

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

<b>Concerned Products:</b>	<b>Customer Notification</b>	Date: Nov. 8, 1999
<i>μPD703033</i> <i>μPD703033Y</i> <i>μPD70F3033</i> <i>μPD70F3033Y</i>		NEC-Electronics (Europe) GmbH EAD –Technical Product Support
<b>Bug Report</b>		Source Doc: HW 260199 HW 080299 SBG-T-1719 SBG-T-1777
		Author: H.-W. Hoefft, Udo Wenzel
Date of initial issue: Feb 19 <sup>th</sup> 99		Doc. No.: TPS-HE-B-2120

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

**(A) BUG LIST**

Bug No.	Outline	μPD703033 μPD703033Y μPD70F3033 μPD70F3033Y			
		ES1.1	V2.0		
1	Output on port pin P33	✋	✓		
5	Sub operation current	✋	✓		
2	TM 2-5 operational defect	✋	✓		
4	IIC-Bug restart	✋	✓		
5	STOP conflict	✋	✓		
6	ROM-CPU interface	✋	✓		
7	CSI4 baud rate restriction	🚫	🚫		
8	Cascaded 8-bit timers	🚫	🚫		
9	Double interrupt execution	🚫	🚫		
10	Limitation on frequency / bus-voltage	👉	👉		

- ✓: No problem
- ✋: Bug (will be corrected by next version upgrade)
- 🚫: Bug (restriction, not corrected by version upgrade)
- 👉: Bug/restriction future status pending

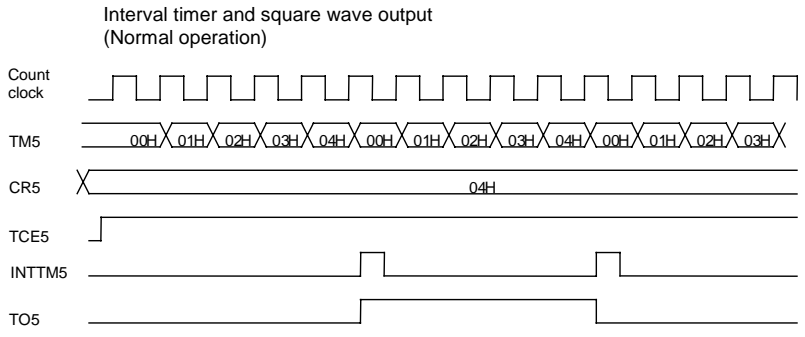
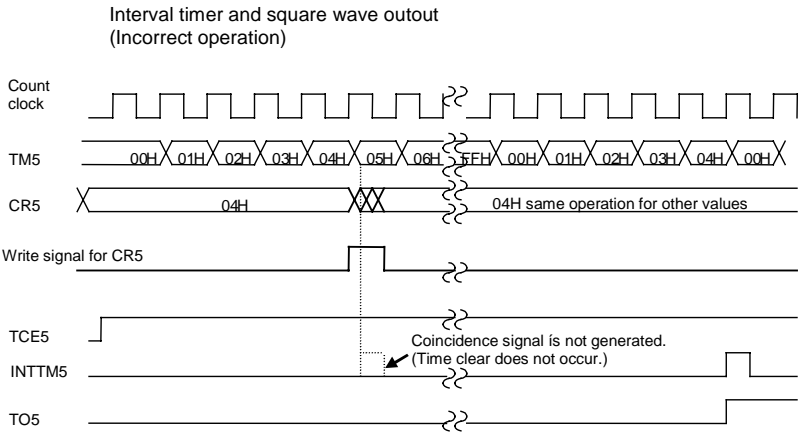
V850/SB1 Mask	μPD703033 μPD703033Y
V850/SB1 Flash	μPD70F3033 μPD70F3033Y

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

**(B) BUG DESCRIPTION**

Bug No.	Outline	Description									
1	Output on port pin P33	<p><u>Details</u></p> <p>When P33 is used for SO4, P31 does not work correctly as output port.</p> <p><u>Workaround</u></p> <p>Use the port pin P31 only as inport pin, when port pin P33 is used for SO4.</p>									
2	Sub operation current	<p><u>Details</u></p> <p>The current consumption during subclock mode is higher than target specification:</p> <table data-bbox="539 757 1145 851"> <tr> <td>subclock</td> <td>normal operation</td> <td>typ 280µA</td> </tr> <tr> <td></td> <td>HALT mode</td> <td>typ 190µA</td> </tr> <tr> <td></td> <td>IDLE mode</td> <td>typ 180µA</td> </tr> </table>	subclock	normal operation	typ 280µA		HALT mode	typ 190µA		IDLE mode	typ 180µA
subclock	normal operation	typ 280µA									
	HALT mode	typ 190µA									
	IDLE mode	typ 180µA									

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

<p>3</p>	<p>TM 2-5 operational defect</p>	<p><u>Details:</u></p> <p>When used in interval timer mode          If the compare register (CR20-50) is written when the timer register (TM2-5) and compare register (CR20-50) coincide, a coincidence signal is not generated. For this reason, the software must be designed to prevent timer coincidence and writing the compare register occurring at the same time. (Example: use a Vector interrupt)</p> <p>When used in square wave output mode          If the compare register (CR20-50) is written when the timer register (TM2-5) and compare register (CR20-50) coincide, a coincidence signal is not generated and the output waveform does not invert. For this reason, the software must be designed to prevent timer coincidence and writing the compare register occurring at the same time. (Example: use a Vector interrupt)</p> <div style="text-align: center;"> <p>Interval timer and square wave output (Normal operation)</p>  </div> <div style="text-align: center;"> <p>Interval timer and square wave output (Incorrect operation)</p>  </div> <div style="text-align: center; border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>The timing chart above is intend to show the general operation and does not show precise timing</p> </div>
----------	----------------------------------	---

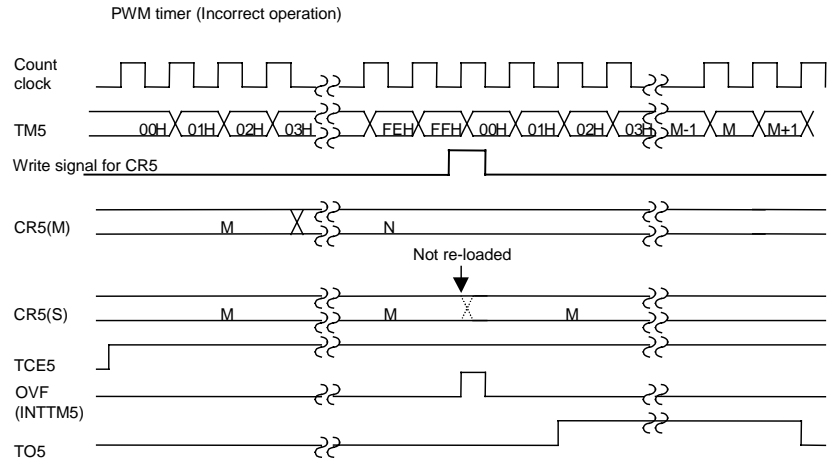
Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

<p>3</p>	<p>TM 2-5 operational defect</p>	<p>When used in PWM output mode</p> <ul style="list-style-type: none"> <li>- If the CR20-50 (master register) is re-written at the same time as the timer register overflows when the CR20-50 (master register) is written, the expected data sometimes cannot be transferred to the CR20-50 (slave register). (Data value is undefined) The software must be designed to prevent the overflow occurring at the same time as writing in the compare register.</li> </ul> <p>PWM timer (Incorrect operation)</p> <p>Count clock</p> <p>TM5</p> <p>Write signal for CR5</p> <p>CR5(M)</p> <p>CR5(S)</p> <p>TCE5</p> <p>OVF (INTTM5)</p> <p>TO5</p> <p>Not re-loaded</p> <p>PWM timer (Incorrect operation)</p> <p>Count clock</p> <p>TM5</p> <p>Read signal for CR5</p> <p>CR5(M)</p> <p>CR5(S)</p> <p>TCE5</p> <p>Coincidence signal</p> <p>OVF (INTTM5)</p> <p>TO5</p> <p>Coincidence signal does not occur</p> <p>Does not become inactive level</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>The Timing chart above is intended to show the general operation, and does not show precise timing.</p> </div>
----------	----------------------------------	--

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

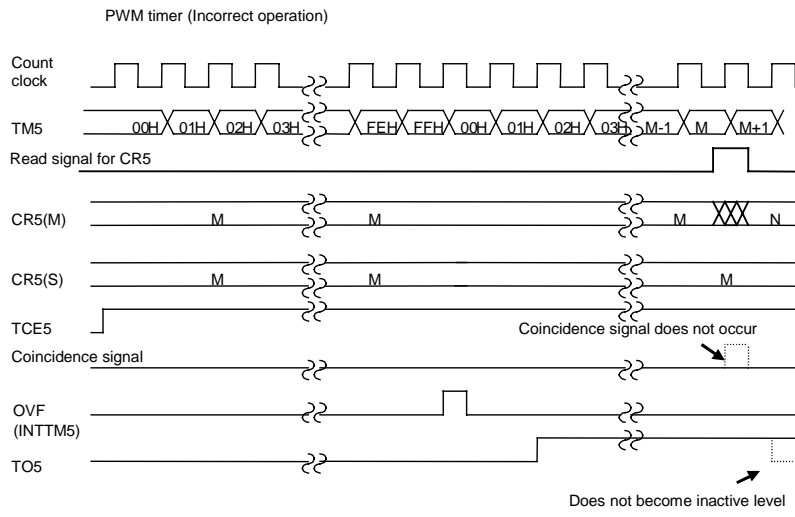
3 TM 2-5 operational defect

- If the CR20-50 register is read when an overflow occurs, the CR20-50 master register will not transfer data to the CR20-50 slave register. The data is transferred when the next overflow occurs.



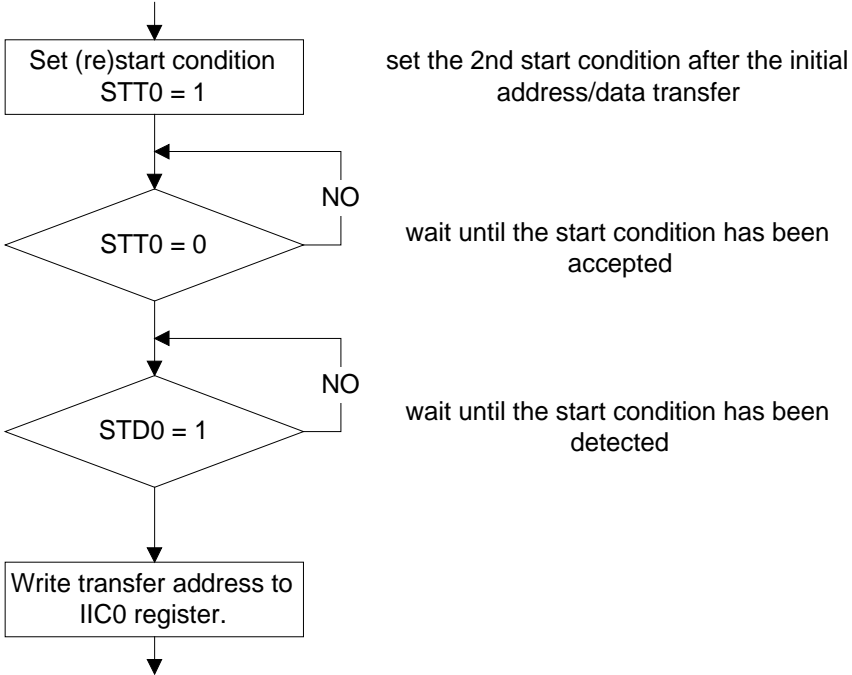
The Timing chart above is intended to show the general operation, and does not show precise timing.

- If the compare register (CR20-50) is written when the timer register (TM2-5) and compare register (CR20-50 [slave register]) coincide, a coincidence signal is not generated and the inactive level is not attained.

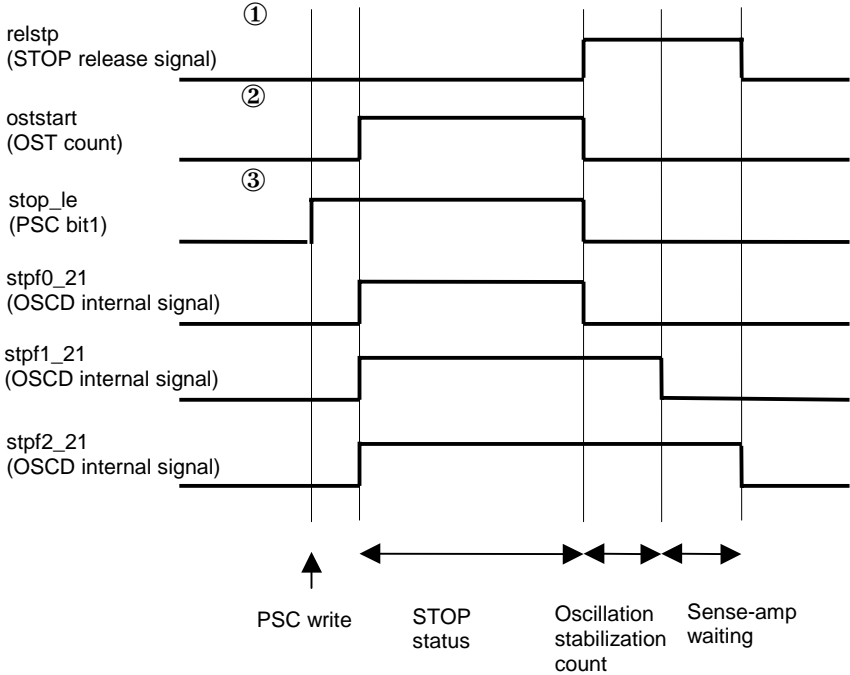


The Timing chart above is intended to show the general operation, and does not show precise timing.

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

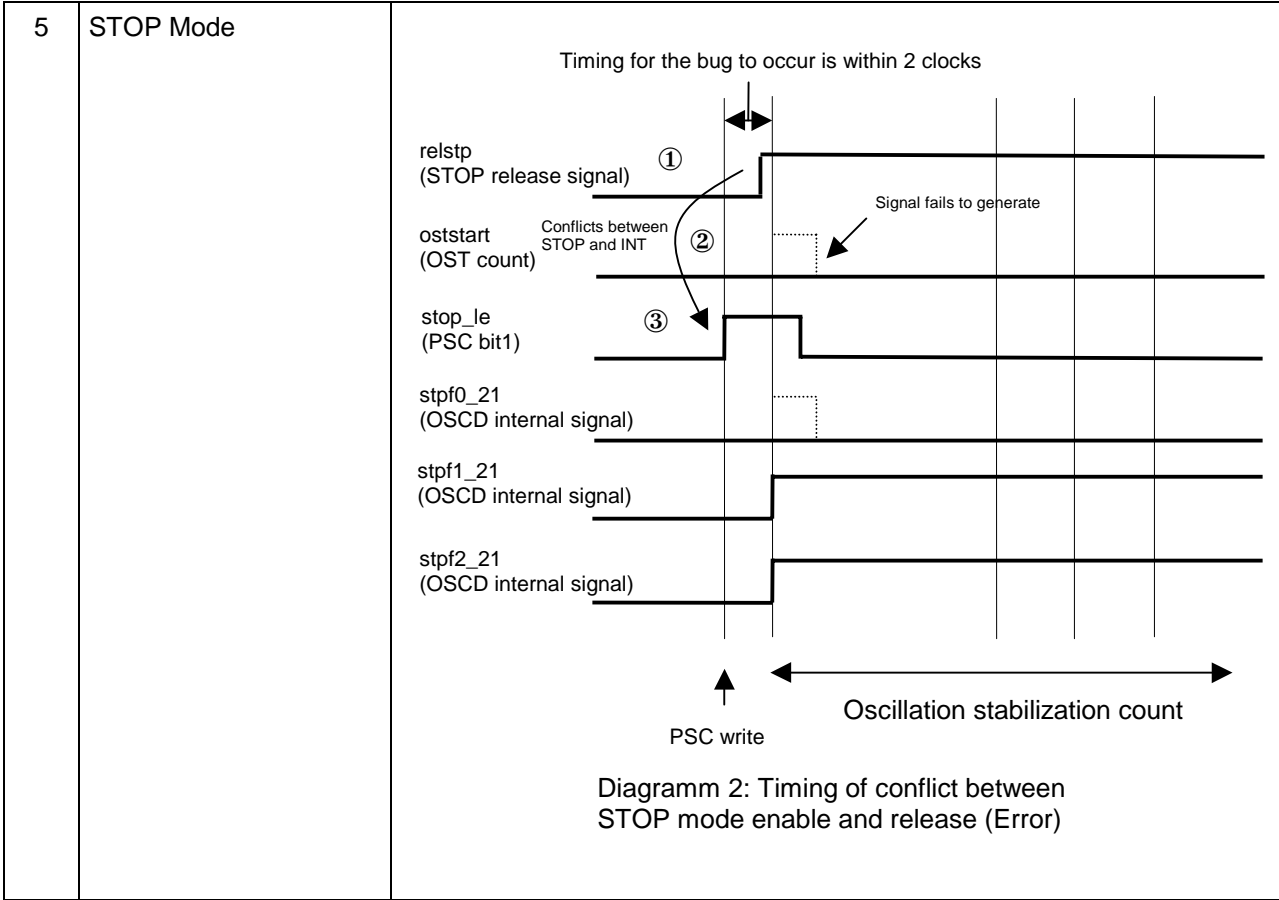
Bug No.	Outline	Description
4	Using the IIC restart condition causes a faulty address transfer.	<p><u>Details</u>                      If the transfer address is set to the shift register (IIC0) immediately after the (re)start is set (STT = 1), the transfer address will be sent simultaneously by rising edge of the serial clock (SCL).                      The serial clock is at low level normally before a restart condition will be set, due to the preceded address/data transfer. To generate a (re)start condition the serial clock (SCL) has to be first on high level (recessive state). By falling edge of the SDA line, when SCL has high level, the (re)start condition can be detected. Afterwards the SCL line will fall to low level and the slave address can be transferred by normal operation.                      Now, if the slave address is written into the shift register (IIC0) immediately after the start condition trigger (STT0) is set, the rising edge on the SCL line resulting in data output before the start condition is set. This causes the bug.</p> <p><u>Workaround</u>                      The transfer address must be set after the confirmation of start condition detection (STT=0, STD = 1).</p>  <pre>                     graph TD                         Start([Start]) --&gt; Set[Set (re)start condition STT0 = 1]                         Set --&gt; STT0{STT0 = 0}                         STT0 -- NO --&gt; Wait1[wait until the start condition has been accepted]                         Wait1 --&gt; STT0                         STT0 -- YES --&gt; STD0{STD0 = 1}                         STD0 -- NO --&gt; Wait2[wait until the start condition has been detected]                         Wait2 --&gt; STD0                         STD0 -- YES --&gt; Write[Write transfer address to IIC0 register.]                         Write --&gt; End([End])                     </pre>

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

Bug No.	Outline	Description
5	STOP Mode	<p>The CPU becomes deadlocked, if the timing of a STOP mode enable and a Stop mode release are in conflict</p> <p><u>Details</u></p> <p>With the V850/SA1, the STOP mode is enable by writing (stop_le<sup>③</sup>) to the PSC register (Power Save Control register). To release the Stop mode, the count initialization signal (oststart:<sup>②</sup>) is generated to start the WDT (watchdog timer) that counts out the oscillation stabilization time (refer to diagram 1). However, if the stop mode release signal (relstp:<sup>①</sup>) because of an interrupt, etc. conflicts with writing to the PSC register (the timing for the bug to occur is within 2 clocks), then the oststart signal is not generated and WDT does not start counting. This means that the microcomputer remains waiting for the oscillation stabilization time to elapse, and even if the stop mode release signal is generated, STOP mode cannot be released (refer to diagram 2).</p> <p><u>Workaround</u></p> <p>As a temporary measure it is suggest that you use the IDLE mode or sub-IDLE mode instead of the STOP mode.</p>  <p style="text-align: center;">Diagramm1: STOP release operation timing (Normal)</p>



Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).



Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

Bug No.	Outline	Description
6	ROM – CPU interface problem	<p><u>Details</u></p> <p>Instructions located at address corresponding to the border between Flash macros (Flash areas) (address 1FFFCH) will not be fetched into the CPU correctly if a previous instruction located between address 1FFEE and 1FFF8 delayed the pipeline in any stage.</p> <p>When the execution of an instruction takes more than one clock cycle (for example read/write access to external memory or peripheral I/O; DIV instruction; BIT instructions), the normal action is for the pipeline to suspend instruction fetch until execution is complete. Due to the flash macro latch select signal changing state at the wrong time at those addresses, this next fetch at this address is accessing the wrong latch and fetching wrong data.</p> <p>This problem is present in the V850/SB1 only and cannot be observed using the emulator.</p> <p>Example:</p> <p>Address: Instruction</p> <p>1FFF8H TST1 7,-4094[r0] &lt;- instruction taking more than 1 clock cycle which suspends normal pipelined fetch operation</p> <p>1FFFCH ADDI 1,r10,r10 &lt;- instruction not being fetched from ROM correctly (this could be any instruction)</p> <p><u>Workaround</u></p> <ol style="list-style-type: none"> <li>1. Block the addresses in the linker file by inserting 'dummy' sections on the two address areas (1FFF8H – 1FFFFH).</li> <li>2. Insert NOP instructions at this position (1FFF0H to 1FFFFH).</li> </ol>

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

Bug No.	Outline	Description
7	CSI4 baud rate restriction	<p><u>Details</u></p> <p>On CSI4 data is not transmitted correctly if the baud rate of variable length CSI (CSI4) is set to a transmission speed faster than or equal to the operating clock of the CPU.</p> <p><u>Example:</u></p> <p>When operating at <math>f_{CPU} = f_{XX}/2</math>, the baud rate <math>f_{XX}/2</math> cannot be selected                      When operating at <math>f_{CPU} = f_{XX}/4</math>, the baud rates <math>f_{XX}/2</math> and <math>f_{XX}/4</math> cannot be selected                      When operating at <math>f_{CPU} = f_{XX}/8</math>, the baud rates <math>f_{XX}/2</math>, <math>f_{XX}/4</math> and <math>f_{XX}/8</math> cannot be selected</p> <p>When operating at <math>f_{CPU} = f_{XX}</math> all baud rates can be selected. This restriction does not apply to CS10 – CS13 or UART0 – UART1.</p> <p><u>Workaround:</u></p> <p>Do not use baud rates faster than or equal to the operating clock of the CPU for CSI4.</p>
8	Cascaded 8-bit timers	<p><u>Details</u></p> <p>When two 8-bit timers are cascaded and used as a 16-bit timer the value of the higher 8-bits becomes undefined if both timers are not stopped when changing the value of the compare register.</p> <p><u>Example:</u></p> <p>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. Then change the value of CR45.</p> <p><u>Workaround:</u></p> <p>If the value of CRmn is to be changed when TMM and TMn are cascaded, first set TCEm and TCEn to 0 in order to stop TMM and TMn counting. Then change the value of CRmn.</p>

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

Bug No.	Outline	Description
9	Double interrupt execution	<p><u>Details</u></p> <p>An interrupt that should occur only once occurs twice if the following three conditions occur simultaneously while interrupts are enabled:</p> <ol style="list-style-type: none"> <li>1) A bit manipulation instruction (set1, clr1, not1 or tst1) is executed on an interrupt request flag (xxIFn) of an interrupt control register (xxICn).</li> <li>2) Interrupt processing involving the hardware of the same register occurs</li> <li>3) There is a DMA startup while executing the above bit operation</li> </ol> <p>Remark: xx: Identification name of a peripheral unit (WDT, P, WTNI, TM, CSI, SER, ST, AD, DMA, WTN, KR, IIC) n: peripheral unit number</p> <p>If the abovementioned condition appears, the Interrupt Request Flag, which is normally reset to 0 at the acknowledge of interrupt processing, will not be reset. Consequently, after returning from interrupt processing (reti instruction), the interrupt processing is executed again. This does not happen if DMA is not used.</p> <p><u>Example:</u></p> <p>During a bit manipulation operation using the clr1 instruction on the interrupt request flag of the CSIC0 register (CSIF0), the non-masked INTCSI0 interrupt occurs at the same time as a DMA startup. As a result the INTCSI0 interrupt processing is executed twice.</p> <p><u>Workaround:</u></p> <ol style="list-style-type: none"> <li>1) Insert a DI instruction before and an EI instruction after executing a bit manipulation instruction on an interrupt request flag (xxIFn) of an interrupt control register (xxICn) to avoid carrying out interrupt processing immediately after executing the bit manipulation instruction.</li> <li>2) When interrupt processing begins, the hardware enters a state where interrupts are disabled. Clear the interrupt request flag in all interrupt processing routines before executing the EI instruction.</li> </ol>
10	Limitation on frequency / bus-voltage	<p><u>Details</u></p> <p>When operating the bus interface (BVDD and EVDD) at voltages &lt;= 4.0V correct bus interface timing cannot be guaranteed for CPU operation frequencies higher than 13 MHz.</p> <p>Workaround:</p> <ol style="list-style-type: none"> <li>a) If BVDD or EVDD are set to voltages &lt;= 4.0V use a maximum operation speed <math>f_{CPU}</math> of 13 MHz.</li> <li>b) If operation speed higher than 13 MHz is used, set BVDD and EVDD to a voltage higher than 4.0V and use only external peripherals that operate at this voltage.</li> </ol>

Exclusively for design purposes; no part of this may be disclosed to a third party without the consent of NEC Electronics(Europe).

**(C) Flash Limitations**

Limitations	μPD70F3033	
	ES1.1	V2.0
W/E cycles	TBD after quality test	TBD after quality test
CPUclk/MHz read/exec	0,25 - 20MHz	2 - 20MHz
CPUclk/MHz write/erase	2 - 20MHz	2 - 20MHz
Ta/°C read/exe	-40 - +85	-40 - +85
Ta/°C write/erase	+25	+25
Data retention	no guarantee target 10years	no guarantee target 10years

**(D) Flash Programming Information**

Programmer settings	μPD70F3033	
	ES1.1	V2.0
Programmer	Flash-Master / Flash Pro2 (V2.35)	Flash-Master using Firmware update E2.01i
Vpp/V	7.8	7.8
Erase/ms	2000	200
Write/μs	50	20
Prewrite	yes	yes
Writeback	-	1ms
Area / Chip	Area / Chip	Area
Area size	2 x 128k	2 x 128k