Product Change Notification

(Notification – P2202008-DI)

(MCP-AC-22-0010 / ANZ012)

February 24, 2022

To: *Our Valued Customer (Insert Customer Name Here)*

The purpose of this notification is to communicate a product change of select Renesas Electronics America, Inc. (REA) devices.

This notification announces new part revision to improve I^{2}C / I3C bus interface functions for select RA2E2 devices.

For the restrictions and precautions of MCU Ver.1, the workarounds and precautions are described in RA2E2 Group User's Manual Hardware Rev.1.00 (R01UH0919EJ0100). To improve the function of the I^{2}C / I3C bus interface, the products have been revised and the MCU version was changed from MCU Ver.1 to MCU Ver.2. The software workarounds of MCU Ver.1 can be executed on MCU Ver.2 without any changes. See the Appendix for details.

There is a new part number for the new part version. There is no impact to form, fit, quality & reliability of the products.

Affected Products: A review of our records indicates the list of products in the Appendix may affect your company.

Part numbers given in this list are for active part numbers in REA database at the time of this notification.

Key Dates:

Shipments from REA of the new revision devices begins.	July 1, 2022

Response: No response is required. REA will consider this notification approved 30 days after its issue. If you anticipate volumes beyond your regular rate prior to the transition date, please contact your REA sales representative with a forecast of your requirements.

If the customer provides a timely acknowledgement, the customer shall have 90 days (an additional 60 days) from the date of receipt of this notification in which to make any objections to the notification. If the customer does not make any objections to this notification within 90 days of the receipt of the notification, then Renesas will consider the notification as approved. If customer cannot accept the notification, then the customer must provide Renesas with a last time buy demand and purchase order.

Please contact your REA sales representative for any questions or comments. Thank you for your attention.

Sincerely,

Renesas Electronics America, Inc.

Appendix A: Affected Part Number List

Booking Part Number	Replacement PN	Booking Part Number	Replacement PN
R7FA2E2A72DNK#AA0	R7FA2E2A72DNK#AA1	R7FA2E2A72DNJ#AA0	R7FA2E2A72DNJ#AA1
R7FA2E2A72DNK#HA0	R7FA2E2A72DNK#HA1	R7FA2E2A72DNJ#HA0	R7FA2E2A72DNJ#HA1
R7FA2E2A73CNK#AA0	R7FA2E2A73CNK#AA1	R7FA2E2A73CNJ#AA0	R7FA2E2A73CNJ#AA1
R7FA2E2A73CNK#HA0	R7FA2E2A73CNK#HA1	R7FA2E2A73CNJ#HA0	R7FA2E2A73CNJ#HA1
R7FA2E2A74CNK#AA0	R7FA2E2A74CNK#AA1	R7FA2E2A74CNJ#AA0	R7FA2E2A74CNJ#AA1
R7FA2E2A74CNK#HA0	R7FA2E2A74CNK#HA1	R7FA2E2A74CNJ#HA0	R7FA2E2A74CNJ#HA1
R7FA2E2A52DNK#AA0	R7FA2E2A52DNK#AA1	R7FA2E2A52DNJ#AA0	R7FA2E2A52DNJ#AA1
R7FA2E2A52DNK#HA0	R7FA2E2A52DNK#HA1	R7FA2E2A52DNJ#HA0	R7FA2E2A52DNJ#HA1
R7FA2E2A53CNK#AA0	R7FA2E2A53CNK#AA1	R7FA2E2A53CNJ#AA0	R7FA2E2A53CNJ#AA1
R7FA2E2A53CNK#HA0	R7FA2E2A53CNK#HA1	R7FA2E2A53CNJ#HA0	R7FA2E2A53CNJ#HA1
R7FA2E2A54CNK#AA0	R7FA2E2A54CNK#AA1	R7FA2E2A54CNJ#AA0	R7FA2E2A54CNJ#AA1
R7FA2E2A54CNK#HA0	R7FA2E2A54CNK#HA1	R7FA2E2A54CNJ#HA0	R7FA2E2A54CNJ#HA1
R7FA2E2A32DNK#AA0	R7FA2E2A32DNK#AA1	R7FA2E2A32DNJ#AA0	R7FA2E2A32DNJ#AA1
R7FA2E2A32DNK#HA0	R7FA2E2A32DNK#HA1	R7FA2E2A32DNJ#HA0	R7FA2E2A32DNJ#HA1
R7FA2E2A33CNK#AA0	R7FA2E2A33CNK#AA1	R7FA2E2A33CNJ#AA0	R7FA2E2A33CNJ#AA1
R7FA2E2A33CNK#HA0	R7FA2E2A33CNK#HA1	R7FA2E2A33CNJ#HA0	R7FA2E2A33CNJ#HA1
R7FA2E2A34CNK#AA0	R7FA2E2A34CNK#AA1	R7FA2E2A34CNJ#AA0	R7FA2E2A34CNJ#AA1
R7FA2E2A34CNK#HA0	R7FA2E2A34CNK#HA1	R7FA2E2A34CNJ#HA0	R7FA2E2A34CNJ#HA1

Appendix B: Change Summary

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 ENTDAA 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)

Appendix C: Product Identification

The last digit of the booking part numbers is changed from 0 to 1 between MCU Ver.1 and MCU Ver.2. Therefore, they can be distinguished by the booking part numbers.

Example: R7FA2E2A72DNK

Product	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

Appendix D: Item #1 - I^2C 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.

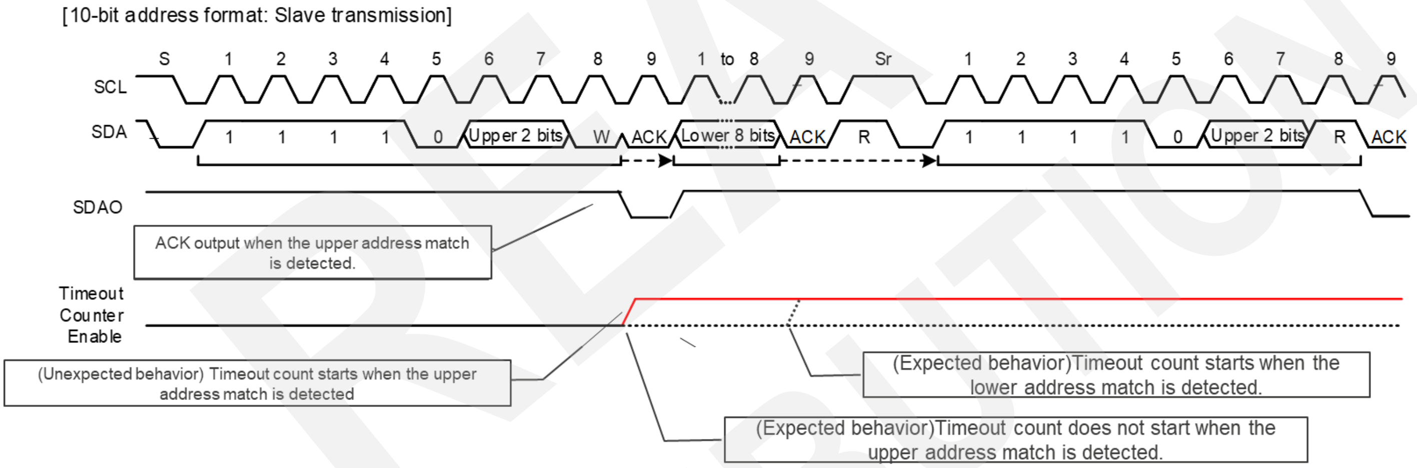
When sending data:

- 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.

Appendix E Item #2 - I^{2}C 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.



ACK output when the upper address match is detected.

(Unexpected behavior) Timeout count starts when the upper address match is detected

(Expected behavior)Timeout count starts when the lower address match is detected.

(Expected behavior)Timeout count does not start when 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 I^{2}C Slave, during 10-bit address communication, the timeout count starts when the upperaddress 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.

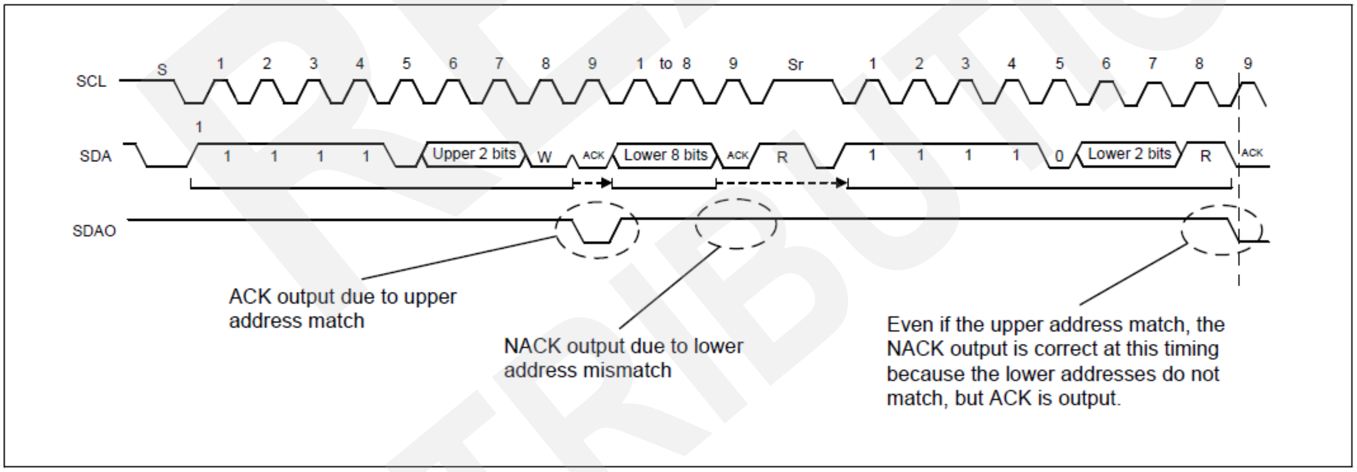
Note: **MCU Ver.1 has the following restriction. The restriction is not required for MCU Ver.2.**
When working with I^{2}C Slave, during 10-bit address communication, the timeout count starts when the upper address match is detected.

Appendix F: Item #3 - I^{2}C 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


SCL
S
1 2 3 4 5 6 7 8 9
1 to 8
9
Sr
1 2 3 4 5 6 7 8 9
SDA
1
1 1 1 1
Upper 2 bits
W
ACK
Lower 8 bits
ACK
R
1 1 1 1
0
Lower 2 bits
R
ACK
SDAO
ACK output due to upper address match
NACK output due to lower address mismatch
Even if the upper address match, the NACK output is correct at this timing because the lower addresses do not match, but ACK is output.

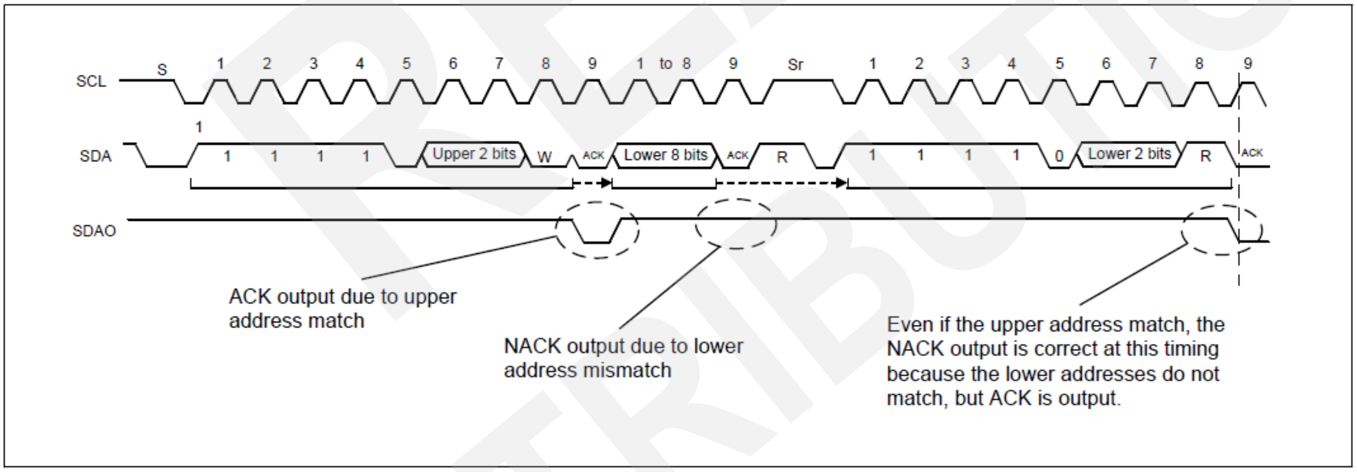
When multiple I^{2}C slaves are connected to the I^{2}C bus and there is a possibility that an I^{2}C Slave other than thismodule will NACK for the upper address / R after Repeated START, there are the following restrictions and workarounds.

- 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 I^{2}C 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 I^{2}C Master side. If 0xFF is valid data, use restriction (1).

Appendix F (cont.): Item #3 - I^{2}C Slave; ACK Output Operation When 10-bit Address Communication and the Lower Address Mismatch is Detected

After the changes :

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

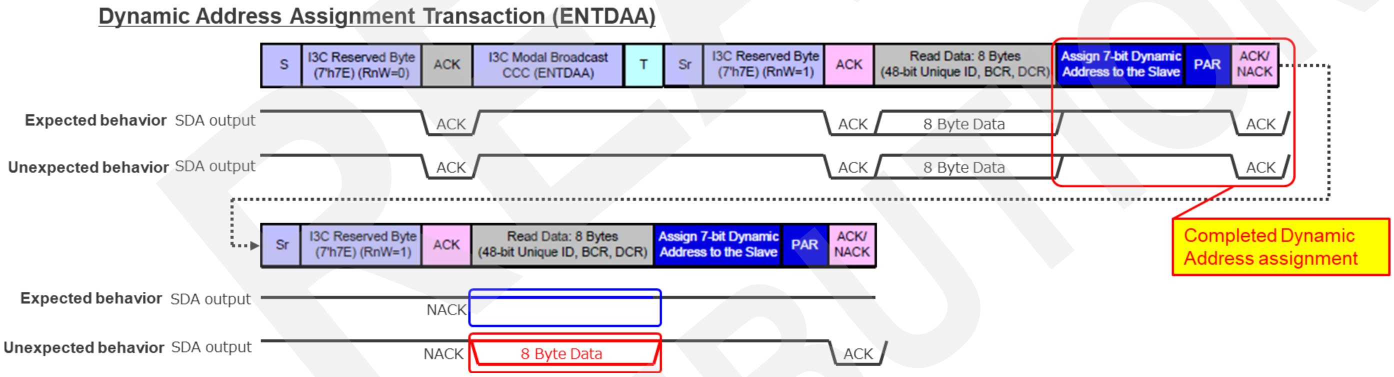

SCL
S
1 2 3 4 5 6 7 8 9
1 to 8
9
Sr
1 2 3 4 5 6 7 8 9
SDA
1
1 1 1 1
Upper 2 bits
W
ACK
Lower 8 bits
ACK
R
1 1 1 1
0
Lower 2 bits
R
ACK
SDAO
ACK output due to upper address match
NACK output due to lower address mismatch
Even if the upper address match, the NACK output is correct at this timing because the lower addresses do not match, but ACK is output.

When multiple I^{2}C slaves are connected to the I^{2}C bus and there is a possibility that an I^{2}C Slave other than thismodule 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 I^{2}C 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 I^{2}C Master side. If 0xFF is valid data, use restriction (1).

Appendix G: Item #4 - I3C Slave; Arbitration Operation When Dynamic Address is Assigned by the ENTDAA command

In MCU Ver.1, if a dynamic address is assigned by the ENTDAA 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.


Dynamic Address Assignment Transaction (ENTDAA)
S
I3C Reserved Byte (7'h7E) (RnW=0)
ACK
I3C Modal Broadcast CCC (ENTDAA)
T
Sr
I3C Reserved Byte (7'h7E) (RnW=1)
ACK
Read Data: 8 Bytes (48-bit Unique ID, BCR, DCR)
Assign 7-bit Dynamic Address to the Slave
PAR
ACK/ NACK
Expected behavior SDA output
Unexpected behavior SDA output
ACK
8 Byte Data
Completed Dynamic Address assignment
NACK

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 ENTDAA command.

After the changes :

MCU Ver.1 has the following restriction and workaround. Thisrestriction 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 ENTDAA command.

Appendix H: Item #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 accordingto 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.

Appendix H (cont.): Item #5 - I3C Master / I3C Slave; Error Recovery Operation

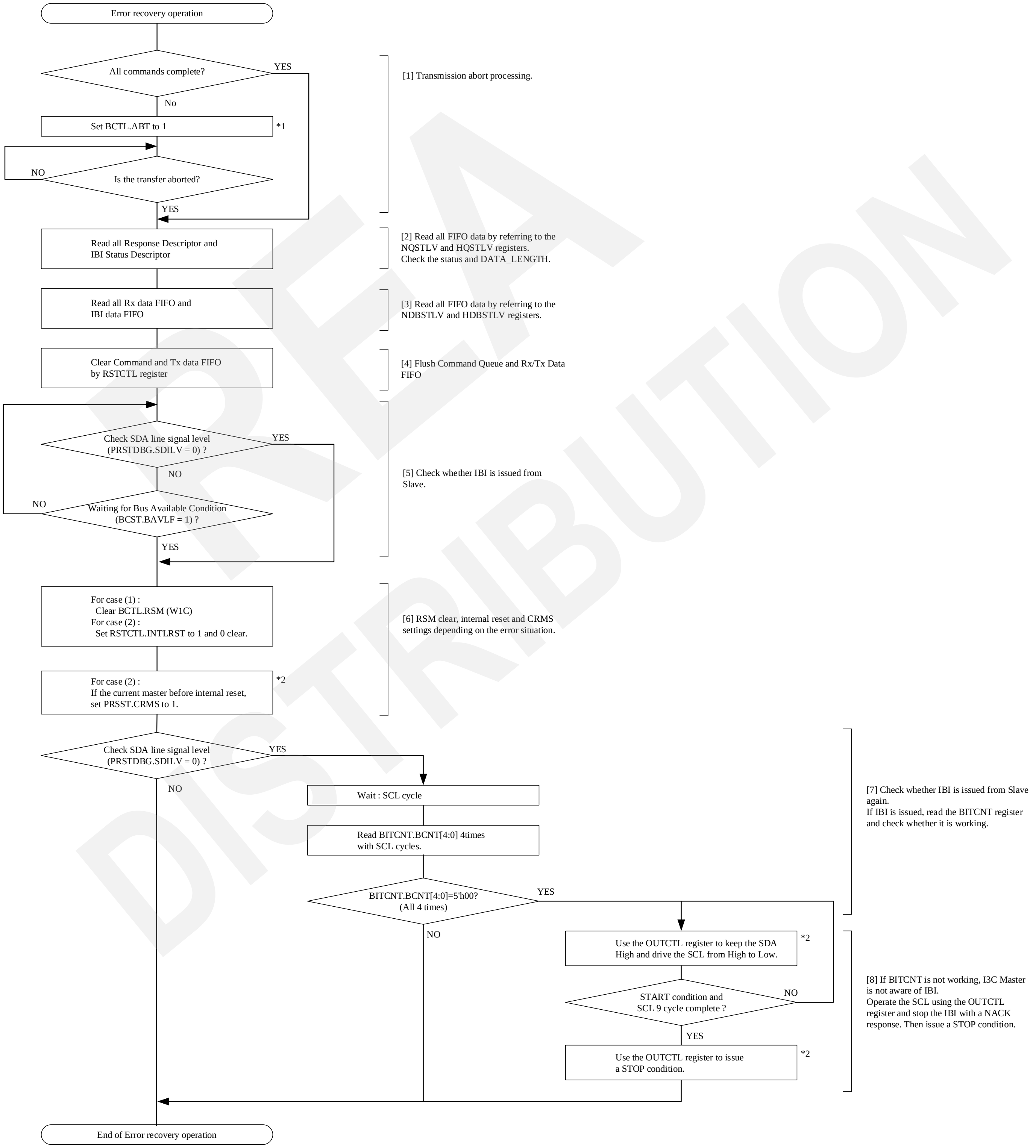

Error recovery operation
All commands complete?
YES
No
Set BCTL.ABT to 1
*1
NO
Is the transfer aborted?
YES
[1] Transmission abort processing.
Read all Response Descriptor and IBI Status Descriptor
[2] Read all FIFO data by referring to the NQSTLV and HQSTLV registers. Check the status and DATA_LENGTH.
Read all Rx data FIFO and IBI data FIFO
[3] Read all FIFO data by referring to the NDBSTLV and HDBSTLV registers.
Clear Command and Tx data FIFO by RSTCTL register
[4] Flush Command Queue and Rx/Tx Data FIFO
Check SDA line signal level (PRSTDBG.SDILV = 0) ?
YES
NO
NO
Waiting for Bus Available Condition (BCST.BAVLF = 1) ?
YES
[5] Check whether IBI is issued from Slave.
For case (1) : Clear BCTL.RSM (W1C)
For case (2) : Set RSTCTL.INTLRST to 1 and 0 clear.
[6] RSM clear, internal reset and CRMS settings depending on the error situation.
For case (2) : If the current master before internal reset, set PRSST.CRMS to 1.
*2
Check SDA line signal level (PRSTDBG.SDILV = 0) ?
YES
NO
Wait : SCL cycle
Read BITCNT.BCNT[4:0] 4times with SCL cycles.
[7] Check whether IBI is issued from Slave again. If IBI is issued, read the BITCNT register and check whether it is working.
BITCNT.BCNT[4:0]=5'h00? (All 4 times)
YES
NO
Use the OUTCTL register to keep the SDA High and drive the SCL from High to Low.
*2
START condition and SCL 9 cycle complete ?
NO
YES
Use the OUTCTL register to issue a STOP condition.
*2
[8] If BITCNT is not working, I3C Master is not aware of IBI. Operate the SCL using the OUTCTL register and stop the IBI with a NACK response. Then issue a STOP condition.
End of Error recovery operation

Figure 25.96 Example of error recovery operation flowchart for I3C Master

Appendix H (cont.): Item #5 - I3C Master / I3C Slave; Error Recovery Operation

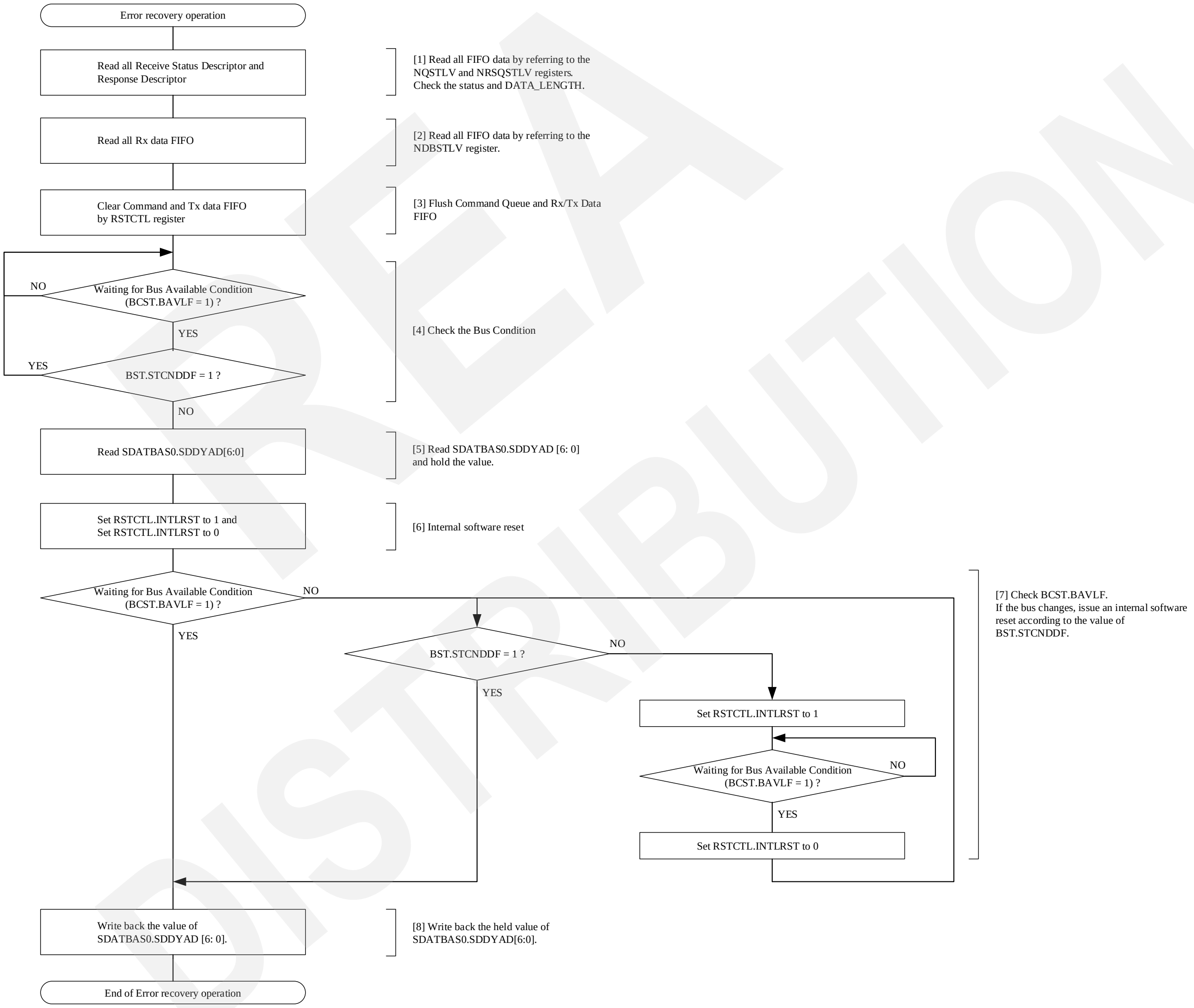

Error recovery operation
Read all Receive Status Descriptor and Response Descriptor
[1] Read all FIFO data by referring to the NQSTLV and NRSQSTLV registers. Check the status and DATA_LENGTH.
Read all Rx data FIFO
[2] Read all FIFO data by referring to the NDBSTLV register.
Clear Command and Tx data FIFO by RSTCTL register
[3] Flush Command Queue and Rx/Tx Data FIFO
NO
Waiting for Bus Available Condition (BCST.BAVLF = 1) ?
YES
YES
BST.STCNDDF = 1 ?
NO
[4] Check the Bus Condition
Read SDATBAS0.SDDYAD[6:0]
[5] Read SDATBAS0.SDDYAD [6: 0] and hold the value.
Set RSTCTL.INTLRST to 1 and Set RSTCTL.INTLRST to 0
[6] Internal software reset
Waiting for Bus Available Condition (BCST.BAVLF = 1) ?
NO
YES
BST.STCNDDF = 1 ?
NO
YES
Set RSTCTL.INTLRST to 1
Waiting for Bus Available Condition (BCST.BAVLF = 1) ?
NO
YES
Set RSTCTL.INTLRST to 0
[7] Check BCST.BAVLF. If the bus changes, issue an internal software reset according to the value of BST.STCNDDF.
Write back the value of SDATBAS0.SDDYAD [6: 0].
[8] Write back the held value of SDATBAS0.SDDYAD[6:0].
End of Error recovery operation

Figure 25.97 Example of error recovery operation flowchart for I3C slave

Appendix H (cont.): Item #5 - I3C Master / I3C Slave; Error Recovery Operation

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 accordingto 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.

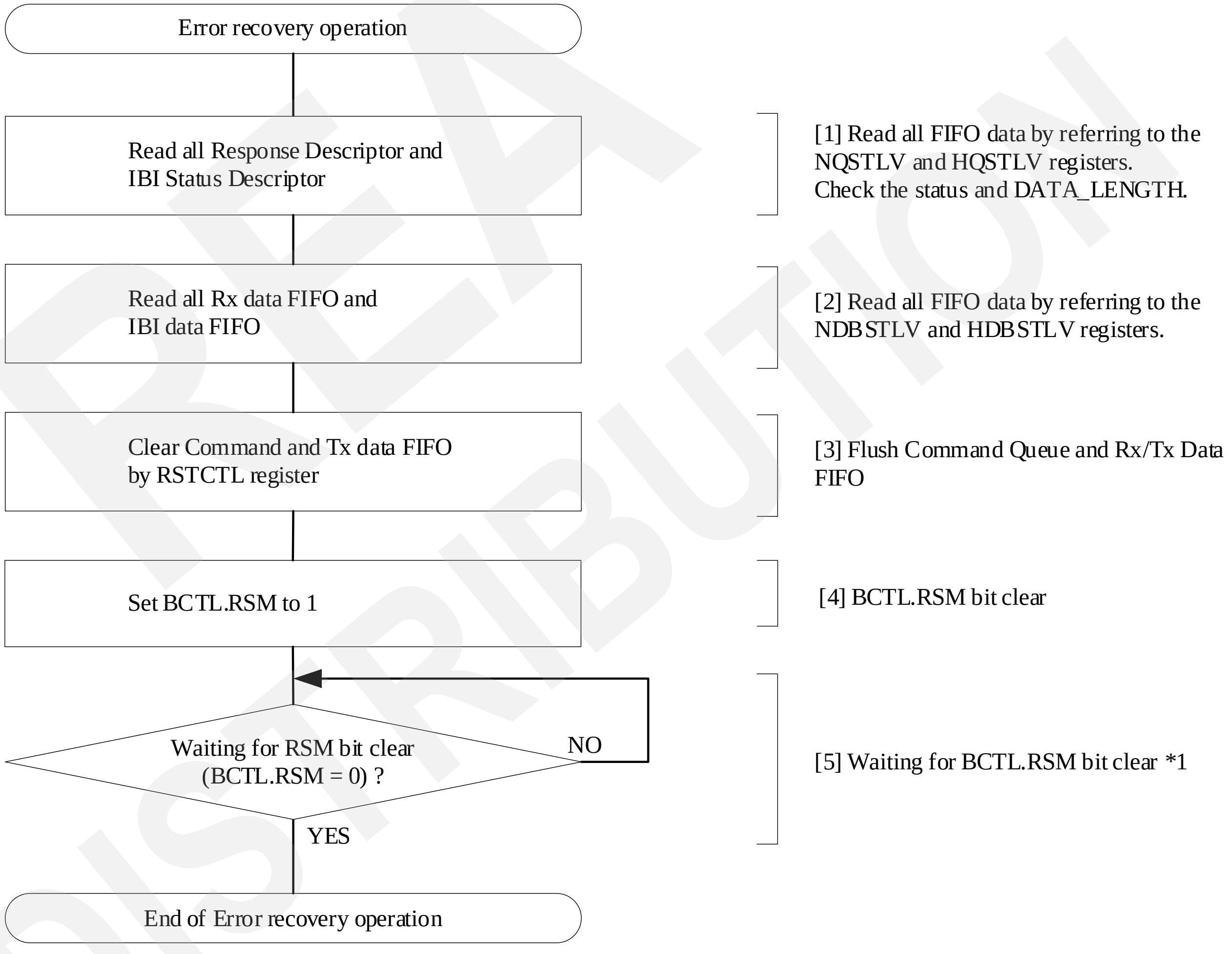
Figure 25.96-2 and Figure 25.97-2 show the error recovery flow of MCU Ver.2.

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

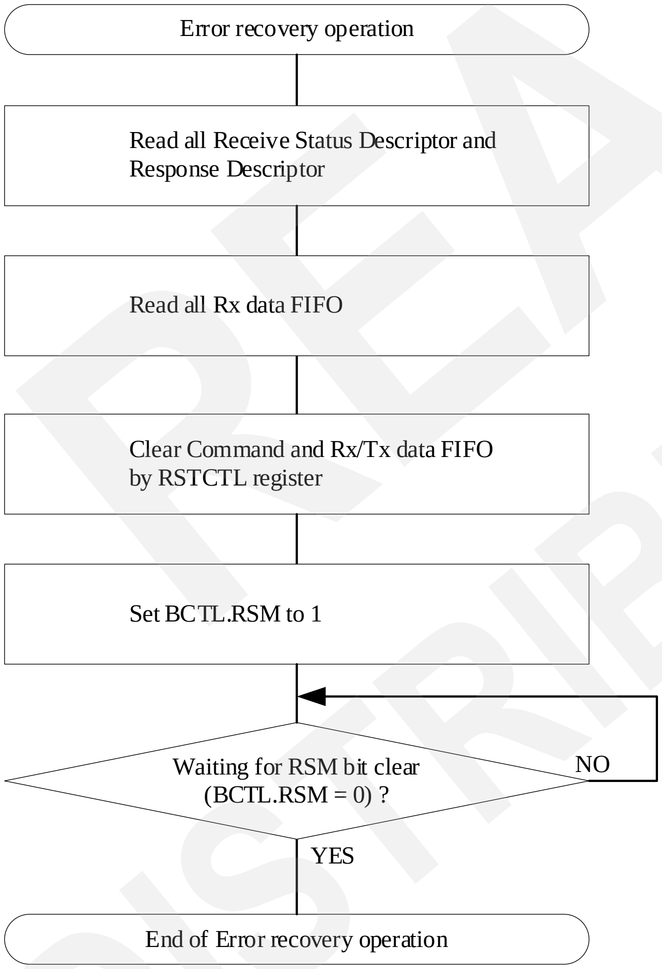
Appendix H (cont.): Item #5 - I3C Master / I3C Slave; Error Recovery Operation


Error recovery operation
Read all Response Descriptor and IBI Status Descriptor
Read all Rx data FIFO and IBI data FIFO
Clear Command and Tx data FIFO by RSTCTL register
Set BCTL.RSM to 1
Waiting for RSM bit clear (BCTL.RSM = 0) ?
NO
YES
End of Error recovery operation
[1] Read all FIFO data by referring to the NQSTLV and HQSTLV registers. Check the status and DATA_LENGTH.
[2] Read all FIFO data by referring to the NDBSTLV and HDBSTLV registers.
[3] Flush Command Queue and Rx/Tx Data FIFO
[4] BCTL.RSM bit clear
[5] Waiting for BCTL.RSM bit clear *1

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

Appendix H (cont.): Item #5 - I3C Master / I3C Slave; Error Recovery Operation


Error recovery operation
Read all Receive Status Descriptor and Response Descriptor
Read all Rx data FIFO
Clear Command and Rx/Tx data FIFO by RSTCTL register
Set BCTL.RSM to 1
Waiting for RSM bit clear (BCTL.RSM = 0) ?
NO
YES
End of Error recovery operation

[1] Read all FIFO data by referring to the NQSTLV and NRSQSTLV registers. Check the status and DATA_LENGTH.

[2] Read all FIFO data by referring to the NDBSTLV register.

[3] Flush Command Queue and Rx/Tx Data FIFO

[4] BCTL.RSM bit clear

[5] Waiting for BCTL.RSM bit clear *1

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

Appendix I: Item #6 - I3C Master / I3C Slave; Error Recovery Operation

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 require 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 following Data, and it is notified to the I3C Master that it is the final data.
   Legacy I^{2}C 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].

Appendix I (cont.): Item #6 - I3C Master / I3C Slave; Error Recovery Operation

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 following Data, and it is notified to the I3C Master that it is the final data.
   Legacy I^{2}C 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].

Appendix J: Item #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.