

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

RENESAS TECHNICAL NEWS

No.M16C-92-0306

M16C/60 Series, M16C/30 Series, M16C/20 Series Cautions for Writing to DMA Enable Bit in DMAi Control Register

Classification

Corrections and supplementary
explanation of documents

✓Notes

Knowhow

Others

Products Effected

M16C/60 Series

M16C/30 Series

M16C/20 Series

1. Cautions

Read this if you make changes to the DMiCON register while the DMA is enabled or in operation. While the DMAE bit in the DMiCON register is a "1" (the DMAi is in an active state), if you set the DMAE bit to a "1" and a DMA request occurs while your changing the bit, the DMAi will continue operating instead of returning to the initial state^(*). The DMAi will return to the initial state^(*), if a DMA request does not occur at the same time the DMAE bit is written to.

Note 1: The initial state is the state described below.

When the DMAE bit is set to "1", the following occurs:

- The DMAi goes into an active state.
- The value of the pointer (SARi register or DARi register), whichever is specified as the forward pointer by the DSD bit or DAD bit in the DMiCON register, is reloaded into the forward address pointer.
- The value in the transfer counter reload register is reloaded into the TCRi register.

2. Countermeasures

While the DMAE bit in the DMiCON register remains set to "1"(the DMAi is in an active state), after setting the DMAE bit to a "1" (*2) again, verify in your program that the DMAi returned to the initial state by reading the TCRi register that it has the reloaded value. If the DMAi does not return to the initial state, write a "1" (*2) to the DMAE bit repeatedly until the DMAi returns to the initial state.

Note 2: Supplemental Description of DMAS bit When Writing a "1" to DMAE bit

When a "0" is written to the DMAS bit, this bit is set to "0". However, when a "1" is written, the DMAS bit remains unchanged. The DMAS bit is set to "1" only when DMA requests occur.

When a "1" is written to the DMAE bit using a BSET instruction, the operation is as follows:

When executing an RMW(or Read-Modify-Write) instruction such as BSET or OR instruction (refer to "Table 11. Read-Modify-Write Instructions" in Technical News No. M16C -55-0006), the data is read out in byte (or word) units, then the specified bits are modified, then the modified value is written back in byte (or word) units. Accordingly, the unspecified bits are written back as read.

When the DMAS bit is set to "0", if an RMW instruction is executed on the DMiCON register to set the DMAE bit to "1" and a DMA request occurs during the execution, then the DMAS bit behaves as follows:

- (1) When data is read in an RMW instruction: a "0" is read out
- (2) When a DMA request occurs: a "1" is read out
- (3) When writing data with an RMW instruction: "0" is written (DMA request is cleared)

Using an RMW instruction to set the DMAE bit to "1" may clear a DMA request. If this cause a problem, use the OR or MOV instruction to change the DMiCON and set both DMAE and DMAS bits "1". (The DMA request is retained because the DMAS bit does not change.)

The following section 3 describes the countermeasure and shows an example of writing a "1" to the DMAE bit using an OR instruction.

3. Example of Countermeasure

Figure 1 shows a flow chart of countermeasure example.

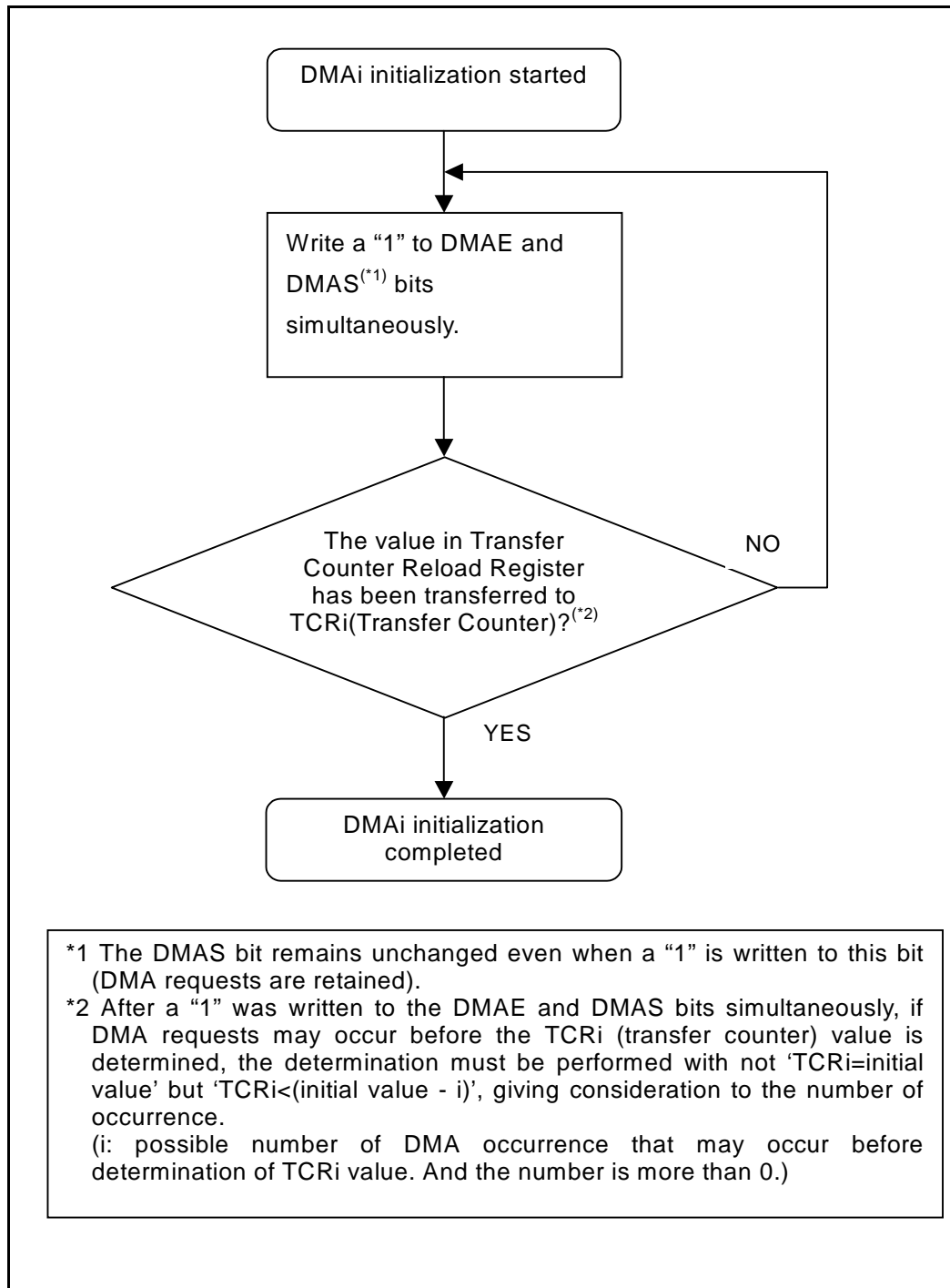


Figure 1. Example of Countermeasure

<Assembler>

```

tcr0_ini_val .blkb      2      ; TCR0 initial value store buffer
...
Initial setting { mov.w   #xxx-1,tcr0_ini_val ; Store TCR0 determination data (initial value-1)("xxx": initial value)
                 mov.w   #xxx,tcr0      ; Set initial value to TCR0
                 ...
DM0CON_REWRITE:
Countermeas- { or.b     #0ch,dm0con    ; Write a 1 to DMAE and DMAS bits simultaneously
ure by       { cmp.w    tcr0,tcr0_ini_val ; If tcr0 < (initial value - 1), then determine DMAi is initialized(*),
software     { jc       DM0CON_REWRITE ; and then exit loop

```

<C language>

```

unsigned short tcr_init_val; // TCR0 initial value store buffer
//((Declare global variables)
...
Initial setting { tcr0_ini_val = xxx; // Store TCR0 initial value ("xxx": initial value)
                 tcr0 = tcr0_ini_val; // Set initial value to TCR0
                 ...
Countermeas- { do {
ure by       { dm0con |= 0x0c; //Write a 1 to DMAE and DMAS bits simultaneously
software     { } while( tcr0 < tcr0_ini_val-1 ) // If tcr0 < (initial value - 1), then determine DMAi is initialized(*),
// and then exit loop

```

*: When a "1" was written to the DMAE and DMAS bits in the DM0CON register simultaneously, DMA requests may occur before determining the TCR0 value. This example uses not 'TCR0<initial value>' but 'TCR0<(initial value - 1)>' so that the DMA will be determined to have been initialized even when the DMA request occurs only once.

4. Groups Affected

Table 1 lists the affected groups.

Table 1. Groups Affected

Series	Group
M16C/60 Series	M16C/60 Group, M16C/61 Group, M16C/62 Group (/62A, /62L, /62M, M3062GF8NFP/GP, /62N, /62P included) M16C/6N Group, M16C/6V Group, M16C/6K Group, M16C/6H Group
M16C/30 Series	M16C/30 Group (/30L included)
M16C/20 Series	M16C/21 Group, M16C/22 Group, M16C/24 Group