

SuperH RISC engine C/C++コンパイラ Ver. 8 使用上のご注意

SHC コンパイラ Ver. 8 には、以下に記載しました不具合があります。SHC コンパイラ Ver. 8 をご使用の場合には、ご注意ください。よろしくお願いいたします。

なお、本ページに記載されている不具合は、コンパイラ Ver. 8.0.01 以降（コンパイラパッケージ Ver. 8.0.01 以降）では、全て修正されています。製品をリビジョンアップしていただきますよう、よろしくお願いいたします。

ビットフィールドメンバ比較不正

該当製品

製品型名	パッケージバージョン	コンパイラバージョン
P0700CAS8-MWR	8.0.00	8.0.00
P0700CAS8-SLR	8.0.00	8.0.00
P0700CAS8-H7R	8.0.00	8.0.00

現象

volatile 指定された構造体において1ビットのビットフィールドメンバをポインタ参照して定数と比較した場合、比較結果が不正になる場合があります。

[例]

```
typedef volatile struct { /* 構造体の型は volatile 修飾されている */
    unsigned int a:19,
    b:1; /* メンバは volatile 修飾されていない */
} ST;
```

```
int f(ST *p) {
    if (p->b) { /* ポインタ間接参照式 p->b を使用している */
        return 1;
    }
    return 0;
}
```

```
MOV.L @R4, R0
TST #16, R0 ; mask 値が不正
MOVT R2
MOV R2, R0
RTS
XOR #1, R0
```

発生条件

次の条件を全て満たす時、発生することがあります。

(1) optimize=1 を指定している。

- (2) if 文の条件式中にビットフィールドの定数比較式がある。
- (3) 上記ビットフィールド式はポインタ間接参照式である。
- (4) 定数 0 との ==, != 比較、または定数 1 と 1 ビット unsigned 型ビットフィールドメンバの == 比較である。
- (5) 構造体の型は volatile 修飾されているが、メンバは volatile 修飾されていない。
- (6) ポインタの型は volatile 修飾されていない。
- (7) ポインタはスカラ変数である。
- (8) ポインタはアドレス参照されていない。
- (9) ビットフィールドの宣言型は (unsigned) (short | int | long) のいずれかである。

回避策

該当箇所が存在した場合、以下のいずれかの方法で回避してください。

- (1) ビットフィールドメンバの型を volatile 修飾する。
- (2) 同一ファイル中にポインタのアドレス参照式を置く。
- (3) ポインタ間接参照式を p->b から (*&p)->b に書き換える。
- (4) optimize=0 を指定する。

