The following document contains information on Cypress products. Although the document is marked with the name "Spansion" and "Fujitsu", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

Continuity of Specifications
There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

Continuity of Ordering Part Numbers
Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information
Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress
Cypress (NASDAQ: CY) delivers high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. With a broad, differentiated product portfolio that includes NOR flash memories, F-RAM™ and SRAM, Traveo™ microcontrollers, the industry's only PSoC® programmable system-on-chip solutions, analog and PMIC Power Management ICs, CapSense® capacitive touch-sensing controllers, and Wireless BLE Bluetooth® Low-Energy and USB connectivity solutions, Cypress is committed to providing its customers worldwide with consistent innovation, best-in-class support and exceptional system value.

Errata

This errata sheet is for MB91635A Series Hardware Manual Rev. 2 (CM71-10153-2E).

FR80
32-BIT MICROCONTROLLER
MB91635A Series
HARDWARE MANUAL

2011.12.16
: Corrected part

Date	Page	Item	Description
2011/ 6/29	7	1.2	Table 1.2-1 was changed as indicated by the shading below. (Error) Built-in RAM capacity (Correct) Built-in RAM (Instruction execution enabled) [mcu_doc1068]
2011/ 6/29	9	1.3	Figure 1.3-1 was changed as indicated by the shading below. (Error) RAM (Correct) Built-in RAM (Instruction execution enabled) [mcu_doc1068]
2011/ 6/29	95	3.1	Figure 3.1-1 was changed as indicated by the shading below. (Error) Built-in RAM area (Correct) Built-in RAM area(Instruction execution enabled) [mcu_doc1068]
2010/ 3/23	126	3.11.4	The following description was added to the end of the page. If an interrupt is received while executing an instruction to set I flag to "0", there is a delay for 1 cycle from execution of an instruction for I flag and ILM to change. Therefore, I flag becomes "0" although processing moves to the interrupt processing routine. At this time, if multiple interrupts are generated, I flag can not receive any interrupt because it is "0", and processing of multiple interrupts is not performed. I flag itself is updated when executing an instruction. Therefore, a value of I flag after update is saved to the stack, and when the value of the stack is returned, the value of I flag after update is reflected to PS register. To receive a new interrupt within the interrupt routine, it is required to set software to make I flag to "1" at the beginning of the interrupt routine. [mcu_doc1056]
2011/ 6/29	237	9.4.1	Figure 9.4-1 was deleted as indicated by the shading below. Reset Resource / Initial Value Timeout for software reset: 1XXXXXX1 Register reading: 00000000 [mcu_doc1088]
2010/ 6/7	314	13.4.2	The following description of [bit2] was deleted as indicated by shading below. When the address output is address shift output mode (ADTY=1), this bit specifies a bus type. [mcu_doc1078]

Date	Page	Item	Description
2010/ 6/7	318, 319, 320, 321, 322, 353, 357, 360, 363, 365	13.4.3 13.6.5, 13.6.7, 13.6.9, 13.6.10, 13.6.11	The following description of address data split bus was deleted as indicated by shading below. … (ADTY=0 or ADTY=1 and BSTY=0) … The following description of address data multiplex bus was deleted as indicated by shading below. …(ADTY=1 and BSTY=1) … [mcu_doc1078]
2011/ 6/29	318	13.4.3	<Notes> was added as indicated by the shading below. <Notes> ・Since all chip select signals are disabled ("H" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins) and the D15 to D00 pins become Hi-Z during a read access idle cycle, the next access does not begin until the read access idle cycle ends. ・No read access idle cycle is inserted during continuous read access of one CS area for which the address data split bus is set as the bus type by the BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) in the corresponding area configuration register (ACR0 to ACR3). ・The function at least guarantees more than the period of this specified cycle. It is sure not to necessarily agree to the specified idol cycle. [mcu_doc1087]
2010/ 6/7	326	13.5.1	The following description was deleted as indicated by shading below. In the explanation of the protocol, the address data split bus is set as the bus type by the ADTY/BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) … [mcu_doc1078]
2010/ 6/7	332	13.5.2	The following description was deleted as indicated by shading below. In the explanation of the protocol, the address data multiplex bus is set as the bus type by the ADTY/BSTY bit (ADTY=1 and BSTY=1) … [mcu_doc1078]
2011/ 6/29	345	13.6.3	<Notes> was added as indicated by the shading below. <Notes> ・No read access idle cycle is inserted during continuous read access of the same CS area of the address data split bus. ・The function at least guarantees more than the period of this specified cycle. It is sure not to necessarily agree to the specified idol cycle. [mcu_doc1087]
2010/ 3/23	395	13.12	The following description was added under Procedure 7. To wait for the CS area settings to be reflected in the subsequent access operations by reading the area setting register (ASR0 to ASR3) that was the last one set, and compare the setting values and read values. Reading and comparison are dummy processing. There is no effect to the comparison results. [mcu_doc1060]

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2010/
3/23</td><td>396</td><td>13.12</td><td>The description was corrected as indicated by the shading below.

・CS0 area settings
CS0 area setting register (ASR0): ASZ3 to ASZ0 = 0010_B
CS0 area setting register (ASR0): SADR31 to SADR16 = $002C_H$
→ 002C 0000_H to 002F $FFFF_H$ is the CS0 area.
・CS1 area settings
CS1 area setting register (ASR1): ASZ3 to ASZ0 = 0000_B
CS1 area setting register (ASR1): SADR31 to SADR16 = 0026_H
→ 0026 0000_H to 0026 $FFFF_H$ is the CS1 area.
・CS2 area settings
(The descriptions of "A 1-MB space from 0011 0000 $_H$ is allocated." were deleted.)
CS2 area setting register (ASR2): ASZ3 to ASZ0 = 0100_B
CS2 area setting register (ASR2): SADR31 to SADR16 = 0030_H
→ 0030 0000_H to 003F $FFFF_H$ is the CS2 area.

The following <Note> was added to the end of the page.

For example, the space between 0031 0000_H and 1M byte can not be allocated.
Supposedly performing the following settings, CS2 area will be 0030 0000_H to 003F $FFFF_H$.
CS2 area setting register (ASR2) : ASZ3 to ASZ0=0100_B
CS2 area setting register (ASR2) : SADR31 to SADR16=0031_H
In these settings, SADR31 to SADR20 bits are valid and SADR19 to SADR16 bits are not subjected to compare the address.

[mcu_doc1060]</td></tr>
<tr><td>2010/
3/23</td><td>397</td><td>13.12</td><td>"Figure 13.12-2 CS area example" was changed as indicated by the shading below.

(Error)
<u>Setting example</u>
0000 0000_H
0006 0000_H
0007 0000_H
000C 0000_H
0010 0000_H
0020 0000_H

(Corrected)
<u>Setting example</u>
0000 0000_H
0026 0000_H
0027 0000_H
002C 0000_H
0030 0000_H
0040 0000_H

[mcu_doc1060]</td></tr>
<tr><td>2011/
6/29</td><td>803,
804</td><td>25.6</td><td>Table 25.6-4 was deleted as indicated by the shading below.

<table>
<tr><th rowspan="2">Register value (N)
STx5 to STx0</th><th colspan="4">Sampling Time [µs]</th><th colspan="4">Maximum External Impedance [kΩ]</th></tr>
<tr><th>PCLK=
30MHz</th><th>PCLK=
32MHz</th><th>PCLK=
33MHz</th><th>PCLK=
40MHz</th><th>PCLK=
30MHz</th><th>PCLK=
32MHz</th><th>PCLK=
33MHz</th><th>PCLK=
40MHz</th></tr>
<tr><td>0</td><td>Setting
prohibit
ed</td><td>Setting
prohibit
ed</td><td>Setting
rohibit
ed</td><td>Setting
rohibit
ed</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2.133</td><td>2.000</td><td>1.939</td><td>1.600</td><td>26.073</td><td>24.112</td><td>23.220</td><td>18.229</td></tr>
</table>

[mcu_doc1061]</td></tr>
</table>

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2011/
6/29</td><td>805,
806</td><td>25.6</td><td>Table 25.6-5 was deleted as indicated by the shading below.

<table>
<tr><td rowspan="2">Register value (N)
STx5 to STx0</td><td colspan="4">Sampling Time [µs]</td><td colspan="4">Maximum External Impedance [kΩ]</td></tr>
<tr><td>PCLK= 30MHz</td><td>PCLK= 32M z</td><td>PCLK= 33MHz</td><td>PCLK= 40MHz</td><td>PCLK= 30MHz</td><td>PCLK= 32MHz</td><td>PCLK= 33MHz</td><td>PCLK= 40MHz</td></tr>
<tr><td>0</td><td>Setting prohibit ed</td><td>Setting prohibit ed</td><td>Setting rohibit ed</td><td>Setting rohibit ed</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>63</td><td>17.067</td><td>16.000</td><td>15.515</td><td>12.800</td><td>245.680</td><td>229.994</td><td>222.864</td><td>182.935</td></tr>
</table>
[mcu_doc1061]</td></tr>
<tr><td>2010/
3/23</td><td>834</td><td>27.1</td><td>The following <Notes> was corrected as indicated by the shading below.

・The operation mode must be set first. Otherwise, the part of registers of the same channel will be initialized when the operation mode is changed. For the registers to be initialized, see the notes for serial mode register (SMR) of each operation mode.
[mcu_doc1058]</td></tr>
<tr><td>2010/
3/23</td><td>847,
901</td><td>27.4.2,
27.13.2</td><td>The following <Note> was corrected as indicated by the shading below.

The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed.
・Serial Control Register (SCR)
・Extended Serial Control Register (ESCR)
Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.
[mcu_doc1058]</td></tr>
<tr><td>2011/
12/16</td><td>947</td><td>27.16.1</td><td>The following <Notes> was corrected as indicated by the shading below.

・When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the SCINV bit settings. When the reload value is odd-numbered,the "L" width is the same as the "H" width.
- When SPI is set to "0" and SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
- When SPI is set to "0" and SCINV is set to "1", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer.
- When SPI is set to "1" and SCINV is set to "0", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer.
- When SPI is set to "1" and SCINV is set to "1", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
[mcu_doc1144]</td></tr>
<tr><td>2010/
3/23</td><td>967</td><td>27.21.2</td><td>The following <Note> was corrected as indicated by the shading below.

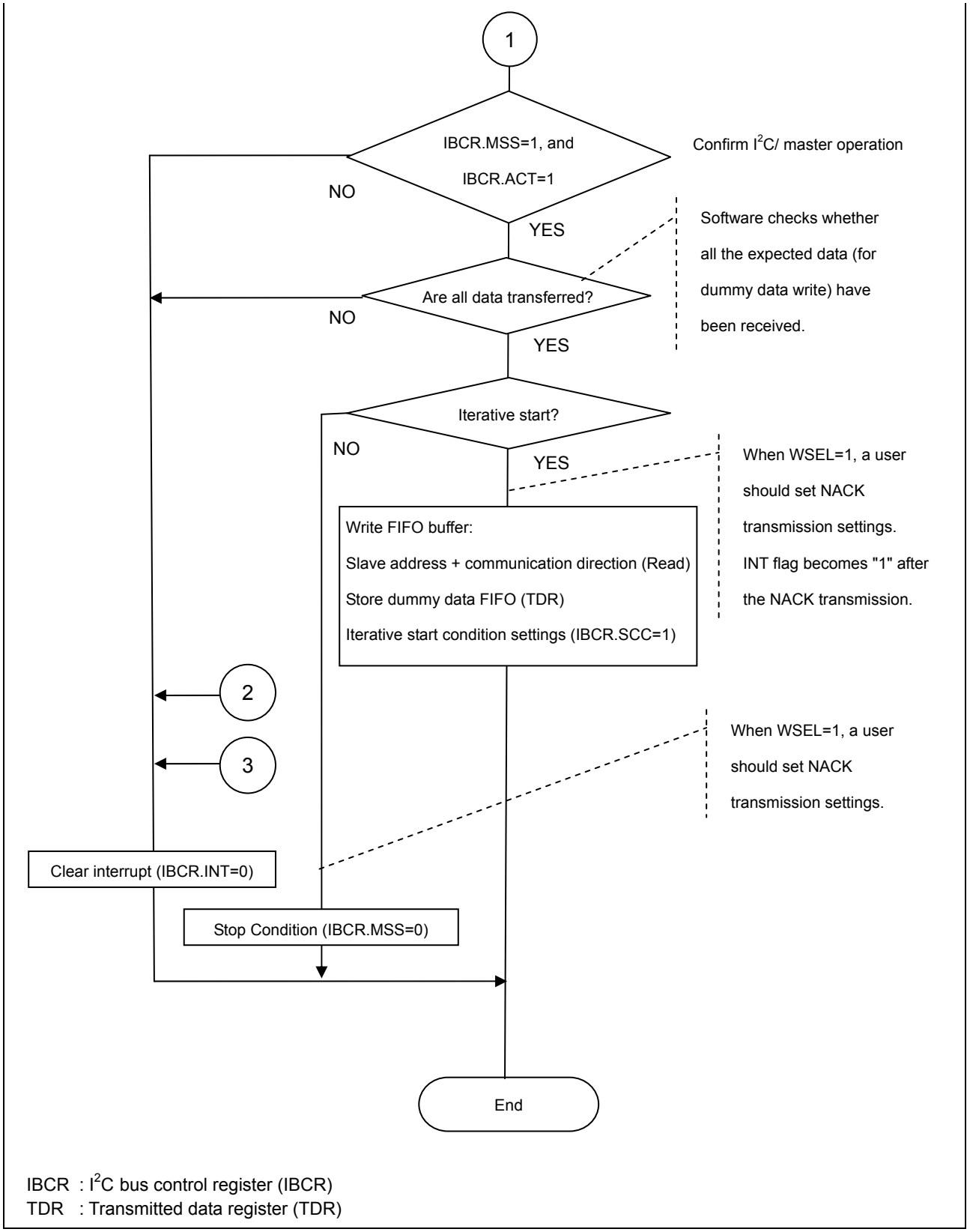
The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed.
・I^{2}C Bus Control Register (IBCR)
・I^{2}C Bus Status Register (IBSR)
Note, however, that when IBCR and SMR are written simultaneously with 16-bit write access, IBCR reflects the written content.
[mcu_doc1058]</td></tr>
<tr><td>2010/
6/7</td><td>1016,
1022</td><td>27.23.1</td><td>The figure was corrected as < Attached document 1-1 > and < Attached document 1-2 >.
[mcu_doc1081]</td></tr>
</table>

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2011/
6/29</td><td>1157</td><td>31.6.2</td><td>■ Notes on Data Writing was corrected as indicated by the shading below.

(Error)
・ Data to which "0" is once written cannot be restored to "1". Rewriting "0" to "1" results in one of the following:
- An element is judged as defective by the data polling algorithm.
- The write time limit is exceeded and the timing limit overrun flag DQ5 (TLOV) of hardware sequence flag changes to "1".

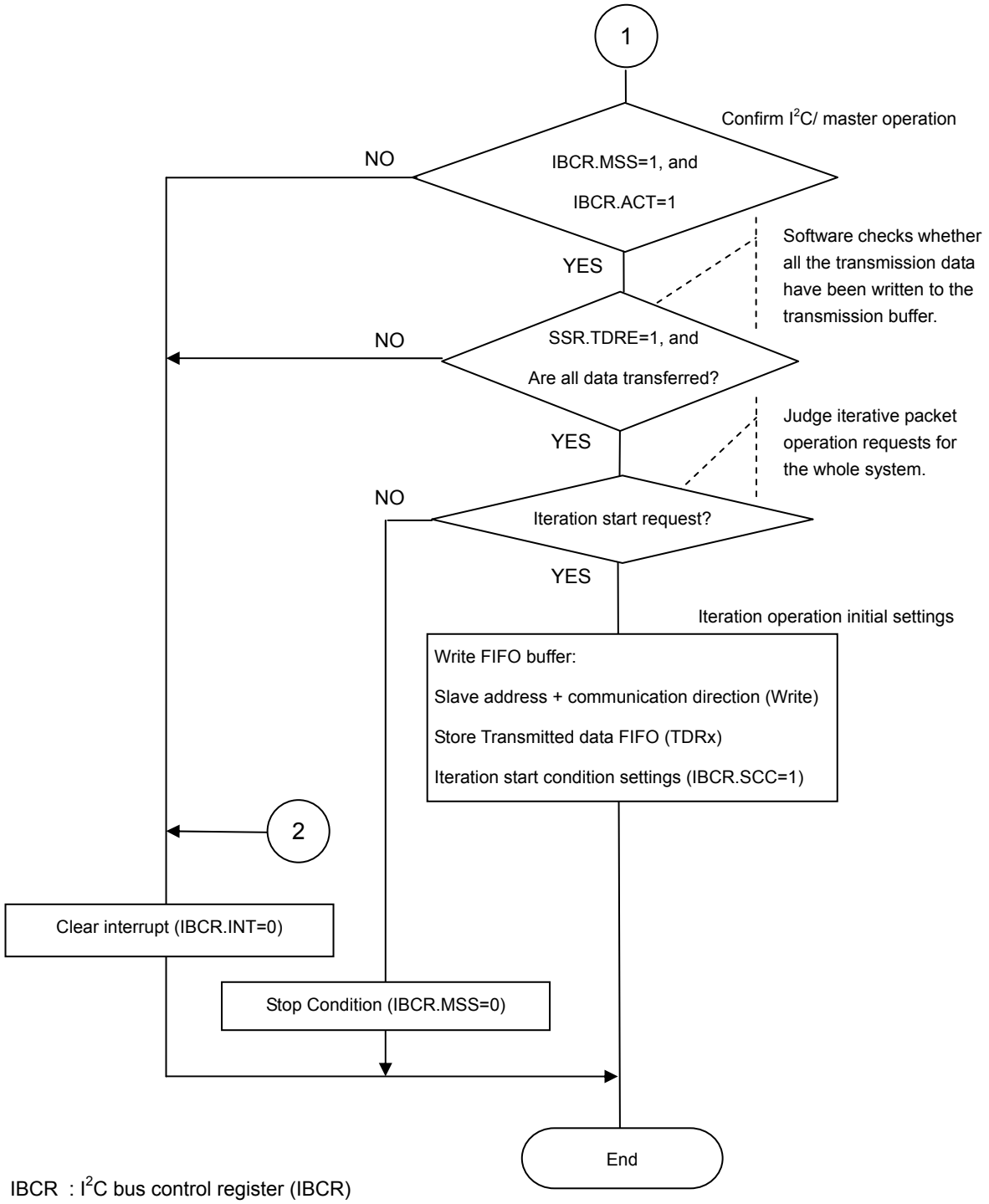
(Correct)
・ Data to which "0" is once written cannot be restored to "1". If "0" is rewritten in "1", an element is judged as defective by the data polling algorithm and the flash memory becomes either of the following states:
- The write time limit is exceeded and the timing limit overrun flag DQ5 (TLOV) of hardware sequence flag changes to "1".
[mcu_doc1072]</td></tr>
<tr><td>2010/
3/23</td><td>1146</td><td>31.5.1</td><td>The following lines were added to "Table 31.5-1 Command Sequence".
<table>
<tr><th rowspan="2">Command</th><th rowspan="2">Writing Count</th><th colspan="2">1st Time</th><th colspan="2">2nd Time</th><th colspan="2">3rd Time</th><th colspan="2">4th Time</th><th colspan="2">5th Time</th><th colspan="2">6th Time</th></tr>
<tr><th>Addr</th><th>Data</th><th>Addr</th><th>Data</th><th>Addr</th><th>Data</th><th>Addr</th><th>Data</th><th>Addr</th><th>Data</th><th>Addr</th><th>Data</th></tr>
<tr><td>Continuous mode</td><td>3</td><td>F5556H</td><td>AAAAH</td><td>FAAAAH</td><td>5555H</td><td>F5556H</td><td>2020H</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>Continuous writing</td><td>2</td><td>FXXXXH</td><td>A0A0H</td><td>PA</td><td>PD</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>Reset continuous mode</td><td>2</td><td>FXXXXH</td><td>9090H</td><td>FXXXXH</td><td>F0F0H or 0000H</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
</table>
[mcu_doc1069]</td></tr>
<tr><td>2011/
6/29</td><td>1167</td><td>31.6.7</td><td>< Attached document 2 > was added next to Chapter 31.6.7
[mcu_doc1081]</td></tr>
</table>

< Attached document 1-1 > Master Reception Interrupt Process



IBCR : I^{2}C bus control register (IBCR)
TDR : Transmitted data register (TDR)

< Attached document 1-2 > Master Transmission Interrupt Process



IBCR : I^{2}C bus control register (IBCR)
TDR : Transmitted data register (TDR)
SSR : Serial status register (SSR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

31.6.7 Continuous Mode Operation

This section explains the continuous mode operation.

■ Continuous mode

In this mode, if the continuous mode commands are written, the unlock cycle in usual command sequence is unnecessary. Therefore, it is possible to write data by the bus operation of two times instead of the bus operation of four times in the continuous mode (do not write the erase command.). Moreover, the read is usually operation and do not write the commands other than the continuous write command/the continuous mode reset command during the continuous mode.

To end this mode, write the continuous mode reset command. Therefore, it is not possible to terminate the continuous mode even when the reset command ($F0_H$) is written in this mode.

When the continuous mode reset command is written, it returns to usual read mode (For details, see Figure 31.6-3.).

■ Continuous write mode

It is possible to write data by the bus operation of two times in the continuous mode. The automatic write algorithm is started by writing the write setup command ($A0_H$) and the write data cycle (PA/PD) in the continuous mode. This command has the same function as usual write operation except data writing by the bus operation of two times (For details, see Figure 31.6-3.).

Figure 31.6-3 shows an example of the flash memory continuous write operation.

Figure 31.6-3 Continuous Write Procedure Example