

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

Concerned Products:	Customer Notification		Date: May 3, 1999
μPD789011 μPD789012 μPD78P9014			Bug Report
Jan. 28th, 98		Doc. No.: TPS-LE-B-S001	
1 st revision	: May 3, 99	Doc. No.: TPS-LE-B-S001-1	

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

(A) BUG LIST

Bug No.	Outline	μPD789011	μPD789012	μPD78P9014
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)

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

(B) BUG DESCRIPTION

Bug No.	Outline	Description
1	Bit and logical operation at ports 2 and 3	<p>Details</p> <p>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.</p> <p>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.</p> <p>We have no plans to change the device circuits.</p> <p>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.</p>

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

Bug No.	Outline	Description
2	Serial / general purpose port switch over	<p>Details</p> <p>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.</p> <p>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).</p> <p>Example 1: Three-wire SIO transmission</p> <pre> MOV CSIM0, #02H MOV BRGC, #00H MOV ASIM, #00H MOV TXS, #0FFH CLR1 CSIE </pre> <p>Upon writing data into TXS, the SO pin immediately turns high (after 4 clocks). However, the clocks are transmitted to the SCK clock pin.</p> <p>Example 2: UART transmission</p> <pre> MOV CSIM0, #00H MOV BRGC, #00H MOV ASIM, #80H MOV TXS, #0FFH CLR1 TXE </pre> <p>The SO pin turns Hi after 16 to 32 clocks after writing data into TXS. With this method, the SCK pin remains Low.</p> <p>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.</p> <p>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.</p> <p>Example to switch over to the general-purpose output port after UART transmission is ended:</p> <pre> 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 </pre>

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

Bug No.	Outline	Description			
3	Read data from UART	<p>Details</p> <p>Do not read the RXB register immediately after occurrence of a reception interrupt, because an overrun error may occur.</p> <p>Instead of this, wait several clock cycles as indicated in the "Clock Count Until RXB Read" table shown below, before reading RXB register!</p>			
		Clock Count Until RXB Read			
		BRGC setting	Transfer rate @ 4.9152 MHz	High speed PCCI = 0	Mid speed PCCI = 1
		00H	153.6Kbps	0	0
		10H	76.8Kbps	0	0
		20H	38.4Kbps	0	0
		30H	19.2Kbps	7	2
		40H	9.6Kbps	23	6
		50H	4.8Kbps	55	14
		60H	2.4Kbps	119	30
		70H	1.2Kbps	247	62
		80H	<p>In the case of an external clock, make sure that the waiting time is satisfying the following expression:</p> $EXCL1(Hz) > f_{CPU}(Hz) / (9 \text{ clocks} + X \text{ clocks})$ <p>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.</p> <p>Example, the CPU operates at 1MHz by inputting 4.8KHz clocks from EXCK1:</p> $4.8KHz > 1MHz / (9 + X)$ $X > (1MHz / 4.8KHz) - 9$ $X > 199.3$ <p>Accordingly, reading the RXB register in the interrupt routine must be performed after 200 clocks.</p>		

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

Bug No.	Outline	Description
4	8-bit timer (TM0/TM1) restrictions	<u>Detail restriction 1:</u> 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 immediately. (In the case that interrupts are permitted, interrupt requests will occur.)

Attachement 1

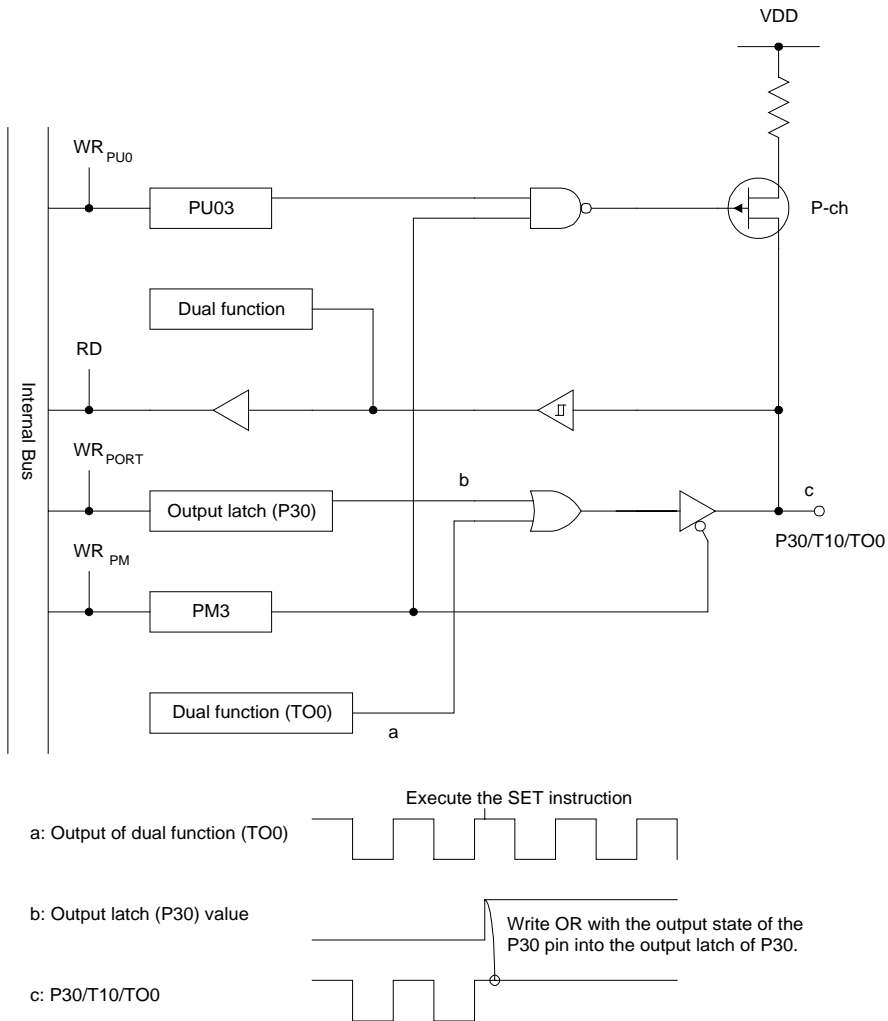


Figure P30 Block Diagram

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.)

Exclusively for design purposes! No part of this may be disclosed to a third party without the consent of NEC Electronics

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