Concerned Products:	**Customer Notification**	Date: Sep. 21, 2000 NEC-Electronics (Europe) GmbH EAD –Technical Product Support
μPD703031A ***μPD703031AY*** ***μPD703032A*** ***μPD703032AY*** ***μPD703033A*** ***μPD703033AY*** ***μPD70F3032A*** ***μPD70F3032AY*** ***μPD70F3033A*** ***μPD70F3033AY*** ***μPD70F3033*** ***μPD70F3033Y***	**Bug Report**	Source Doc: HW 260199 HW 080299 SBG-T-1719 SBG-T-1777 SBG-T-2107 Author: H.-W. Hoeft, Udo Wenzel
Date of initial issue: Feb 19th 99		Doc. No.: TPS-HE-B-2122

(A) BUG LIST

Bug No.		Outline	µPD70F3033, µPD70F3033Y				µPD70F3032A, µPD70F3032AY µPD70F3033A, µPD70F3033AY			µPD703033A, µPD703033AY, µPD703031A, µPD703031AY, µPD703032A, µPD703032AY		
			ES1.1	2.0	2.1	2.2	2.3			2.0		
1		Output on port pin P31	✋	✓	✓	✓	✓			✓		
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	💣	💣	💣	💣	💣			💣		
11		16-bit timer (one-shot pulse mode)	💣	💣	💣	💣	💣			💣		
12		EI instruction	💣	💣	💣	💣	💣			💣		
13		Limitation on frequency / bus-voltage	✋	✋	✋	✋	✋			✋		
14		A/Dconverter	💣	💣	💣	💣	💣			💣		

✓: No problem

✋: Bug (will be corrected by next version upgrade)

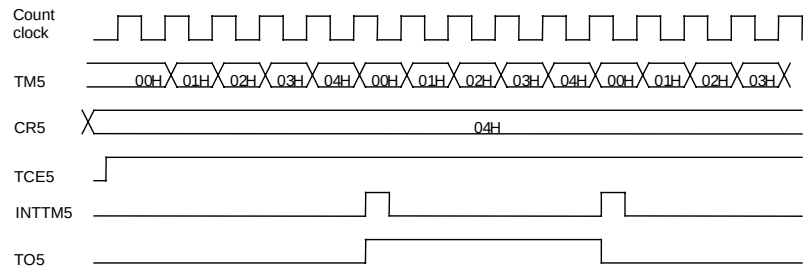
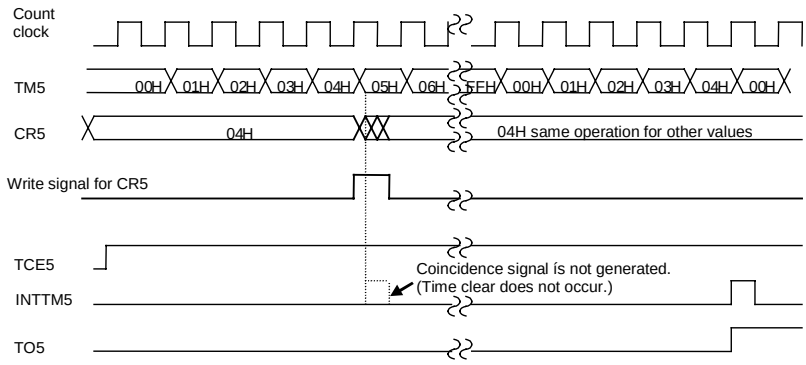
💣: Bug (restriction, not corrected by version upgrade)

☝: Bug/restriction future status pending

Name	Order number	Memory
V850/SB1	µPD703030A µPD703030AY	384Mask
	µPD703031A µPD703031AY	128Mask
	µPD703032A µPD703032AY	512Mask
	µPD703033A µPD703033AY	256Mask
	µPD70F3032A µPD70F3032AY	512Flash
	µPD70F3033A µPD70F3033AY	256Flash

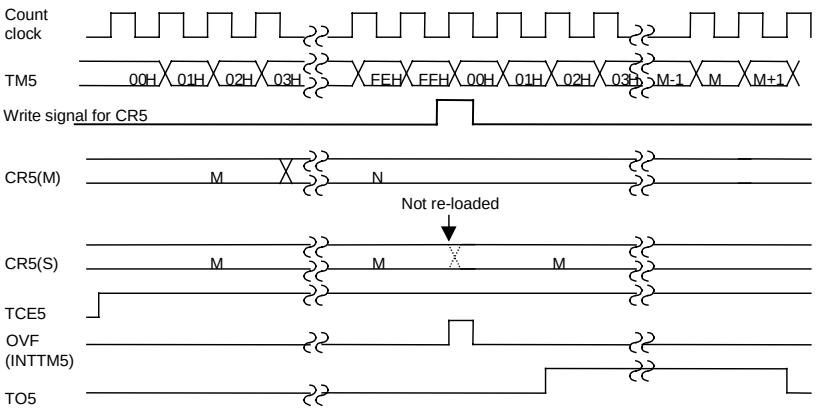
(B) BUG DESCRIPTION

Bug No.	Outline	Description
1	Output on port pin P31	Details When P33 is used for SO4, P31 does not work correctly as output port. Workaround Use the port pin P31 only as inport pin, when port pin P33 is used for SO4.
2	Sub operation current	Details The current consumption during subclock mode is higher than target specification: subclock normal operation typ 280µA HALT mode typ 190µA IDLE mode typ 180µA

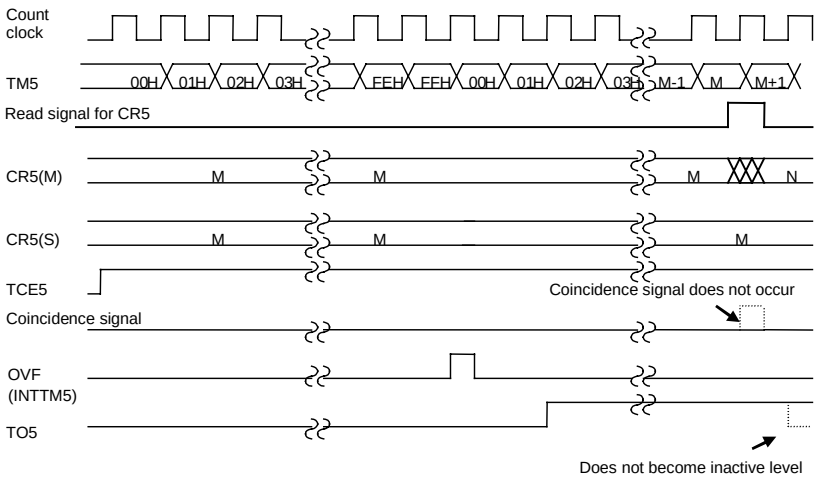
3	TM 2-5 operational defect	Details: When used in interval timer mode If the compare register (CR20-50) is written when the timer register (TM2-5) and compare register (CR20-50) coincide, a coincidence signal is not generated. For this reason, the software must be designed to prevent timer coincidence and writing the compare register occurring at the same time. (Example: use a Vector interrupt) 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) Interval timer and square wave output (Normal operation)  Interval timer and square wave outout (Incorrect operation)  The timing chart above is intend to show the general operation and does not show precise timing

3	TM 2-5 operational defect	When used in PWM output mode - 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.

PWM timer (Incorrect operation)

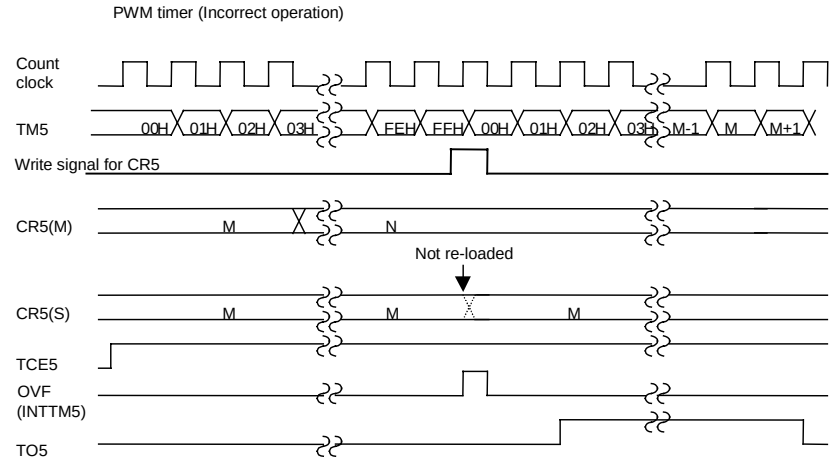


PWM timer (Incorrect operation)



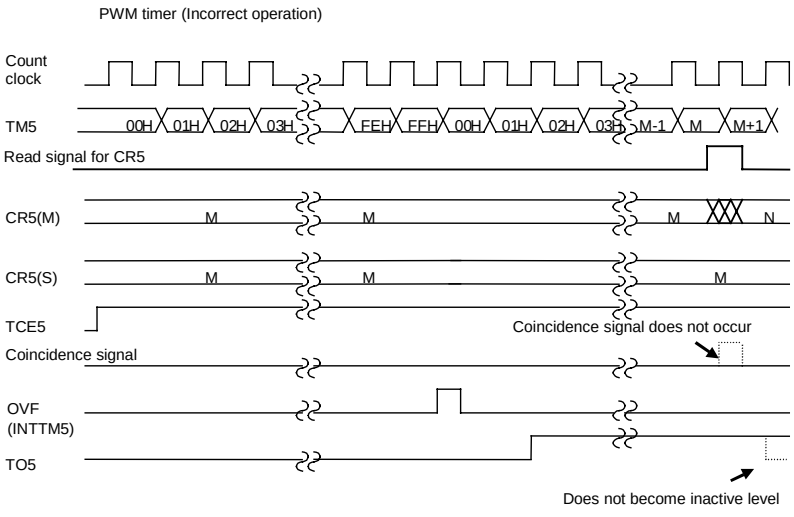
The Timing chart above is intendend to show the general operation, and does not show precise timing.

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.

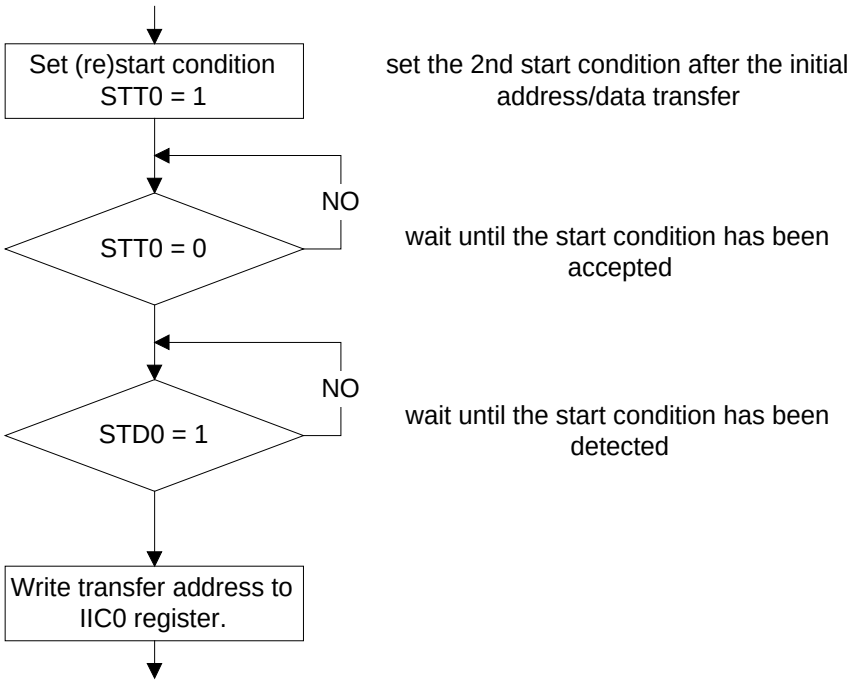


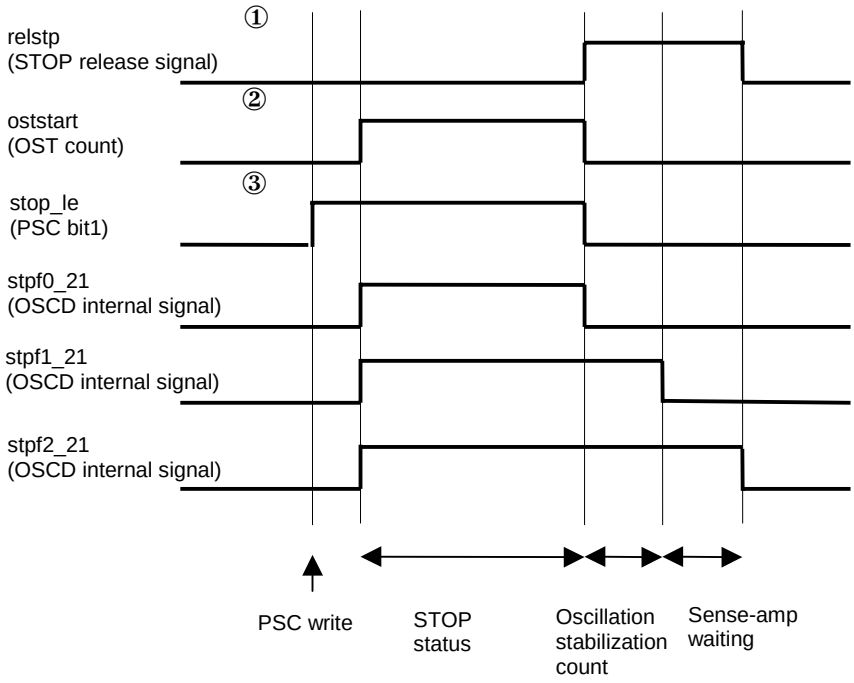
The Timing chart above is intendend to show the general operation, and does not show precise timing.

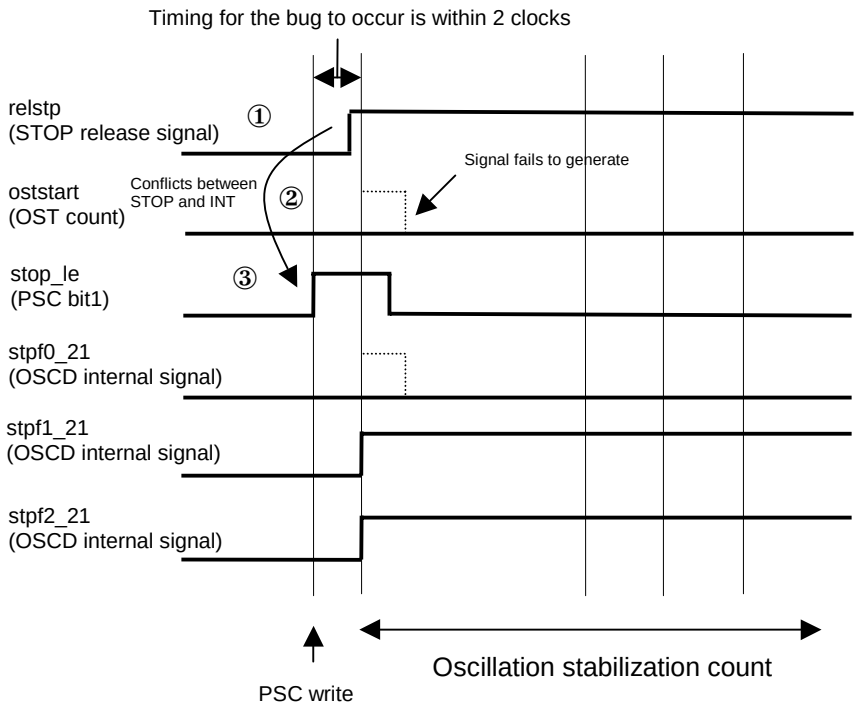
- If the compare register (CR20-50) is written when the timer register (TM2-5) and compare register (CR20-50 [slave register]) coincide, a coincidence signal is not generated and the inactive level is not attained.



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

Bug No.	Outline	Description
4	Using the IIC restart condition causes a faulty address transfer.	*Details* 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. *Workaround* The transfer address must be set after the confirmation of start condition detection (STT=0, STD = 1). 

Bug No.	Outline	Description
5	STOP Mode	The CPU becomes deadlocked, if the timing of a STOP mode enable and a Stop mode release are in conflict *Details* With the V850/SA1, the STOP mode is enable by writing (stop_le③) to the PSC register (Power Save Control register). To release the Stop mode, the count initialization signal (oststart:②) 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:①) 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). *Workaround* As a temporary measure it is suggest that you use the IDLE mode or sub-IDLE mode instead of the STOP mode.  Diagramm1: STOP release operation timing (Normal)

5	STOP Mode	 Diagramm 2: Timing of conflict between STOP mode enable and release (Error)

Bug No.	Outline	Description
6	ROM – CPU interface problem	Details Instructions located at addresse corresponding to the boarder between Flash macros (Flash areas) (addresse 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. 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. This problem is present in the V850/SB1 only and cannot be observed using the emulator. Example: Address: Instruction 1FFF8H TST1 7,-4094[r0] <- instruction taking more than 1 clock cycle which suspends normal pipelined fetch operation 1FFFCH ADDI 1,r10,r10 <- instruction not being fetched from ROM correctly (this could be any instruction) Workaround 1. Block the addresses in the linker file by inserting ‘dummy’ sections on the two address areas (1FFF8H – 1FFFFH). 2. Insert NOP instructions at this position (1FFF0H to 1FFFFH).

Bug No.	Outline	Description
7	CSI4 baud rate restriction	Details 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. Example: When operating at $f_{CPU} = f_{XX}/2$, the baud rate $f_{XX}/2$ cannot be selected When operating at fCPU = $f_{XX}/4$, 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 – CS13 or UART0 – UART1. Workaround: Do not use baud rates faster than or equal to the operating clock of the CPU for CSI4.
8	Cascaded 8-bit timers	Details 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. 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. Then change the value of CR45. Workaround: 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.

Bug No.	Outline	Description
9	Double interrupt execution	Details An interrupt that should occur only once occurs twice if the following three conditions occur simultaneously while interrupts are enabled: 1) A bit manipulation instruction (set1, clr1, not1 or tst1) is executed on an interrupt request flag (xxIFn) of an interrupt control register (xxICn). 2) Interrupt processing involving the hardware of the same register occurs 3) There is a DMA startup while executing the above bit operation Remark: xx: Identification name of a peripheral unit (WDT, P, WTNI, TM, CSI, SER, ST, AD, DMA, WTN, KR, IIC) n: peripheral unit number 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. Example: 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. Workaround: 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. 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.

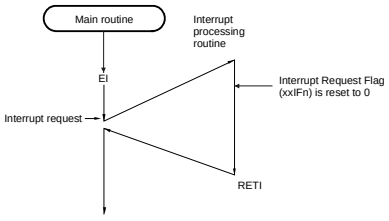
<table>
<tr>
<td></td>
<td></td>
<td>


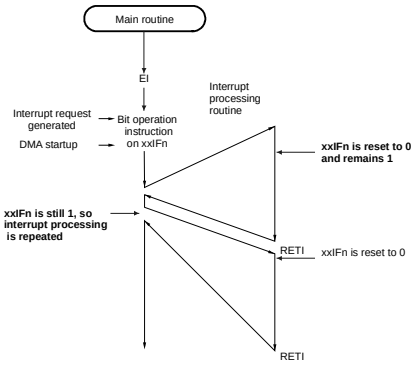
Figure 1. Normal Interrupt Processing


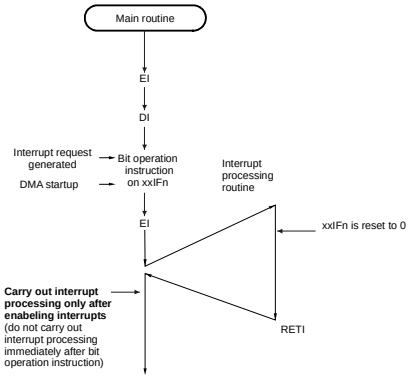
Figure 2. Interrupt Processing Affected by This Phenomen


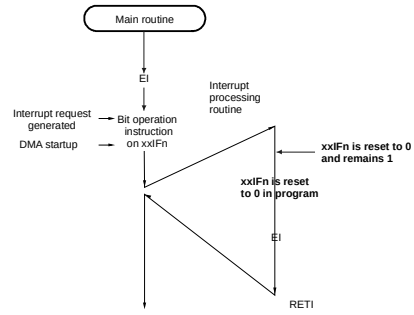
Figure 3. Countermeasure (1)


Figure 4. Countermeasure (2)
Remarks xx: Identification name of each peripheral unit (WDT, P, WTNI, TM, CS, SER, ST, AD, DMA, WTM, KR, IIC, IEB)
N: Peripheral unit number (0 to 7)
</td>
</tr>
<tr>
<td>11</td>
<td>16-bit timer (one-shot pulse mode)</td>
<td>16-bit timer one-shot pulse output function

<u>Details</u>

When using the one-shot pulse function of timers 0,1 and 7 via a software trigger, the level of the T1pin or its alternate function port cannot be changed.
Because the trigger is also enable in this case, the trigger will inadvertently clear & start even if the level of T1 pin or its alternate function port is changed, causing a pulse to be out´üut at an unintended timing.</td>
</tr>
</table>

12	EI instruction	Interrupt servicing acknowledgement after EI instruction Details 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 cause all interrupts to be held pending until the re-execution of the EI instruction (interrupt enable). Since this determination time is also 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 instruction. - 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 DI : ; MK flag = 0 (interrupt enable) : ; Interrupt request generation (IF flag = 1) EI JR, LP1 ┐ : │ 7 clcoks have not elapsed between EI an DI instructions : │ (3 clocks) LP1: │ DI ◄┘ : Workaround example: DI : ; MK flag = 0 (interrupt enable) : ; Interrupt request generation (IF flag = 1) EI ┐ NOP │ ; 1system clock NOP │ ; 1system clock NOP │ ; 1system clock NOP │ ; 1system clock JR, LP1 │ ; 3 system clocks (branch to LP1 routine) : │ : │ LP1: │ DI ◄┘ ; Interrupt servicing executed on 8th clock : after EI instruction

(C) Limitations

13	Limitation on frequency / bus-voltage	Details When operating the bus interface (BVDD and EVDD) at voltages <= 4.0V correct bus interface timing cannot be guaranteed for CPU operation frequencies higher than 13 MHz. Workaround: a) If BVDD or EVDD are set to voltages <= 4.0V use a maximum operation speed f_{CPU} of 13 MHz. 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.

(D) Cautions

14	A/Dconverter	Details When not using the A/D converter, change the A/D conversion time mode to low speed conversion mode and cut off the tap selector and voltage supply block (AVdd). – Set the ADPS bit of the ADM1 register to1 – Set the IEAD bit of the ADM2 register to 0 Failure to make this settings will cause a current of 2mA to flow. Note that if ADPS is set to 0 again (high-speed conversion mode), stabilization time up to the start of the A/D conversion operation will be required (up to 5 μs). A wait of at least 5 μs must therefore be secured by software

Note:
The above mentioned cautions will be incorporated in the revised user's manual.