

Product Change Notice (PCN)

Subject: RA2E2 group I3C function revision

Publication Date: 2/24/2022

Effective Date: 7/1/2022

Revision Description: Initial Release

Description of Change:

The applicable products (MCU Ver.1) listed in the Affected Product List have restrictions and precautions on the I²C / I3C bus interface.

Regarding the restrictions and precautions of MCU Ver.1, the workarounds and precautions have already been described in RA2E2 Group User's Manual Hardware Rev.1.00 (R01UH0919EJ0100). However, in order to improve the function of the I²C / I3C bus interface, the products were revised and the MCU version was changed from MCU Ver.1 to MCU Ver.2. For details of the changes, please refer to "Appendix, Details of changes".

The software workarounds of MCU Ver.1 can be executed on MCU Ver.2 without any changes.

Affected Product List:

Applicable products : RA2E2 Group

R7FA2E2A72DNK#AA0	R7FA2E2A72DNK#HA0	R7FA2E2A73CNK#AA0	R7FA2E2A73CNK#HA0
R7FA2E2A74CNK#AA0	R7FA2E2A74CNK#HA0	R7FA2E2A52DNK#AA0	R7FA2E2A52DNK#HA0
R7FA2E2A53CNK#AA0	R7FA2E2A53CNK#HA0	R7FA2E2A54CNK#AA0	R7FA2E2A54CNK#HA0
R7FA2E2A32DNK#AA0	R7FA2E2A32DNK#HA0	R7FA2E2A33CNK#AA0	R7FA2E2A33CNK#HA0
R7FA2E2A34CNK#AA0	R7FA2E2A34CNK#HA0	R7FA2E2A72DNJ#AA0	R7FA2E2A72DNJ#HA0
R7FA2E2A73CNJ#AA0	R7FA2E2A73CNJ#HA0	R7FA2E2A74CNJ#AA0	R7FA2E2A74CNJ#HA0
R7FA2E2A52DNJ#AA0	R7FA2E2A52DNJ#HA0	R7FA2E2A53CNJ#AA0	R7FA2E2A53CNJ#HA0
R7FA2E2A54CNJ#AA0	R7FA2E2A54CNJ#HA0	R7FA2E2A32DNJ#AA0	R7FA2E2A32DNJ#HA0
R7FA2E2A33CNJ#AA0	R7FA2E2A33CNJ#HA0	R7FA2E2A34CNJ#AA0	R7FA2E2A34CNJ#HA0

Reason for Change:

Improved I²C / I3C bus interface function

Impact on Package Dimensions, Function, Quality & Reliability:

No impact other than changing the I²C / I3C bus interface function.

Product Identification:

As shown in the example below, the last digit of the booking part numbers are changed from 0 to 1 between MCU Ver.1 and MCU Ver.2. Therefore, they can be distinguished by the booking part numbers.

In case of R7FA2E2A72DNK

Product part number	Packing	Booking part number	
		MCU Ver.1	MCU Ver.2
R7FA2E2A72DNK	Tray	R7FA2E2A72DNK#AA0	R7FA2E2A72DNK#AA1
	T&R	R7FA2E2A72DNK#HA0	R7FA2E2A72DNK#HA1

In addition, since the MCU version information stored in the MCU version register (MCUVER) is changed as follows, it can be identified by software.

MCU Ver.1: MCUVE[7:0]=01h

MCU Ver.2: MCUVE[7:0]=02h

Qualification Status: N/A

Sample Availability Date: 2/28/2022

The following ES samples are prepared as samples for checking the function change of I²C/I³C. There is no difference in I²C/I³C functions between mass-produced products and the ES samples.

Booking part number	Package	Packing	Code Flash	Operating temperature	Sample Availability Date
R7FA2E2A72DNK#YK2	24-pin HWQFN	Tray	64KB	-40 to +85°C	2022/2/28
R7FA2E2A72DNJ#YK2	20-pin HWQFN	Tray	64KB	-40 to +85°C	2022/4/1

Device Material Declaration: Please contact our sales and distributors.



PCN#[MCP-AC-22-0010]

Note:

1. Acknowledgement must be received by Renesas within 30 days or Renesas will consider the change as approved.
2. If timely acknowledgement is provided by Customer, then Customer shall have 90 days from the date of receipt of this PCN to make any objections to this PCN. If Customer fails to make objections to this PCN within 90 days of the receipt of the PCN then Renesas will consider the PCN changes as approved.
3. If customer cannot accept the PCN then customer must provide Renesas with a last time buy demand and purchase order.

For additional information regarding this notice, please contact your Renesas sales representative.

Appendix. Details of changes:

The restrictions / precautions of MCU Ver.1 and the changes of MCU Ver.2 are as follows.

#	Operation mode	Restrictions / precautions of MCU Ver.1
1	I ² C master	Operation when writing transmission data
2	I ² C slave	Timeout count operation when 10-bit address communication and the upper address match is detected.
3	I ² C slave	ACK output operation when 10-bit address communication and the lower address mismatch is detected.
4	I3C slave	Arbitration operation when Dynamic Address is assigned by the ENTDAAC command
5	I3C master / I3C slave	Error recovery operation
6	I3C master	Error recovery operation when receiving irregular data in I3C master receive mode
7	I3C master / I3C slave	Operation when using unsupported Common Command Code (CCC)

#1 : I²C master; Operation when writing transmission data

In MCU Ver.1, when the output timing of the first data of the frame and the transmission data write timing are conflicted, the transmission of the first data to be transmitted (b7) will start with a deviation of 1 bit.

MCU Ver.1 requires the software workaround when writing transmission data, but MCU Ver.2 does not require the software workaround.

There is no problem even if the same software workaround of MCU Ver.1 is executed on MCU Ver.2.

Before the changes : User's Manual: Hardware Rev.1.00 P699-P610

Note: The following processing is required when checking NTST.TDBEF0 = 1 in Steps [3] and [4] of the I²C master transmission flowchart as shown in Figure 25.104.

When sending a slave address:

After confirming NTST.TDBEF0 = 1, confirm that PRSTDBG.SCILV = 0 (check the status of SCL) before writing the transmission data.

When sending data:

- If BITCNT.BCNT = other than 0 after confirming NTST.TDBEF0 = 1, write the transmission data immediately
- If BITCNT.BCNT = 0 after confirming NTST.TDBEF0 = 1, check that BST.TENDF = 1 and PRSTDBG.SCILV = 0 (check the status of SCL) before writing the transmission data.

After the changes :

Note: [MCU Ver.1 has the following restrictions and workarounds. These restrictions and workarounds are not required for MCU Ver.2.](#)

The following processing is required when checking NTST.TDBEF0 = 1 in Steps [3] and [4] of the I²C master transmission flowchart as shown in Figure 25.104.

When sending a slave address:

After confirming NTST.TDBEF0 = 1, confirm that PRSTDBG.SCILV = 0 (check the status of SCL) before writing the transmission data.

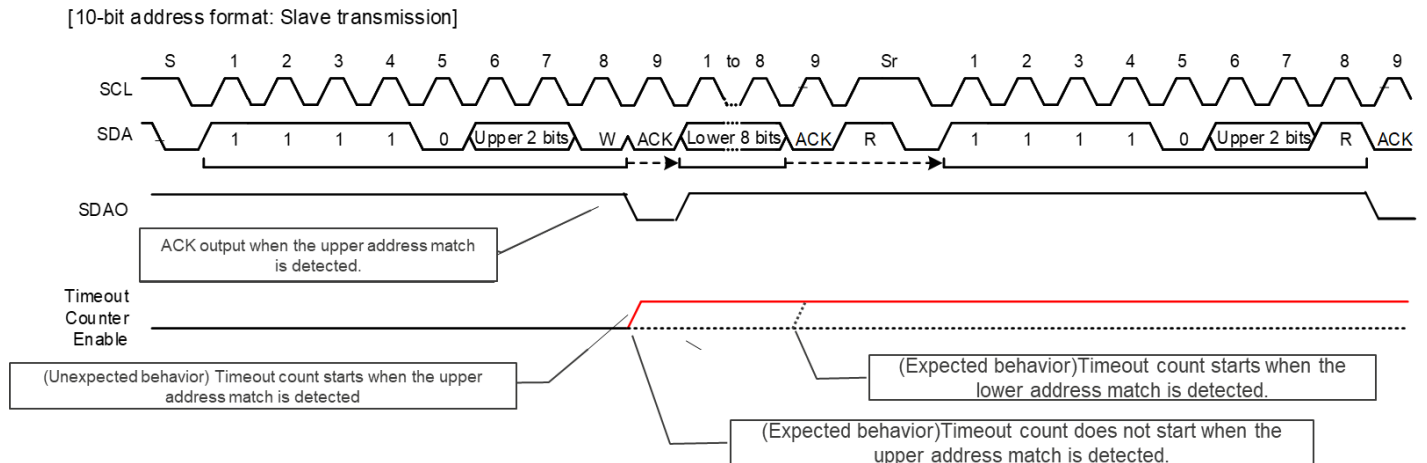
When sending data:

- If BITCNT.BCNT = other than 0 after confirming NTST.TDBEF0 = 1, write the transmission data immediately
- If BITCNT.BCNT = 0 after confirming NTST.TDBEF0 = 1, check that BST.TENDF = 1 and PRSTDBG.SCILV = 0 (check the status of SCL) before writing the transmission data.

#2 : I²C slave; Timeout count operation when 10-bit address communication and the upper address match is detected

In MCU Ver.1, the timeout count starts when 10-bit address communication and the upper address match is detected.

MCU Ver.2 is revised the circuit so that the timeout count does not start until the lower address match is detected after the upper address match is detected.



Before the changes : User's Manual: Hardware Rev.1.00 P558

TOMDS[1:0] bits (Timeout Operation Mode Selection)

These bits are used to select the detection condition for timeout when the timeout function is enabled.

Note: When working with I²C Slave, during 10-bit address communication, the timeout count starts when the upper address match is detected.

After the changes :

TOMDS[1:0] bits (Timeout Operation Mode Selection)

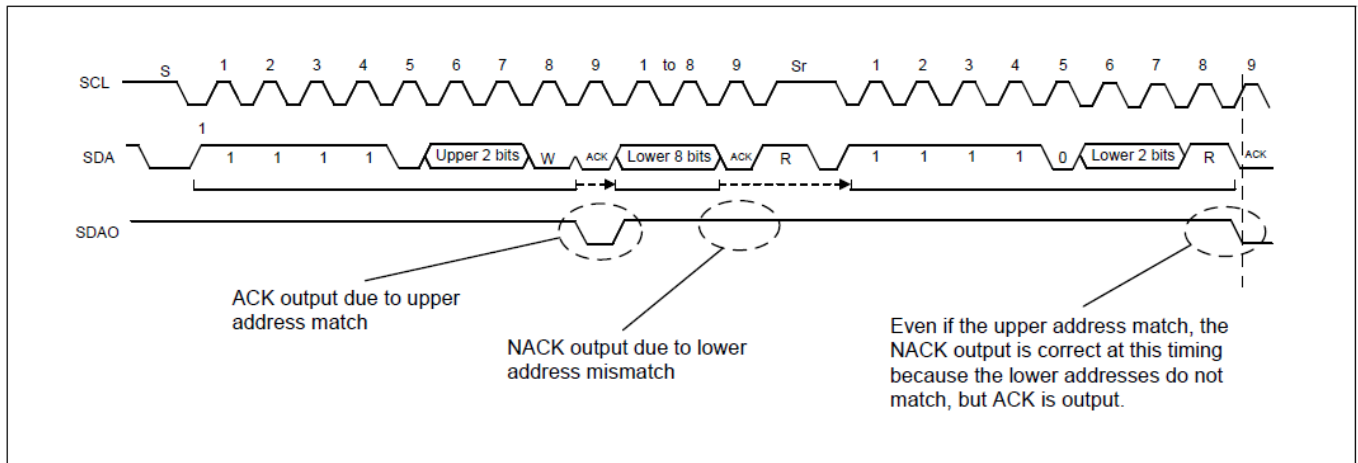
These bits are used to select the detection condition for timeout when the timeout function is enabled.

Note: **MCU Ver.1 has the following restriction. The restriction is not required for MCU Ver.2.**
When working with I²C Slave, during 10-bit address communication, the timeout count starts when the upper address match is detected.

#3 : I²C slave; ACK output operation when 10-bit address communication and the lower address mismatch is detected

MCU Ver.1 has the following restrictions and workarounds when using 10-bit address communication. The restrictions and workarounds are not required for MCU Ver.2

Before the changes : User's Manual: Hardware Rev.1.00 P641

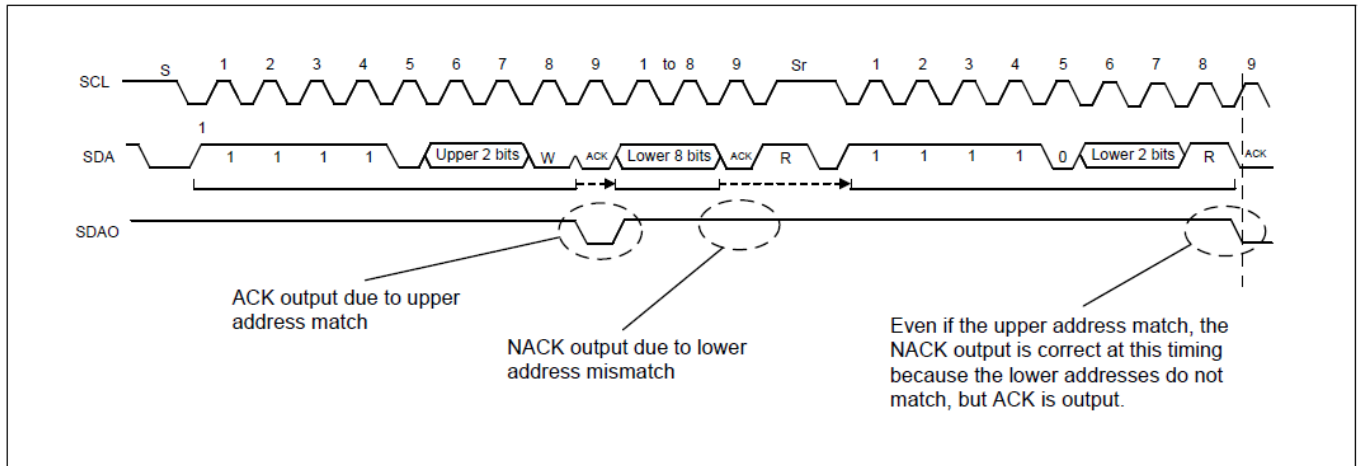


When multiple I²C slaves are connected to the I²C bus and there is a possibility that an I²C Slave other than this module will NACK for the upper address / R after Repeated START, there are the following restrictions and workarounds.

- Work around : Set the upper 2 bits of the 10-bit address assigned to this module to a value different from other Slave. If the address is exhausted and cannot be set to a different value, use restriction (1).
- Restriction (1) : 10-bit address not available.
- Restriction (2) : After the ACK response in the red circle in the above figure, none of the slaves respond to data, so the SDA keeps the high level and the I²C Master receives the 0xFF data. In the case of a system that can handle 0xFF as abnormal data, 0xFF is read and discarded on the I²C Master side. If 0xFF is valid data, use restriction (1).

After the changes :

MCU Ver.1 has the following restrictions and workarounds. The restrictions and workarounds are not required for MCU Ver. 2.

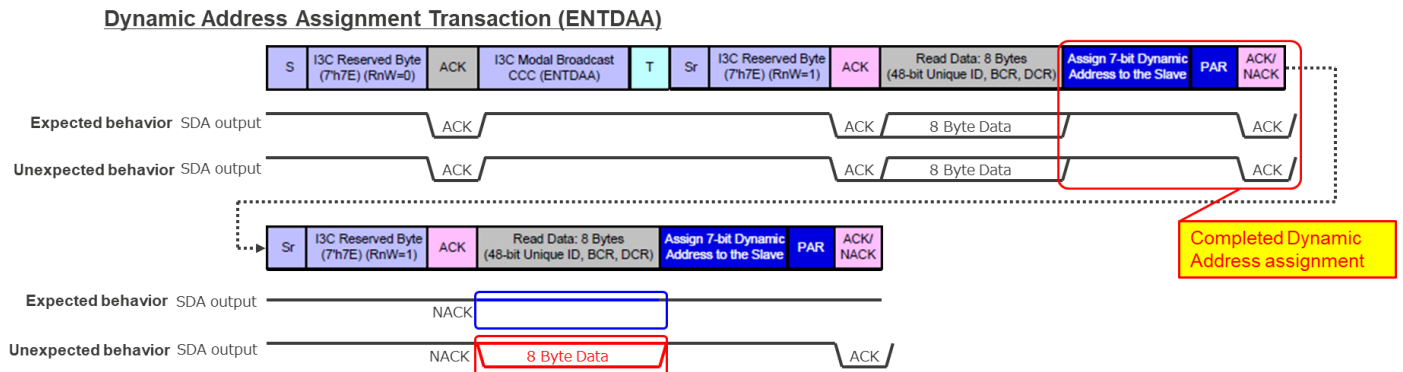


When multiple I²C slaves are connected to the I²C bus and there is a possibility that an I²C Slave other than this module will NACK for the upper address / R after Repeated START, there are the following restrictions and workarounds.

- Work around : Set the upper 2 bits of the 10-bit address assigned to this module to a value different from other Slave. If the address is exhausted and cannot be set to a different value, use restriction (1).
- Restriction (1) : 10-bit address not available.
- Restriction (2) : After the ACK response in the red circle in the above figure, none of the slaves respond to data, so the SDA keeps the high level and the I²C Master receives the 0xFF data. In the case of a system that can handle 0xFF as abnormal data, 0xFF is read and discarded on the I²C Master side. If 0xFF is valid data, use restriction (1).

4 : I3C slave; Arbitration operation when Dynamic Address is assigned by the ENTDAAs command

In MCU Ver.1, if a dynamic address is assigned by the ENTDAAs command in I3C slave mode, I3C will participate in dynamic address arbitration in response to the ACK response of other slaves even after the dynamic address assignment.



MCU Ver.1 requires the restrictions and workarounds when assigning dynamic address, but the restrictions and workarounds are not required for MCU Ver.2.

There is no problem even if the same software workaround of MCU Ver.1 is executed on MCU Ver.2.

Before the changes : User's Manual: Hardware Rev.1.00 P641

For RA2E2, the version that has not been modified by ECO has the following restrictions and workarounds. This restriction / workaround is not required for the version modified by ECO.

Note: When multiple I3C (I3C Slave) are connected on the I3C Bus assign Dynamic Addresses in the following order.

1. Set the SDCTPIDH and SDCTPIDL registers (6 Bytes) of the I3C (I3C Slave) to a value (All 1 etc.) that has a lower priority than other Slave Devices by Dynamic Address Arbitration.
2. After setting the Static Address in the I3C (I3C Slave), assign the Dynamic Address using the SETDASA / SETAASA command.
3. Assign a Dynamic Address to an I3C Slave Device other than the I3C (I3C Slave) using the ENTDAAs command.

After the changes :

MCU Ver.1 has the following restriction and workaround. This restriction and workaround are not required for MCU Ver.2.

Note: When multiple I3C (I3C Slave) are connected on the I3C Bus assign Dynamic Addresses in the following order.

1. Set the SDCTPIDH and SDCTPIDL registers (6 Bytes) of the I3C (I3C Slave) to a value (All 1 etc.) that has a lower priority than other Slave Devices by Dynamic Address Arbitration.
2. After setting the Static Address in the I3C (I3C Slave), assign the Dynamic Address using the SETDASA / SETAASA command.
3. Assign a Dynamic Address to an I3C Slave Device other than the I3C (I3C Slave) using the ENTDAAs command.

5 : I3C master / I3C slave; Error recovery operation

In order to improve the complexity of the error recovery flow, MCU Ver.2 is revised the circuit so that I3C will be suspended (BCTL.RSM becomes 1) when an error occurs. After I3C is suspended, the application must write the value 1 to the BCTL.RSM bit to resume I3C operation and recover form suspended state.

In MCU Ver.2, the error recovery flow shown in Figures 25.96 and Figures 25.97 are simplified by the above improvements.

There is no problem even if the same error recovery flow of MCU Ver.1 is executed on MCU Ver.2.

Before the changes : User's Manual: Hardware Rev.1.00 P690-P692

25.3.2.4.6 Error Recovery Operation [I3C mode]

When an error occurs, the INST.INEF, NTST.TEF, NTST.TABTF, HTST.TEF and HTST.TABTF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled.)

There is a possibility of communication error or internal module error.

The I3C master must perform an error recovery flow according to the following case:

- When TEF is detected.

Figure 25.96 and Figure 25.97 show the error recovery flow.

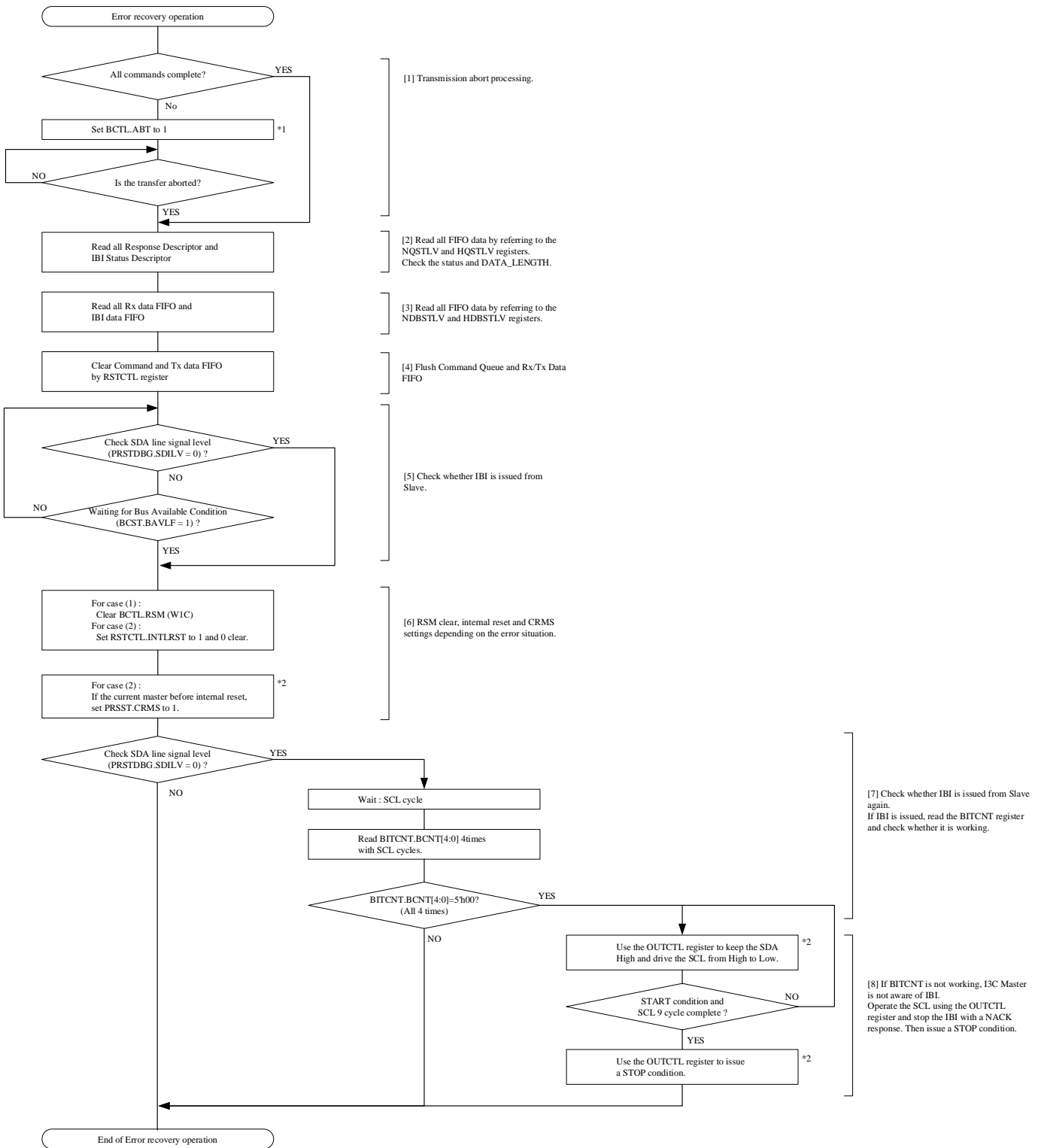


Figure 25.96 Example of error recovery operation flowchart for I3C Master

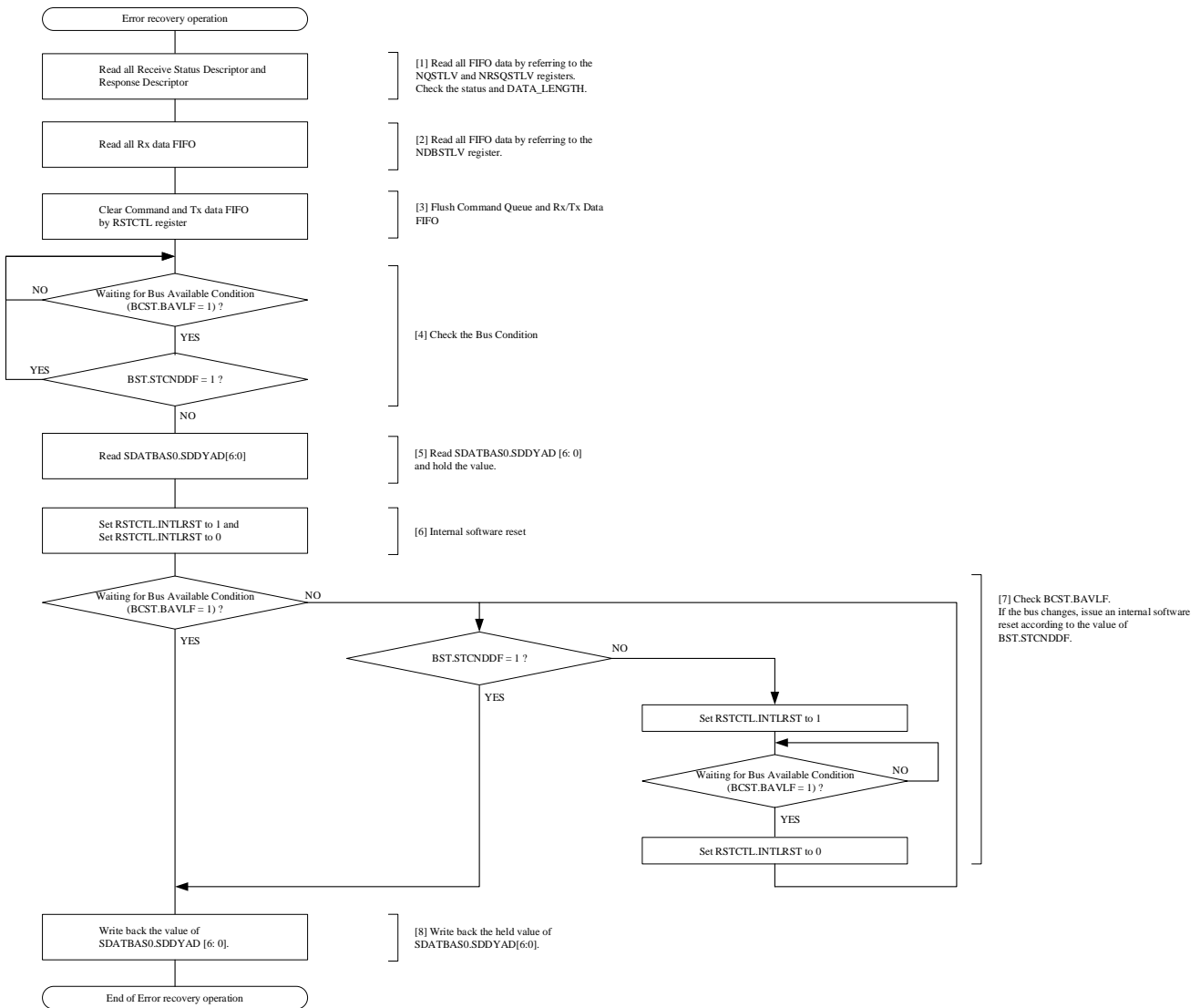


Figure 25.97 Example of error recovery operation flowchart for I3C slave

After the changes :

25.3.2.4.6 Error Recovery Operation [I3C mode]

When an error occurs, the INST.INEF, NTST.TEF, NTST.TABTF, HTST.TEF and HTST.TABTF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled.)

There is a possibility of communication error or internal module error.

Note: For MCU Ver.1, apply the following error recovery flow.

The I3C master / I3C slave must perform an error recovery flow according to the following case:

- When TEF is detected.

Figure 25.96-1 and Figure 25.97-1 show the error recovery flow of MCU Ver.1.

Note: For MCU Ver.2, apply the following error recovery flow.

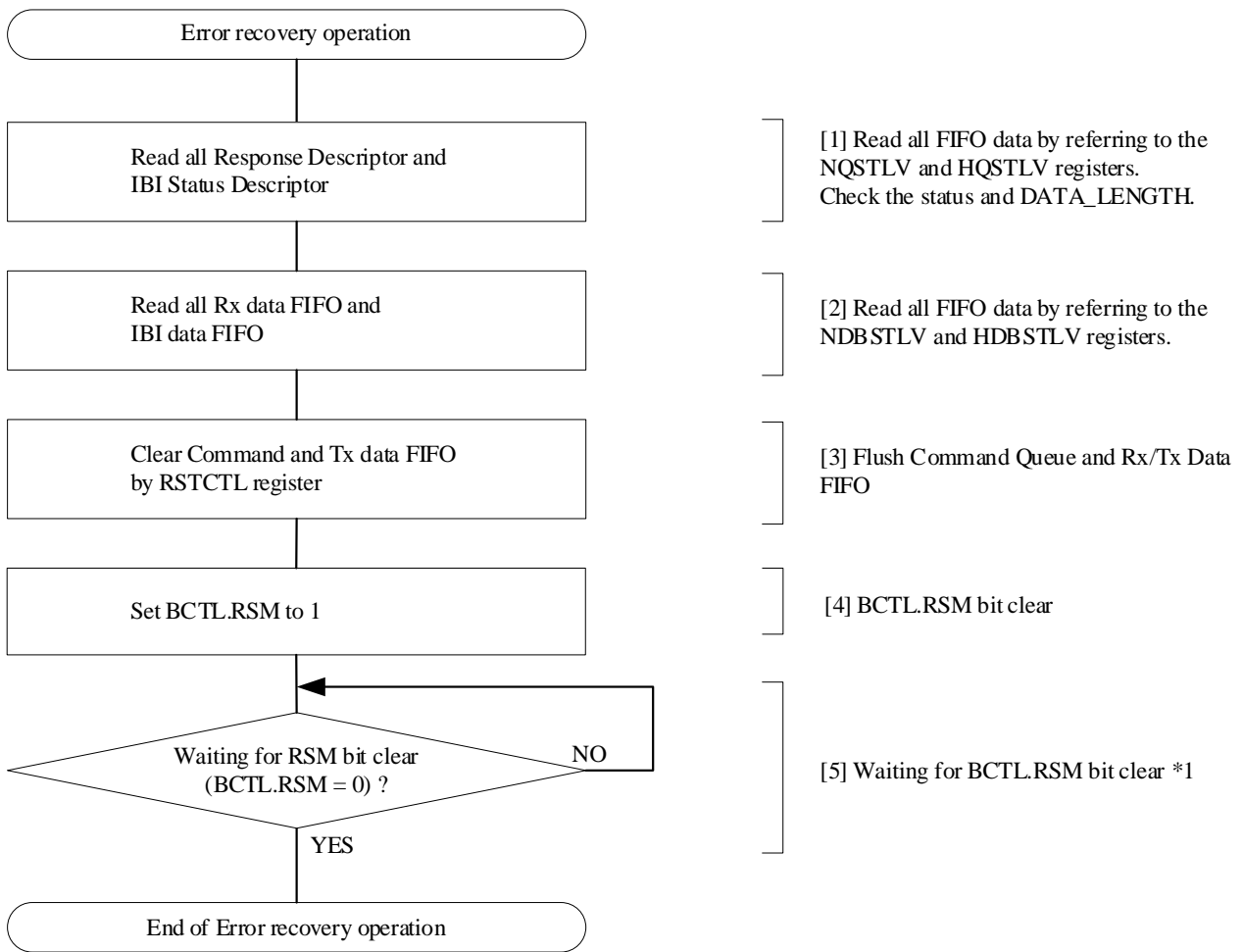
If an error occurs, I3C will be suspended. (BCTL.RSM becomes 1.) After I3C is suspended, the application must write the value 1 to the BCTL.RSM bit to resume I3C operation and recover from the suspended state.

Figure 25.96-2 and Figure 25.97-2 show the error recovery flow of MCU Ver.2.

There is no problem even if the error recovery flow of MCU Ver.1 is executed on MCU Ver.2.

(Omitted because of the same figure as Figure 25.96)

Figure 25.96-1 Example of error recovery operation flowchart for I3C Master

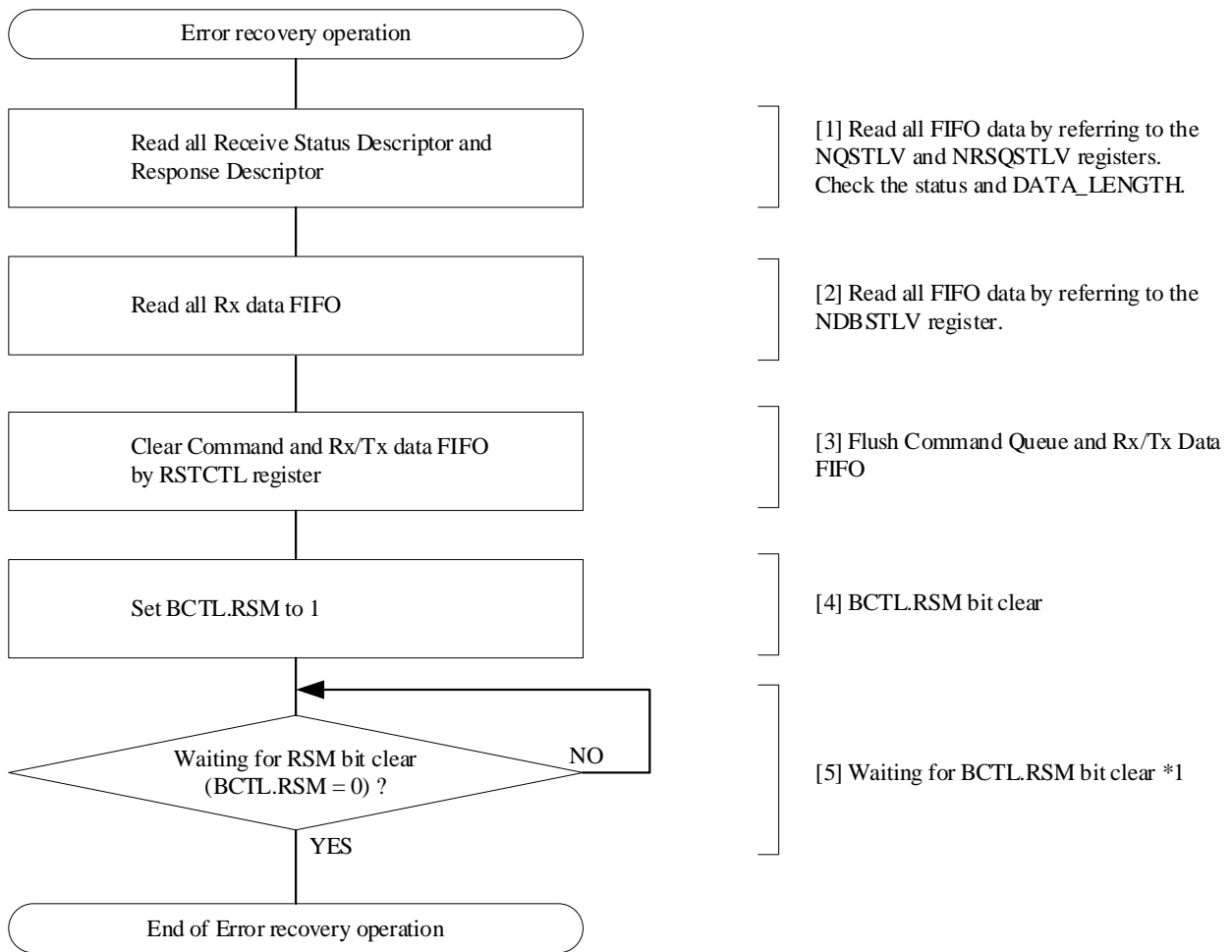


Note 1. It is possible to set the Command Descriptor while waiting for the RSM bit to be cleared.

Figure 25.96-2 Example of error recovery operation flowchart for I3C Master

(Omitted because of the same figure as Figure 25.97)

Figure 25.97-1 Example of error recovery operation flowchart for I3C Slave



Note 1. It is possible to set the Command Descriptor for issuing IBI while waiting for the RSM bit to be cleared.

Figure 25.97-2 Example of error recovery operation flowchart for I3C Slave

6 : I3C master; Error recovery operation when receiving irregular data in I3C master receive mode

In MCU Ver.1, if the data transmitted from the slave is less than the received data length (number of bytes) set in the command descriptor when receiving the I3C master, I3C is required to perform an internal reset. MCU Ver.2 does not required to perform the internal reset.

There is no problem even if the same software workaround of MCU Ver.1 is executed on MCU Ver.2.

Before the changes : User's Manual: Hardware Rev.1.00 P645

(c) SDR Data Read Transfer

1. Write the data requested from the I3C Master to the Transmit Data Buffer via the NTDTBPn register.
2. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own SlaveAddress, and if it matches, I3C responds with ACK.

When a Transaction is received, if the Transmit Data Buffer is EMPTY, I3C Slave responds with NACK with theAddress Header.

In preparation for retrying the I3C Master, write data to the Transmit Data Buffer via the NTDTBPn register.

3. Transmit the data stored in the Transmit Data Buffer.
4. If data to be transmitted still remains, write the data to be transmitted with an interrupt by TDBEF0 = 1 to the TransmitData Buffer via the NTDTBPn register.

5. SDR:

When the transmission of the data stored in the Transmit Data Buffer is completed, Low is output to the T-bit followingData, and it is notified to the I3C Master that it is the final data.

Legacy I²C Message:

When NACK is detected, data transmission is terminated.

6. When a Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in theReceive Status Buffer.
7. Read the Receive Status Descriptor via NRSQP and check the status.

If the data length does not match, set the RSTCTL.INTLRST bit to 1 and then reset the internal states of this module. For details, see section 25.3.2.4.6. Error Recovery Operation [I3C mode].

After the changes :

(c) SDR Data Read Transfer

1. Write the data requested from the I3C Master to the Transmit Data Buffer via the NTDTBPn register.
2. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own SlaveAddress, and if it matches, I3C responds with ACK.

When a Transaction is received, if the Transmit Data Buffer is EMPTY, I3C Slave responds with NACK with theAddress Header.

In preparation for retrying the I3C Master, write data to the Transmit Data Buffer via the NTDTBPn register.

3. Transmit the data stored in the Transmit Data Buffer.
4. If data to be transmitted still remains, write the data to be transmitted with an interrupt by TDBEF0 = 1 to the TransmitData Buffer via the NTDTBPn register.

5. SDR:

When the transmission of the data stored in the Transmit Data Buffer is completed, Low is output to the T-bit followingData, and it is notified to the I3C Master that it is the final data.

Legacy I²C Message:

When NACK is detected, data transmission is terminated.

6. When a Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in theReceive Status Buffer.
7. Read the Receive Status Descriptor via NRSQP and check the status.

MCU Ver.1 has the following restriction and workaround. This restriction and workaround are not required for MCU Ver.2.

If the data length does not match, set the RSTCTL.INTLRST bit to 1 and then reset the internal states of this module. For details, see section 25.3.2.4.6. Error Recovery Operation [I3C mode].

7 : I3C master / I3C slave; Operation when using unsupported Common Command Code(CCC)

There are restrictions and workarounds for MCU Ver.1 and MCU Ver.2 when using common command codes (CCC) that are not supported by the I3C master / I3C slave.

Before the changes : User's Manual: Hardware Rev.1.00 P687

25.3.2.3.11 Common Command Codes (CCC) [I3C mode]

Command Code 0xE0-0xFE Vendor Extension-Direct CCCs defined are not supported.

The MIPI reserved area and Vendor Extension area of Command Code are not supported.

Do not use an unsupported CCC when using this module with I3C Slave.

If the I3C Master must use an unsupported CCC, use the added CCC after using ENTASx CCC to put this module to Sleepmode.

After the changes :

25.3.2.3.11 Common Command Codes (CCC) [I3C mode]

For the common command code (CCC), refer to 5.1.9 Common Command Codes (CCC) in MIPI I3C Specification v1.0. This I3C is based on Table 15 I3C Common Command Codes in 5.1.9.3 Common Command Definitions of MIPI I3C Specification v1.0.

Note: MCU Ver.1 has the following restriction and workaround. The restriction and workaround are not required for MCU Ver.2.

Command Code 0xE0-0xFE Vendor Extension-Direct CCCs defined are not supported.

The MIPI reserved area and Vendor Extension area of Command Code are not supported.

Do not use an unsupported CCC when using this module with I3C Slave.

If the I3C Master must use an unsupported CCC, use the added CCC after using ENTASx CCC to put this module to Sleepmode.

Note: For MCU Ver.2, the MIPI Reserved area and Vendor Extension area of Command Code are described below.

I3C Master mode :

**When sending CCCs in the MIPI Reserved area and Vendor Extension area from the I3C Master, only Broadcast / Direct SET CCCs using the Immediate Transfer Command can be sent.
Sending Direct GET CCC is not supported.**

I3C Slave mode :

**Only Broadcast / Direct SET CCC can be received for CCC in MIPI Reserved area and Vendor Extension area.
Receiving Direct GET CCC is not supported.**