

以下の制限事項を対策しました。

ビットフィールドメンバ比較不正

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る。

```
[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;
}
```

```
[出力コード]
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) (b)の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) ポインタはスカラー変数である。
- (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。