

M3T-NC308WA ご使用上のお願い

M32C/80, M16C/80, M16C/70シリーズ用CコンパイラパッケージM3T-NC308WAの使用上の注意事項を連絡します。

- long long型変数またはlong long型定数の除算および剰余算に関する注意事項

1. 該当製品

M3T-NC308WA V.5.00 Release 1 ~ V.5.20 Release 1

2. 内容

long long型変数またはlong long型定数の除算または剰余算を行うとアドレスレジスタの内容を不正に書きかえます。

3. 発生条件

以下の条件を満たす場合に発生します。

- (1) 除算または剰余算がある。
- (2) (1)の演算の被除数または除数のいずれかが、以下の変数または定数である。
 - signed long long型
 - unsigned long long型

3.1 発生例

```
-----#include
<stdio.h>const char far ch[10] = "Hello¥n";long long func1(long long ll1,long long
ll2){    long long ll3;    ll3 = ll1 / ll2;    /* 発生条件(1)、(2) */    return ll3;}int
main(void){    long long ll;    int far *i;    i = ch;    ll = func1( 2000, 1000
);    printf( "%s¥n",i );}-----
```

4. 回避策

インラインアセンブル機能の#pragma ASM～#pragma ENDASM または asm()関数を使用して、除算および剰余算の前にアドレスレジスタA0およびA1を退避し、除算および剰余算の直後に復帰してください。

```
例-----#include
<stdio.h>const char far ch[10] = "Hello¥n";long long func1(long long ll1,long long ll2){
long long ll3;    asm( "    pushm A0,A1" ); /* A0,A1を退避する */    ll3 = ll1 / ll2;
asm( "    popm A0,A1" ); /* A0,A1を復帰する */    return ll3;}int main(void){    long
long ll;    int far *i;    i = ch;    ll = func1( 2000, 1000 );    printf( "%s¥n",i );}-
-----
```

5. 恒久対策

次期バージョンアップの際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。