GRADE **A**

MESC TECHNICAL NEWS No.M16C-66-0012

MESC TECHNICAL NEWS 'No.M16C-64-0011' Replacement

Classification	**Products Effected**
Corrections and supplementary explanation of document ✔ Notes Knowhow Others	M16C/80 Group

MESC TECHNICAL NEWS 'No.M16C-64-0011' requires additional information.
Please replace old Technical News 'No.M16C-64-0011' with this corrected Technical News 'M16C/80 Group Precautions When Using Address Match Interrupts' (M16C-66-0012).

[Additional contents]

The contents of section 3.4 is increased.

< Additional instructions >

```
mov.w    6[SP],R0              ; Read FLG on stack
  -->
mov.w    6[SP],R0              ; Read FLG on stack (In high speed interrupt,  stc   SVF,R0 )
```

[Attached]

Corrected Technical News 'No.M16C-66-0012'
'M16C/80 Group Precautions When Using Address Match Interrupts' ..... 5 pages

GRADE **A**

MESC TECHNICAL NEWS No.M16C-66-0012

M16C/80 Group Precautions When Using Address Match Interrupts

Classification	Products Effected
Corrections and supplementary explanation of document ✔ Notes Knowhow Others	M16C/80 Group

1. Affected devices

- M16C/80 Group

2. Precautions

When using the address match interrupt function, unwanted interrupts may occur. This can happen when the address match interrupt register is set to the first address of an interrupt process routine, or the contents of an interrupt control register is dynamically changed.

3. Detail and countermeasure

If the address match interrupt disabled, this countermeasure is not needed.

3.1 When the address match interrupt register is set to the first instruction of an interrupt process routine, and the interrupt occurs,

Example:

```
     Inta:                            ; Interrupt A routine
       Pushm    R0,R1,R2,R3,A0        ; <-- Address match interrupt register set here
```

(1) If a non-maskable interrupt (watchdog timer or NMI) occurs, the non-maskable interrupt routine is executed twice.

(2) If another maskable interrupt (peripheral I/O interrupt) occurs, it’s process routine will execute, regardless of it’s priority level.

(3) If no other interrupts occur, a software interrupt number 63 will occur.

Countermeasure :

Do not set the address of the first instruction of an interrupt process routine to the address match interrupt register

When you are setting the address match interrupt to an interrupt routine, use the second instruction or later to write to the address match interrupt register.

3.2 If an interrupt occurs immediately after it’s control register is modified (interrupts are enabled), the address match interrupt register is set to the next 7 instructions (rewriting the interrupt priority level to a smaller value or clearing interrupt request bit / interrupt enable flag is “1”).

Example:

```
    mov.b    #0,TA0IC          ;Change TA0 interrupt priority level to 0
    nop                        ; 1st instruction ┐
    nop                        ; 2nd instruction │
    nop                        ; 3rd instruction │
    nop                        ; 4th instruction ├ Occurs when address match interrupt
    nop                        ; 5th instruction │ is set to one of these
    nop                        ; 6th instruction │
    nop                        ; 7th instruction ┘
```

(1) If a non-maskable interrupt occurs immediately before the execution of the instruction address set in the address match interrupt register, the non-maskable interrupt routine is executed twice. **(If ‘I’ flag is cleared to ‘0’, interrupt will not occur.)**

(2) If another maskable Interrupt request occurs immediately before the execution of the instruction address set in the address match interrupt register, it’s service routine will execute, regardless of it’s priority level. **(If ‘I’ flag is cleared to ‘0’, interrupt will not occur.)**

(3) If no interrupt occurs immediately before the execution of the instruction address set in the address match interrupt register, a software interrupt number 63 will occur. **(If ‘I’ flag is cleared to ‘0’, interrupt will not occur.)**

Countermeasure :

Do not set the address in the address match interrupt register to any of the next 7 instructions following a rewrite to the interrupt control register (rewriting the interrupt priority level to a smaller value or clearing the interrupt request bit).

When you need to set the address match interrupt register, either use the address of the interrupt control register rewrite operation or use the 8th or later instruction following the rewrite instruction.

3.3 If an interrupt is pending, then its control register is changed, and then the I flag is set (or rewriting the IPL to a smaller value) (Note), if the address match interrupt register is set to one of the next 3 instructions:

Example:

```
        fclr        I
        ••••
        ••••
        bclr 3,TA0IC                ; Clear TA0 interrupt request bit
        fset I                      ; Set I flag (interrupt enabled)
        nop                         ; 1st instruction
        nop                         ; 2nd instruction  Occurs when address match interrupt
        nop                         ; 3rd instruction  is set to one of these
```

(1) If another non-maskable Interrupt request occurs immediately before the execution of the instruction address set in the address match interrupt register, non-maskable Interrupt routine is executed twice.

(2) If another maskable Interrupt request occurs or is pending immediately before the execution of the instruction address set in the address match interrupt register, it's service routine will execute, regardless of it's priority level. **(If 'I' flag is cleared to '0', interrupt will not occur.)**

(3) If no interrupt occurs immediately before the execution of the instruction address set in the address match interrupt register, a software interrupt number 63 will occur.

Countermeasure :

Do not set the address in the address match interrupt register to any of the next 3 instructions following a setting to the interrupt enable flag (I flag) or rewriting the IPL to a smaller value (Note).

Note : Instructions related to setting the I flag: FSET I, LDC FLG, POPC FLG, REIT and FREIT. Refer to section 3.4 about REIT and FREIT for detail.
Instructions related to rewriting the IPL to a smaller value: LDC FLG, LDIPL, POPC FLG, REIT and FREIT. Refer to section 3.4 about REIT and FREIT for detail.

3.4 If an interrupt B's interrupt control register is rewritten in interrupt A's interrupt routine, then returns to the address set in the address match interrupt register by a return instruction from interrupt:

```
Interrupt_A:
      ••••
      bclr    3,TA0IC             ; Rewrite interrupt B's interrupt control register
      ••••
      reit                        ; When return to the address set in address match interrupt
                                  ; register by a return instruction from interrupt
```

(1) If another non-maskable Interrupt request occurs immediately before the execution of the instruction address set in the address match interrupt register, non-maskable Interrupt routine will execute twice.

(2) If another maskable Interrupt request occurs or is pending immediately before the execution of the instruction address set in the address match interrupt register, it’s service routine will execute, regardless of it’s priority level. **(If ‘I’ flag is cleared to ‘0’, interrupt will not occur.)**

(3) If no interrupt occurs immediately before the execution of the instruction address set in the address match interrupt register, a software interrupt number 63 will occur.

Countermeasure :

Do not rewrite the interrupt control register in the interrupt process routine.

When rewriting the interrupt control register in interrupt process routine, add the following instructions at the end of the interrupt (immediately before reit and freit instructions). When multiple interrupts are enabled in another interrupt, add the same instructions as above at the end of the interrupt routine.

Also, when rewriting the interrupt control register in nonmaskable interrupt process routine, add the following instructions at the end of all interrupts.

< Additional instructions >

```
                                  ; Add after register return instruction (popm instruction)
      fclr     U                  ; Select ISP (Not needed if ISP is already selected.)
      pushm    R0                 ; Store R0
      mov.w    6[SP],R0           ; Read FLG on stack (In high speed interrupt,  stc   SVF,R0 )
      ldc      R0,FLG             ; Set to FLG
      popm     R0                 ; Restore R0
      nop                         ; Dummy
      reit                        ; Interrupt end (Freit when high speed interrupt)
```

Example:
When the interrupt B's interrupt control register is rewritten in interrupt A routine and multiple interrupts are enabled in interrupt C routine, the countermeasure is needed at end of interrupt A and C routine.

```
Interrupt_A:
        pushm   R0,R1,R2,R3,A0,A1 ; Store registers
        ••••
        bclr    3,TA0IC           ; Rewrite interrupt B's interrupt control register
        ••••
        popm    R0,R1,R2,R3,A0,A1 ; Restore registers
        fclr    U                 ; Select ISP  (Not needed if ISP is already selected.)
        pushm   R0                ; Store R0
        mov.w   6[SP],R0          ; Read FLG on stack
        ldc     R0,FLG            ; Set to FLG
        popm    R0                ; Restore R0
        nop                       ; Dummy
        reit                      ; Interrupt end

Interrupt_C:
        pushm   R0,R1,R2,R3,A0,A1 ; Store registers
        fset    I                 ; Multiple interrupt enabled
        ••••
        ••••
        popm    R0,R1,R2,R3,A0,A1 ; Restore registers
        fclr    U                 ; Select ISP  (Not needed if ISP is already selected.)
        pushm   R0                ; Store R0
        mov.w   6[SP],R0          ; Read FLG on stack
        ldc     R0,FLG            ; Set to FLG
        popm    R0                ; Restore R0
        nop                       ; Dummy
        reit                      ; Interrupt end
```