

SuperH RISC engine ファミリ用C/C++コンパイラパッケージ ご使用上のお願い

SuperH RISC engine ファミリ用C/C++コンパイラパッケージ使用上の注意事項を 連絡します。

1. アセンブラのリテラルプール自動生成機能に関する注意事項
2. 最適化リンケージエディタでリテラルプールを参照する命令を最適化した 場合の注意事項

1. アセンブラのリテラルプール自動生成機能に関する注意事項

1.1 該当製品

SuperH RISC engine ファミリ C/C++コンパイラパッケージ
V.9.00 Release 00 ~ V.9.01 Release 01

1.2 内容

絶対アドレス形式のコードセクション内にリテラルプール自動生成機能によってリテラルプールが生成される命令が書かれたアセンブリプログラムの場合、CPU種別としてSH2AまたはSH2AFPUを選択してアSEMBルすると、リテラルプールの内容が正しくない場合があります。

1.3 発生条件

以下の条件をすべて満たした場合に発生することがあります。

- (1) アSEMBル時に、CPU種別としてSH2AまたはSH2AFPUを選択している。
- (2) 絶対アドレス形式で宣言されたコードセクションが存在する。
- (3) (2)のセクション内に以下6つのMOV命令のいずれか1つ以上が存在する。
MOV.B @(disp, R0),Rn MOV.B Rm,@(disp, R0)
MOV.W @(disp, R0),Rn MOV.W Rm,@(disp, R0)
MOV.L @(disp, R0),Rn MOV.L Rm,@(disp, R0)
- (4) (3)の命令のディスプレイメント幅(:12)を明記していない。
- (5) (3)の命令のdisp値は4ビットディスプレイメントで表現できる値である。
- (6) (3)の最初の命令より後にリテラルプール自動生成機能によりリテラルプールが生成される命令が存在する。
- (7) (6)の命令に対して生成されるリテラルはラベル(アドレス値)である。
- (8) (7)のラベルは(3)の最初の命令と(6)の命令の間に存在する。

例：

```
-----  
.SECTION P, CODE, LOCATE=H'00000600 ; 発生条件(2)  
MOV.L @(0,R0),R2 ; 発生条件(3)(4)(5)  
NOP  
LAB01: ; 発生条件(7)(8)  
NOP  
MOV.L #LAB01,R1 ; 発生条件(6)  
NOP  
.POOL  
.END  
-----
```

1.4 回避策

以下のいずれかの方法で回避してください。

- (1) 発生条件(2)のコードセクションを絶対アドレス形式でなく、
相対アドレス形式で宣言する。
- (2) リテラルプール自動生成機能を使用しない。
- (3) 発生条件(3)のMOV命令のディスプレイメント幅(:12)を明記する。

2. 最適化リンケージエディタでリテラルプールを参照する命令を最適化した 場合の注意事項

2.1 該当製品

SuperH RISC engine ファミリ C/C++コンパイラパッケージ
V.9.00 Release 00 ~ V.9.01 Release 01

2.2 内容

最適化を行うことによって、誤ったオブジェクトコードを生成する場合があります。

2.3 発生条件

以下の条件をすべて満たした場合に発生することがあります。

- (1) コンパイル時にオプション-cpu=SH2Aまたは-cpu=SH2AFPUを使用している。
- (2) コンパイル時にオプション-goptimizeを使用している。
- (3) リンク時に最適化を行っている。(-nooptimizeを使用していない。)
- (4) ある関数内に以下のいずれかを参照する命令がある。
 - ・変数/関数のアドレス
 - ・3バイト以上の定数
 - ・変数
- (5) (4)の参照命令を含む関数はあるプログラム領域の最後にある。

(6) (4)の命令が参照するアドレスまたは定数値がプログラム領域セクションのリテラルプールの先頭に配置されている。

(7) コンパイルにより、(5)の関数内で以下のいずれかの命令列が生成されている。

```
mov.l ; (6)のリテラルプールの先頭を参照するデータ転送命令  
rts/n ; 遅延スロットなしサブルーチンプロシージャからの復  
      ; 帰命令
```

または、

```
mov.l ; (6)のリテラルプールの先頭を参照するデータ転送命令  
rtv/n ; レジスタ値転送付き遅延スロットなしサブルーチン  
      ; プロシージャからの復帰命令
```

補足：

(7)に該当するかどうかは、コンパイルリストファイル(.lst)を参照することで確認できます。コンパイルリストを生成するには、コンパイル時に、-list オプションを使用してください。

発生例:

ファイル「example.c」

```
-----  
long a;          /* 発生条件(6) プログラム領域、Pセクションの */  
                /*      リテラルプールの先頭に変数aの */  
                /*      アドレスが出力される。      */  
long * mov();  
  
void func2(void);  
void func1()  
{  
    a++;  
    func2();  
}  
long * mov()     /* 発生条件(5) mov()はexample.c内のプログラム */  
                /*      領域、Pセクションの最後に配置 */  
{  
    return &a;   /* 発生条件(4)および(7) */  
}  
#pragma section 1 /* 発生条件(5) */  
void func2()     /* func2()はプログラム領域、P1セクションに配置 */  
{  
    .....  
}  
-----
```

2.4 回避策

以下のいずれかの方法で回避してください。

- (1) 該当ファイルのコンパイル時-goptimizeオプションを使用しない。
- (2) リンク時に-nooptimizeオプションを使用する。

3. 恒久対策

上記注意事項2件については、V.9.02 Release 00 (2月20日から公開予定)で改修されています。改修後のバージョンをご使用ください。

V.9.02 Release 00の詳細はRENASAS TOOL NEWS 資料番号 : 080216/tn2を参照ください (2月20日から公開予定)。

<http://tool-support.renesas.com/jpn/toolnews/080216/tn2.htm>

[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。