CUSTOMER NOTIFICATION Power Save Function of V850 Series Emulator IE-703002-MC Notes on Use	SUD-T-5132-E January 29, 2001 Yoichi Hirasawa, Expert Microcomputer Engineering Dept. Solution Engineering Div. NEC Electron Devices NEC Corporation

CP(K), O

Be sure to read this document before using the product.

1. Affected products
   - IE-703002-MC

2. Bug details

   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.

[Bug operation]

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

3. Countermeasures

Please implement one of the following countermeasures.

Although the product can technically be modified to correct this bug, in consideration of the specification modifications and risks that would accompany such a modification, caused by problems in the timing relationship between the CPU and the bus controller, customers are requested to treat this bug as a usage restriction. We apologize for any inconvenience this may cause.

[Countermeasures]

(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 countermeasures 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
```

• Example of program with bug

Address	Data	DisAsm
00000000	80070010	jr 0x1000
•	•	•
00001000	20560000	movea 0x0, r0, r10
00001004	60576000	st.h r10, 0x60[r0]
00001008	20560000	movea 0x0, r0, r10
0000100C	605762f0	st.h r10, BCC
00001010	20560700	movea 0x7, r0, r10
00001014	40574cf0	st.b r10, MM
00001018	c007c2f0	set1 0x0, EGN0
0000101C	c007c0f0	set1 0x0, EGP0
00001020	40561000	movhi 0x10, r0, r10
00001024	204e8000	movea 0x80, r0, r9
00001028	e92f2000	ldsr r9, 0x5
0000102C	80070600	jr 0x1032
00001030	0000	nop
00001032	400770f1	st.b r0, PRCMD
00001036	c01770f0	set1 0x2, PSC
0000103A	0000	nop
0000103C	0000	nop
0000103E	0000	nop
00001040	0000	nop
00001042	010a	mov 0x1, r1
00001044	0212	mov 0x2, r2
00001046	031a	mov 0x3, sp
00001048	0422	mov 0x4, gp
0000104A	052a	mov 0x5, tp
0000104C	0632	mov 0x6, r6
0000104E	073a	mov 0x7, r7
00001050	0000	nop
00001052	0000	nop

• Example of trace data with bug

Fram	Time	Address	Data	Status	Address	Data	Status	ExtProbe	DisAsm
00000	20	00000000	80070010	BRM1				00	jr 0x1000
00001	29	00001000	20560000	BRM1				00	movea 0x0, r0, r10
00002	11	00001004	605760F0	M1	00FFF060	0000	W	00	st.h r10, DWC
00003	8	00001008	20560000	M1				00	movea 0x0, r0, r10
00004	8	0000100C	605762F0	M1	00FFF062	0000	W	00	st.h r10, BCC
00005	6	00001010	20560700	M1				00	movea 0x7, r0, r10
00006	3	00001014	40574CF0	M1	00FFF04C	07	W	00	st.b r10, MM
00007	3				00FFF0C2	00	R	00	
00008	5	00001018	C007C2F0	M1	00FFF0C2	01	W	00	set1 0x0, EGN0
00009	3				00FFF0C0	00	R	00	
00010	7	0000101C	C007C0F0	M1	00FFF0C0	01	W	00	set1 0x0, EGP0
00011	6	00001020	40561000	M1				00	movhi 0x10, r0, r10
00012	2	00001024	204E8000	M1				00	movea 0x80, r0, r9
00013	1	00001028	E92F2000	M1				00	ldsr r9, 0x5
00014	3	0000102C	80070600	M1				00	jr 0x1032
00015	21	00001032	400770F1	BRM1	00FFF170	00	W	00	st.b r0, PRCMD
00016	3				00FFF070	C0	R	00	
00017	8	00001036	C01770F0	M1	00FFF070	C4	W	00	set1 0x2, PSC
00018	1	0000103A	00000000	M1				00	nop
00019	1	0000103C	00000000	M1				00	nop
00020	65535	0000103E	00000422	M1				00	nop
00021	5	00001040	0422052A	M1				00	mov 0x4, gp ←+
00022	1	00001042	052A0632	M1				00	mov 0x5, tp ←+ Instruction is illegally executed
00023	5	00001044	0632073A	M1				00	mov 0x6, r6 ←+
00024	1	00001046	073A0000	M1				00	mov 0x7, r7 ←+
00025	5	00001048	00000000	M1				00	nop
00026	1	0000104A	00000000	M1				00	nop
00027	5	0000104C	00000000	M1				00	nop
00028	1	0000104E	00000000	M1				00	nop
00029	5	00001050	00000000	M1				00	nop
00030	1	00001052	00000000	M1				00	nop

• Example of bug-removed program

```
Address          Data          DisAsm
_____________________________________
00000000          80070010     jr 0x1000
    •                 •           •
00001000          20560000     movea 0x0, r0, r10
00001004          60576000     st.h r10, 0x60[r0]
00001008          20560000     movea 0x0, r0, r10
0000100C          605762f0     st.h r10, BCC
00001010          20560700     movea 0x7, r0, r10
00001014          40574cf0     st.b r10, MM
00001018          c007c2f0     set1 0x0, EGN0
0000101C          c007c0f0     set1 0x0, EGP0
00001020          40561000     movhi 0x10, r0, r10
00001024          204e8000     movea 0x80, r0, r9
00001028          e92f2000     ldsr r9, 0x5
0000102C          80070600     jr 0x1032
00001030          0000         nop
00001032          400770f1     st.b r0, PRCMD
00001036          c01770f0     set1 0x2, PSC
0000103A          0000         nop
0000103C          0000         nop
0000103E          0000         nop          ← Insert 6 nop instructions
00001040          0000         nop          ←
00001042          0000         nop          ←
00001044          0000         nop          ←
00001046          0000         nop          ←
00001048          0000         nop          ←
0000104A          9505         br 0x104c    ← br instruction to solve discrepancy with PC
0000104C          0000         nop
0000104E          0000         nop
00001050          0000         nop
00001052          0000         nop
```

• Example of bug-removed trace data

```
Fram  Time  Address   Data      Status  Address    Data  Status  ExtProbe  DisAsm
_____________________________________________________________________________
00000   20  00000000  80070010  BRM1                               00        jr 0x1000
00001   29  00001000  20560000  BRM1                               00        movea 0x0, r0, r10
00002   11  00001004  605760F0  M1      00FFF060   0000    W       00        st.h r10, DWC
00003    8  00001008  20560000  M1                                 00        movea 0x0, r0, r10
00004    8  0000100C  605762F0  M1      00FFF062   0000    W       00        st.h r10, BCC
00005    6  00001010  20560700  M1                                 00        movea 0x7, r0, r10
00006    3  00001014  40574CF0  M1      00FFF04C   07      W       00        st.b r10, MM
00007    3                             00FFF0C2   00      R       00
00008    5  00001018  C007C2F0  M1      00FFF0C2   01      W       00        set1 0x0, EGN0
00009    3                             00FFF0C0   00      R       00
00010    7  0000101C  C007C0F0  M1      00FFF0C0   01      W       00        set1 0x0, EGP0
00011    6  00001020  40561000  M1                                 00        movhi 0x10, r0, r10
00012    2  00001024  204E8000  M1                                 00        movea 0x80, r0, r9
00013    1  00001028  E92F2000  M1                                 00        ldsr r9, 0x5
00014    3  0000102C  80070600  M1                                 00        jr 0x1032
00015   21  00001032  400770F1  BRM1    00FFF170   00      W       00        st.b r0, PRCMD
00016    3                             00FFF070   C0      R       00
00017    8  00001036  C01770F0  M1      00FFF070   C4      W       00        set1 0x2, PSC
00018    1  0000103A  00000000  M1                                 00        nop
00019    1  0000103C  00000000  M1                                 00        nop
00020  65535 0000103E  00000000  M1                                 00        nop
00021    1  00001040  00009505  M1                                 00        nop
00022    7  00001042  95050000  M1                                 00        br 0x1044  ← 0x104Ah instruction is mistaken as
                                                                              fetch from 0x1042h
00023   12  00001044  00000000  BRM1                               00        nop
00024    1  00001046  00000000  M1                                 00        nop
00025    1  00001048  00009505  M1                                 00        nop
00026    7  0000104A  95050000  M1                                 00        br 0x104c  ← br instruction to solve discrepancy with PC
00027   12  0000104C  00000000  BRM1                               00        nop
00028    1  0000104E  00000000  M1                                 00        nop
```