

## M3T-NC308WA, M3T-NC30WA ご使用上のお願い

Cコンパイラ(アセンブラ・統合化開発環境付き) M3T-NC308WA, M3T-NC30WAの使用上の注意事項を連絡します。

- 自動変数の配置に関する注意事項

### 1. 該当製品

M32C/80, M16C/80, M16C/70シリーズ用

M3T-NC308WA V.5.00 Release 1

M16C/60, M16C/30, M16C/20, M16C/10シリーズ用

M3T-NC30WA V.5.00 Release 1 ~ V.5.00 Release 2

### 2. 内容

関数内の自動変数を、誤ったアドレスに配置することがあります。

#### 2.1 発生条件

以下の条件をすべて満たす場合に発生する可能性があります。

- ※ 以下の条件を満たしていても、他の変数の使用状況や最適化の実行結果によっては現象が発生しない場合があります。

- (1) 自動変数を使用している関数は、以下のいずれの拡張機能も指定されていない。

#pragma INTERRUPT, #pragma INTF, #pragma HANDLER, #pragma  
INTHANDLER, #pragma CYCHANDLER, #pragma ALMHANDLER

M3T-NC30WAの場合はさらに、この関数から別の関数(inline関数も含む) を呼び出している。

- (2) (1)の関数内で偶数サイズの自動変数と奇数サイズの自動変数をそれぞれ1つ以上宣言している。
- (3) 引数が(1)の関数に、スタック渡しで渡されている。

- (4) (3)の引数は、ポインタ型でなく、かつ、アドレス演算子 "&"で演算されていない。
- (5) 以下のいずれかの条件に該当する。
- a) volatile修飾されていない引数が存在し、その引数のサイズ以上である偶数サイズの自動変数(\*1)が存在する。  
(\*1) 例えば、引数がint型の場合、自動変数はint型、long型などである。
  - b) 引数のサイズ以下の奇数サイズの自動変数(\*2)が存在し、かつ、その自動変数はvolatile修飾されていない。  
(\*2) 例えば、引数がint型の場合、自動変数はchar型である。
- (6) (5)-a)または、(5)-b)の自動変数は、以下の条件すべてに該当する。
- a) 自動変数は、ある関数内で最後に(3)の引数が使用されている個所以前では使用されていない。
  - b) 自動変数は、アドレス演算子 "&"で演算されていない。
  - c) 自動変数は、asm関数の引数として使用されていない。
  - d) 自動変数は、ポインタ型でない。
- (7) 上記の引数および自動変数は、関数内で少なくとも1回以上参照され、かつレジスタに割り付けられていない。
- (8) 最適化オプション -O, -O[1-5], -OR, -OSのいずれか1つ以上を指定し、M3T-NC30WAの場合はさらに、-OSFA(-Ostack\_frame\_align)を併用している。  
※ M3T-NC308WAには、オプション -OSFA はありません。

## 2.2 発生例

[C言語ソース]

構造体変数での発生例を示します。

```
-----
typedef struct { int i; } s2; /* 2バイトの構造体型 */
typedef struct { char c; } s1; /* 1バイトの構造体型 */

void sub(void);
/* ↓変数の使用区間 */
void func(s2 a) /* a 発生条件(1),(3),(4) */
{ /* | */
  s2 dmy; /* | 偶数サイズの自動変数 発生条件(2) */
  s1 b; /* | 奇数サイズの自動変数 発生条件(2),(5)-b), */
/* | (6)-c),(6)-d) */
  dmy.i = idmy; /* | dmy */
/* | | */
  ia = a.i; /* a | a の最後の使用個所 */
/* | */
}
```

```

b.c = cb;    /* b | b の最初の使用箇所 発生条件(6)-a), */
b.c++;      /* | | (6)-b),(6)-c),(6)-d)*/
cb = b.c;    /* b | */
            /* | */
dmy.i++;     /* | 発生条件(7) */
idmy = dmy.i; /* dmy */

sub();      /* 発生条件(1) */
}

```

-----  
[NC308による生成アセンブリ語ソース]  
-----

```

;### # FUNCTION func
;### # FRAME AUTO ( dmy) size 2, offset -2
;### # FRAME AUTO ( b) size 1, offset 3
;### # FRAME ARG ( a) size 2, offset 8
;### # ARG Size(2) Auto Size(2) Context Size(8)

```

```

.glb _func
_func:
enter #02H
mov.w _idmy:16,-2[FB] ; dmy.i = idmy;
mov.w 8[FB],_ia:16 ; ia = a.i;
mov.b _cb:16,3[FB] ; b.c = cb; <- ※
add.b #01H,3[FB] ; b.c++;
mov.b 3[FB],_cb:16 ; cb = b.c;
add.w #0001H,-2[FB] ; dmy.i++;
mov.w -2[FB],_idmy:16 ; idmy = dmy;
jsr _sub
exitd

```

-----  
※ "b.c = cb;" のコンパイル結果である "mov.b \_cb:16,3[FB]" のオペランド "3[FB]" は当該関数が呼び出される前のFBレジスタの値がenter命令によって退避された領域です。したがって、この例では、自動変数 "b" が誤ったアドレスに配置されています。

### 3. 回避策

以下のいずれかの方法により回避してください。

- (1) 自動変数の宣言の直後に、ダミーのasm関数を挿入し、そのasm関数の引数に不正アドレスに割り付けられた自動変数 "b" を使用する。

なお、自動変数bが構造体の場合には、その先頭メンバを使用してください。

(2) M3T-NC30WAの場合、オプション "-OSFA" を使用しない。

[回避策(1)の例]

```
-----  
void func(s2 a)  
{  
    str2 dmy;  
    str1 b;  
  
    asm(";$@", b.c); /* ダミーasm関数を挿入し、構造体メンバを記述 */  
                    /* 自動変数bを示すオペランドがコメントとして */  
                    /* 出力される。 */  
  
    dmy.i = idmy;  
  
    /* 省略 */  
}
```

-----

#### 4. 恒久対策

本内容については、次期バージョンアップ時に改修する予定です。

---

#### [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。