RENESAS ツールニュース

RENESAS TOOL NEWS 2015年06月01日：150601/tn3

RH850ファミリ用Cコンパイラパッケージのご使用上のお願い

RH850ファミリ用Cコンパイラパッケージ CC-RHの使用上の注意事項を連絡します。

1. アセンブリ命令のオペランドに即値を指定した場合の注意事項 (No.2)
2. ラベルの定義と参照が同一アセンブル単位にある場合の注意事項 (No.3)
3. アセンブリ命令のベース・レジスタ省略時の注意事項 (No.4)
4. 仕様外のアセンブリ記述時の注意事項(No.5)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

---

1. アセンブリ命令のオペランドに即値を指定した場合の注意事項 (No.2)

1.1 該当製品

CC-RH V1.02.00

1.2 内容

アセンブリソース上でアセンブリ命令のオペランドに即値を指定した場合に
意図しないコードを生成する場合があります。
なお、コンパイラが生成したアセンブリ命令は非該当です。

1.3 発生条件

以下(1)～(6)のいずれかを満たす場合に発生します。
(1) ld、あるいはst命令のdispに16ビット幅より大きい即値を指定し、
dispの下位16ビットが0である。

発生条件例:

```
-------------------------------------------
st.w  r10, 0xfff80000[r0]
-------------------------------------------
```

(2) prepare命令のimm1に16ビット幅より大きい即値を指定し、
imm1の下位16ビットが0である。

(3) mulu命令のimmに0x8000～0xFFFFの即値を指定している。

(4) prepare、あるいはdispose命令のimm1に4の倍数でなく、
かつ、128以上の即値を指定している。

発生条件例:

```
-------------------------------------------
prepare  r20, 0x83
-------------------------------------------
```

(5) mov、cmov、cmp、add、mulh、satadd命令のオペランドに
分離演算子(LOW または HIGH)を使用し、その分離演算子の項が
以下(a)～(b)のいずれかを満たす即値を指定している(注)。
(a) LOWの場合、第4ビットが1である。
(b) HIGHの場合、第12ビットが1である。
注: 即値の最下位ビットを第0ビットとします。

発生条件例:

```
-------------------------------------------
add  low(0x10), r12
-------------------------------------------
```

V1.01.00以前ではr12に-0x10が加算されますが、
V1.02.00ではr12に0x10が加算されます。

(6) ld、st、set1、clr1、not1、tst1、movea、addi、satsubi命令のオペランド
に分離演算子(LOWW、HIGHW、HIGHW1)を使用し、その分離演算子の項が以下
(c)～(e)のいずれかを満たす即値を指定している(注)。
(c) LOWWの場合、第15ビットが1である。
(d) HIGHWの場合、第31ビットが1である。
(e) HIGHW1の場合、第15ビットまたは第31ビットが1である。
注: 即値の最下位ビットを第0ビットとします。

発生条件例:

```
-------------------------------------------
movea  loww(0x8000), r0, r12
-------------------------------------------
```

V1.01.00以前ではr12に-0x8000が格納されますが、
V1.02.00ではr12に0x8000が格納されます。

1.4 回避策

以下のいずれかで回避可能です。

(1) 発生条件(1)～(3)の場合、アセンブラによる命令展開が正しく行われない
ため、意図しないコードとなります。
従って、ユーザー自身で命令展開したコードを記述してください。

回避策の適用例:

```
-----------------------------------------------
movhi  highw(0xfff80000), r0, r5
st.w   r10, loww(0xfff80000)[r5]
-----------------------------------------------
```

(2) 発生条件(4)の場合、imm1を4の倍数に丸めてコード生成する必要があるにも
関わらず、丸めずにコードを生成します。
従って、ユーザー自身で丸めた値でimm1を指定してください。

回避策の適用例:

```
-----------------------------------------------
prepare  r20, 0x80
-----------------------------------------------
```

(3) 発生条件(5)～(6)の場合、分離演算子を使用せずに期待値を直接指定して
ください。

回避策の適用例:

```
-----------------------------------------------
add     -0x10, r12
movea  -0x8000, r0, r12
-----------------------------------------------
```

1.5 恒久対策

次期バージョンで改修する予定です。

2. ラベルの定義と参照が同一アセンブル単位にある場合の注意事項 (No.3)

2.1 該当製品

CC-RH V1.02.00

2.2 内容

アセンブリソース上でラベル定義と、そのラベルに対する .extern疑似命令が
同一アセンブル単位に記述されている場合に、アセンブルエラーまたはシンボル
アドレスが不正になります。
アセンブル単位とはINCLUDE制御命令により挿入展開したアセンブリ命令も含み

ます。
なお、コンパイラが生成したアセンブリ命令は非該当です。

2.3 発生条件
以下(1)～(2)の全てを満たす場合に発生します。
(1) アセンブリソース上でラベルを定義している。
(2) そのラベルに対する.extern疑似命令を同一アセンブル単位に記述している。

発生条件例:
```
-----------------------------------------------
.extern _sym       -->(a)
  nop
_sym:              -->(b)
  jarl _sym, lp    -->(c)
-----------------------------------------------
```
(c)の分岐命令で_symの定義行(b)に分岐せず、(a)に分岐します。

2.4 回避策
.extern疑似命令を削除してください。

2.5 恒久対策
次期バージョンで改修する予定です。

3. アセンブリ命令のベースレジスタ省略時の注意事項 (No.4)

3.1 該当製品
CC-RH V1.02.00

3.2 内容
アセンブリソース上でメモリ参照命令のベースレジスタを省略した場合に、
不正なコードを生成します。
なお、コンパイラが生成したアセンブリ命令は非該当です。

3.3 発生条件
以下(1)～(2)の全てを満たす場合に発生します。
(1) set1、clr1、tst1、not1、あるいはst命令のベースレジスタ[reg1]を省略
    している。
(2) (1)の命令のdispに分離演算子を使用している。

発生条件例:
```
-----------------------------------------------
st.w  r10, loww(#_sym)
-----------------------------------------------
```
#_symによる参照のため、ベースレジスタに[r0]が指定されたものとみなします。

このときV1.02.00では分離演算子LOWWが無視されます。

3.4 回避策
ベースレジスタを省略しないでください。

回避策の適用例:
```
-----------------------------------------------
st.w r10, loww(#_sym)[r0]
-----------------------------------------------
```

3.5 恒久対策
次期バージョンで改修する予定です。

4. 仕様外のアセンブリ記述時の注意事項(No.5)

4.1 該当製品
CC-RH V1.02.00

4.2 内容
アセンブリソース上で式が記述できるオペランドにおいて、本来はアセンブル
エラーとすべき記述がエラーとならない場合があります。
その結果、意図しないコードを生成します。
なお、コンパイラが生成したアセンブリ命令は非該当です。

4.3 発生条件
式が記述できるオペランドまたはオペランドの一部が、以下(1)～(4)の
いずれかを満たす場合に発生する場合があります。
(1) 以下のいずれかの文字
*、/、+、-、<、>、&、"、|、^、~

発生条件例:
```
-----------------------------------------------
br      +
-----------------------------------------------
```

(2) 以下のいずれかの文字列
&&、==、«、»、()、""

(3) ()の中に(1)～(2)の文字または文字列を含む文字列

発生条件例:
```
-----------------------------------------------
mov (&&), r10
-----------------------------------------------
```

(4) 項が空の分離演算子またはセクション集合演算子

発生条件例:

```
-----------------------------------------------
add      low(), r10
-----------------------------------------------
```

4.4 回避策

正しいアセンブリ記述に変更してください。

4.5 恒久対策

次期バージョンで改修する予定です。
