

M3T-CC32R ご使用上のお願い -- ビットフィールドを使った変数の変更について --

M32Rファミリ用CコンパイラパッケージM3T-CC32Rの使用上の注意事項を連絡します。

- 変数をビットフィールドを使って変更する場合の注意事項

1. 該当製品

M3T-CC32R V.4.00 Release 1 ~ V.4.20 Release 1A

2. 内容

以下の発生条件に該当する整数型または浮動小数点型の変数を参照する場合、誤った最適化によって、参照される値が不正になります。

2.1 発生条件

以下の条件をすべて満たす場合、問題が発生します。

- (1) 最適化オプション-O4, -O5, -O6, および-O7のいずれかを指定している。
または、-Ospace または -Otimeを単独で指定している。
- (2) 整数型または浮動小数点型の変数に代入している。
- (3) ビットフィールドを含む構造体へのポインタ (&変数が返す値も含む) を用いてビットフィールドに代入している。
- (4) (3)のポインタが、(2)の変数の格納場所と同じアドレスを指している。
- (5) (3)の実行の後に、(2)の変数を参照している。

2.2 発生例

```
ソースファイル例1 : sample1.c-----  
-----unsigned long src1, dst1;struct btf1 { unsigned long x:1; unsigned long
```

```
y:3;};void func01(void){ src1 = 0x80000000; /* 発生条件(2) */ ((struct
btf1 *)&src1)->y = 2; /* 発生条件(3)、(4) */ dst1 = src1; /* 発生条件(5)
*/}-----ソースファイル
```

例2 : sample2.c-----

```
unsigned long src2, dst2;struct btf2 { unsigned long x:1; unsigned long y:3;}
*ptr2;void initialize(void){ ptr2 = (struct btf2 *) &src2; /* 発生条件(4) */}void
func02(void){ /* この関数の実行までに、initialize関数を実行している */ src2 = 0;
/* 発生条件(2) */ ptr2->y = 2; /* 発生条件(3) */ dst2 = src2;
/* 発生条件(5) */}-----
```

- ソースファイル例3 : sample3.c-----

```
-----long src3, dst3;struct btf3 { unsigned long x:1; unsigned long y:3;}
*ptr3;long func03(void){ src3 = 255; /* 発生条件(2) */ ptr3 = (struct
btf3 *) &src3; /* 発生条件(4) */ ptr3->y = 2; /* 発生条件(3) */ return
src3; /* 発生条件(5) */}-----
```

-----ソースファイル例4 : sample4.c-----

```
-----long src4, dst4;struct btf4 { unsigned long x:1;
unsigned long y:3;} *ptr4;long func04(long arg4){ src4 = arg4; /* 発生
条件(2) */ ptr4 = (struct btf4 *) &src4; /* 発生条件(4) */ ptr4->y = 2;
/* 発生条件(3) */ return src4; /* 発生条件(5) */}-----
```

3. 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(3)のポインタが、変数のアドレスの場合、変数の元の型とビットフィールドを含む構造体の型の両方のメンバを持つ共用体を作り、発生条件(3)の代入を、共用体のメンバへの代入に変更する。

ソースファイル例1の回避例-----

```
-----unsigned long dst1;struct btf1 { unsigned long x:1; unsigned long
y:3;};union { /* src1 を共用体に変更 */ unsigned long word; struct
btf1 bit;} src1;void func01(void){ src1.word = 0x80000000; /* src1 を src1.word
に置き換える */ src1.bit.y = 2; /* src1をビット操作で書き変える */ dst1 =
src1.word; /* src1 を src1.word に置き換える */}-----
```

- (2) 発生条件(5)で参照する変数が代入式の右辺である場合
発生条件(5)の代入式の左辺および右辺をvolatileへのポインタを使った間接参照にする。

ソースファイル例2の回避例-----

```
-----unsigned long src2, dst2;struct btf2 { unsigned long x:1; unsigned long
y:3;} *ptr2;void initialize(void){ ptr2 = (struct btf2 *) &src2; /* 発生条件(4)
*/}void func02(void){ /* この関数の実行までに、initialize関数を実行している */ src2
= 0; ptr2->y = 2; /* 書き込みと読み出しをvolatile型に変更する */ *((volatile
```

```
unsigned long*)&dst2) = *((volatile unsigned long*)&src2);}-----  
-----
```

- (3) -Ospaceまたは-Otimeを使用する場合
同時に-O0, -O1, -O2, および-O3のいずれかを指定する。

4. 恒久対策

本問題は、次期バージョンで改修する予定です。

[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。

© 2010-2016 Renesas Electronics Corporation. All rights reserved.