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.

GRADE A

RENESAS TECHNICAL NEWS

No.ASSP-03-0309

M66291 Dn_FIFO Usage Limitations

Classification	Concerned Products
Corrections and supplementary explanation of document √ Notes Knowhow Others	USB ASSP M66291GP/M66291HP

1. Phenomenon

In a system that uses D0_FIFO or D1_FIFO (Dn_FIFO hereon), data that is read from Dn_EP (CPU bus side) or written to Dn_EP (data sent to USB bus) may be incorrect, as described below:

(1) When Dn_EP is OUT:
When the CPU bus reads Dn_EP, some of the data may be incorrect.

(2) When Dn_EP is IN:
Some of the Dn_EP send data on the USB bus may be incorrect.

2. Occurring Conditions

This limitation does not affect systems that do not use Dn_EP. In addition, even when using Dn_EP, in applications in which accesses to Dn_EP and another endpoint do not occur at the same time (example: when only using the default configuration specified in the sub class of the mass storage device class), this phenomenon will not occur and the limitation described here does not apply. Furthermore, if the system has an internal check for data compatibility in the upper protocol, even when this phenomenon does occur, the problem will be worked around in the transfer retry.

This phenomenon occurs when the conditions listed in Type (1) or Type (2) are present.

Type (1): When all of the following 3 conditions are present,and
Condition 1: the system uses both Dn_EP and CPU_EP,and
Condition 2: either one or both of Dn_EP and CPU_EP are set to the OUT direction,and
Condition 3: both [data receive complete on USB bus] and [CPU bus data read or write] occur simultaneously (indicated by (1) in Table 1).

Table 1: Cases corresponding to "Occurring Conditions Type (1)"

USB Side / CPU Side		EP0 OUT (Receive Complete)	CPU_EP OUT (Receive Complete)	D0_EP OUT (Receive Complete)	D1_EP OUT (Receive Complete)
EP0	IN(write)	-	-	-	-
	OUT(read)	-	-	-	-
CPU_EP	IN(write)	-	-	(1)	(1)
	OUT(read)	-	-	(1)	(1)
D0_EP	IN(write)	-	(1)	-	-
	OUT(read)	-	(1)	-	-
D1_EP	IN(write)	-	(1)	-	-
	OUT(read)	-	(1)	-	-

Type (2): When all of the following 3 conditions are present, and

Condition 1: the system uses Dn_EP, and

Condition 2: either a control-write transfer or a control-read transfer occurs in the system during a Dn_EP data transfer, and

Condition 3: ["data receive complete on USB bus" for Dn_EP or control-write transfer data stage] and [control transfer data stage, or CPU read from Dn_EP, or CPU write to Dn_EP CPU] occur simultaneously (indicated by (2) in Table 2).

Table 2: Cases corresponding to "Occurring Conditions Type (2)"

USB Side / CPU Side		EP0 OUT (Receive Complete)	CPU_EP OUT (Receive Complete)	D0_EP OUT (Receive Complete)	D1_EP OUT (Receive Complete)
EP0	IN(write)	-	-	(2)	(2)
	OUT(read)	-	-	(2)	(2)
CPU_EP	IN(write)	-	-	-	-
	OUT(read)	-	-	-	-
D0_EP	IN(write)	(2)	-	-	-
	OUT(read)	(2)	-	-	-
D1_EP	IN(write)	(2)	-	-	-
	OUT(read)	(2)	-	-	-

3. Solutions

(1)For applications that fit the descriptions in Occurring Conditions Types (1) and (2):

Use method #1 or #2 to work around the phenomenon.

#1 Assign the endpoint that is assigned to Dn_EP to CPU_EP, and then perform the transfer.

#2 While the CPU bus is accessing Dn_EP, set the CPU_EP PID and EP0 PID to NAK.
In addition, when the CPU bus accesses either the CPU_EP or EP0, set the Dn_EP PID to NAK.
When setting the PID to NAK, wait a period of 60us after setting NAK and then perform the data access via the CPU bus.

Note: Use Solution #1 for isochronous transfers.

(2) For applications that fit the descriptions in Occurring Conditions Type (1) only:
Use method #1, #2 or #3 to work around the phenomenon.
#1 Assign the endpoint that is assigned to Dn_EP to CPU_EP, and then perform the transfer.
#2 Assign the endpoint that is assigned to CPU_EP to Dn_EP, and then perform the transfer.
#3 While the CPU bus is accessing Dn_EP, set the CPU_EP PID to NAK. In addition, while the CPU bus is accessing EP0, set the Dn_EP PID to NAK.
When setting PID to NAK, wait a period of 60µs after setting NAK and then perform the data access via the CPU bus.

Note: Use Solution #1 or #2 for isochronous transfers.

(3) For applications that fit the descriptions in Occurring Conditions Type (2) only:
Use method #1 or #2 to work around the phenomenon.
#1 Assign the endpoint that is assigned to Dn_EP to CPU_EP, and then perform the transfer.
#2 While the CPU bus is accessing Dn_EP, set the EP0 PID to NAK (in other words, do not execute the control transfer data stage process). In addition, while the CPU bus is accessing EP0, set the Dn_EP PID to NAK.
When setting PID to NAK, wait a period of 60µs after setting NAK and then perform the data access via the CPU bus.

Note: Use Solution #1 for isochronous transfers.

4. Related Terminology

EP0: Endpoint 0
CPU_EP: The endpoint selected in the CPU_FIFO Select Register.
Dn_EP: The endpoint selected in the D0_FIFO or D1_DIDO Select Register.
IN: IN direction transfer
OUT:: OUT direction transfer
CPU bus: External bus that accesses M66291 from the external CPU or DMAC.