[B]、ES:!word[C]、または
ES:word[BC] を持つ命令
- DWまたはDG疑似命令

7.3 発生条件

下記の条件を全て満たす場合に発生します。

(1) RL78-S1コアのマイコンを使用している。

(2) 絶対番地指定のセグメントに所属するシンボルを参照している。

(3) (2)の参照形式が、以下のいずれかである。

(a) ES:!addr16

(b) ES:word[B]

(c) ES:word[C]

(d) ES:word[BC]

(e) #word

(f) DW word

(g) DG lword

7.4 発生例

[Sample.ASM] (アセンブラ・ソース・ファイル)

```
------------------------------------------------------------------------
C1    CSEG    AT    30H     ; 発生条件(2)
TAB1:
    DB    1
    DB    2

    CSEG
main:
    MOV       A,ES:!TAB1          ; 発生条件(3)の(a)
    MOV       A,ES:TAB1[B]        ; 発生条件(3)の(b)
    MOV       A,ES:TAB1[C]        ; 発生条件(3)の(c)
    MOV       A,ES:TAB1[BC]       ; 発生条件(3)の(d)
    MOVW      AX,#TAB1            ; 発生条件(3)の(e)
    MOVW      AX,ES:!TAB1         ; 発生条件(3)の(a)
    MOVW      AX,ES:TAB1[B]       ; 発生条件(3)の(b)
    MOVW      AX,ES:TAB1[C]       ; 発生条件(3)の(c)
    MOVW      AX,ES:TAB1[BC]      ; 発生条件(3)の(d)
    RET
;
    DW    TAB1              ; 発生条件(3)の(f)
    DG    TAB1              ; 発生条件(3)の(g)
    END
------------------------------------------------------------------------
```

[Sample.P] (アセンブル・リスト・ファイル)　　発生条件

```
------------------------------------------------------------------------------
-----------------------
ALNO  STNO  ADRS    OBJECT      M I       SOURCE STATEMENT
1       1       -----    C1            CSEG    AT        30H         (2)
```

```
2       2       00030   TAB1:
3       3       00030   01          DB      1
4       4       00031   02          DB      2
5       5
6       6       -----   CSEG
7       7       000CE   main:
8       8       000CE   118F3080    MOV     A,ES:!TAB1          (3)の(a)
9       9       000D2   11093080    MOV     A,ES:TAB1[B]        (3)の(b)
10      10      000D6   11293080    MOV     A,ES:TAB1[C]        (3)の(c)
11      11      000DA   11493080    MOV     A,ES:TAB1[BC]       (3)の(d)
12      12      000DE   303080      MOVW    AX,#TAB1            (3)の(e)
13      13      000E1   11AF3080    MOVW    AX,ES:!TAB1         (3)の(a)
14      14      000E5   11593080    MOVW    AX,ES:TAB1[B]       (3)の(b)
15      15      000E9   11793080    MOVW    AX,ES:TAB1[C]       (3)の(c)
16      16      000ED   11793080    MOVW    AX,ES:TAB1[BC]      (3)の(d)
17      17      000F1   D7          RET
18      18      ;
19      19      000F2   3080        DW      TAB1                (3)の(f)
20      20      000F4   30800F00    DG      TAB1                (3)の(g)
21      21      END
-------------------------------------------------------------------------------
-----------------------
```

7.5 回避策

セグメントの絶対番地指定を解除し、代わりにリンク・ディレクティブ・
ファイルで配置指定を行ってください。

[Sample.ASM] (アセンブラ・ソース・ファイル)

```
-----------------------------------------------------------------------
C1    CSEG    ;AT    30H     ; セグメントの絶対番地指定を解除する
TAB1:
    DB    1
    DB    2

    CSEG
main:
    MOV     A,ES:!TAB1          ; 発生条件(3)の(a)
    MOV     A,ES:TAB1[B]        ; 発生条件(3)の(b)
    MOV     A,ES:TAB1[C]        ; 発生条件(3)の(c)
    MOV     A,ES:TAB1[BC]       ; 発生条件(3)の(d)
    MOVW    AX,#TAB1            ; 発生条件(3)の(e)
    MOVW    AX,ES:!TAB1         ; 発生条件(3)の(a)
    MOVW    AX,ES:TAB1[B]       ; 発生条件(3)の(b)
    MOVW    AX,ES:TAB1[C]       ; 発生条件(3)の(c)
    MOVW    AX,ES:TAB1[BC]      ; 発生条件(3)の(d)
    RET
;
```

```
    DW    TAB1            ; 発生条件(3)の(f)
    DG    TAB1            ; 発生条件(3)の(g)
    END
  ------------------------------------------------------------------------
```

[Sample.DR] (リンク・ディレクティブ・ファイル)

```
  ------------------------------------------------------------------------
  MERGE C1 : AT (30H)       ; リンク・ディレクティブ・ファイルで
                            ; 配置指定を行う
  ------------------------------------------------------------------------
```

[Sample.P] (アセンブル・リスト・ファイル)

```
                                                                    発生条件
------------------------------------------------------------------------------
----------------------
ALNO  STNO  ADRS   OBJECT      M I     SOURCE STATEMENT
1     1     -----  C1          CSEG    AT      30H       (2)
2     2     00030  TAB1:
3     3     00030  01          DB      1
4     4     00031  02          DB      2
5     5
6     6     -----  CSEG
7     7     000CE  main:
8     8     000CE  R118F3000   MOV     A,ES:!TAB1        (3)の(a)
9     9     000D2  R11093000   MOV     A,ES:TAB1[B]      (3)の(b)
10    10    000D6  R11293000   MOV     A,ES:TAB1[C]      (3)の(c)
11    11    000DA  R11493000   MOV     A,ES:TAB1[BC]     (3)の(d)
12    12    000DE  R303000     MOVW    AX,#TAB1          (3)の(e)
13    13    000E1  R11AF3000   MOVW    AX,ES:!TAB1       (3)の(a)
14    14    000E5  R11593000   MOVW    AX,ES:TAB1[B]     (3)の(b)
15    15    000E9  R11693000   MOVW    AX,ES:TAB1[C]     (3)の(c)
16    16    000ED  R11793000   MOVW    AX,ES:TAB1[BC]    (3)の(d)
17    17    000F1  D7          RET
18    18    ;
19    19    000F2  R3000       DW      TAB1              (3)の(f)
20    20    000F4  R30000000   DG      TAB1              (3)の(g)
21    21    END
------------------------------------------------------------------------------
----------------------
```

8. CALL疑似命令で誤ったコードを出力する注意事項

8.1 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
RA78K0R V1.31 ～ V1.80 (統合開発環境 PM+)

8.2 内容

CALL疑似命令を使用した時、分岐先までの相対距離が -8000H未満、

または +7FFFHを超える場合に、誤ったコード CALL $!addr20 (注) を
生成します。そのため、正しく分岐しません。
Cソースの場合、ASM文でCALL疑似命令を使用したときも該当します。

注: CALL $!addr20は3バイト命令です。

8.3 発生条件

以下のすべての条件を満たす場合に発生します。
(1) アセンブラでCALL疑似命令を記述した時、または、CソースのASM文で
CALL疑似命令を記述している。
(2) 対象となるCALL疑似命令と分岐先シンボルが、同一ファイル内で、同一の
アブソリュートまたは再配置属性BASEのセグメントに所属している。
(3) 分岐先シンボルの同一セグメント内で、下位のアドレスにCALL疑似命令、
またはBR疑似命令が存在し、そのCALL疑似命令、またはBR疑似命令が以下の
いずれかを満たす。
- 下位のアドレスに存在するCALL疑似命令、またはBR疑似命令のサイズが
最小となる場合の分岐先の配置アドレスは、 0H ～ 0FFFFHの範囲内。
- 下位のアドレスに存在するCALL疑似命令、またはBR疑似命令のサイズが
最大となる場合の分岐先の配置アドレスは、 0H ～ 0FFFFHの範囲外。
(4) CALL疑似命令から分岐先までの相対距離が、-8000H未満、または +7FFFHを
超えている。

8.4 回避策

該当するCALL疑似命令を、以下のイミーディエト・アドレッシングを使用した
CALL命令で記述してください。

回避例:

```
-----------------------------------------
. . . . . .
   CALL   !!addr20
. . . . . .
-----------------------------------------
```

addr20 には、分岐先のラベルを記述してください。

9. BR疑似命令, CALL疑似命令でエラーになる注意事項

9.1. 該当製品およびバージョン

CA78K0R V1.20 ～ V1.60 (統合開発環境 CubeSuite+)
CA78K0R V1.00 ～ V1.10 (統合開発環境 CubeSuite)
RA78K0R V1.31 ～ V1.80 (統合開発環境 PM+)

9.2. 内容

BR疑似命令、またはCALL疑似命令を使用したとき、エラー「E2410: Phase error」
が発生する場合があります。
Cソースの場合、ASM文でBR疑似命令、またはCALL疑似命令を使用したときも該当
します。

9.3 発生条件

以下のすべての条件を満たす場合に発生します。

(1) アセンブラでBR疑似命令、またはCALL疑似命令を記述している。または、CソースのASM文でBR疑似命令、またはCALL疑似命令を記述している。

(2) 1つのアブソリュートセグメント内に、分岐先となるラベルを前方参照するBR疑似命令、またはCALL疑似命令と分岐先があり、分岐先までの相対距離が80H 以上である。

(3) (2)と同じアブソリュートセグメント内に、(2)のBR疑似命令、またはCALL疑似命令と別のBR疑似命令、またはCALL疑似命令がある。この疑似命令の分岐先が(1)と別のアブソリュートセグメント内にあり、以下の(a)～(c)のいずれかに該当している。

(a) BR疑似命令、またはCALL疑似命令の分岐先が 10000H 番地付近である。
10000H 番地付近とは以下の位置を指します。
- BR/CALL !addr16 から BR/CALL !!addr20 に切り替わる位置
- BR/CALL !addr16 から BR/CALL $!addr20 に切り替わる位置

(b) BR疑似命令で分岐先までの相対距離が 80H 付近である。
80H 付近とは以下の位置を指します。
- BR $addr20 から BR $!addr20 に切り替わる位置
- BR $addr20 から BR !addr16 に切り替わる位置

(c) BR疑似命令、またはCALL疑似命令で分岐先までの相対距離が 8000H 付近である。
8000H 付近とは以下の位置を指します。
- BR/CALL $!addr20 から BR/CALL !!addr20 に切り替わる位置

(4) (2)と別のアブソリュートセグメント内に、BR疑似命令、またはCALL疑似命令があり、その命令の配置アドレスが、(3)の分岐先ラベルより小さいアドレスに存在する。

9.4 発生例

以下のコードの場合、L1 ラベルの行で E2410: Phase errorが発生します。

[Sample.ASM] (アセンブラ・ソース・ファイル)

```
------------------------------------------------------------------------
        ORG     8000H     ; アブソリュートセグメントA
        BR      L1        ; 発生条件(2)のBR疑似命令
        DS      (80H)
  L1:                     ; 発生条件(2)の分岐先(ラベル)
        CALL    L2        ; 発生条件(3)のCALL疑似命令

        ORG     10000H    ; アブソリュートセグメントB
        DS      (82H)
        BR      L3        ; 発生条件(4)のBR疑似命令
  L2:                     ; 発生条件(3)の(c)の分岐先
        NOP
  L3:
------------------------------------------------------------------------
```

9.5 回避策

セグメントの絶対番地指定を解除し、代わりにリンク・ディレクティブ・
ファイルで配置指定を行ってください。

回避例:

[Sample.ASM] (アセンブラ・ソース・ファイル)

```
------------------------------------------------------------------------
  A1    CSEG              ; リロケータブルセグメントA1
        ;ORG    8000H     ; アブソリュートセグメントA
                          ; セグメントの絶対番地指定を解除する
        BR      L1        ; 発生条件(2)のBR疑似命令
        DS      (80H)
  L1:                     ; 発生条件(2)の分岐先(ラベル)
        CALL    L2        ; 発生条件(3)のCALL疑似命令

  B1    CSEG              ; リロケータブルセグメントB1
        ;ORG    10000H    ; アブソリュートセグメントB
                          ; セグメントの絶対番地指定を解除する
        DS      (82H)
        BR      L3        ; 発生条件(4)のBR疑似命令
  L2:                     ; 発生条件(3)の(c)の分岐先
        NOP
  L3:
------------------------------------------------------------------------
```

[Sample.DR] (リンク・ディレクティブ・ファイル)

```
------------------------------------------------------------------------
MERGE   A1 : AT (8000H)  ; リロケータブルセグメントA1の配置指定
MERGE   B1 : AT (10000H) ; リロケータブルセグメントB1の配置指定
------------------------------------------------------------------------
```

10. 恒久対策

これら9件の注意事項は、CubeSuite+ CA78K0Rコンパイラ V1.70で改修します。
(2014年1月30日リリース予定)
なお、CubeSuite (注) CA78K0Rコンパイラ、PM+ CC78K0Rコンパイラ、および
PM+ RA78K0Rアセンブラの改修予定はありません。

注: CubeSuiteは保守製品です。

[免責事項]
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。