Microcontroller Technical Information

RX78K0R 78K0R Real-Time OS Document Modifications	Document No.	ZBG-CD-08-0053	1/1
	Date issued	December 17, 2008	
	Issued by	Development Tool Solution Group Multipurpose Microcomputer Systems Division Microcomputer Operations Unit NEC Electronics Corporation	
Related documents RX78K0R Functionalities: U18317EJ2 (2nd) RX78K0R Internal Structure: U18318EJ2 (2nd) RX78K0R Task Debugger: U18454EJ2 (2nd) AZ78K0R System Performance Analyzer: U18802EJ1 (1st)	Notification classification		Usage restriction
			Upgrade
		√	Document modification
			Other notification

1. Affected product

   RX78K0R Ver. 4.20

2. Affected documents

   - RX78K0R Functionalities (Document number: U18317EJ2V0)
   - AZ78K0R System Performance Analyzer (Document number: U18802EJ1V0)

3. Modifications

   See the attachment.

4. Action

   - RX78K0R Functionalities:

     Will be revised in January 2009.

   - AZ78K0R System Performance Analyzer:

     Will not be revised. Use this document together with the user's manual.

Modifications in RX78K0R User's Manual

1. Modifications

No.	Modified Items	Document Number	Page
1	Exclusion of reset and software interrupts from the interrupts that activate interrupt handlers	U18317EJ2V0	97
2	Correction of the interrupt status when the interrupt handler is activated	U18317EJ2V0	102, 103
3	Deletion of MINICUBE2 from AZ78K0R execution environment	U18802EJ1V0	15

2. Modification Details

No. 1 Exclusion of reset and software interrupts from the interrupts that activate interrupt handlers

- Document: *RX78K0R Functionalities* (U18317EJ2V0)
- Location: *10.1 Outline* on page 97

[Description]
Reset and software interrupts have been deleted from the description because they cannot be assigned to interrupt handlers.

Before change:

The RX78K0R provides as interrupt management functions related to the interrupt handlers activated when an interrupt (maskable interrupt, software interrupt, reset interrupt) is occurred.

After change:

The RX78K0R provides as interrupt management functions related to the interrupt handlers activated when a maskable interrupt is occurred.

No. 2 Correction of the interrupt status when the interrupt handler is activated

- Document: *RX78K0R Functionalities* (U18317EJ2V0)
- Location: (1) *10.3.3 Internal processing of interrupt handler* on page 102
  (2) *10.4 Multiple Interrupts* on page 103

(1) *10.3.3 Internal processing of interrupt handler* on page 102

The interrupt status when an interrupt handler is activated has been corrected.

Before change:

- Interrupt status

Maskable interrupt acknowledgement is prohibited in the RX78K0R when control is passed to the interrupt handler. To change (enable) the interrupt status in the interrupt handler, writing of #pragma EI directive and calling of the EI function are therefore required.

After change:

- Interrupt status

The RX78K0R goes into the following state when passing control to an interrupt handler.

Consequently, after control has passed to an interrupt handler, if an interrupt occurs with a higher precedence than the current level, then multiple interrupts can be processed.

- Acceptance of maskable interrupts is permitted
- Interrupts with the precedence below are disabled

A level-2 interrupt handler process is ongoing: ISP1 = 0, ISP = 1

A level-3 interrupt handler process is ongoing: ISP1 = 1, ISP = 0

Note It is not possible to define Interrupt handlers as level 0 or 1.

Note Even if the acceptance of maskable interrupts is disabled inside an interrupt handler (IE = 0), it will be enabled (IE = 1) after control returns from the interrupt handler.

(2) *10.4 Multiple Interrupts* on page 103

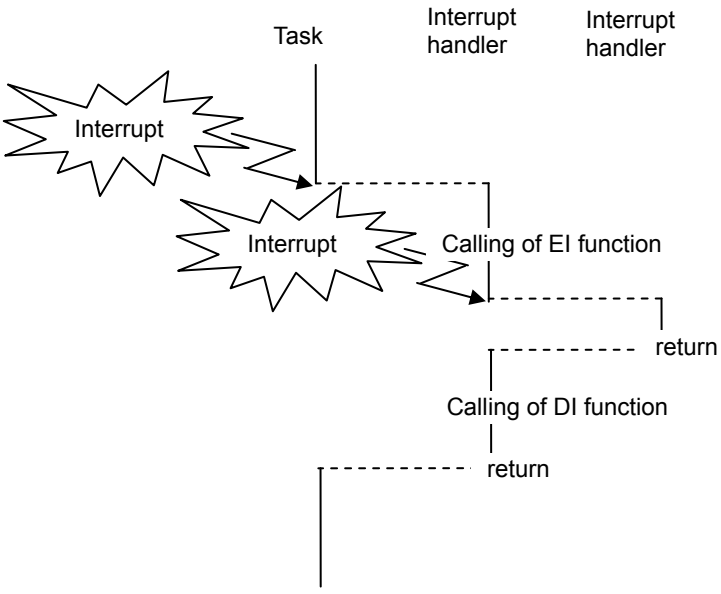
Calling of the EI function, which was required for generating multiple interrupts, is no longer required.

Before change:

The reoccurrence of an interrupt within an interrupt handler is called "multiple interrupt". However, since acknowledgment of a maskable interrupt is prohibited in the RX78K0R when control is passed to an interrupt handler, in order to create multiple interrupts, it is necessary to explicitly change the interrupt state within the interrupt handler (enable: #pragma EI directive code and EI function call).

The following shows the flow of the processing for handling multiple interrupts.

Figure 10-2 Multiple Interrupts

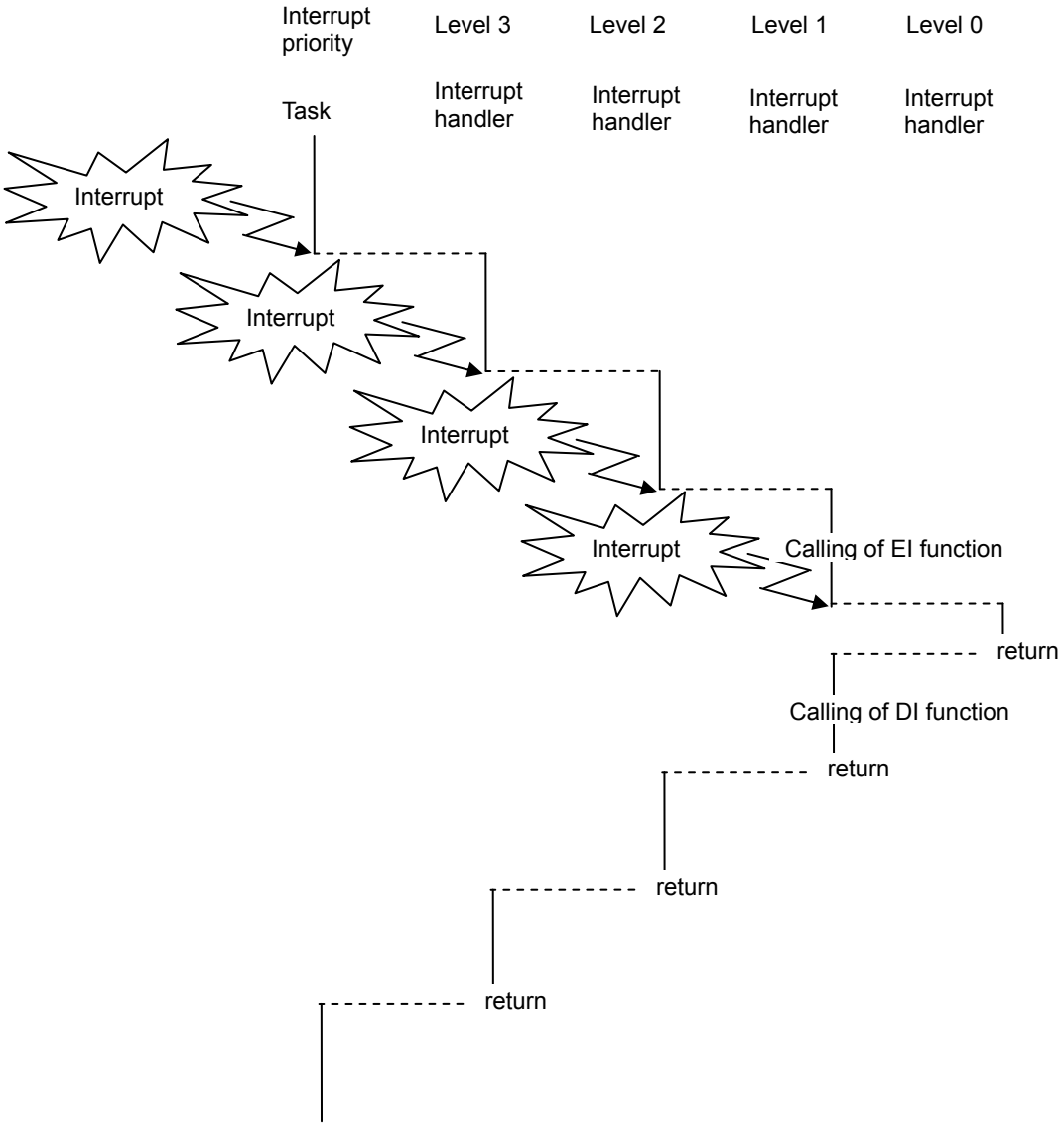


After change:

The reoccurrence of an interrupt within an interrupt handler is called "multiple interrupt".

The following shows the flow of the processing for handling multiple interrupts.

Figure 10-2 Multiple Interrupts



When control moves to an interrupt handler, then the state changes to acceptance of maskable interrupts enabled ("IE = 1"). For this reason, multiple interrupts are generally accepted from interrupt handlers. Multiple interrupts are likewise accepted from timer interrupts and cyclic handlers called from them.

When control moves to an interrupt process, then the state changes to acceptance of maskable interrupts disabled (because the RX78K0R does not mediate, the behavior is in accordance with that of the 78K0R microcontroller). For this reason, multiple interrupts are generally not accepted from interrupt processes. To enable the acceptance of multiple interrupts, it is necessary to call the EI function from the interrupt process. It is not allowed to accept multiple interrupt handlers from an interrupt process, and behavior is not guaranteed if this occurs.

If a user application enables multiple interrupts, then it is necessary to set the interrupt level of the interrupt handler/process as shown below.

Table 10-2 Settable Interrupt Level (Enabling Multiple Interrupts From the User Application)

	Interrupt Handler	Interrupt Servicing
Interrupt level 0	Not available	Available
Interrupt level 1	Not available	Available
Interrupt level 2	Available	Not available
Interrupt level 3	Available	Not available

If a user application disables multiple interrupts, then it is necessary to set the interrupt level of the interrupt handler/process to one of the patterns shown below.

Pattern 1: Set the level of all interrupt handlers and interrupt processes to 2.

Pattern 2: Set the level of all interrupt handlers and interrupt processes to 2.

Pattern 3: Set the level of all interrupt handlers to 2, and the level of all interrupt processes to either 2 or 3. Interrupts are disabled during an interrupt process with an interrupt level of 3 (IE = 0).

Table 10-3 Settable Interrupt Level (Disabling Multiple Interrupts From the User Application)

	Pattern 1		Pattern 2		Pattern 3	
	Interrupt Handler	Interrupt Servicing	Interrupt Handler	Interrupt Servicing	Interrupt Handler	Interrupt Servicing
Interrupt level 0	Not available	Not available	Not available	Not available	Not available	Not available
Interrupt level 1	Not available	Not available	Not available	Not available	Not available	Not available
Interrupt level 2	Available	Available	Not available	Not available	Available	Available
Interrupt level 3	Not available	Not available	Available	Available	Not available	Available (*)

(*) Interrupts are disabled during this interrupt process (IE = 0).

No. 3 Deletion of MINICUBE2 from AZ78K0R execution environment

- Document: *AZ78K0R System Performance Analyzer* (U18802EJ1V0)
- Location: *1.4 Execution Environment* on page 15

[Description]

Descriptions related to the on-chip debug emulator MINICUBE2 have been deleted because the AZ78K0R does not operate with MINICUBE2.

Deleted description:

- On-chip debug emulator

  MINICUBE2 (from NEC Electronics Corporation)

[Note] On-chip debug emulators other than the above can be connected to the AZ78K0R, as long as they support TIP.