Integrated version of restriction notification

This document is a compilation of the restrictions of the corresponding products that have already been reported, and will be utilized in the NEC microcomputer technical document browsing service. All the restrictions as of October 4, 2001 are included.

NEC Microcomputer Technical Information

CP(K), O

32-Bit Microcontroller V850/SB1, SB2 Usage Restrictions		Document No.	SBG-DT-0007-E		1/1
		Date issued	October 4, 2001		
		Issued by	Microcomputer Group Sales Engineering Div. NEC Electron Devices NEC Corporation		
Related documents	User's manual -Hardware: U13850EJ4 User's manual -Architecture: U10243EJ7 Data sheet (V850/SB1 ROM:128,256K): U14734EJ2 Data sheet (V850/SB1 ROM:384,512K): U14893EJ1 Data sheet (V850/SB2 ROM:128,256K): U14780EJ2 Data sheet (V850/SB2 ROM:384,512K): U14894EJ1	Notification classification	√	Usage restriction	
				Upgrade	
				Document modification	
				Other notification	

1. Affected products

   V850/SB1 Series

   μPD70F3033A(Y)/F3032A(Y), 3033A(Y), 3031A(Y), 3032A(Y), 3030A(Y)

   V850/SB2 Series

   μPD70F3035A(Y)/F3037A(Y), 3034A(Y), 3035(Y), 30326(Y), 30307(Y)

   IE-703037-MC-EM1 (emulation board)

2. List of restrictions

   A list of restrictions in the V850/SB1, SB2, including the revision history and detailed information, is described on the following pages.

List of Restrictions in V850/SB1, SB2

1. Product Version

V850/SB1

μPD703030A: Rank K
μPD703031A: Rank K
μPD703032A: Rank K
μPD703033A: Rank K
μPD70F3032A: Rank K, E
μPD70F3033A: Rank K, E

V850/SB2

μPD703034A: Rank K
μPD703035A: Rank K
μPD703036A: Rank K
μPD703037A: Rank K
μPD70F3035A: Rank K, E, P
μPD70F3037A: Rank K, E, P

* The rank is indicated by the letter appearing as the 5th digit from the left in the lot number marked on each product.

Emulation board

IE-703037-MC-EM1

2. Product History

V850/SB1mask ROM version D703030A(Y), 3031A(Y), 3032A(Y), 3033A(Y)

No.	Bugs and Restrictions	Rank
		K
Restriction 1	Restriction on TM2 to TM7 in 16-bit cascade mode	Δ
Restriction 2	Restriction on baud rate setting for variable-length serial interface CSI4	Δ
Restriction 3	Restriction on interrupts	Δ
Restriction 4	Restriction on 16-bit timer one-shot pulse output function	Δ
Restriction 5	Restriction on interrupt servicing acknowledgement after EI instruction	Δ
Restriction 6	Restriction on power save function	Δ
Restriction 7	Restriction related to CLKOUT	–
Restriction 8	Restriction on I/O register illegal access break	–
Restriction 9	Restriction on the initial values of PM6 and PM9	–
Restriction 10	Restriction on P11 when set as output port	–
Restriction 11	Restriction on interrupts in STOP/IDLE mode	–
Bug 12	Bug in initial value of pull-up resistor option register 10 (PU10)	–

V850/SB2 mask ROM version D703034A(Y), 3035A(Y), 3036A(Y), 3037A(Y)

No.	Bugs and Restrictions	Rank
		K
Restriction 1	Restriction on TM2 to TM7 in 16-bit cascade mode	Δ
Restriction 2	Restriction on baud rate setting for variable-length serial interface CSI4	Δ
Restriction 3	Restriction on interrupts	Δ
Restriction 4	Restriction on 16-bit timer one-shot pulse output function	Δ
Restriction 5	Restriction on interrupt servicing acknowledgement after EI instruction	Δ
Restriction 6	Restriction on power save function	Δ
Restriction 7	Restriction related to CLKOUT	–
Restriction 8	Restriction on I/O register illegal access break	–
Restriction 9	Restriction on the initial values of PM6 and PM9	–
Restriction 10	Restriction on P11 when set as output port	–
Restriction 11	Restriction on interrupts in STOP/IDLE mode	–
Bug 12	Bug in initial value of pull-up resistor option register 10 (PU10)	–

V850/SB1 flash memory version D70F3032A(Y), F3033A(Y)

No.	Bugs and Restrictions	Rank
		K, E
Restriction 1	Restriction on TM2 to TM7 in 16-bit cascade mode	Δ
Restriction 2	Restriction on baud rate setting for variable-length serial interface CSI4	Δ
Restriction 3	Restriction on interrupts	Δ
Restriction 4	Restriction on 16-bit timer one-shot pulse output function	Δ
Restriction 5	Restriction on interrupt servicing acknowledgement after EI instruction	Δ
Restriction 6	Restriction on power save function	Δ
Restriction 7	Restriction related to CLKOUT	–
Restriction 8	Restriction on I/O register illegal access break	–
Restriction 9	Restriction on the initial values of PM6 and PM9	–
Restriction 10	Restriction on P11 when set as output port	–
Restriction 11	Restriction on interrupts in STOP/IDLE mode	–
Bug 12	Bug in initial value of pull-up resistor option register 10 (PU10)	–

V850/SB2 flash memory version D70F3035A(Y), F3037A(Y)

No.	Bugs and Restrictions	Rank
		K, E, P
Restriction 1	Restriction on TM2 to TM7 in 16-bit cascade mode	Δ
Restriction 2	Restriction on baud rate setting for variable-length serial interface CSI4	Δ
Restriction 3	Restriction on interrupts	Δ
Restriction 4	Restriction on 16-bit timer one-shot pulse output function	Δ
Restriction 5	Restriction on interrupt servicing acknowledgement after EI instruction	Δ
Restriction 6	Restriction on power save function	Δ
Restriction 7	Restriction related to CLKOUT	–
Restriction 8	Restriction on I/O register illegal access break	–
Restriction 9	Restriction on the initial values of PM6 and PM9	–
Restriction 10	Restriction on P11 when set as output port	–
Restriction 11	Restriction on interrupts in STOP/IDLE mode	–
Bug 12	Bug in initial value of pull-up resistor option register 10 (PU10)	–

Emulation board

No.	Bugs and Restrictions	Rank			
		A	B	C	D
Restriction 1	Restriction on TM2 to TM7 in 16-bit cascade mode	Δ	Δ	Δ	Δ
Restriction 2	Restriction on baud rate setting for variable-length serial interface CSI4	Δ	Δ	Δ	Δ
Restriction 3	Restriction on interrupts	Δ	Δ	Δ	Δ
Restriction 4	Restriction on 16-bit timer one-shot pulse output function	Δ	Δ	Δ	Δ
Restriction 5	Restriction on interrupt servicing acknowledgement after EI instruction	Δ	Δ	Δ	Δ
Restriction 6	Restriction on power save function	Δ	Δ	Δ	Δ
Restriction 7	Restriction related to CLKOUT	Δ	Δ	Δ	Δ
Restriction 8	Restriction on I/O register illegal access break	–	Δ	Δ	Δ
Restriction 9	Restriction on the initial values of PM6 and PM9	Δ	Δ	Δ	Δ
Restriction 10	Restriction on P11 when set as output port	Δ	Δ	Δ	Δ
Restriction 11	Restriction on interrupts in STOP/IDLE mode	Δ	Δ	Δ	Δ
Bug 12	Bug in initial value of pull-up resistor option register 10 (PU10)	×	×	×	√

√: Corrected, Δ: Restriction will apply in future, ×: Bug occurs, –: Not relevant

Restriction 1. Restriction on TM2 to TM7 in 16-bit cascade mode

[Description]

When two 8-bit timers are cascaded and used as a 16-bit timer, and the value of the compare register is to be changed, stop both the 8-bit cascaded timers before changing the value of the compare register. If the timers are not stopped, the value of the higher 8 bits becomes undefined.

Example) If the value of CR45 is to be changed when TM4 and TM5 are cascaded, first set TCE4 and TCE5 to 0, stop TM4 and TM5 counting, and then change the value of CR45.

Restriction 2. Restriction on baud rate setting for variable-length serial interface CSI4

[Description]

Do not set the baud rate of the variable-length serial interface CSI4 to a transmission speed faster than the operating clock of the CPU. Data is not transmitted correctly under these conditions.

Example)

When operating at $f_{cpu} = f_{xx}/2$, the baud rate $f_{xx}/2$ cannot be selected.
When operating at $f_{cpu} = f_{xx}/4$, the baud rates $f_{xx}/2$ and $f_{xx}/4$ cannot be selected.
When operating at $f_{cpu} = f_{xx}/8$, the baud rates $f_{xx}/2$, $f_{xx}/4$ and $f_{xx}/8$ cannot be selected.

When operating at $f_{cpu} = f_{xx}$, all baud rates can be selected.

This restriction does not apply to CSI0 to CSI3, or UART0 to UART1.

Restriction 3. Restriction on interrupts

[Description]

If the three conditions below occur at the same time while interrupts are enabled (EI), an interrupt that should only occur once occurs twice.

(1) A bit manipulation instruction (set1, clr1, not1, tst1) is executed on an interrupt request flag (xxIFn) of an interrupt control register (xxICn).
(2) Interrupt servicing occurs involving the hardware of the same register as in (1).
(3) DMA is started while executing the above bit manipulation instruction.

Remark xx: Identifier of each peripheral unit (WDT, P, WTI, TM, CS, SER, SR, ST, AD, DMA, WT)
n: Peripheral unit number (0 to 7)

When the above conditions (1) to (3) match and interrupt servicing begins, the interrupt request flag that should be reset by hardware is not reset, so that after returning from the interrupt servicing routine (RETI instruction), the interrupt servicing is executed again. (See Figures 3-1 and 3-2.)

Example) During a bit manipulation on the interrupt request flag (CSIF0) of the CSIC0 register using the clr1 instruction, the non-masked INTCSI0 interrupt occurs, which then conflicts with the start of DMA. In this case, the INTCSI0 interrupt servicing is executed twice.

[Workaround]

(1) Issue the DI instruction before executing a bit manipulation instruction on the interrupt request flags (xxIFn) of an interrupt control register (xxICn) by software. Then issue an EI instruction afterwards, and do not execute interrupt servicing immediately after the bit manipulation instruction. (See Figure 3-3.)

(2) When interrupt servicing begins, interrupts are disabled by hardware (the DI state), so clear the interrupt request flag in every interrupt processing routine before executing the EI instruction. (See Figure 3-4.)

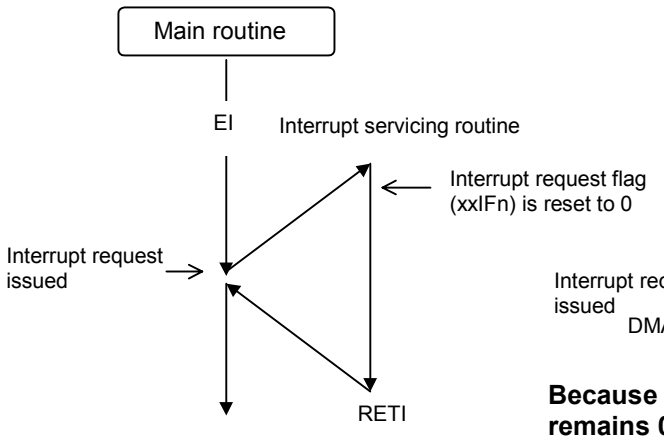

Figure 3-1. Normal Interrupt Servicing

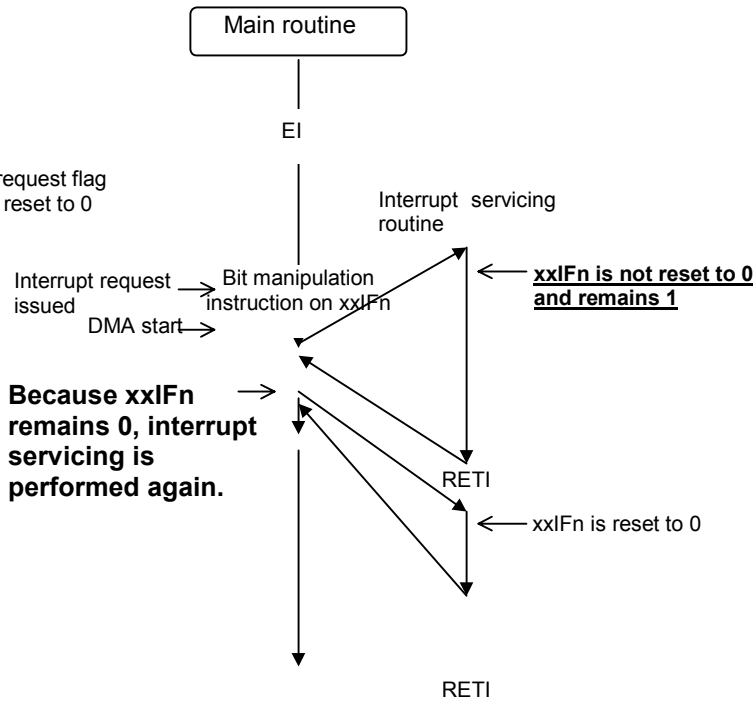

Figure3-2. Interrupt Servicing Causing This Bug

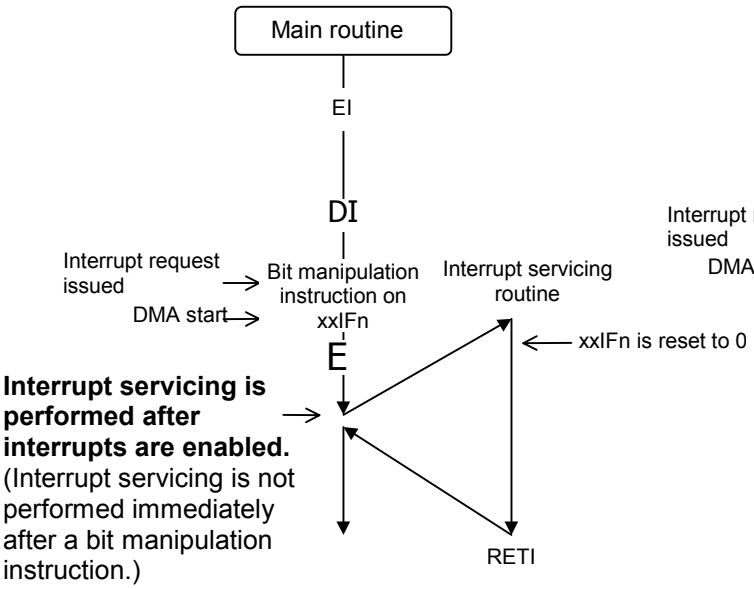

Figure 3-3. Workaround (1)

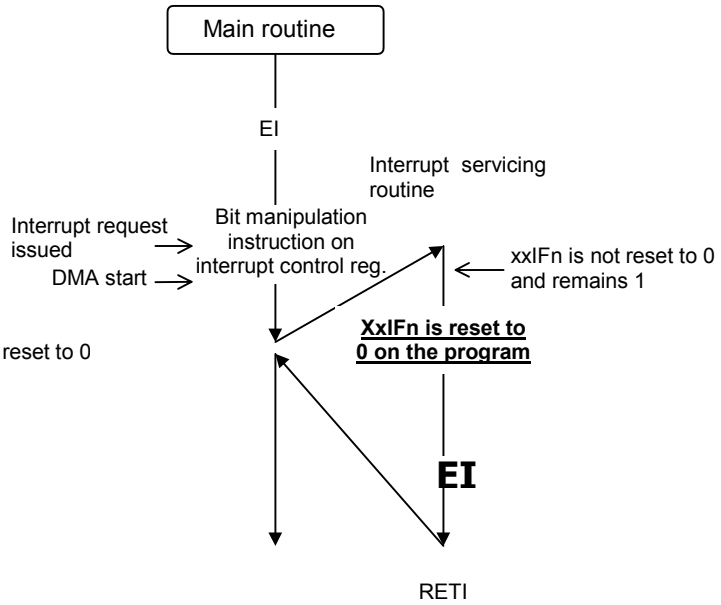

Figure 3-4. Workaround (2)

Remark xx: Identifier of each peripheral unit (WDT, P, WTI, TM, CS, SER, SR, ST, AD, DMA, WT)
n: Peripheral unit number (0 to 7)

Restriction 4. Restriction on 16-bit timer one-shot pulse output function

[Description]

When using the one-shot pulse function of timers 0, 1, and 7 via a software trigger, the level of the T1 pin or its alternate function port cannot be changed.

Because the external trigger is also enabled in this case, the trigger will inadvertently clear & start even if the level of the T1 pin or its alternate function port is changed, causing a pulse to be output at an unintended timing.

Restriction 5. Restriction on interrupt servicing acknowledgement after EI instruction

[Description]

In this product, at least 7 clocks are required as determination time between the generation of an interrupt and its acknowledgement. Because instructions continue to be executed in this period, if the DI instruction (interrupt disable) is executed, interrupts become disabled. This causes all interrupts to be held pending until the re-execution of the EI instruction (interrupt enable).

Since this determination time is also required when the EI instruction is executed, at least 7 clocks must be allowed before interrupts can be acknowledged after execution of the EI instruction. Consequently, if the DI instruction is executed before these 7 clocks have elapsed, interrupts will be held pending and not acknowledged.

To ensure proper acknowledgement of interrupts therefore, insert an instruction (other than those below) of at least 7 execution clocks between the EI and DI instructions.

- IDLE/STOP mode setting
- EI, DI instruction
- RETI instruction
- LDSR instruction (for PSW register)
- Access to interrupt control register (xxICn)

Example) When EI instruction processing is invalid

[Program example]

```
        DI
          :             ; MK flag = 0 (interrupts enabled)
          :             ; Interrupt request generated (IF flag = 1)
        EI
        JR  LP1
          :             7 clocks have not elapsed between EI and DI instructions (3 clocks)
          :
        LP1:
        DI              ; Interrupt request is not acknowledged.
          :
```

[Workaround example]

```
    DI
     :              ; MK flag = 0 (interrupts enabled)
     :              ; Interrupt request generated (IF flag = 1)
    EI
    NOP             ; 1 system clock
    NOP             ; 1 system clock
    NOP             ; 1 system clock
    NOP             ; 1 system clock
    JR   LP         ; 3 system clocks (branch to LP1 routine)
     :
    LP1:
    DI              ; Interrupt servicing is executed at 8th clock cycle after the EI instruction
     :
```

Restriction 6. Restriction on power save function

[Description]

If the affected products are used under the following conditions, a discrepancy may occur between the address indicated by the program counter (PC) and the address at which the instruction is actually read following the release of a power save mode.

This may result in the CPU ignoring a 4- or 8-byte instruction from between 4 bytes and 16 bytes after an instruction is executed to write to the PSC register, which could in turn result in the execution of an erroneous instruction. Note that this bug only occurs if all of conditions (1) to (3) below are met.

[Conditions]

(1) A power save mode (IDLE or STOP) is set while an instruction is being executed on the external ROM.

(2) The power save mode is released by an interrupt.

(3) The next instruction is executed while interrupts are in a pending state following the release of the power save mode.

Note that interrupts are held pending under any of the following conditions.

<1> The NP flag of the PSW register is 1. (NMI servicing in progress/set by software)

<2> The ID flag of the PSW register is 1. (Interrupt servicing in progress/DI instruction/set by software)

<3> The EI (interrupt enable) state had been set during interrupt servicing to enable multiple interrupt servicing, but was released by an interrupt with the same or lower priority than the interrupt being serviced.

The operation of the bug is shown below using the power save mode setting example from the user's manual.

(rD: PSC setting value, rX: Value written to PSW, rY: Value written back to PSW, assuming PSW has been set)

```
ldsr  rX,5          ; Sets PSW to the value of rX
st.b  r0,PRCMD[r0]      ; Writes to PRCMD
st.b  rD,PSC[r0]        ; Sets the PSC register        (PSC setting)
ldsr  rY,5              ; Returns the value of PSW     (After 4 bytes)
nop                     ; 2 to 5 NOP instructions      (After 6 bytes)
nop                     ;                              (After 8 bytes)
nop                     ;                              (After 10 bytes)
nop                     ;                              (After 12 bytes)
nop                     ;                              (After 14 bytes)
(Next instruction)      ;                              (After 16 bytes)
```

Bug occurs here
<1>Discrepancy with PC
<2>Instructions ignored

[Workaround]

(1) Do not use a power save mode (IDLE or STOP) while an instruction is being executed on the external ROM.

(2) If it is necessary to use a power save mode (IDLE or STOP) while an instruction is being executed on the external ROM, take the software workaround shown below.

<1> Insert 6 NOP instructions 4 bytes after an instruction that writes to the PSC register.

<2> Insert the br $+2 instruction after the NOP instructions to eliminate the PC discrepancy.

[Program example]

(rD: PSC setting value, rX: Value written to PSW, rY: Value written back to PSW, assuming PSW has been set)

```
ldsr  rX,5              ; Sets PSW to the value of rX
st.b  r0,PRCMD[r0]      ; Writes to PRCMD
st.b  rD,PSC[r0]            ; Sets the PSC register
ldsr  rY,5              ; Returns the value of PSW

nop                     ; <1> 6 or more NOP instructions
nop
nop
nop
nop
nop

br $+2                  ; <2> Eliminates PC discrepancy
```

Restriction 7. Restriction related to CLKOUT

[Description]

Output of CLKOUT is not disabled even if the reset signal is input.

Restriction 8. Restriction on I/O register illegal access break

[Description]

There are some addresses that are reserved areas but for which an I/O register illegal access break cannot be detected. These addresses are shown in Table 8-1.

Table 8-1. Register Addresses That Cannot Detect Illegal Access Break

Access Address	Emulation of SB1 (Except -Y Model)	Emulation of SB1 (-Y Model)	Emulation of SB2 (Except -Y Model)	Emulation of SB2 (-Y Model)
0xFFF138	×	√	×	√
0xFFF340	×	√	×	√
0xFFF342	×	√	×	√
0xFFF344	×	√	×	√
0xFFF346	×	√	×	√
0xFFF348	×	√	×	√
0xFFF34A	×	√	×	√
0xFFF34C	×	√	×	√
0xFFF350	×	√	×	√
0xFFF352	×	√	×	√
0xFFF354	×	√	×	√
0xFFF356	×	√	×	√
0xFFF358	×	√	×	√
0xFFF35A	×	√	×	√
0xFFF35C	×	√	×	√
0xFFF142	×	×	√	√
0xFFF144	×	×	√	√
0xFFF3E0	×	×	√	√
0xFFF3E2	×	×	√	√
0xFFF3E4	×	×	√	√
0xFFF3E6	×	×	√	√
0xFFF3E8	×	×	√	√
0xFFF3EA	×	×	√	√
0xFFF3EC	×	×	√	√
0xFFF3EE	×	×	√	√
0xFFF3F0	×	×	√	√
0xFFF3F2	×	×	√	√
0xFFF3F4	×	×	√	√
0xFFF3F6	×	×	√	√
0xFFF3F8	×	×	√	√

√: I/O register illegal break is detected.

×: I/O register illegal break is not detected.

Restriction 9. Restriction on the initial values of PM6 and PM9

[Description]

The read values of PM6 and PM9 after reset are different from those of the real chip.

Table 9-1. R Values of PM6 and PM9 After Reset

I/O Register Name	Address	Emulator Read Value	Real Chip Read Value
PM6	0xFFF02C	FF	3F
PM9	0xFFF032	FF	7F

Restriction 10. Restriction on P11 when set as output port

[Description]

If P11 is read when it is set as output port, the pin status is read instead of the port register value.

Restriction 11. Restriction on interrupts in STOP/IDLE mode

[Description]

The emulator is dead-locked under the following conditions.

If the device is shifted to the STOP/IDLE mode while the interrupt request flag is set by an interrupt that is not masked.

[Workaround]

Be sure to clear the non-masked interrupt request flag before shifting to the STOP/IDLE mode. If the device is inadvertently shifted before clearing the flag, execute "Run" → "Stop" or click the  button on the debugger. The program is forcibly terminated. Then, execute "Run" → "CPU Reset" or click the  button.

Bug 12. Bug in initial value of pull-up resistor option register 10 (PU10)

[Description]

On-chip pull-up resistors are connected to ports 100 to 107 following debugger startup or after a reset (including the time from immediately after emulator power-on to debugger startup), even though the value of pull-up resistor option register 10 (PU10) is displayed as 00h (on-chip pull-up resistors not connected) on the debugger at that time.

[Workaround]

Write 00h to pull-up resistor option register 10 (PU10) during initialization immediately after a reset or in the I/O register window.

This workaround has been implemented in products with control code D and later.