

SH C/C++コンパイラ V.9.00.01 改修内容

以降 12 点の問題を改修しました。

1. 以下の DSP ライブラリ関数を CPU コア SH4AL-DSP 用のプログラムで使用した場合に正しく動作しない場合がある。

FftComplex, FftReal, IfftComplex, IfftReal, FftInComplex,
FftInReal, IfftInComplex, IfftInReal, Fir, ConvComplete,
ConvCyclic, ConvPartial, Correlate, CorrCyclic, MatrixMult,
MsPower, および Variance

本改修により、CPU コア SH4AL-DSP 用のライブラリファイル名が以下の通り変更になります。

指定オプション ライブラリ

```
-pic=0 -endian=big sh4aldspnb.lib  
-pic=1 -endian=big sh4aldsppb.lib  
-pic=0 -endian=little sh4aldspnl.lib  
-pic=1 -endian=little sh4aldsppl.lib
```

2. 条件式を含むループにおいて、ループが不正に実行される場合がある (SHC-0008)。

例 1

```
-----  
int b[100];  
unsigned int c=0;  
void func1() {  
    unsigned int i;  
    for(i=0;i<=100;i++) { // 無限ループになる  
        if(i != c) {  
            b[i]=0;  
        }  
    }  
}
```

例 2

```
-----  
int b[100];  
unsigned int c=0;  
void func2() {  
    unsigned int i;  
    for(i=0;i<c;i++) { // c=0 の時もループを 1 回以上まわる  
        if(i != 5) {  
            b[i]=0;  
        }  
    }  
}
```

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) optimize=1 オプションを使用している。
- (2) ループが存在する。

- (3) (2)のループ制御変数の増分値が1または-1である。
- (4) (2)のループ内に if 文が存在する
- (5) 以下(a)、または(b)のいずれかの条件を満たす。
 - (a) (4)の if 文の条件式が(2)のループ制御変数と、(2)のループ内で値が不変の式(例1の変数 c)の比較である。
 - (b) (2)のループ制御変数の初期値または上限値がループ内で値が不変の式(例2の変数 c)である。
- (6) (5)のループ内不変式が整数型である。

3. do-while ループで、正しくは1回で終了するものが2回以上ループする場合がある(SHC-0010)。

例：

```
-----  
int func() {  
    int count=0;  
    int limit=0x60000000;  
    do {  
        count++;  
        limit += 0x10000000;  
    } while(limit < -0x60000000); // 正しい動作では1回目のループ後の  
                                // 判定で偽となり、ループを抜ける。  
  
    return (count);           // count=1 が正しいが、違う値になる。  
}  
-----
```

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) optimize=1 オプションを使用している。
- (2) do-while ループが存在する。
- (3) (2)のループ制御変数の型が int, signed int, long, または signed long である。
- (4) (2)のループ判定式が、定数値との大小比較(<, <=, >, >=) である。
- (5) (2)のループの比較演算子、制御変数の初期値、更新値、および判定式の比較値が以下(a)~(d)のいずれかの条件を満たす。
 - (a) 判定式が、ループ制御変数<定数 の場合に以下全てに該当する。
 - ・更新値が正
 - ・比較値<=初期値
 - ・ $0x00000000 \leq (\text{比較値} - \text{初期値} - 1) <= 0x7FFFFFFF$
 - (b) 判定式が、ループ制御変数<=定数 の場合に以下全てに該当する。
 - ・更新値が正
 - ・比較値<初期値
 - ・ $0x00000000 \leq (\text{比較値} - \text{初期値} - 1) <= 0x7FFFFFFF$
 - (c) 判定式が、ループ制御変数>定数 の場合に以下全てに該当する。
 - ・更新値が負
 - ・初期値<=比較値
 - ・ $0x00000000 \leq (\text{初期値} - \text{比較値} - 1) <= 0x7FFFFFFF$

(d) 判定式が、ループ制御変数>=定数 の場合に以下全てに該当する。

- ・ 更新値が負
- ・ 初期値<比較値
- ・ $0x00000000 \leq (\text{初期値} - \text{比較値} - 1) \leq 0x7FFFFFFF$

4. 1ビットの符号付きビットフィールドと1との比較を行ったとき、またはある比較結果に対して演算を行いその結果を1と比較したときに、比較結果が間違っている場合がある(SHC-0011)。

例1:

```
-----  
struct {  
    char b0:1;  
} ST;  
void func() {  
    if (ST.b0 != 1) {  
        . . . . .  
    }  
}
```

例2:

```
-----  
int a;  
void func2() {  
    int t;  
    t = ((a & 0x40) == 0);  
    t = t - 1;  
    t = -t;  
    if (~t==1) {  
        a = 1;  
    } else {  
        a = 2;  
    }  
}
```

発生条件

以下(1)、(2)のいずれかの条件を満たした場合に発生することがあります。

(1) 以下のすべての条件を満たした場合。

- (a) optimize=1 を指定している。
- (b) 1ビットの符号付きビットフィールドを使用している。
- (c) (b)と1の比較(==, !=)を行っている。

(2) 以下のすべての条件を満たした場合。

- (a) optimize=1 を指定している。
- (b) 比較結果に対して以下のいずれかの演算を行っている。
比較結果と1の減算、比較結果と1の排他的論理和、比較結果の符号反転、または比較結果のビット反転
- (c) (b)の演算結果と1の比較(==, !=)を行っている。

5. 0 との加算および減算、または 1 との乗算をした結果を他の演算で使用したときに、変数の値を誤って更新する場合がある (SHC-0012)。

例 :

```
-----  
int a[4], b;  
void func() {  
    a[3&(b-0)]=0;  
}
```

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) 変数と 0 の加算および減算、または変数と 1 の乗算をしている。
- (2) (1) の結果を以下の演算で使用している。

加算、減算、論理演算 (&, |, ^)、除算、剰余算、またはシフト

6. pack=1 を指定した構造体または共用体の double 型メンバを参照したときに、レジスタ R2 の値を誤って変更する場合がある (SHC-0013)。

例 :

```
-----  
#pragma pack 1  
struct {  
    double d;  
} ST;  
int t;  
double d[2];  
void func() {  
    d[t]=ST.d;  
}
```

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) pack=1 を指定した構造体または共用体がある。
- (2) (1) の構造体または共用体に double 型のメンバが存在する。
- (3) cpu=sh4, sh4a, sh2afpu を指定している。
- (4) size または unaligned=runtime オプションを使用している。
- (5) 実行時ルーチン _pack1_ld64 の呼び出しがある。

7. 定数乗算および定数除算を共に含む式において、演算結果が間違っている場合がある (SHC-0015)。

例 :

```
-----  
unsigned int a=65536;  
unsigned int b;  
void func() {  
    b=(a*65536)/8; // b=0 ((65536*65536)/8→0/8=0) が正しい
```

```
} // 結果だが、b=65535<<13 と置き換えられる。  
-----
```

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) unsigned 型の式と定数の乗算式が存在する。
- (2) (1)を含む式を(1)の定数の正の約数で除算している。
- (3) (1)の乗算結果がその式の型の最大値を超える。

8. 被シフト数のビットサイズ未満の定数シフトを複数回行い、そのシフト数の合計が被シフト数のビットサイズ以上になった場合、演算結果が間違っている場合がある (SHC-0016)。

```
例 :  
-----  
int x,y;  
void func() {  
    x=y<<31<<1;  
    // シフト数の合計は 32 で、int 型のビットサイズ  
    // 以上になる  
}
```

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) cpu オプションのパラメータに sh1, sh2, sh2e, または sh2dsp 以外を指定している。
- (2) 以下 (a)、(b)のいずれかの条件を満たす。
 - (a) 左シフトもしくは 2 のべき乗の乗算が 2 回以上連続し、かつ個々のシフト数、乗数は被シフト数のビットサイズ未満である。
 - (b) 右シフトもしくは 2 のべき乗の除算が 2 回以上連続し、かつ個々のシフト数、乗数は被シフト数のビットサイズ未満である。
- (3) (2)のシフト数と乗数の値の合計が被シフト数のビットサイズ以上となる。

9. 関数の最後が関数呼び出しの場合、呼び出し先関数のアドレスをレジスタ R0 に誤って書き込む場合がある。

```
例 :  
-----  
char a[3];  
void func2() {}  
#pragma section A  
void func() {  
    ++a[2];  
    func2();  
}
```

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) 呼び出し元関数の最後で関数を呼び出している。
- (2) 呼び出し先関数の定義が(1)と同一ファイル内にある。
- (3) (2)の関数定義と(1)の関数呼び出しは別セクションにある、もしくは4096バイト以上離れた位置にある。
- (4) 呼び出し元関数内に(1)以外の関数呼び出しがない。

10. ポインタを介してビットフィールドメンバに対して演算を行った場合、その演算結果が間違っている場合がある (SHC-0023)。

例 :

```
-----  
typedef union {  
    unsigned int a;  
    unsigned int b:32;  
} UN;  
UN un;  
void func() {  
    int *p=(int *)&un.a;  
    un.b=1;  
    *p+=1;  
}  
-----
```

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) 構造体または共用体型を使用している。
- (2) (1)のメンバを直接参照(例では un.b)している。
- (3) (2)と同じ領域のメンバ(例では un.a)を指すポインタ変数(例では p)が存在する。
- (4) (3)のポインタはローカル変数である。
- (5) (3)のポインタの間接参照(例では *p)が存在する。
- (6) 当該領域の値を更新する。
- (7) 構造体または共用体自身の参照を関数内で行っていない。

11. 構造体または共用体コピーを行った場合、スタック領域が必要以上に確保される場合がある (SHC-0021)。

例 :

```
-----  
typedef struct {  
    unsigned char c;  
} ST0;  
typedef struct {  
    ST0 s;  
} ST;  
extern ST A[1000];  
-----
```

```
extern unsigned short i;
void func(ST *d) {
    A[i-1].s=d->s; // スタックが A[1000]分余分に確保される
}
-----
```

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) 構造体または共用体型のメンバを持つ構造体または共用体配列が存在する。
- (2) (1)の配列の要素数は2以上である。
- (3) 構造体または共用体代入文が存在する。
- (4) (3)の左辺は(1)の構造体または共用体配列の構造体または共用体型メンバである。

12. C4098、および C4099 インターナルエラーが発生する場合がある (SHC-0007, SHC-0017, SHC-0022, SHC-0025)。

