Microcontroller Technical Information

CP(K), O

32-Bit Microcontrollers V850E2/ME3 Usage Restrictions		Document No.	ZBG-CC-06-0039	1/2
		Date issued	November 2, 2006	
		Issued by	2nd Solution Development Group Multipurpose Microcomputer Systems Division 4th Systems Operations Unit NEC Electronics Corporation	
Related documents	Hardware User's Manual Architecture User's Manual	Notification classification	√	Usage restriction
				Upgrade
				Document modification
				Other notification

1. Affected products

V850E2/ME3

- μPD703500

2. New restrictions

The following restrictions (No. 1 and No. 2) have been newly found. For details, refer to the attachment.

No. 1 Restriction of DMA transfer by an internal USB interrupt

DMA transfer may not be correctly executed by a USB interrupt.

No. 2 USB-related registers cannot be accessed unless an external clock is supplied to the UCLK pin (the CPU must be reset because it keeps waiting for a response of a USB register that is not operating).

3. Workarounds

- Workaround for restriction No. 1

When starting DMA by a USB interrupt, set the DTFRn register to 0x7EH and execute DMA transfer in the single-step transfer mode (setting of 0x7EH is prohibited in the User's Manual, but it was confirmed that the operation is performed without problem even when the register is set to 0x7EH instead of 0x7FH).

- Workaround for restriction No. 2

If the clock is not supplied to UCLK, set a register (to be specified) that supplies the internal clock as the USB clock.

4. Action

The above restrictions No. 1 and No. 2 will be added to usage restrictions. We apologize for the inconvenience.

5. List of restrictions

The restriction history and detailed information is described in the attachment.

6. Document revision history

V850E2/ME3 - Usage Restrictions

Document Number	Date Issued	Description
ZBG-CC-06-0039 (Latest version)	November 2, 2006	Addition of new bug (No. 1 and 2)

List of Usage Restrictions in V850E2/ME3

1. Product Version

μPD703500: Rank K, E

* The rank is indicated by the letter appearing as the 5th digit from the left in the lot number marked on each product.

2. Product History

- μPD703500

No.	Bugs	Rank
		K, E
1	Restriction of DMA transfer by an internal USB interrupt	Δ
2	Restriction of USB register access when USB clock is not connected	Δ

√: Bug does not occur, Δ: Bug will also apply in future, ×: Bug occurs

3. Restriction Details

No. 1 Restriction of DMA transfer by an internal USB interrupt

[Description]

DMA transfer may not be correctly executed by a USB interrupt because of the combination of the following problems <1> and <2>.

Problem <1>

If a USB interrupt is specified (if the DTFRn register is set to 0x7FH) as a transfer trigger for two or more DMA channels, the USB interrupt source set for the DMA channel of the highest priority is transferred to the other DMA channel in the CPU, and all the DMA requests that set the DTFRn register to 0x7FH are generated.

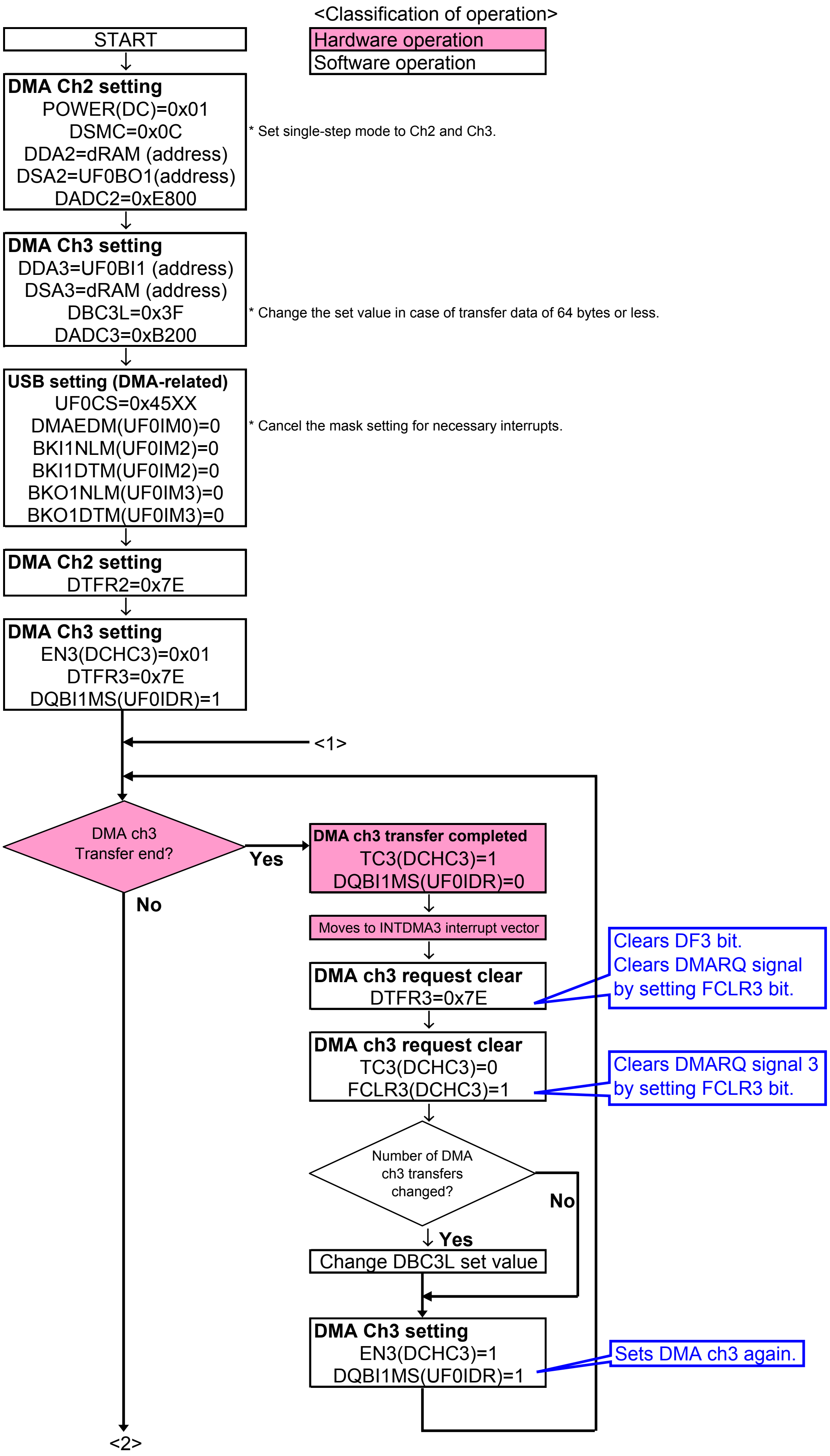
Problem <2>

If a DMA request for USB is generated in the CPU while DMA transfer for internal peripheral I/O is executed, the CPU cannot correctly control the DMA request and acknowledge the request; consequently, DMA transfer is not executed in some cases.

[Workaround]

When starting DMA by a USB interrupt, set the DTFRn register to 0x7EH and execute DMA transfer in the single-step transfer mode. This can avoid both of the above problems <1> and <2> (a flowchart is shown on the next page). The User's Manual prohibits setting of 0x7EH, but it was confirmed that the operation is performed without problem even when the register is set to 0x7EH instead of 0x7FH.

DMA transfer EP2 → dRAM (DMA: ch2)/dRAM → EP1 (DMA: ch3) (1/2)



* Set single-step mode to Ch2 and Ch3.

* Change the set value in case of transfer data of 64 bytes or less.

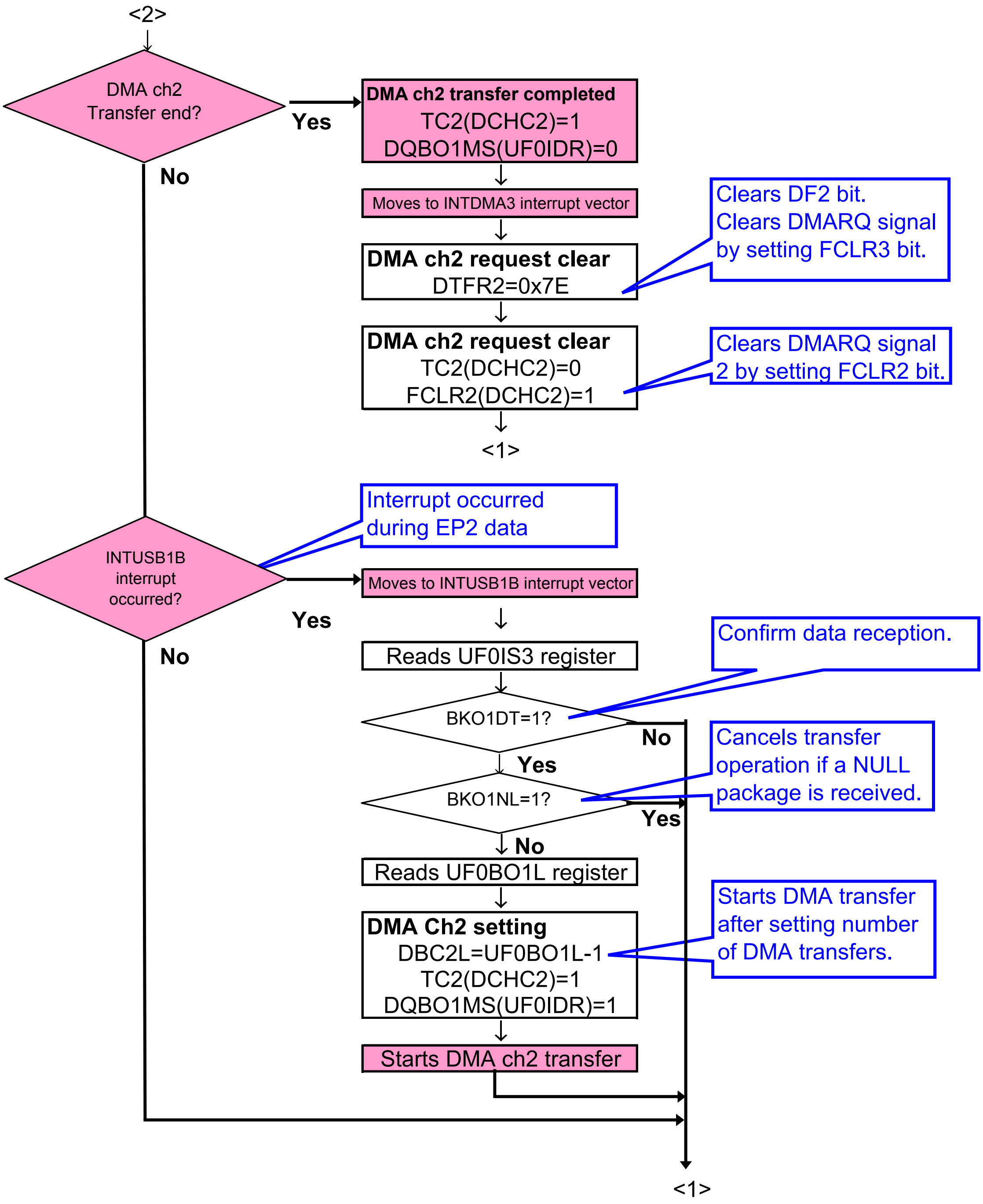
* Cancel the mask setting for necessary interrupts.

Clears DF3 bit.
Clears DMARQ signal by setting FCLR3 bit.

Clears DMARQ signal 3 by setting FCLR3 bit.

Sets DMA ch3 again.

DMA transfer EP2 → dRAM (DMA: ch2)/dRAM → EP1 (DMA: ch3) (2/2)



No. 2 Restriction of USB register access when USB clock is not connected

[Description]

When an external clock is not supplied to the UCLK pin*, the registers related to USB cannot be accessed (the CPU must be reset because it keeps waiting for a response of a USB register that is not operating).

* The external clock is not supplied to the UCLK pin in either of the following cases.

- The external clock is not connected.
- Before the PFC10 bit of the PFC1 register is switched to the UCLK input mode with the external clock connected

[Workaround]

The USB macro of the above product is of a configuration always requiring clock supply. As a register that supplies the internal clock of ME3 to the USB macro, therefore, the USB clock control register (UCKC) is stipulated. Set this register in the following procedure, depending on how the USB function is used.

[USB clock control register (UCKC)]

This register selects the USB clock source. When USB is not used, set the UCDIV1 bit to 1 to supply the internal clock. This register can be read or written in 8- or 1-bit units. To write this register, be sure to clear bits 0 and 2 to 7 to "0". The operation cannot be guaranteed if a value other than 0 is set to these bits.

Caution Be sure to supply the external clock when using USB. Do not use the USB function while the internal clock is supplied.

	7	6	5	4	3	2	1	0	Address	Reset value
UCKC	0	0	0	0	0	0	UCDIV1	0	FFFFF82EH	00H
	R	R	R	R	R	R	R/W	R		

UCDIV1	Selection of USB clock
0	External clock (supplied by UCLK pin input) (initial value)
1	Internal clock

- When USB is not used

  Set the UCKC register in the following procedure in the initialization routine.

  <1> UCKC.UCDIV1 = 1; Sets USB clock as internal clock.
  <2> After that, execute other initialization processing.

- When USB is used

  The above register does not have to be set.
  As described in the User's Manual, however, set the PFC1 and PMC1 registers using the following procedure to enable supply of the external clock, and then use the USB function (including accesses to registers related to USB).

  <1> PM1. PM10 = 1; Sets P10 in the input mode[Note].
  <2> PFC1.PFC10 = 1; Sets the UCLK pin in the input mode.
  <3> PMC1.PMC10 = 1; Specifies the alternate function of the P10 pin as the operation mode.
  <4> After that, execute other initialization processing.

  Note The PM1 register does not have to be set after reset, because its default value is FFH.