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: Sep.30.2008

RENESAS TECHNICAL UPDATE

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

Product Category	MPU&MCU		Document No.	TN-SH7-A676A/E	Rev.	1.00
Title	SH7764 on-chip SSI（DMAC portion）trouble and additional notes of hardware manual（Rev1.0)		Information Category	Technical Notification		
Applicable Product	R5S77640P300BG R5S77640D300BG R5S77640N300BG R5S77641P300BG R5S77641D300BG R5S77641N300BG	Lot No. All lots	Reference Document	SH7764 Group Hardware Manual (REJ09B0360-0100)		

Please be informed of SH7764 on-chip SSI (DMAC portion) trouble and additional notes of hardware manual (Rev1.0).

- Notes -

1. DMAC on SSI trouble phenomenon

There is a possibility to make following troubles if you restart DMAC on SSI (hereafter "DMAC") in receiving mode after it is reset by software reset (set DMRST bit of SSIDMCOR register).

Trouble phenomenon 1

DMAC transferring pointer backs to the top address of receiving buffer (which is allocated on RAM) before the pointer reaches its bottom address. Please refer figure.1. As the result, the read address of the received data which software calculates by SSIBLCNT and SSIBLNCNT counters will be different from the transferring address of the received data by DMAC.

As the system level trouble, it will make a difference of write and read pointers in the recording music data buffer. And the difference will make below troubles according to the amount of the difference.

Trouble phenomenon 2

DMAC transferring pointer is incremented/decremented to outside of receiving buffer range. As the result, DMAC transfers data to outside of receiving buffer. Please refer figure.1. If you copy the program to RAM, DMAC transfer reaches to the area and it might make a system hung-up.

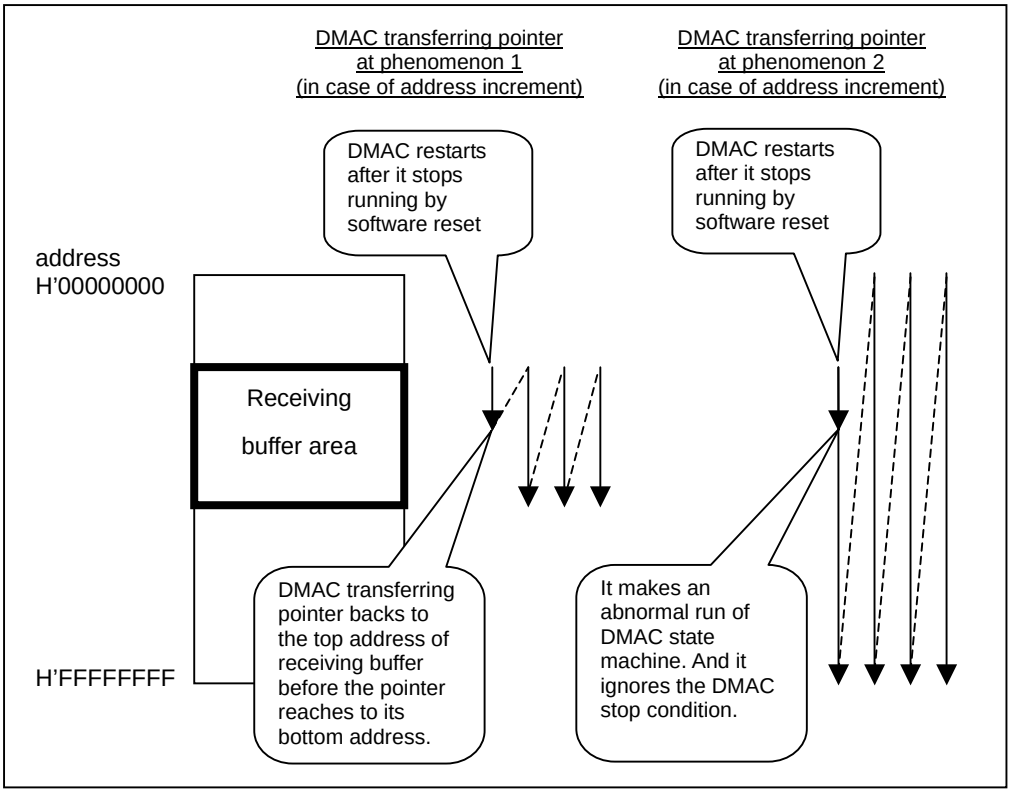

Figure.1 DMAC trouble phenomenon

2. Countermeasure of DMAC trouble

It is possible to make the countermeasure as followings operation, if you select 1-burst (8-byte) or 2-burst (16-byte) at WDMBSZ bit in SSIDMMR. Selecting 4-burst (32-byte) at this bit is prohibited.

(a) Clear RPTMD bit in SSIDMCOR

(b) Clear DMEN bit in SSIDMCOR

(c) Wait DMEND interruption or polling DMEND flag in SSIDMINTSR

(d) Clear DMEN bit in SSICR

(e) Clear EN bit in SSICR

(f) Set DMRST bit in SSIDMCOR

(g) Set RXRST bit in SSIDMCOR

(h) Clear RXRST bit in SSIDMCOR

(i) Restart SSI and SSIDMAC

Note 1: Please keep the order of each operation.

Note 2: It is okay to do (a) and (b) operations at same time.

Note 3: It is okay to set/clear TXRST bit too in (g) and (h) operations.

Note 4: It is okay to do (h) and (i) operations at same time.

Note 5: It is okay to put the other operations between each operation which does not relate to SSI and SSI-DMAC operations. Software wait routine is also okay to put.

Note: Issue of software reset makes no troubles when DMAC is not running or idle after above operation.

3. Additional notes of hardware manual (These will be written in the next edition of hardware manual.)

Additional note 1

SSIBLNCNT counter is cascaded with SSIBLCNT counter. Around the timing of SSIBLCNT counter produces a carry, there is a case that both counters do not match to the cascade process. In that case, the transferred block number is not calculated correctly, if you read SSIBLCNT counter and SSIBLNCNT counter only one time. As the countermeasure, if you read SSIBLNCNT counter before/after you read SSIBLCNT counter and you add a decision, you can calculate the transferred block number correctly in any case.

Please perform following operations continuously to calculate transferred block number by SSIBLCNT and SSIBLNCNT counters correctly.
Read SSIBLNCNT at 1st. Read SSIBLCNT at 2nd. Read SSIBLNCNT at 3rd. You can calculate transfer block number correctly by following algorism.

If 1st read value = 3rd read value
Transferred data number
= (3rd read value) x (SSIBLNCNTSR register value) x (SSIBLCNTSR register value)
+ (2nd read value) x (SSIBLCNTSR register value)

If 1st read value != 3rd read value
Transferred data number
= (3rd read value) x (SSIBLNCNTSR register value) x (SSIBLCNTSR register value)

Additional note 2

When you stop DMAC running by the operation of DMEN bit and RPTMD bit in SSIDMCOR register, please perform followings operation.

When DMAC is running in non-repeat mode (RPTMD bit=0 in SSIDMCOR register)
DMAC automatically stops when the remaining transfer word counter *1 becomes "0". At the timing, DMEND bit of SSIDMINTSR register is set.

When DMAC is running in repeat mode (RPTMD bit=1 in SSIDMCOR register)
(a) Please clear DMEN and RPTMD bits in SSIDMCOR register. After the clear, DMAC keeps running till the remaining transfer word counter *1 becomes "0". And it automatically stops when the counter becomes "0". But this stopping method needs to take care of below additional note 4.
(b) Or please clear only RPTMD bit in SSIDMCOR register. After the clear, DMAC keeps running till the remaining transfer word counter *1 becomes "0". And it automatically stops when the counter becomes "0".

*1 : This counter is non-accessible from software. It is reloaded the setting value of SSIWDMCNTR/SSIRDMCNTR at DMAC start timing or at the repeat timing in repeat mode. It is decremented at each DMAC transfer and becomes "0" when DMAC reaches the bottom of receiving/sending buffer.

Additional note 3

When you stop DMAC running in SSI sending mode by the operation of DMRST and TXRST bits in SSIDMCOR register, please perform the followings operation.

(a) Clear DMEN bit in SSIDMCOR
(b) Clear DMEN bit in SSICR
(c) Clear EN bit in SSICR
(d) Set DMRST bit in SSIDMCOR
(e) Set TXRST bit in SSIDMCOR
(f) Clear TXRST bit in SSIDMCOR

Note 1: Please keep the order of each operation.
Note 2: It is okay to do (b) and (c) operation at same time.
Note 3: It is okay to also set/clear RXRST bit at same time in (e) and (f) operation.
Note 4: It is okay to do (h) and (i) operations at same time.
Note 5: It is okay to put the other operations between each operation which does not relate to SSI and SSI-DMAC operations. Software wait routine is also okay to put.

Additional note 4

There is a lack of explanation about the note of SSIBLCNT counter, SSIBLNCNT counter, BLKEND and BLKNEND bits of SSIDMINTSR register.

During DMEN bit in SSIDMCOR is “0”, neither of SSIBLCNT and SSIBLNCNT is updated. And neither of BLKEND and BLKNEND bits in SSIDMINTSR is set. So even though DMAC is running till the remaining transfer word counter *1 becomes "0" after DMEN bit is cleared, neither of SSIBLCNT and SSIBLNCNT is updated and neither of BLKEND and BLKNEND bits in SSIDMINTSR is set.

If you restart DMAC in this condition, the counter over-flows because the SSIBLCNT counter has a bigger value than the setting value of SSIBLNCNTSR. And BLKEND bit in SSIDMINTSR will not bet set till the counter matches to the setting value of SSIBLNCNTSR again. So you need to initialize SSIBLCNT and SSIBLNCNT before you restart DMAC.

In the case that you stopped DMAC by clearing DMEN and RPTMD bits in SSIDMCOR at same timing (In the case that you stopped DMAC in repeat mode by method (a) in additional note 2), please perform DMAC software reset to initialize SSIBLCNT and SSIBLNCNT after DMEND bit in SSIDMINTSR is set.

*1 : This counter is non-accessible from software. It is reloaded the setting value of SSIWDMCNTR/SSIRDMCNTR at DMAC start timing and at the repeat timing in repeat mode. It is decremented at each DMAC transfer and becomes "0" when DMAC reaches the bottom of receiving/sending buffer.

Additional note 5

There is a lack of explanation about the initialization condition of IIRQ and IDST bits in SSISR register. The note in "Fig. 18.18 Transition Diagram between Operation Modes" is also lacked.

To initialize IIRQ and IDST bits, you need to make a clock on the serial bit clock line in SH7764 after a power-on-reset. These bits are not initialized by only the power-on-reset. There are following 2 methods to make a clock on the serial bit clock line in SH7764.

(a) You input a clock at SSISCK pin from outside of SH7764 after you assert SSISCK pin of the target of SSI channel by the port control register operation and select "serial clock = INPUT" at SCKD bit in ISSICR register

(b) You input a clock at AUDIO_CLK pin from outside of SH7764 after you assert AUDIO_CLK pin of the target of SSI channel by the port control register operation and select "serial clock = OUTPUT" at SCKD bit in ISSICR register.

As above, IDST bit in SSISR register is not initialized by only the power-on-reset. After the hardware reset, you do not need to confirm that IDST bit is the initialization value to move to SSI enable mode. This note is not written in "Fig. 18.18 Transition Diagram between Operation Modes".

End