GRADE A

MESC TECHNICAL NEWS No.M16C-40-9912

M16C/80 Group Cautions for Using Decimal Arithmetic Instruction (DSUB, DSBB, DADD or DADC)

1. Affected devices

• M16C/80 Group

2. Cautions

When DMA transfer occurs when executing decimal operation instruction (DSUB, DSBB, DADD or DADC), the result of the operation will not be correct.

3. Countermeasure

When using DMA, do not use decimal operation instruction.
The examples programs for converting HEX to decimal and decimal to HEX are shown in the next pages.
When you want to perform a decimal operation, execute the operation using HEX, then convert the results to decimal.

HEX - decimal exchange program (2 bytes)

```
.SECTION    PROG,         CODE
.GLB  HEXtoBCD_2byte
;==========================================================================
;        Title : Exchange HEX code toBCD code
;        Outline : Exchange HEX 2 bytes code to BCD 4 bytes code
;        Input : -----------------------> Output :
;        R0  ()                      R0  (Low order 4-digit of BCD code)
;        R1  (HEX code input)        R1  (Undefined)
;        R2  ()                      R2  ( High order 1-digit of BCD code)
;        R3  ()                      R3  (Undefined)
;        A0  ()                      A0  (Not used)
;        A1  ()                      A1  (Not used)
;        Stack : Not used
;==========================================================================
HEXtoBCD_2byte:                                 ; HEXtoBCD_2byte{
         mov.w #0,R0                            ; BCD area initialization
         mov.w #0,R2                            ;
HEXtoBCD_2byte5:                                ; Count 5th digit
         mov.w R1,R3                            ; Save R1 to R3 before subtraction
         sub.w  #10000,       R1                ; R1 = R1 - 10000
         jltu    HEXtoBCD_2byte4                ; After 4th digit
         inc.w   R2                             ; 5th digit ++
         jmp             HEXtoBCD_2byte5
HEXtoBCD_2byte4:                                ; Count 4th digit
         mov.w R3,     R1                       ; Restore R1
HEXtoBCD_2byte4Loop:
         mov.w R1,     R3                       ; Save R1
         sub.w  #1000,R1                        ; R1 = R1 - 1000
         jltu    HEXtoBCD_2byte3                ; if( R1 < 1000 ), go to 3rd digit
         inc.b   R0H                            ;4th digit ++
         jmp     HEXtoBCD_2byte4Loop
HEXtoBCD_2byte3:                                ; Count 3rd digit
         mov.w R3,     R1                       ; Restore R1
         shl.b   #4,     R0H                    ; R0H << 4 (rotate 4th digit)
HEXtoBCD_2byte3Loop:
         mov.w R1,     R3                       ; Restore R1
         sub.w  #100,R1
         jltu    HEX_toBCD_2byte2               ; if( R1 < 100 ), go to 2nd digit
         inc.b   R0H                            ; 3rd digit++
         jmp     HEXtoBCD_2byte3Loop
HEX_toBCD_2byte2:                               ; Count 2nd digit
         mov.w R3,     R1                       ; Restore R1
HEXtoBCD_2byte2Loop:
         mov.w R1,     R3                       ; Restore R1
         sub.w  #10, R1                         ; R1 = R1 - 10
         jltu    HEX_toBCD_2byte1               ; if( R1 < 10 ), go to 1st digit
         inc.b   R0L                            ; 2nd digit ++
         jmp     HEXtoBCD_2byte2Loop
HEX_toBCD_2byte1:
         shl.b   #4,R0L                         ; R0L << 4 (rotate 2nd digit)
         add.w  R3,R0                           ; Add rest of 1st digit
rts                                             ; }
.END
```

Decimal - HEX exchange program (2 bytes)

```
.SECTION    PROG,         CODE
.GLB   BCDtoHEX_2byte
;============================================================================
;         Title : Exchange BCD code to HEX code
;         Outline : Exchange BCD 2 bytes code to HEX 2 bytes code
;         Input : -----------------------> Output :
;         R0 (BCD code input )          R0 (Undefined)
;         R1 ()                         R1 (Loop variable)
;         R2 ()                         R2 (HEX code output)
;         R3 ()                         R3 (Not used)
;         A0 ()                         A0 (Not used)
;         A1 ()                         A1 (Not used)
;         Stack : Not used
;============================================================================
BCDtoHEX_2byte:                                      ; BCDtoHEX_2byte{
         mov.w #0,      R2                           ;         R2 = 0 (HEX area initialization )
         mov.b #16,     R1H                          ;         R1H = 16 ( Set rotate loop cycle )

BCDtoHex_2byte_Rotate:                               ;         while( R1H ){
         shl.w   #-1,    R0                          ;                  Right rotate R0, remainder --> C flag
         rorc.w R2                                   ;                  Right rotate with C flag
         mov.b #4,               R1L                 ;                  R1L = 4 (set rotate loop cycle)
BCDtoHex_2byte_Rot:                                  ;                  while( R1L ){
         btst     3,       R0L                       ;                           Correction check
         jnc               BCDtoHEX_2byte_NoAdj      ;                           if ( b3 == 1 ){
         sub.w  #3,      R0                          ;                                    Correct
BCDtoHEX_2byte_NoAdj:                                ;                           }
         rot.w   #-4, R0                             ;                           Rotate 4 digits
         adjnz.b          #-1, R1L, BCDtoHex_2byte_Rot   ;                       R1L--
                                                     ;                  }
         adjnz.b          #-1, R1H, BCDtoHex_2byte_Rotate        ;               R1H--
                                                     ;         }
rts                                                  ; }/* RTS */

.END
```

Decimal - HEX exchange program (1 byte)

```
.SECTION    PROG,         CODE
.GLB  BCDtoHEX_1byte
;===========================================================================
;         Title : Exchange BCD code to HEX code
;         Outline : Exchange BCD 1 byte code to HEX 1 byte code
;         Input : ----------------------->Output :
;         R0L ()                          R0L (HEX code)
;         R0H (BCD code)                  R0H (Undefined)
;         R1L ()                          R1L (Undefined)
;         R1H ()                          R1H (Not used)
;         R2  ()                          R2  (Not used)
;         R3  ()                          R3  (Not used)
;         A0  ()                          A0  (Not used)
;         A1  ()                          A1  (Not used)
;         Stack : Not used
;===========================================================================
BCDtoHEX_1byte:
         mov.b  #0,R0L                                  ; HEX area initialization
         mov.b  #8,R1L                                  ; Set loop cycle
BCDtoHEX_1byte_10:
         shl.b   #-1,R0H                                ; Rotate MSB
         rorc.b  R0L                                    ;
         btst    3,R0H                                  ;
         jeq     BCDtoHEX_1byte_20                      ;
         sub.b   #3,R0H                                 ;
BCDtoHEX_1byte_20:
         adjnz.b         #-1,R1L,BCDtoHEX_1byte_10      ; --> Go to next BCD digit
         rts

.END
```

HEX- decimal exchange program (1 byte)

```
.SECTION    PROG,        CODE
.GLB  HEXtoBCD_1byte
;==========================================================================
;        Title : Exchange HEX code to BCD code
;        Outline : Exchange HEX 1 byte code to BCD 2 byte code
;        Input : ------------------------>Output :
;        R0  ()                          R0  (BCD code)
;        R1L (HEX code)                  R1L (Undefined)
;        R1H ()                          R1H ((Not used)
;        R2  ()                          R2  (Undefined)
;        R3  ()                          R3  ((Not used)
;        A0  ()                          A0  ((Not used)
;        A1  ()                          A1  ((Not used)
;        Stack : Not used
;==========================================================================
HEXtoBCD_1byte:
        mov.w #0,R0                     ; BCD area initialization
HEXtoBCD_1byte_10:
        mov.w R1,R2                     ; Save HEX data
        sub.b  #0Ah,R1L                 ; Check if there is carry
        jltu   HEXtoBCD_1byte_END       ; If no carry, go to end
        add.b  #10H,R0L                 ; Add 1 to higher digit of BCD
        cmp.b  #0A0H,R0L                ; Check if there is carry at 2nd digit
        jne    HEXtoBCD_1byte_10        ; If less than 090H, jump
        add.b  #01H,R0H                 ; Add 1 to highest digit of BCD
        mov.b  #0H,R0L                  ; Reset lower digit of BCD
        jmp    HEXtoBCD_1byte_10        ; Repeat until there is no carry

HEXtoBCD_1byte_END:
        mov.w R2,R1                     ; Restore HEX data
        add.b  R1L,R0L                  ; Set the lower digit of BCD
        rts                             ;

.END
```