To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Date: Dec.28. 2005

RENESAS TECHNICAL UPDATE

Nippon Bldg.,2-6-2,Ohte-machi,Chiyoda-ku,Tokyo 100-0004,Japan
Renesas Technology Corp.

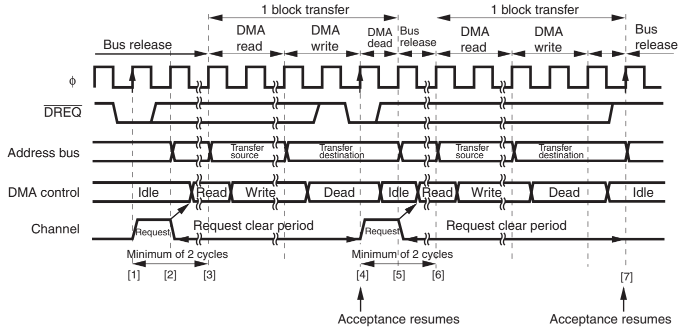
<table>
<tr><td><b>Product Category</b></td><td colspan="2">MPU&MCU</td><td><b>Document No.</b></td><td>TN-H8*-A317A/E</td><td><b>Rev.</b></td><td>1.00</td></tr>
<tr><td><b>Title</b></td><td colspan="2">The description on the DMAC block transfer mode by a USB request is corrected and added in the H8S/2215 Group and the H8S/2218, H8S/2212 Group manuals</td><td><b>Information Category</b></td><td colspan="3">Technical Notification</td></tr>
<tr><td rowspan="2"><b>Applicable Product</b></td><td rowspan="2">H8S/2215 Group,
H8S/2218, H8S/2212 Group</td><td><b>Lot No.</b></td><td rowspan="2"><b>Reference Document</b></td><td colspan="3" rowspan="2">H8S/2215 Group Hardware Manual
Rev. 5.00 (REJ09B0140-0500O)
H8S/2218, H8S/2212 Group
Hardware Manual
Rev. 4.00 (REJ09B0074-0400O)</td></tr>
<tr><td>All lots</td></tr>
</table>

We would like to inform you of the following changes made in the single chip microcomputer H8S/2215 Group and the H8S/2218, H8S/2212 Group manuals. The changes involve error correction on the DMAC block transfer mode by a USB request and additions about DMAC auto-request of USB data.

1. H8S/2215 Group

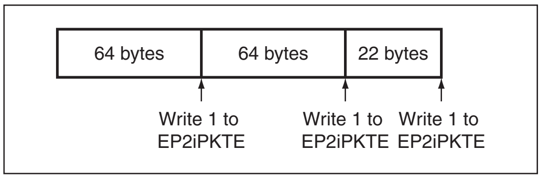
<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td rowspan="2">6.10.2 Bus Transfer Timing</td><td rowspan="2">146</td><td>(Incorrect)
In the case of a USB request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.</td></tr>
<tr><td>(Correct)
In case of a USB request in normal mode, and in short address mode or in cycle steal mode, the DMAC releases the bus after a single transfer.</td></tr>
<tr><td rowspan="2">7.3.4 DMA Control Register (DMACR)</td><td rowspan="2">159</td><td>(Incorrect)
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>3</td><td>DTF3</td><td>0</td><td>R/W</td><td>Data Transfer Factor</td></tr>
<tr><td>2</td><td>DTF2</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td>1</td><td>DTF1</td><td>0</td><td>R/W</td><td>In normal mode</td></tr>
<tr><td>0</td><td>DTF0</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>In block transfer mode
:
0011: Activated by $\overline{\text{DREQ}}$ signal’s low level input from USB (USB request)
:</td></tr>
</table>
</td></tr>
<tr><td>(Correct)
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>3</td><td>DTF3</td><td>0</td><td>R/W</td><td>Data Transfer Factor</td></tr>
<tr><td>2</td><td>DTF2</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td>1</td><td>DTF1</td><td>0</td><td>R/W</td><td>In normal mode</td></tr>
<tr><td>0</td><td>DTF0</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>In block transfer mode
:
0011: —
:</td></tr>
</table>
</td></tr>
</table>

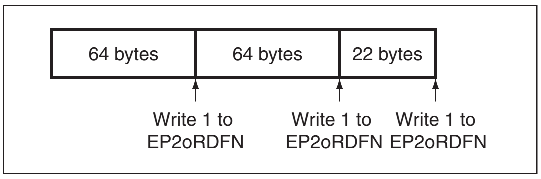
<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td>7.4.1 Transfer Modes
Table 7.2 DMAC Transfer Modes</td><td>170</td><td>(Incorrect)
<table>
<tr><th colspan="2">Transfer Mode</th><th>Transfer Source</th><th>Remarks</th></tr>
<tr><td>Full address mode</td><td>(4) Normal mode</td><td>• USB request
• Auto-request</td><td>• Max. 2-channel operation, combining channels A and B
• With auto-request, burst mode transfer or cycle steal transfer can be selected</td></tr>
<tr><td></td><td>(5) Block transfer mode</td><td>• TPU channel 0 to 2 compare match/input capture A interrupts
• SCI transmission complete interrupt
• SCI reception complete interrupt
• A/D conversion end interrupt
• USB request</td><td></td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>(Correct)
<table>
<tr><th colspan="2">Transfer Mode</th><th>Transfer Source</th><th>Remarks</th></tr>
<tr><td>Full address mode</td><td>(4) Normal mode</td><td>• USB request
• Auto-request</td><td>• Max. 2-channel operation, combining channels A and B
• With auto-request, burst mode transfer or cycle steal transfer can be selected</td></tr>
<tr><td></td><td>(5) Block transfer mode</td><td>• TPU channel 0 to 2 compare match/input capture A interrupt
• SCI transmission complete interrupt
• SCI reception complete interrupt
• A/D conversion end interrupt</td><td></td></tr>
</table>
</td></tr>
<tr><td>7.4.7 DMAC Activation Sources
Table 7.8 DMAC Activation Sources</td><td>187</td><td>(Incorrect)
<table>
<tr><th colspan="2" rowspan="2">Activation Source</th><th rowspan="2">Short Address Mode</th><th colspan="2">Full Address Mode</th></tr>
<tr><th>Normal Mode</th><th>Block Transfer Mode</th></tr>
<tr><td>USB request</td><td>Low level input of the $\overline{\text{DERQ}}$ signal</td><td>×</td><td>○</td><td>○</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>(Correct)
<table>
<tr><th colspan="2" rowspan="2">Activation Source</th><th rowspan="2">Short Address Mode</th><th colspan="2">Full Address Mode</th></tr>
<tr><th>Normal Mode</th><th>Block Transfer Mode</th></tr>
<tr><td>USB request</td><td>Low level input of the $\overline{\text{DERQ}}$ signal</td><td>×</td><td>○</td><td>×</td></tr>
</table>
</td></tr>
</table>

Item	Page	Revision
7.4.9 DMAC Bus Cycles (Dual Address Mode)	195	(Incorrect) Figure 7.21 shows an example of $\overline{\text{DREQ}}$ level activated block transfer mode transfer.  1 block transfer 1 block transfer Bus release DMA read DMA write DMA dead Bus release DMA read DMA write Bus release ϕ $\overline{\text{DREQ}}$ Address bus Transfer source Transfer destination Transfer source Transfer destination DMA control Idle Read Write Dead Idle Read Write Dead Idle Channel Request Request clear period Request Request clear period Minimum of 2 cycles Minimum of 2 cycles [1] [2] [3] [4] [5] [6] [7] Acceptance resumes Acceptance resumes [1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of ϕ, and the request is held. [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC. [3] [6] Start of DMA cycle. [4] [7] Acceptance is resumed after the dead cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of ϕ, and the request is held.) **Figure 7.21 Example of $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Transfer** $\overline{\text{DREQ}}$ signal sampling is performed every cycle, with the rising edge of the next ϕ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point. When the $\overline{\text{DREQ}}$ signal low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. Acceptance resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ signal low level sampling is performed again, and this operation is repeated until the transfer ends. Note: The $\overline{\text{DREQ}}$ signal of this chip is an internal signal of chip, so it is not output from the pin.
		(Correct) This page deleted.
15.3.3 USB DMAC Transfer Request Register (UDMAR)	494	(Incorrect) UDMAR is set when data transfer by means of on-chip DMAC is performed for data registers UEDR2i, UEDR2o, UEDR4i, and UEDR4o corresponding to EP2i, EP2o, EP4i, and EP4o used for Bulk transfer, respectively.
		(Correct) UDMAR is set when data transfer by means of a USB request of the on-chip DMAC is performed for data registers UEDR2i, UEDR2o, UEDR4i, and UEDR4o corresponding to EP2i, EP2o, EP4i, and EP4o used for Bulk transfer, respectively.
		(Added) Note: As the $\overline{\text{DREQ}}$ signal is not used in the data transfer by auto request of the on-chip DMAC, set UDMAR to H'00.

<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td rowspan="2">15.4 Interrupt Sources
Table 15.5 SCI Interrupt Sources</td><td rowspan="2">528</td><td>(Incorrect)<table><tr><th>Register</th><th>Bit</th><th>Transfer Mode</th><th>Interrupt Source</th><th>Description</th><th>Interrupt Request Signal</th><th>DMAC Activation</th></tr></table></td></tr>
<tr><td>(Correct)<table><tr><th>Register</th><th>Bit</th><th>Transfer Mode</th><th>Interrupt Source</th><th>Description</th><th>Interrupt Request Signal</th><th>DMAC Activation by USB Request*^{7}</th></tr></table></td></tr>
<tr><td rowspan="2">15.4 Interrupt Sources
Table 15.5 SCI Interrupt Sources</td><td rowspan="2">529</td><td>(Incorrect)
Notes: 1. EP0 interrupts must be assigned to the same interrupt request signal.
2. An EP2i DMA transfer request is specified by the EP2iT1 and EP2iT0 bits of UDMAR.
3. An EP2o DMA transfer request is specified by the EP2oT1 and EP2oT0 bits of UDMAR.
4. An EP4i DMA transfer request is specified by the EP4iT1 and EP4iT0 bits of UDMAR.
5. An EP4oDMA transfer request is specified by the EP4oT1 and EP4oT0 bits of UDMAR.
6. The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (IRQ6SCB, A = 01 in ISCRH) by the interrupt controller register.</td></tr>
<tr><td>(Correct)
Notes: 1. EP0 interrupts must be assigned to the same interrupt request signal.
2. An EP2i DMA transfer by a USB request is specified by the EP2iT1 and EP2iT0 bits of UDMAR.
3. An EP2o DMA transfer by a USB request is specified by the EP2oT1 and EP2oT0 bits of UDMAR.
4. An EP4i DMA transfer by a USB request is specified by the EP4iT1 and EP4iT0 bits of UDMAR.
5. An EP4o DMA transfer by a USB request is specified by the EP4oT1 and EP4oT0 bits of UDMAR.
6. The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (IRQ6SCB, A = 01 in ISCRH) by the interrupt controller register.
7. The $\overline{\text{DREQ}}$ signal is not used for auto-request. The CPU can activate the DMAC using any flags and interrupts.</td></tr>
</table>

Item	Page	Revision
15.6 DMA Transfer Specifications	560	(Incorrect) 15.6 DMA Transfer Specifications 15.6.1 Overview This module incorporates the interface that supports dual-address transfer by means of the on-chip DMAC. Endpoints that can be transferred by the on-chip DMAC are EP2 and EP4 in Bulk transfer (corresponding registers are UEDR2i, UEDR2o, UEDR4i, and UEDR4o).
		(Correct) 15.6 DMA Transfer Specifications Two methods of USB request and auto request are available for the DMA transfer of USB data. 15.6.1 DMA Transfer by USB Request (1) Overview Only normal mode in full address mode (cycle steal mode) supports the transfer by a USB request of the on-chip DMAC. Endpoints that can be transferred by the on-chip DMAC are EP2 and EP4 in Bulk transfer (corresponding registers are UEDR2i, UEDR2o, UEDR4i, and UEDR4o). : The section number of Overview was changed from 15.6.1 to (1) as shown above. The following sections were also changed as appropriate.

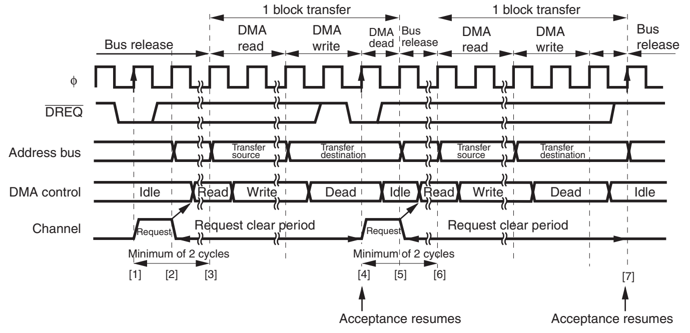
Item	Page	Revision
15.6.2 DMA Transfer by Auto-Request	562	(Added) (1) Overview Burst mode transfer or cycle steal transfer can be selected for the on-chip DMAC auto-request transfer. Endpoints that can be transferred by the on-chip DMAC are all registers (UEDR0s, UEDR0i, UEDR0o, UEDR1i, UEDR2i, UEDR2o, UEDR3i, UEDR3o, UEDR4i, UEDR4o, and UEDR5i). Confirm flags and interrupts corresponding to each data register before activating the DMA. As UDMAR is not used in auto-request mode, set UDMAR to H'00. (2) On-Chip DMAC Settings The on-chip DMAC must be specified as follows: Auto-request, byte size, full-address mode transfer, and number of transfers equal to or less than the maximum packet size of the data register. After completing the DMAC transfers of specified time, the DMAC automatically stops. (3) EPni DMA Transfer (n = 0 to 5) • EPniPKTE Bits of UTRG (n = 0 to 5) Note that 1 is not automatically written to EPniPKTE in case of auto-request transfer. Always write 1 to EPniPKTE by the CPU. The following example shows when 150-byte data is transmitted from EP2i to the host. In this case, 1 should be written to EP2iPKTE three times as shown in the figure below. • EP2i DMA Transfer Procedure The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint. 1. Confirm that UIFR1/EP2iEMPTY flag is 1. 2. DMAC settings for EP2i data transfer (such as auto-request and address setting). 3. Set the number of transfers for 64 bytes (the maximum packet size or less) in the DMAC. 4. Activate the DMAC (write 1 to DTE after reading DTE as 0). 5. DMA transfer. 6. Write 1 to the UTRG0/EP2iPKTE bit after the DMA transfer is completed. 7. Repeat steps 1 to 6 above. 8. Confirm that UIFR1/EP2iEMPTY flag is 1. 9. Set the number of transfer for 22 bytes in the DMAC. 10. Activate the DMAC (write 1 to DTE after reading DTE as 0). 11. DMA transfer. 12. Write 1 to the UTRG0/EP2iPKTE bit after the DMA transfer is completed.  64 bytes 64 bytes 22 bytes Write 1 to EP2iPKTE Write 1 to EP2iPKTE Write 1 to EP2iPKTE Additional Figure 1 EP2iPKTE Operation in UTRG0 (Auto-Request)

Item	Page	Revision
15.6.2 DMA Transfer by Auto-Request	562	(Added) (4) EPno DMA Transfer (n = 0, 2, 4) • EPnoRDFN Bits of UTRG (n = 0, 2, 4) Note that 1 is not automatically written to EPnoRDFN in case of auto-request transfer. Always write 1 to EPnoRDFN by the CPU. The following example shows when EP2o receives 150-byte data from the host. In this case, 1 should be written to EP2oRDFN three times as shown in the figure below. • EP2o DMA Transfer Procedure The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint. 1. Wait for the UIFR1/EP2oREADY flag to be set. 2. DMAC settings for EP2o data transfer (such as auto-request and address setting). Read value of UESZ2o and specify number of transfers to match size of received data (64 bytes or less). 3. Activate the DMAC (write 1 to DTE after reading DTE as 0). 4. DMA transfer (transfer of 64 bytes or less). 5. Write 1 to the UTRG0/EP2oRDFN bit after the DMA transfer is completed. 6. Repeat steps 1 to 5 above.  64 bytes 64 bytes 22 bytes Write 1 to EP2oRDFN Write 1 to EP2oRDFN Write 1 to EP2oRDFN Additional Figure 2 EP2oRDFN Operation in UTRG0 (Auto-Request)

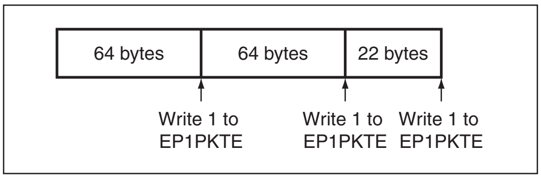
2. H8S/2218, H8S/2212 Group

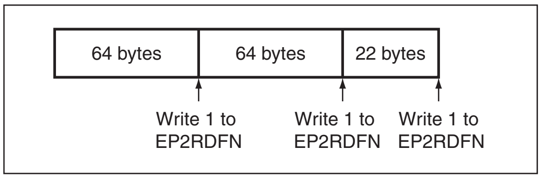
<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td rowspan="2">6.10.2 Bus Transfer Timing</td><td rowspan="2">152</td><td>(Incorrect)
In the case of a USB request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.</td></tr>
<tr><td>(Correct)
In case of a USB request in normal mode, and in short address mode or in cycle steal mode, the DMAC releases the bus after a single transfer.</td></tr>
<tr><td rowspan="2">7.3.4 DMA Control Register (DMACR)</td><td rowspan="2">166</td><td>(Incorrect)
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>3</td><td>DTF3</td><td>0</td><td>R/W</td><td>Data Transfer Factor</td></tr>
<tr><td>2</td><td>DTF2</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td>1</td><td>DTF1</td><td>0</td><td>R/W</td><td>In normal mode</td></tr>
<tr><td>0</td><td>DTF0</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>In block transfer mode</td></tr>
<tr><td></td><td></td><td></td><td></td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0011: Activated by DREQ signal's low level input from USB (USB request)</td></tr>
<tr><td></td><td></td><td></td><td></td><td>:</td></tr>
</table>
</td></tr>
<tr><td>(Correct)
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>3</td><td>DTF3</td><td>0</td><td>R/W</td><td>Data Transfer Factor</td></tr>
<tr><td>2</td><td>DTF2</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td>1</td><td>DTF1</td><td>0</td><td>R/W</td><td>In normal mode</td></tr>
<tr><td>0</td><td>DTF0</td><td>0</td><td>R/W</td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>In block transfer mode</td></tr>
<tr><td></td><td></td><td></td><td></td><td>:</td></tr>
<tr><td></td><td></td><td></td><td></td><td>0011: —</td></tr>
<tr><td></td><td></td><td></td><td></td><td>:</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td rowspan="2">7.4.1 Transfer Modes
Table 7.2 DMAC Transfer Modes</td><td rowspan="2">175</td><td>(Incorrect)
<table>
<tr><th colspan="2">Transfer Mode</th><th>Transfer Source</th><th>Remarks</th></tr>
<tr><td rowspan="2">Full address mode</td><td>(4) Normal mode</td><td>• USB request
• Auto-request</td><td rowspan="2">• Max. 2-channel operation, combining channels A and B
• With auto-request, burst mode transfer or cycle steal transfer can be selected</td></tr>
<tr><td>(5) Block transfer mode</td><td>• TPU channel 0 to 2 compare match/input capture A interrupts
• SCI transmission complete interrupt
• SCI reception complete interrupt
• A/D conversion end interrupt
• USB request</td></tr>
</table>
</td></tr>
<tr><td>(Correct)
<table>
<tr><th colspan="2">Transfer Mode</th><th>Transfer Source</th><th>Remarks</th></tr>
<tr><td rowspan="2">Full address mode</td><td>(4) Normal mode</td><td>• USB request
• Auto-request</td><td rowspan="2">• Max. 2-channel operation, combining channels A and B
• With auto-request, burst mode transfer or cycle steal transfer can be selected</td></tr>
<tr><td>(5) Block transfer mode</td><td>• TPU channel 0 to 2 compare match/input capture A interrupt
• SCI transmission complete interrupt
• SCI reception complete interrupt
• A/D conversion end interrupt</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">7.4.7 DMAC Activation Sources
Table 7.8 DMAC Activation Sources</td><td rowspan="2">192</td><td>(Incorrect)
<table>
<tr><th colspan="2" rowspan="2">Activation Source</th><th rowspan="2">Short Address Mode</th><th colspan="2">Full Address Mode</th></tr>
<tr><th>Normal Mode</th><th>Block Transfer Mode</th></tr>
<tr><td>USB request</td><td>Low level input of the $\overline{\text{DERQ}}$ signal</td><td>×</td><td>○</td><td>○</td></tr>
</table>
</td></tr>
<tr><td>(Correct)
<table>
<tr><th colspan="2" rowspan="2">Activation Source</th><th rowspan="2">Short Address Mode</th><th colspan="2">Full Address Mode</th></tr>
<tr><th>Normal Mode</th><th>Block Transfer Mode</th></tr>
<tr><td>USB request</td><td>Low level input of the $\overline{\text{DERQ}}$ signal</td><td>×</td><td>○</td><td>×</td></tr>
</table>
</td></tr>
</table>

Item	Page	Revision
7.4.9 DMAC Bus Cycles (Dual Address Mode)	199 to 200	(Incorrect) Figure 7.20 shows an example of $\overline{\text{DREQ}}$ level activated block transfer mode transfer.  1 block transfer 1 block transfer Bus release DMA read DMA write DMA dead Bus release DMA read DMA write Bus release ϕ DREQ Address bus Transfer source Transfer destination Transfer source Transfer destination DMA control Idle Read Write Dead Idle Read Write Dead Idle Channel Request Request clear period Request Request clear period Minimum of 2 cycles Minimum of 2 cycles [1] [2] [3] [4] [5] [6] [7] Acceptance resumes Acceptance resumes [1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of ϕ, and the request is held. [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC. [3] [6] Start of DMA cycle. [4] [7] Acceptance is resumed after the dead cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of ϕ, and the request is held.) **Figure 7.20 Example of $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Transfer** $\overline{\text{DREQ}}$ signal sampling is performed every cycle, with the rising edge of the next ϕ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point. When the $\overline{\text{DREQ}}$ signal low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. Acceptance resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ signal low level sampling is performed again, and this operation is repeated until the transfer ends. Note: The $\overline{\text{DREQ}}$ signal of this chip is an internal signal of chip, so it is not output from the pin.
		(Correct) This page deleted.
14.3.2 USB DMAC Transfer Request Register (UDMAR)	469	(Incorrect) UDMAR is set when data transfer by means of the on-chip DMAC is performed for data registers UEDR1 and UEDR2 corresponding to EP1 and EP2 respectively used for Bulk transfer.
		(Correct) UDMAR is set when data transfer by means of a USB request of the on-chip DMAC is performed for data registers UEDR1 and UEDR2 corresponding to EP1 and EP2 respectively used for Bulk transfer.
		(Added) Note: As the $\overline{\text{DREQ}}$ signal is not used in the data transfer by auto request of the on-chip DMAC, set UDMAR to H'00.

<table>
<tr><th>Item</th><th>Page</th><th>Revision</th></tr>
<tr><td>14.4 Interrupt Sources
Table 14.4 Interrupt Sources</td><td>494</td><td>(Correct)<table><tr><th>Register</th><th>Bit</th><th>Transfer Mode</th><th>Interrupt Source</th><th>Description</th><th>Interrupt Request Signal</th><th>DMAC Activation</th></tr></table></td></tr>
<tr><td></td><td></td><td>(Incorrect)<table><tr><th>Register</th><th>Bit</th><th>Transfer Mode</th><th>Interrupt Source</th><th>Description</th><th>Interrupt Request Signal</th><th>DMAC Activation by USB Request*[5]</th></tr></table></td></tr>
<tr><td></td><td>495</td><td>(Incorrect)
Notes: 1. EP0 interrupts must be assigned to the same interrupt request signal.
2. An EP1 DMA transfer request is specified by the EP1T1 and EP1T0 bits in UDMAR.
3. An EP2 DMA transfer request is specified by the EP2T1 and EP2T0 bits in UDMAR.
4. The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (IRQ6SCB and IRQ6SCA in ISCRH = 01) by the interrupt controller register.</td></tr>
<tr><td></td><td></td><td>(Correct)
Notes: 1. EP0 interrupts must be assigned to the same interrupt request signal.
2. An EP1 DMA transfer by a USB request is specified by the EP1T1 and EP1T0 bits in UDMAR.
3. An EP2 DMA transfer by a USB request is specified by the EP2T1 and EP2T0 bits in UDMAR.
4. The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (IRQ6SCB and IRQ6SCA in ISCRH = 01) by the interrupt controller register.
5. The $\overline{\text{DREQ}}$ signal is not used for auto-request. The CPU can activate the DMAC using any flags and interrupts.</td></tr>
<tr><td>14.6 DMA Transfer Specifications</td><td>519</td><td>(Incorrect)
14.6 DMA Transfer Specifications
14.6.1 Overview
This module incorporates the interface that supports dual-address transfer by means of the on-chip DMAC. Endpoints that can be transferred by the on-chip DMAC are EP1 and EP2 in Bulk transfer (corresponding registers are UEDR1 and UEDR2).</td></tr>
<tr><td></td><td></td><td>(Correct)
14.6 DMA Transfer Specifications
Two methods of USB request and auto request are available for the DMA transfer of USB data.
14.6.1 DMA Transfer by USB Request
(1) Overview
Only normal mode in full address mode (cycle steal mode) supports the transfer by a USB request ($\overline{\text{DREQ}}$ level activated) of the on-chip DMAC. Endpoints that can be transferred by the on-chip DMAC are EP1 and EP2 in Bulk transfer (corresponding registers are UEDR1 and UEDR2).
:
The section number of Overview was changed from 14.6.1 to (1) as shown above. The following sections were also changed as appropriate.</td></tr>
</table>

Item	Page	Revision
14.6.2 DMA Transfer by Auto-Request	521	(Added) (1) Overview Burst mode transfer or cycle steal transfer can be selected for the on-chip DMAC auto-request transfer. Endpoints that can be transferred by the on-chip DMAC are all registers (UEDR0s, UEDR0i, UEDR0o, UEDR1, UEDR2, and UEDR3). Confirm flags and interrupts corresponding to each data register before activating the DMA. As UDMAR is not used in auto-request mode, set UDMAR to H'00. (2) On-Chip DMAC Settings The on-chip DMAC must be specified as follows: Auto-request, byte size, full-address mode transfer, and number of transfers equal to or less than the maximum packet size of the data register. After completing the DMAC transfers of specified time, the DMAC automatically stops. (3) EP0i, EP1, and EP3 DMA Transfer • EPnPKTE Bits of UTRG0 (n = 0i, 1, 3) Note that 1 is not automatically written to EPnPKTE in case of auto-request transfer. Always write 1 to EPnPKTE by the CPU. The following example shows when 150-byte data is transmitted from EP1 to the host. In this case, 1 should be written to EP1PKTE three times as shown in the figure below. • EP1 DMA Transfer Procedure The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint. 1. Check that UIFR1/EP1EMPTY flag is 1. 2. DMAC settings for EP1 data transfer (such as auto-request and address setting). 3. Set the number of transfers for 64 bytes (the maximum packet size or less) in the DMAC. 4. Activate the DMAC (write 1 to DTE after reading DTE as 0). 5. DMA transfer. 6. Write 1 to the UTRG0/EP1PKTE bit after the DMA transfer is completed. 7. Repeat steps 1 to 6 above. 8. Confirm that the UIFR1/EP1EMPTY flag is 1. 9. Set the number of transfer for 22 bytes in the DMAC. 10. Activate the DMAC (write 1 to DTE after reading DTE as 0). 11. DMA transfer. 12. Write 1 to the UTRG0/EP1PKTE bit after the DMA transfer is completed.  64 bytes 64 bytes 22 bytes Write 1 to EP1PKTE Write 1 to EP1PKTE Write 1 to EP1PKTE Additional Figure 1 EP1PKTE Operation in UTRG0 (Auto-Request)

Item	Page	Revision
14.6.2 DMA Transfer by Auto-Request	521	(Added) (4) EP0o and EP2 DMA Transfer • EPnRDFN Bits of UTRG0 (n = 0o, 2) Note that 1 is not automatically written to EPnRDFN in case of auto-request transfer. Always write 1 to EPnRDFN by the CPU. The following example shows when EP2 receives 150-byte data from the host. In this case, 1 should be written to EP2RDFN three times as shown in the figure below. • EP2 DMA Transfer Procedure The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint. 1. Wait for the UIFR1/EP2READY flag to be set. 2. DMAC settings for EP2 data transfer (such as auto-request and address setting). Read value of UESZ2 and specify number of transfers to match size of received data (64 bytes or less). 3. Activate the DMAC (write 1 to DTE after reading DTE as 0). 4. DMA transfer (transfer of 64 bytes or less). 5. Write 1 to the UTRG0/EP2RDFN bit after the DMA transfer is completed. 6. Repeat steps 1 to 5 above.  64 bytes 64 bytes 22 bytes Write 1 to EP2RDFN Write 1 to EP2RDFN Write 1 to EP2RDFN Additional Figure 2 EP2RDFN Operation in UTRG0 (Auto-Request)