Date: Dec. 26, 2018

RENESAS TECHNICAL UPDATE

TOYOSU FORESIA, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RZ*-A055A/E	Rev.	1.00
Title	RZ/T1 Group User's Manual: Hardware Corrections and adding notification for the Ethernet MAC and Ethernet Switch		Information Category	Technical Notification		
Applicable Product	RZ/T1 Group	Lot No. All lots	Reference Document	RZ/T1 Group User's Manual: Hardware Rev1.40 R01UH0483EJ0140 Rev.1.40		

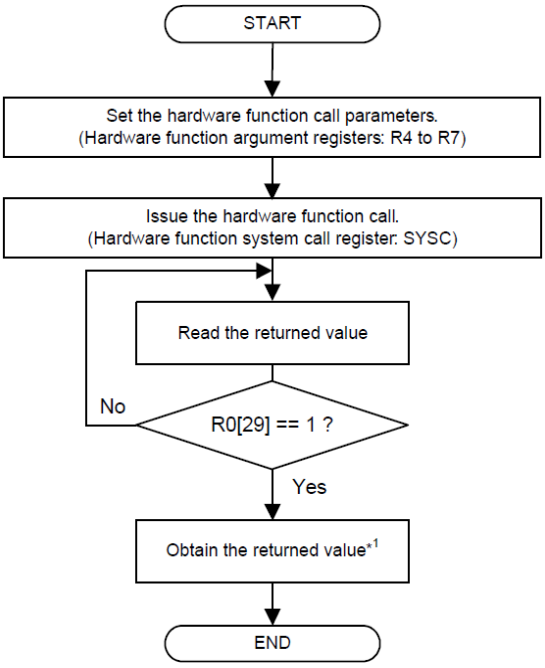
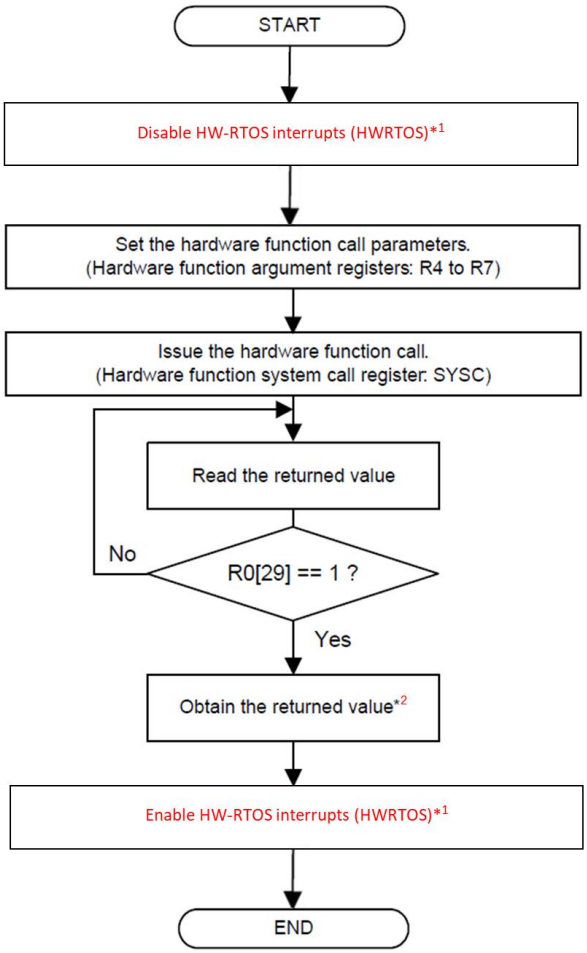
There are corrections and adding notifications for the Ethernet MAC and Ethernet Switch in User's manual.

Please use the Ethernet MAC and Ethernet Switch with the contents as shown below.

[Correction of User's manual]

- Ethernet MAC

No.	Page	Current description	Correct description
1	1329	28.2.2.1 MIIM Register (GMAC_MIIM) Note: The contents of the GMAC_MIIM register are valid for the management interface that is set by the MAC select register (MACSEL). In other conditions, writing to the register is impossible and any values read from the register are undefined.	28.2.2.1 MIIM Register (GMAC_MIIM) Note: The contents of the GMAC_MIIM register are valid **only when the EthernetMAC or the Ethernet Switch Port is selected** by the MAC select register (MACSEL). In other conditions, writing to the register is impossible and any values read from the register are undefined.
2	1334	[Current description] 28.2.2.5 RX MODE Register (GMAC_RXMODE) b29 SFRXFIFO Store & Forward For RX FIFO 1: Store & Forward mode R/W The receive DMA controller does not start operation until data up to the end of the frame is written to RX FIFO. 0: Cut Through mode	[Correct description] 28.2.2.5 RX MODE Register (GMAC_RXMODE) b29 SFRXFIFO Store & Forward For RX FIFO 1: Store & Forward mode R/W **The reception DMA controller starts to operate after data up to the end of the frame is written to the RX FIFO buffer.** 0: Cut Through mode **The reception DMA controller starts to operate after data is written to the RX FIFO buffer.** **The number of data words is specified in the RRTTH[2:0] bits.**

3	1357	[Current description] 28.3.1.2 Figure 28.4 Flow of Processing for Issuing the Hardware Function Call  Note 1. The hardware function recognizes that the result registers were read by reading the R0 and R1 registers. Be sure to read R1 even in the system call which does not use R1 for storing the returned value.
		[Correct description] 28.3.1.2 Figure 28.4 Flow of Processing for Issuing the Hardware Function call  **Note 1. This processing is required only when the hardware real-time OS is used.** **Note 2.** The hardware function recognizes that the result registers were read by reading the R0 and R1 registers. Be sure to read R1 even in the system call which does not use R1 for storing the returned value.

<table>
<tr><td>4</td><td>1358</td><td>28.3.1.3 Buffer Allocator
(1) Functional Overview
To use the buffer RAM, secure the required area (hereafter “buffer”) beforehand, and then issue the hardware function calls provided for the buffer allocator. The CPU attempting to write to an area which has not been secured is ignored, but an exception is generated in the case of access by the DMAC.</td><td>28.3.1.3 Buffer Allocator
(1) Functional Overview
To use the buffer RAM, secure the required area (hereafter “buffer”) beforehand, and then issue the hardware function calls provided for the buffer allocator. When writing to an area which has not been secured, access by the CPU or MAC DMA controller generates an interrupt, and access to such area by the buffer RAM DMA controller generates an interrupt or returns an exception to the return value register R0 depending on the type of hardware function calls.</td></tr>
<tr><td>5</td><td>1361</td><td colspan="2">[Current description]
28.3.1.3 (2) (e)
Table 28.7 HWFNC_LongBuffer_Get
<table><tr><td>R1</td><td>[31:0] First logical address of the buffer</td><td>[31:27] 00001b
[26] 1
[25:18] LLID
[17:0] 0</td></tr></table>
[Correct description]
28.3.1.3 (2) (e)
Table 28.7 HWFNC_LongBuffer_Get
<table><tr><td>R1</td><td>[31:0] First logical address of the buffer</td><td>[31:27] 00001b
[26:24] 100b
[23:18] LLID
[17:0] 0</td></tr></table></td></tr>
<tr><td>6</td><td>1362</td><td colspan="2">[Current description]
28.3.1.3 (2) (e)
Table 28.8 HWFNC_ShortBuffer_Get
<table><tr><td>R1</td><td>[31:0] First logical address of the buffer</td><td>[31:27] 00001b
[26] 0
[25:18] SBID
[17:0] 0</td></tr></table>
[Correct description]
28.3.1.3 (2) (e)
Table 28.8 HWFNC_ShortBuffer_Get
<table><tr><td>R1</td><td>[31:0] First logical address of the buffer</td><td>[31:27] 00001b
[26:25] 00b
[24:18] SBID
[17:0] 0</td></tr></table></td></tr>
<tr><td>7</td><td>1368</td><td>28.3.1.4 (2) (b) Usage
[Example of reading and releasing a buffer]
(1) Read the BUFID register.
(2) Shift the bits [27:16] read from BUFID 16 bits to the right to obtain the number of received words.
(3) The bits [15:0] read from the BUFID are bits [26:11] of the address where the acquired buffer starts. The individual bits of the address where the acquired buffer starts are configured as follows.
[31:27]: 00001b
[26:19]: Equivalent to the bits [15:8] in the BUFID ([26] of the start address is always 1; [25:19] are LLID[6:0])
[18:11]: Equivalent to the bits [7:0] in the BUFID (always 0)
[10: 0]: Always 0
(4) After using the buffer, specify the start address as an argument and issue the buffer release function call to release the buffer.</td><td>28.3.1.4 (2) (b) Usage
[Example of reading and releasing a buffer]
(1) Read the BUFID register.
(2) Shift the bits [27:16] read from BUFID 16 bits to the right to obtain the number of received words.
(3) The bits [15:0] read from the BUFID are bits [26:11] of the address where the acquired buffer starts. The individual bits of the address where the acquired buffer starts are configured as follows.
[31:27]: 00001b
[26:18]: Equivalent to the bits [15:7] in the BUFID
[17:11]: Equivalent to the bits [6:0] in the BUFID
[10: 0]: Always 0
(4) After using the buffer, specify the start address as an argument and issue the buffer release function call to release the buffer.</td></tr>
</table>

<table>
<tr><td>8</td><td>1369</td><td>28.3.1.4 (2) (c) List of hardware function calls
If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.
Access to an access-prohibited area (an area other than the buffer RAM, etc.) while the hardware function call is running leads to the return of an exception code in the return value register R0.</td><td>28.3.1.4 (2) (c) List of hardware function calls
If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.
If an error occurs while the hardware function call is running, an interrupt is generated.</td></tr>
<tr><td>9</td><td>1374</td><td>28.3.1.4 (3) (d) List of hardware function calls
If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.</td><td>28.3.1.4 (3) (d) List of hardware function calls
If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.
If an error occurs while the hardware function call is running, an interrupt is generated.</td></tr>
<tr><td>10</td><td>1374</td><td colspan="2">[Current description]
28.3.1.4 (3) (d)
Table 28.16 HWFNC_MACDMA_TX_Errstat
<table>
<tr><td>R0</td><td>[1:0] Result</td><td>[0]: Memory Access Violation
[1]: Memory Access Timeout</td></tr>
</table>
[Correct description]
28.3.1.4 (3) (d)
Table 28.16 HWFNC_MACDMA_TX_Errstat
<table>
<tr><td>R0</td><td>[1:0] Result</td><td>[0]:
0: Success
1: Memory Access Violation
・ Access to the buffer that is not acquired
・ The number of transfer bytes is not correct
・ The start address of the descriptor is not on a 64-bit boundary.
[1]:
0: Success
1: Memory Access Timeout
・ The start address of a transmission descriptor turns to be an end value (FFFF FFFFh)
・ Releasing the buffer automatically is failed</td></tr>
</table></td></tr>
<tr><td>11</td><td>1376</td><td colspan="2">[Current description]
28.3.1.5 (2) (c)
Table 28.17 HWFNC_Direct_Memory_Transfer
Argument registers
<table>
<tr><td>R4</td><td>[31:0] Address where the destination area for transfer starts</td><td>Specifies the address where the source area for transfer starts.</td></tr>
<tr><td>R5</td><td>[31:0] Address where the source area for transfer starts</td><td>Specifies the address where the destination area for transfer starts.</td></tr>
</table>
[Correct description]
28.3.1.5 (2) (c)
Table 28.17 HWFNC_Direct_Memory_Transfer
Argument registers
<table>
<tr><td>R4</td><td>[31:0] Address where the source area for transfer starts</td><td>Specifies the address where the source area for transfer starts.</td></tr>
<tr><td>R5</td><td>[31:0] Address where the destination area for transfer starts</td><td>Specifies the address where the destination area for transfer starts.</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td>12</td>
<td>1378</td>
<td>
[Current description]
28.3.2
Table 28.19 Interrupts related to Operations for Transmission
<table>
<tr><th>Interrupt Name</th><th>Symbol</th><th>Conditions for Asserting and De-asserting Interrupts</th></tr>
<tr><td>MACDMA transmission error interrupt</td><td>ETHDTIE</td><td>This interrupt is generated when the address field of a descriptor is outside the range of the buffer, the number of bytes for transfer is invalid, or a descriptor is not set to start on a 64-bit boundary.
Modify the settings of the transmission descriptor for retransmission.
Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</td></tr>
</table>

[Correct description]
28.3.2
Table 28.19 Interrupts related to Operations for Transmission
<table>
<tr><th>Interrupt Name</th><th>Symbol</th><th>Conditions for Asserting and De-asserting Interrupts</th></tr>
<tr><td>MACDMA transmission error interrupt</td><td>ETHDTIE</td><td>This interrupt is generated when an error occurs while the transmission MAC DMA is operating. There are several error sources and HWFNC_MACDMA_TX_Errstat is used to obtain the error source.
Modify the settings of the transmission descriptor for retransmission.
Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.</td></tr>
</table>
</td>
</tr>
</table>

- Ethernet Switch

<table>
<tr><th>No.</th><th>Page</th><th>Contents</th></tr>
<tr><td>1</td><td>1417</td><td>[Current description]
29.2.2.5 Input Learning Blocking Register (INPUT_LEARN_BLOCK)
<table>
<tr><td>b16</td><td>P0LEARNDIS</td><td>Port 0 Address Learning Setting</td><td>Sets the address learning of port 0
0: Invalid
1: Valid</td><td>R/W</td></tr>
<tr><td>b17</td><td>P1LEARNDIS</td><td>Port 1 Address Learning Setting</td><td>Sets the address learning of port 1
0: Invalid
1: Valid</td><td>R/W</td></tr>
<tr><td>b18</td><td>P2LEARNDIS</td><td>Internal Interface Address Learning Setting</td><td>Sets the address learning of the internal interface port (port 2)
0: Invalid
1: Valid</td><td>R/W</td></tr>
</table>
[Correct description]
29.2.2.5 Input Learning Blocking Register (INPUT_LEARN_BLOCK)
<table>
<tr><td>b16</td><td>P0LEARNDIS</td><td>Port 0 Address Learning Setting</td><td>Sets the address learning of port 0
0: Valid
1: Invalid</td><td>R/W</td></tr>
<tr><td>b17</td><td>P1LEARNDIS</td><td>Port 1 Address Learning Setting</td><td>Sets the address learning of port 1
0: Valid
1: Invalid</td><td>R/W</td></tr>
<tr><td>b18</td><td>P2LEARNDIS</td><td>Internal Interface Address Learning Setting</td><td>Sets the address learning of the internal interface port (port 2)
0: Valid
1: Invalid</td><td>R/W</td></tr>
</table></td></tr>
<tr><td>2</td><td>1448</td><td>[Current description]
29.2.5.3 Port Timestamp Control/Status Register (PORTn_CTRL) (n = 0, 1)
<table>
<tr><td>b0</td><td>TSVALID</td><td>Timestamp Status</td><td>Indicates the status of stored timestamp.
0: Timestamp is valid.
1: Timestamp is invalid.
Writing to the register (with any value) clears the bit.</td><td>R/W</td></tr>
</table>
[Correct description]
29.2.5.3 Port Timestamp Control/Status Register (PORTn_CTRL) (n = 0, 1)
<table>
<tr><td>b0</td><td>TSVALID</td><td>Timestamp Status</td><td>Indicates the status of stored timestamp.
0: Timestamp is invalid.
1: Timestamp is valid.
Writing to the register (with any value) clears the bit.</td><td>R/W</td></tr>
</table></td></tr>
</table>