Concerned Products:	**Customer Notification**	Date: May 3, 1999
IE-789014-NS-EM1		NEC-Electronics (Europe) GmbH EAD -Technical Product Support
	Bug Report	Source Doc: SBG-T-0442 SBG-T-0487 SBG-T-0524 SBG-T-0603
		Author: W. Noll M. Kratz

Jan. 28th, 98	Doc. No.: TPS-LE-B-ST01
1st revision : May 3, 99	Doc. No.: TPS-LE-B-ST01-1

(A) BUG LIST

Bug No.	Outline	IE-789014-NS-EM1
1	Bit and logical operation at ports 2 and 3	💣
2	Serial / general purpose port switch over	💣
3	Read data from UART	💣
4	8-bit timer / interval restriction	💣

✓: No problem

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

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

(B) BUG DESCRIPTION

Bug No.	Outline	Description
1	Bit and logical operation at ports 2 and 3	Details Avoid to use bit operation instructions and logical operation instructions on Ports 2 and 3(dual function pins). Instead of this, be sure to use 8 or 16-bit data transfer instructions to control output ports 2 and 3. Reason for operating precautions Executing bit operation instructions (SET1, CLR1) and logical operation instructions (OR, XOR, etc.) on the ports (2 and 3) which have the dual functions of timer output and serial interface may cause the contents of the dual function pins to be different from the expected values. This is because the bit operation instructions and logical operation instructions of the µPD789011, 789012 and 78P9014 are not for performing operations on the contents of the output latch but for performing operations on the status of the relevant pins. We have no plans to change the device circuits. For your reference, **Attachment 1** shows an example of executing the SET1 instruction on P32 causing P30 to be fixed to “H” and **Attachment 2** is listing the ports and instructions relevant to this precaution.

Bug No.	Outline	Description

2	Serial / general purpose port switch over	Details
		
		Three wire serial I/O mode
		If the operation is suspended (CSIE=0 write) while the system is transmitting/receiving data in three-wire SIO, or if the operation enable flag is cleared (CSIE=0 write) when the system is not performing transmission/reception, SO0's dual-purpose output port cannot be used as a general-purpose output port.
		
		Provisional Remedy:
		Do not clear the CSIE flag until the transmission/reception has ended. When ending the three wire SIO mode, send "FFH" first, before clearing the CSIE flag. Or, send "FFH" in the UART mode before clearing the transmission operation enable flag (TXE).
		Example 1: Three-wire SIO transmission
		MOV CSIM0, #02H
		MOV BRGC, #00H
		MOV ASIM, #00H
		MOV TXS, #0FFH
		CLR1 CSIE
		Upon writing data into TXS, the SO pin immediately turns high (after 4 clocks). However, the clocks are transmitted to the SCK clock pin.
		
		Example 2: UART transmission
		MOV CSIM0, #00H
		MOV BRGC, #00H
		MOV ASIM, #80H
		MOV TXS, #0FFH
		CLR1 TXE
		The SO pin turns Hi after 16 to 32 clocks after writing data into TXS. With this method, the SCK pin remains Low.
		
		UART mode
		If the operation is suspended (TXE=0 write) while the UART system is transmitting data, TXD's dual-purpose output port cannot be used as a general-purpose output port.
		
		Provisional Remedy:
		Do not write "0" into the transmission operation enable flag (TXE) while data is being transmitted in the transmission operation enable (TXE=1) state. When switching over to the general-purpose output port, clear the transmission operation enable flag at the point when the data transmission is completed.
		Example to switch over to the general-purpose output port after UART transmission is ended:
		
		MOV CSIM0, #00H
		MOV BRGC, #40H ; Baud rate:9600 bps
		MOV ASIM, #88H ; Data length: 8 bits; one stop bit; no parity
		WAIT:
		BF STIF, SWAIT
		CLR1 TXE

<table>
<tr><th>Bug No.</th><th>Outline</th><th>Description</th></tr>
<tr><td>3</td><td>Read data from UART</td><td>
Details
Do not read the RXB register immediately after occurrence of a reception interrupt, because an overrun error may occur.
Instead of this, wait several clock cycles as indicated in the “Clock Count Until RXB Read” table shown below, before reading RXB register!
<table>
<tr><td></td><td></td><td colspan="2">Clock Count Until RXB Read</td></tr>
<tr><td>BRGC setting</td><td>Transfer rate @ 4.9152 MHz</td><td>High speed PCCI = 0</td><td>Mid speed PCCI = 1</td></tr>
<tr><td>00H</td><td>153.6Kbps</td><td>0</td><td>0</td></tr>
<tr><td>10H</td><td>76.8Kbps</td><td>0</td><td>0</td></tr>
<tr><td>20H</td><td>38.4Kbps</td><td>0</td><td>0</td></tr>
<tr><td>30H</td><td>19.2Kbps</td><td>7</td><td>2</td></tr>
<tr><td>40H</td><td>9.6Kbps</td><td>23</td><td>6</td></tr>
<tr><td>50H</td><td>4.8Kbps</td><td>55</td><td>14</td></tr>
<tr><td>60H</td><td>2.4Kbps</td><td>119</td><td>30</td></tr>
<tr><td>70H</td><td>1.2Kbps</td><td>247</td><td>62</td></tr>
<tr><td>80H</td><td colspan="3">
In the case of an external clock, make sure that the waiting time is satisfying the following expression:

EXCL1(Hz)>f_{CPU}(Hz)/(9 clocks + X clocks)

The external clock frequency EXCL1 is “the transfer rate multiplied by 2” , f_{CPU} is the CPU’s operating frequency.
Nine clocks result because the interrupt processing is starting one clock after the occurrence of the interrupt and eight clocks are used for the interrupt processing. “X clocks” refers to the clock count until the reading is over. The timing of reading the RXB register in the interrupt routine varies from one application to another.

Example, the CPU operates at 1MHz by inputting 4.8KHz clocks from EXCK1:

4.8KHz > 1MHz/(9+X)
X > (1MHz/4.8KHz)-9
X > 199.3

Accordingly, reading the RXB register in the interrupt routine must be performed after 200 clocks.
</td></tr>
</table>
</td></tr>
</table>

Bug No.	Outline	Description
4	8-bit timer (TM0/TM1) restrictions	Detail restriction 1: To use these timers in interval timer mode, rewrite the value of the compare registers (CR00/CR11) in a state where the timer operation is inhibited. Rewriting the value of a compare register (CRxx) in a state where the timer operation is permitted may generate coincidence signals immediatley. (In the case that interrupts are permitted, interrupt requests will occur.)

Attachement 1

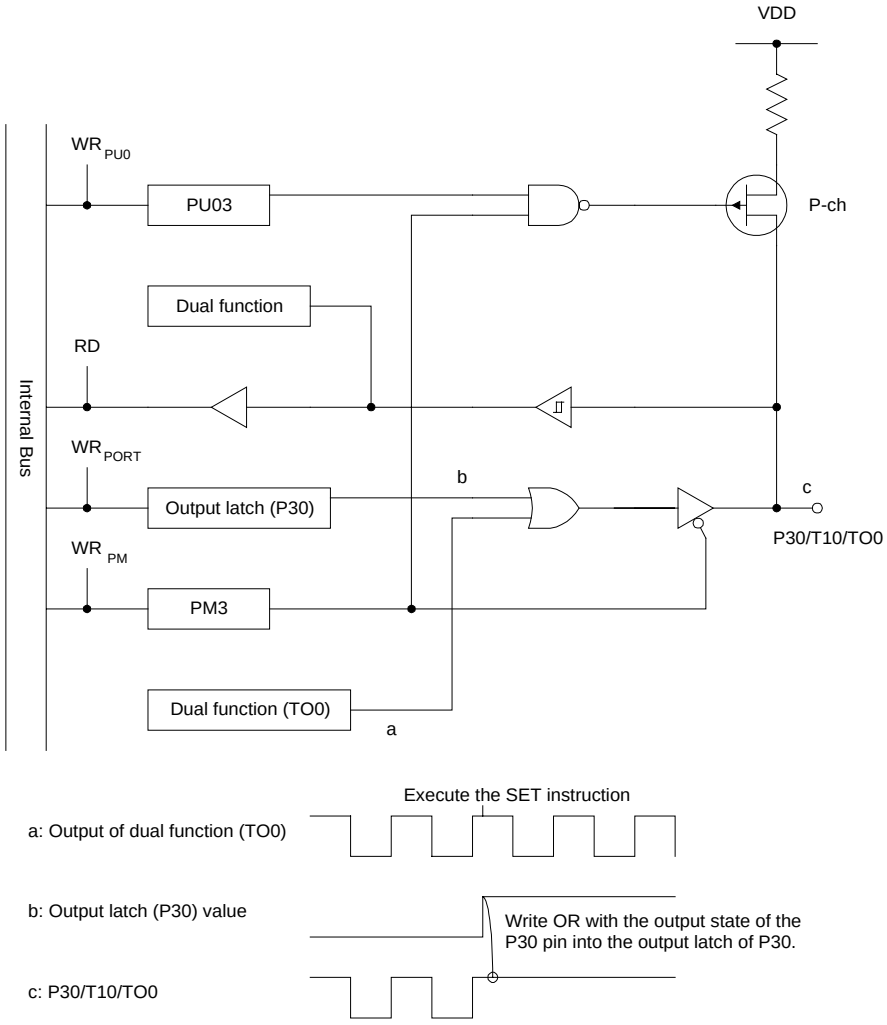

Figure P30 Block Diagram

Example: P3x

① Uses P30 as the TO0 output to execute “SET1 PORT 3.2” while TO0 is outputting a high level.

② The CPU reads all the pin statuses of Port 3 in response to the SET1 operating instruction, sets the relevant bit (P32 in this example) and writes the result to the latch of the Port 3.

③ When this SET1 instruction was executed, pin P30 was high level. This results in a high level being written to the output latch of P30, thus causing the pin´s output to be fixed to high level. (When outputting TO0, it is necessary to set the output latch to low level first.)

Attachement 2

① Relevant ports

Port 2	P20/ASCK/SCK0, P21/TxD/SO0, P22/RxD/SIO
Port 3	P30/INTP0/TIO/TO0, P31/INTP1/TI1/TO0, P32/INTP

② Relevant instructions
* Assembler code
Use of the following instructions may result in the operations shown in the restrictions above.

Relevant Instruction	Description Example
SET1	SET1 P32
CLR1	CLR1 P32
AND	AND P3, #5
OR	OR P2, #2
XOR	XOR P2, #3

* Example in the C language
The example shown below may cause the problems explained above.

C Language	Assembler
P3 | = 0 x 04;	OR P3, #4
P3. 2 = 1;	SET1 P3.2

P3 = P3 | 0 x 04;	MOV A, P3
	OR A, #4
	MOV P3, A

The example below shows an example where only the relevant bits are affected, taking the restrictions into consideration.

C Language	Assembler

P3 = (P3 | 0 x 04) & 0 x 04:	MOV A, P3
	OR A, #4
	AND A, #4
	MOV P3, A