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 MB91640A/645A Series Hardware Manual Rev. 1 (CM71-10154-1E).

FR80
32-BIT MICROCONTROLLER
MB91640A/645A 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 capacity(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	101	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]
2009/ 7/6	158	4.4.3	The table of [bit3 to bit0] was corrected as indicated by the shading below. MOSW3 / MOSW2 / MOSW1 / MOSW0 / Main Clock (MCLK) Oscillation Stabilization Wait Time 0 / 1 / 0 / 0 / 2^8 × Main clock (MCLK) period [mcu_doc1002]
2010/ 3/23	132	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]

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2011/
6/29</td><td>245</td><td>9.4.1</td><td>Figure 9.4-1 was deleted as indicated by the shading below.

<table><tr><th>Reset Resource</th><th>Initial Value</th></tr><tr><td></td><td></td></tr><tr><td>Timeout for software reset</td><td>1XXXXXX1</td></tr><tr><td>Register reading</td><td>00000000</td></tr></table>
[mcu_doc1088]</td></tr>
<tr><td>2010/
6/7</td><td>320</td><td>13.4.2</td><td>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]</td></tr>
<tr><td>2010/
6/7</td><td>324,
325,
326,
327,
328,
359,
363,
366,
369,
371</td><td>13.4.3

13.6.5,
13.6.7,
13.6.9,
13.6.10,
13.6.11</td><td>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]</td></tr>
<tr><td>2011/
6/29</td><td>324</td><td>13.4.3</td><td><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]</td></tr>
<tr><td>2010/
6/7</td><td>332</td><td>13.5.1</td><td>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]</td></tr>
<tr><td>2010/
6/7</td><td>338</td><td>13.5.2</td><td>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]</td></tr>
<tr><td>2011/
6/29</td><td>351</td><td>13.6.3</td><td><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]</td></tr>
<tr><td>2010/
3/23</td><td>401</td><td>13.12</td><td>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]</td></tr>
</table>

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2010/
3/23</td><td>402</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>403</td><td>13.12</td><td>"Figure 13.12-2 CS area example" was changed as indicated by the shading below.

(Error)
Setting example
0000 0000_H
0006 0000_H
0007 0000_H
000C 0000_H
0010 0000_H
0020 0000_H

(Corrected)
Setting example
0000 0000_H
0026 0000_H
0027 0000_H
002C 0000_H
0030 0000_H
0040 0000_H

[mcu_doc1060]</td></tr>
<tr><td>2009/
10/23</td><td>610</td><td>22.5.2</td><td>"Table 22.5-4 External Pins Used" was corrected as indicated by the shading below.

<table><tr><th></th><th>Even-numbered Channel</th></tr><tr><td>Input pin</td><td>3</td></tr><tr><td>Output pin</td><td>1</td></tr></table>
[mcu_doc1037]</td></tr>
<tr><td>2009/
10/23</td><td>611</td><td>22.5.2</td><td>"Table 22.5-6 Connections for I/O Mode 1" was corrected as indicated by the shading below.

<table><tr><th>Connection Source</th><th>Connection Destination</th></tr><tr><td>TOUT signal of ch.n</td><td>Output from the TIOAn pin</td></tr><tr><td>Input signal from the TIOBn pin</td><td>Input to ch.n as an ECK signal</td></tr><tr><td>Input signal from the TIOAn+1 pin</td><td>Input to ch.n as a TGIN signal</td></tr><tr><td>Input signal from the TIOBn+1 pin</td><td>Input to ch.n as a TIN signal</td></tr></table>
[mcu_doc1037]</td></tr>
<tr><td>2009/
10/23</td><td>624</td><td>22.5.8</td><td>"Table 22.5-24 Connection for I/O Mode 7" was corrected as indicated by the shading below.

<table><tr><th>Connection Source</th><th>Connection Destination</th></tr><tr><td>TOUT signal of ch.n</td><td>- ……
- Input to ch.n+1 as the TIN/TGIN/ECK signal</td></tr></table>
[mcu_doc1037]</td></tr>
</table>

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2009/
7/6</td><td>715</td><td>24.1</td><td>The description of "· Reload compare function" was corrected as indicated by the shading below.
· Reload/compare clear function: One of the following three types can be selected.
- Compare clear function
Clears the counter at the next up count timing when the specified value matches the counter value.
- Reload function
……
- Reload compare clear function
Compare clear function and reload function can be combined for use.
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>726</td><td>24.4.3</td><td>The description of [bit5] was corrected as indicated by the shading below.
[bit5]:UCRE (Counter clear enable bit)
This bit controls the clear operation of the counter by compare function.
If this bit is enabled, it clears the counter at the next up count timing when the counter value matches the value specified to the reload compare register (RCR0 to RCR3).
<table><tr><th>Written Value</th><th>Explanation</th></tr><tr><td>0</td><td>Disables compare clear function.</td></tr><tr><td>1</td><td>Enables compare clear function.</td></tr></table>
<Note>
This bit can control only the compare clear function. It does not affect comparison result match interrup.
……
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>734</td><td>24.6</td><td>The description of "●Reload/Compare function" was corrected as indicated by the shading below.
●Reload/Compare clear function
8/16-bit up/down counter can enable and disable the reload function and compare clear function by using the RLDE bit and UCRE bit in the counter control register (CCR0 to CCR3).
· Reload function
……
· Compare clear function
……
· Reload compare clear function
this is a function used by combining the reload function and compare clear function. ……
Table 24.6-1 shows how to set the reload function/compare clear function.
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>734</td><td>24.6</td><td>"Table 24.6-1 Setting the reload/compare function" was corrected as indicated by the shading below.
Table 24.6-1 Setting the reload/compare clear function
<table><tr><th>RLDE bit</th><th>UCRE bit</th><th>Explanation</th></tr><tr><td>0</td><td>0</td><td>Disable reload function/compare clear function</td></tr><tr><td>0</td><td>1</td><td>Disable reload function
Enable compare clear function</td></tr><tr><td>1</td><td>0</td><td>Enable reload function
Disable compare clear function</td></tr><tr><td>1</td><td>1</td><td>Enable reload function/compare clear function</td></tr></table>[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>735</td><td>24.6</td><td>The description of "■Clear event" was corrected as indicated by the shading below.
· Clear with the compare clear function
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>739,
742,
744</td><td>24.6.2,
24.6.3,
24.6.4</td><td>The description of "■Overview" was corrected as indicated by the shading below.
· Compare clear function
· Reload compare clear function
[mcu_doc1006]</td></tr>
</table>

<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2009/
7/6</td><td>740</td><td>24.6.2</td><td>The description of "●Operations when the compare function is used" was corrected as indicated by the shading below.
●Operations when the compare clear function is used
……
Figure 24.6-4 shows the operations when the compare clear function is used.
Figure 24.6-4 Operations when the compare clear function is used
……
<Note>
When the compare clear function is used, ……
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>741</td><td>24.6.2</td><td>The description of "●Operations when the reload compare function is used" was corrected as indicated by the shading below.
●Operations when the reload compare clear function is used
……, and during count upward, the compare clear function is used.
Figure 24.6-5 shows the operations when the reload compare clear function is used.
Figure 24.6-5 Operations when the reload compare clear function is used
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>743,
745</td><td>24.6.3,
24.6.4</td><td>The description of "●Operations when the compare function is used" and "●Operations when the reload compare function is used " was corrected as indicated by the shading below.
●Operations when the compare clear function is used
……
●Operations when the reload compare clear function is used
……
[mcu_doc1006]</td></tr>
<tr><td>2009/
7/6</td><td>762,
766,
773,
777</td><td>25.4.3,
25.4.4,
25.4.7,
25.4.8</td><td>The following <Note> was corrected as indicated by the shading below.
(Error) Byte access to these registers must be performed independently, or half word access must be performed with a … register (…).
(Corrected) Do not perform word access to these registers.
[mcu_doc1004]</td></tr>
<tr><td>2011/
6/29</td><td>805,
806</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 prohibited</td><td>Setting prohibited</td><td>Setting rohibited</td><td>Setting rohibited</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>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>
<tr><td>2011/
6/29</td><td>807,
808</td><td>25.6</td><td>Table 25.6-5 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= 32M z</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 prohibited</td><td>Setting prohibited</td><td>Setting rohibited</td><td>Setting rohibited</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>
</table>

Date	Page	Item	Description
2010/ 2/22	822	25.6.4	The following description was added above "Figure 25.6-6 DMA transfer operation (through scan conversion interrupt requests)". …… number for DMA transfer, see "CHAPTER 28 DMA Controller (DMAC)". ・In single conversion mode To perform DMA transfer, set the same value to the DMA block size and the interrupt generation FIFO stage number, and perform the next A/D activation after DMA is completed. ・In repeat conversion mode To perform DMA transfer, set 1 to the DMA block size, and 1 for the interrupt generation FIFO stage number. Figure 25.6-6 shows the DMA transfer operation. "Figure 25.6-6 DMA transfer operation (through scan conversion interrupt requests)" was corrected as 1. in < Attached document 1 >. The following <Note> was added to the lower part of "Figure 25.6-6 DMA transfer operation (through scan conversion interrupt requests)". Set the same value to the DMA block size and the interrupt generation FIFO stage number. Perform the next A/D activation after performing DMA transfer of all FIFO data.
2010/ 2/22	823	25.6.4	"Figure 25.6-7 DMA retransfer operation" was corrected as 2. in < Attached document 1 >. The following <Note> was added to the lower part of "Figure 25.6-7 DMA retransfer operation". Set 1 to block size of DMA, and 1 for the interrupt generation FIFO stage number.
2010/ 3/23	836	27.1	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]
2010/ 3/23	849, 902	27.4.2, 27.12.2	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]
2010/ 2/22	888	27.10	Added < Attached document 2 > as 27.10.
2011/ 12/16	948	27.15.1	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]
2010/ 2/22	952	27.18	Added < Attached document 3 > as 27.18.

Date	Page	Item	Description
2010/ 3/23	967	27.19.2	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^2C Bus Control Register (IBCR) ・I^2C 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]
2010/ 6/7	1016, 1022	27.21.1	The figure was corrected as < Attached document 5-1 > and < Attached document 5-2 >. [mcu_doc1081]
2010/ 2/22	1025	27.24	Added < Attached document 4-1 > and < Attached document 4-2 > as 27.24.
2009/ 7/6	1044	28.4.5	The following description was added at the end of <Note> of [bit26]. ・ When the end interrupt request is generated, the interrupt request will not be cleared, even if AIE is set to "0". Please write "0" to AC to clear the interrupt request. The following <Note> was added to [bit25]. When the transfer suspension interrupt request is generated, the interrupt request will not be cleared, even if SIE is set to "0". Please write "0" to SP to clear the interrupt request. The following <Note> was added to [bit24]. When the normal end interrupt request is generated, the interrupt request will not be cleared, even if NIE is set to "0". Please write "0" to NC to clear the interrupt request. [mcu_doc0997]
2009/ 7/6	1055	28.4.7	"Interrupt Request Level Where DMA Transfers Are Halted" in the table of [bit4 to bit0] was corrected as indicated by the shading below. (Error) Interrupt request of ... or higher (Corrected) Higher level of interrupt request than ... [mcu_doc0997]
2009/ 7/6	1056	28.5	The following description was added at the end of <Notes>. ・ When the interrupt request of DMA controller is generated, the interrupt request will not be dropped even if the interrupt enable bits (AIE, SIE, and NIE) are set to "0". Please write "0" to the interrupt request flags (AC, SP, and NC) to clear the interrupt request. [mcu_doc0997]
2009/ 7/6	1062	28.6.2	The following <Note> was added to the lower part of "Table 28.6-2 Detect Condition of Transfer Requests and Transfer Request Source". The interrupt request of the peripheral function does not start transfer, even if CE is changed from "0" to "1" during the interrupt request is generated because it is for edge detection. Please execute an interrupt enable etc. of the peripheral function after setting CE to "1". [mcu_doc0997]
2009/ 7/6	1077	28.6.6	The description was corrected as indicated by the shading below. (Error) The DMA transfer restarts when the interrupt level reaches or falls below the level … (Corrected) The DMA transfer restarts when the interrupt request is cleared, and the interrupt level reaches or falls below the level … "Interrupt Request Level Where DMA Transfers Are Halted" in "Table 28.6-9 Interrupt Request Level where DMA Transfers are Halted." was corrected as indicated by the shading below. (Error) Interrupt request of ... or higher (Corrected) Higher level of interrupt request than ... [mcu_doc0997]

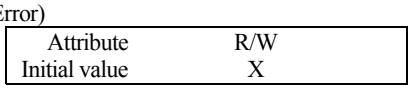
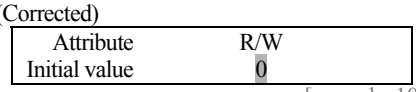
<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2009/
10/23</td><td>1102,
1103</td><td>29.3.7</td><td>The description of [bit2 to bit0] was corrected as indicated by the shading below.
<table>
<tr><th colspan="2">by the DMAC</th><th rowspan="2">Flag Bit That Requests DMA Transfer Stop *</th></tr>
<tr><th></th><th>Flag Bit To Be Cleared</th></tr>
<tr><td></td><td>SSR8: RDRF</td><td>SSR8: ORE
SSR8: FRE
SSR8: PE</td></tr>
<tr><td></td><td>SSR9: RDRF</td><td>SSR9: ORE
SSR9: FRE
SSR9: PE</td></tr>
<tr><td></td><td>SSR10: RDRF</td><td>SSR10: ORE
SSR10: FRE
SSR10: PE</td></tr>
<tr><td></td><td>SSR11: RDRF</td><td>SSR11: ORE
SSR11: FRE
SSR11: PE</td></tr>
<tr><td></td><td>SSR8: TDRE
FCR18: FDRQ</td><td>-</td></tr>
<tr><td></td><td>SSR9: TDRE
FCR19: FDRQ</td><td>-</td></tr>
<tr><td></td><td>SSR10: TDRE
FCR110: FDRQ</td><td>-</td></tr>
<tr><td></td><td>SSR11: TDRE
FCR111: FDRQ</td><td>-</td></tr>
</table>
* When RIE=1, a stop request is generated if either of the flag is 1.
[mcu_doc1035]</td></tr>
<tr><td>2009/
10/23</td><td>1111</td><td>29.3.11</td><td>The description of [bit2 to bit0] was corrected as indicated by the shading below.
<table>
<tr><th>Flag Bit To Be Cleared</th><th>Flag Bit That Requests DMA Transfer Stop *</th></tr>
<tr><td>ICS45: ICP4</td><td rowspan="5">SSR7: ORE
SSR7: FRE
SSR7: PE</td></tr>
<tr><td>ICS45: ICP5</td></tr>
<tr><td>ICS67: ICP6</td></tr>
<tr><td>ICS67: ICP7</td></tr>
<tr><td>SSR7: RDRF</td></tr>
</table>
* When RIE=1, a stop request is generated if either of the flag is 1.</td></tr>
<tr><td>2009/
10/23</td><td>1128</td><td>29.4.1</td><td>The description of "■Operation" was corrected as indicated by the shading below.

3. A DMA transfer request is generated, and the DMA controller (DMAC) is activated.
4. A clear of an interrupt request flag in the peripheral functions is requested from DMA controller (DMAC) for the block size multiplied by the number of transfers per transfer.
5. DMA transfer is finished.

[mcu_doc1035]</td></tr>
</table>

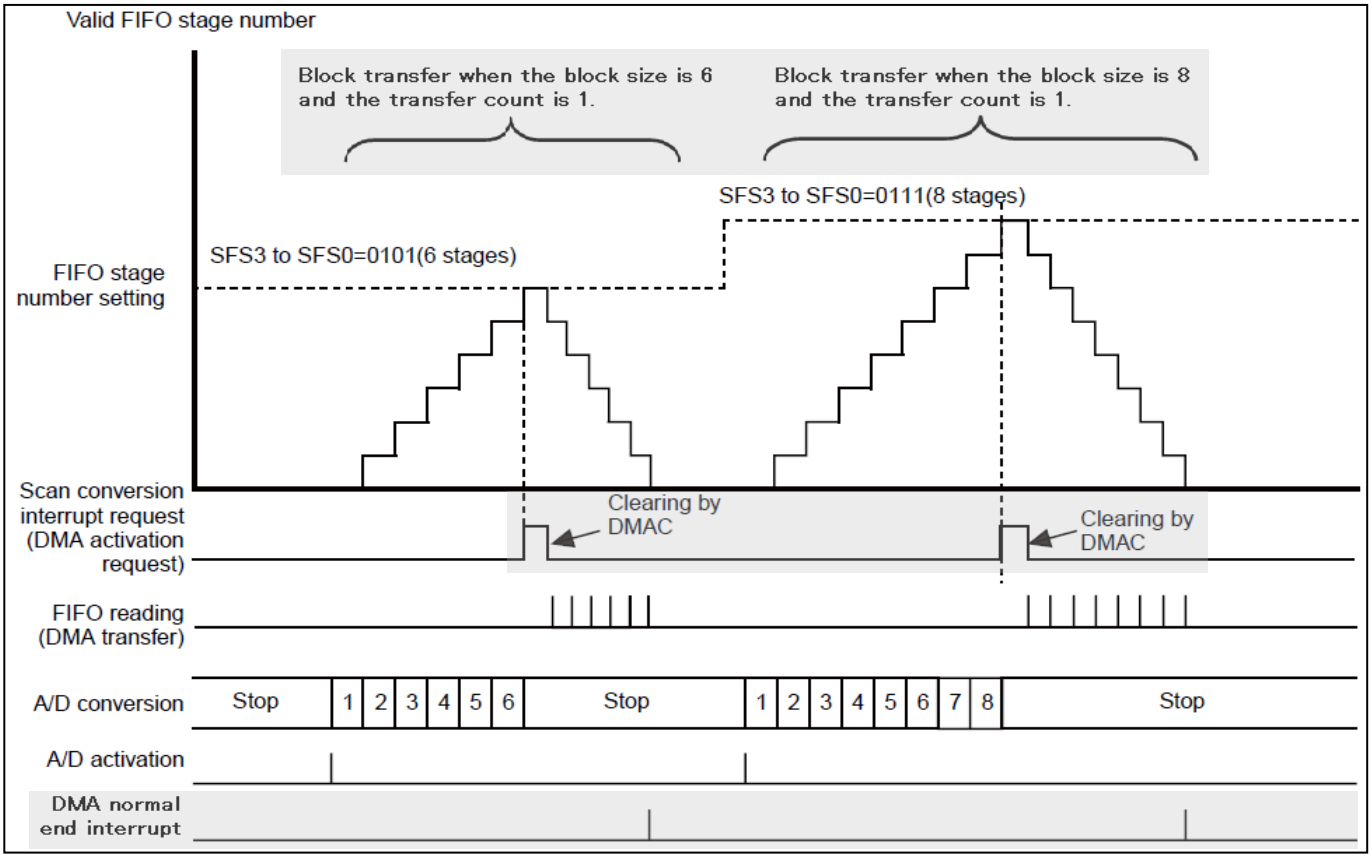
<table>
<tr><th>Date</th><th>Page</th><th>Item</th><th>Description</th></tr>
<tr><td>2011/
6/29</td><td>1155</td><td>31.6.2</td><td>■ Notes on Data Writing was corrected as indicated by the shading below.

(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".

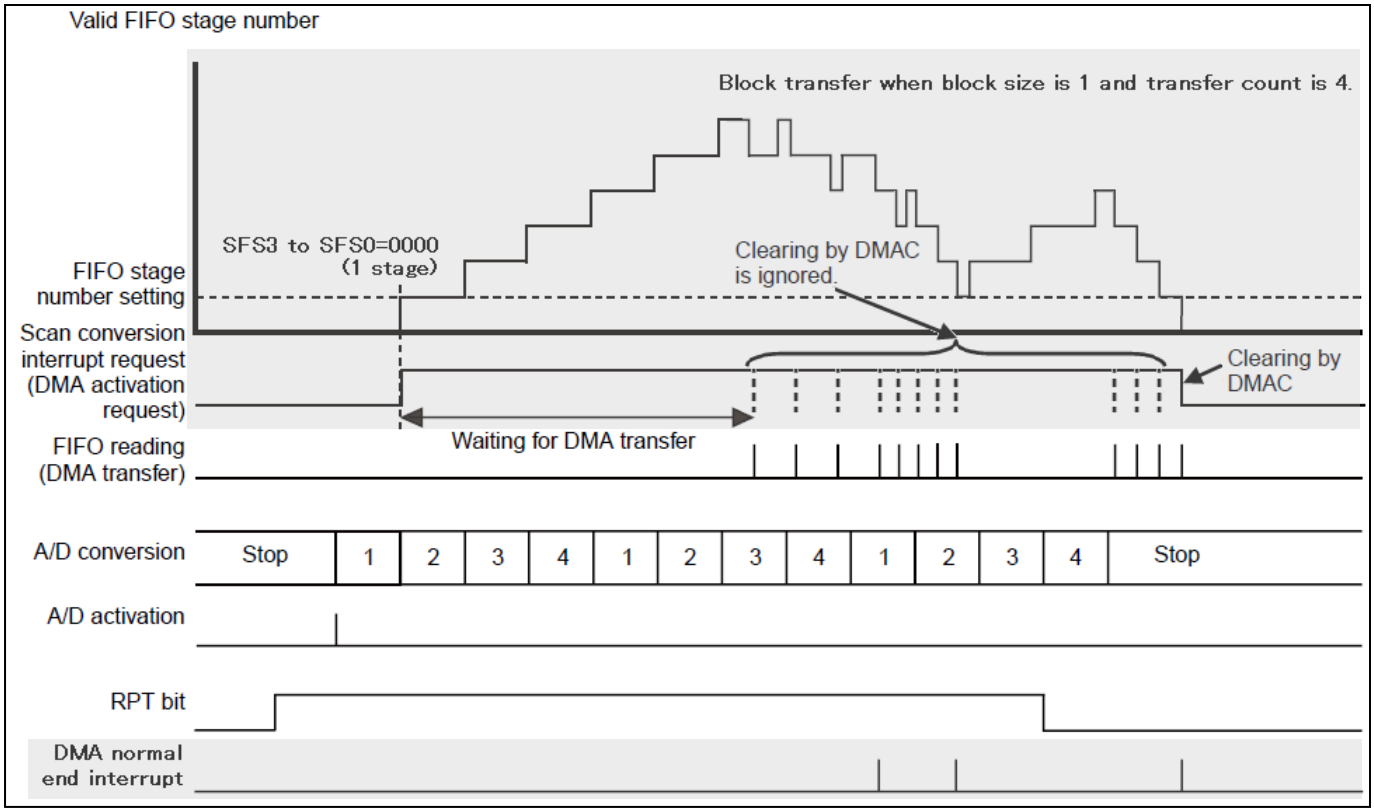
(Correct)
・Data to which "0" is once written cannot be restored to "1". If "0" is rewritten in "1", an element is judged as defective by the data polling algorithm and the flash memory becomes either of the following states:
- The write time limit is exceeded and the timing limit overrun flag DQ5 (TLOV) of hardware sequence flag changes to "1".
[mcu_doc1072]</td></tr>
<tr><td>2011/
6/29</td><td>1160</td><td>31.6.7</td><td>< Attached document 6 > was added next to Chapter 31.6.7
[mcu_doc1081]</td></tr>
<tr><td>2009/
10/23</td><td>1170</td><td>32.3.3</td><td>The initial value in "Figure 32.3-3 Bit Configuration of Wild Register Enable Register (WREN)" was corrected as indicated by the shading below.
(Error)
 
(Corrected)
 
[mcu_doc1039]</td></tr>
</table>

< Attached document 1 > * : Corrected part

1. DMA transfer operation (through scan conversion interrupt requests)



2. DMA retransfer operation



< Attached document 2 >

Notes on UART Mode

The notes for when you use the UART mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.

< Attached document 3 >

Notes on CSIO Mode

The notes for when you use the CSIO mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.

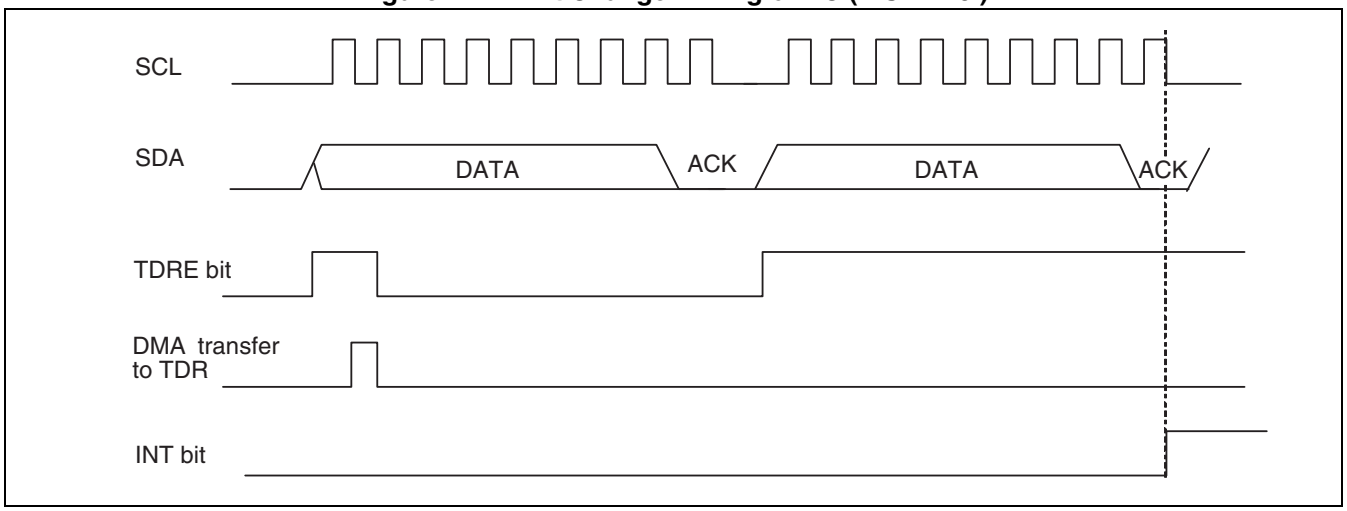
< Attached document 4-1 >

Notes on I²C Mode

The notes for when you use the I²C mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.
- In I²C mode, if there is no valid data in transmission register (TDR), and transmission data empty flag bit (TDRE) is "1", the interrupt flag (INT) becomes "1" as shown in Figure 1 when the data on I²C bus for 9 bits (WSEL=0) or for 8 bits (WSEL=1) is transmitted. When the interrupt flag (INT) becomes "1" during DMA transfer, DMA transfer cannot be continued unless clearing the bit to "0" by software. (Common to master transmission, slave transmission, mater reception, and slave reception.)

Figure 1 INT Bit Change Timing of I²C (WSEL= 0)



To perform DMA transfer in I²C mode, since the specification is as shown above, such operations listed below are required for performing DMA transfer to TDR before the interrupt flag (INT) becomes "1". Below operations are possible to perform to prioritize DMA transfer of I²C.

- Use DMA which has a higher priority (channel number is small). It is enabled to use by fixing the priority setting bit (AT=0).
- Set the value of DMA-halt by interrupt level bit as small as possible (LVL4-LVL0 bit in DILVR register).

< Attached document 4-2 >

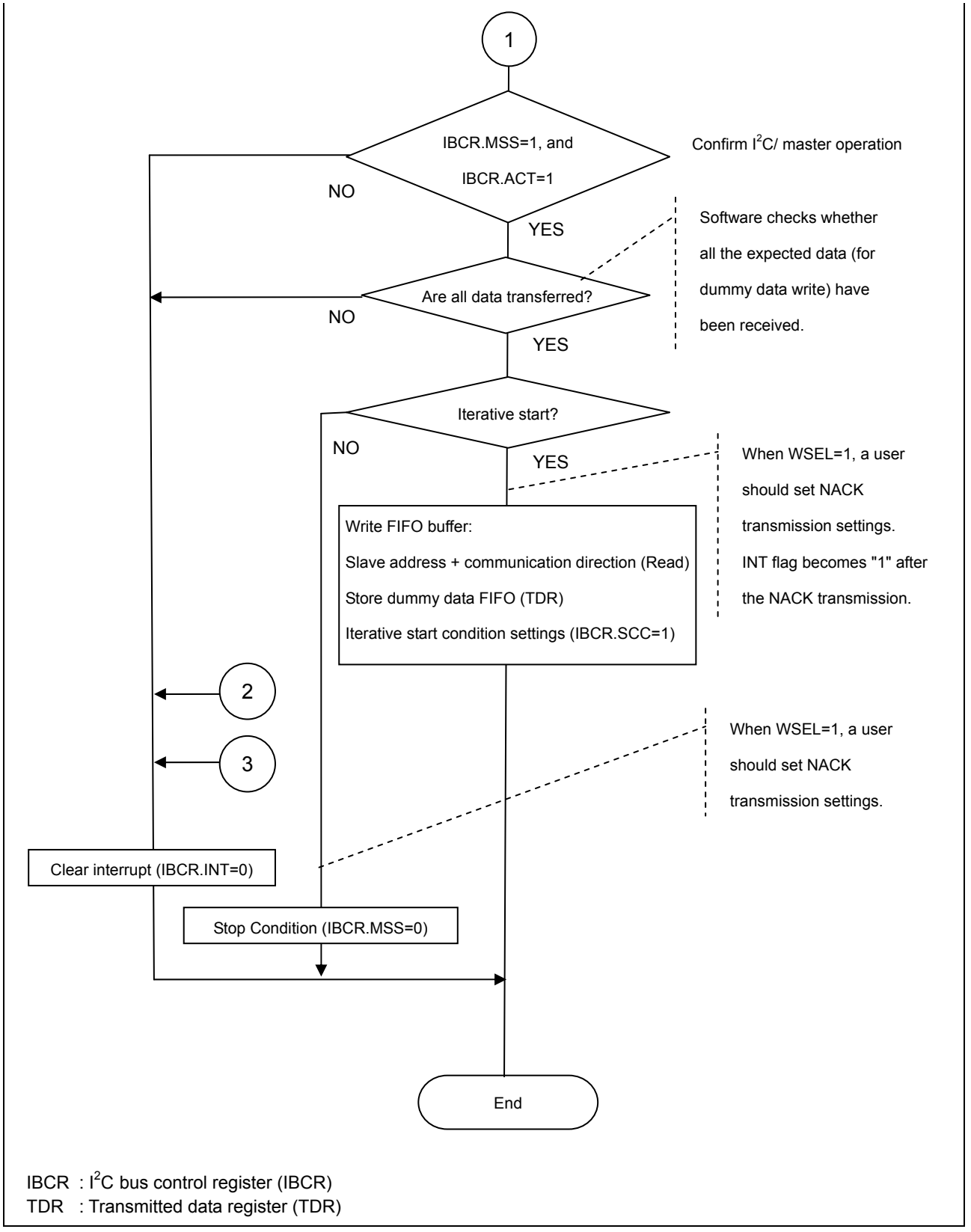
- In case of writing the transmission data to transmission data register (TDR) by DMA transfer after transmission data empty flag (SSR: TDRE) becomes "1", or writing the data by software confirming the transmission data empty flag (SSR:TDRE), transmission data empty flag (SSR:TDRE) may not become "0". Therefore, the transmission data should be written before SCL in ACK field falls. There are no restrictions on writing the transmission data by software after the interrupt flag (IBCR:INT) becomes "1".
  When performing DMA transfer or sending the data by software confirming the transmission data empty flag (SSR:TDRE), please follow below procedures if the data cannot be written before SCL in ACK field falls.
  - Setting

    Set the timing of interrupt flag (IBCR:INT) becoming "1" to the 8th bit (WSEL=1).
  - Procedures

    To transmit or receive data by master, the following procedures are required. To transmit or receive data by slave, it is not required to perform the following.

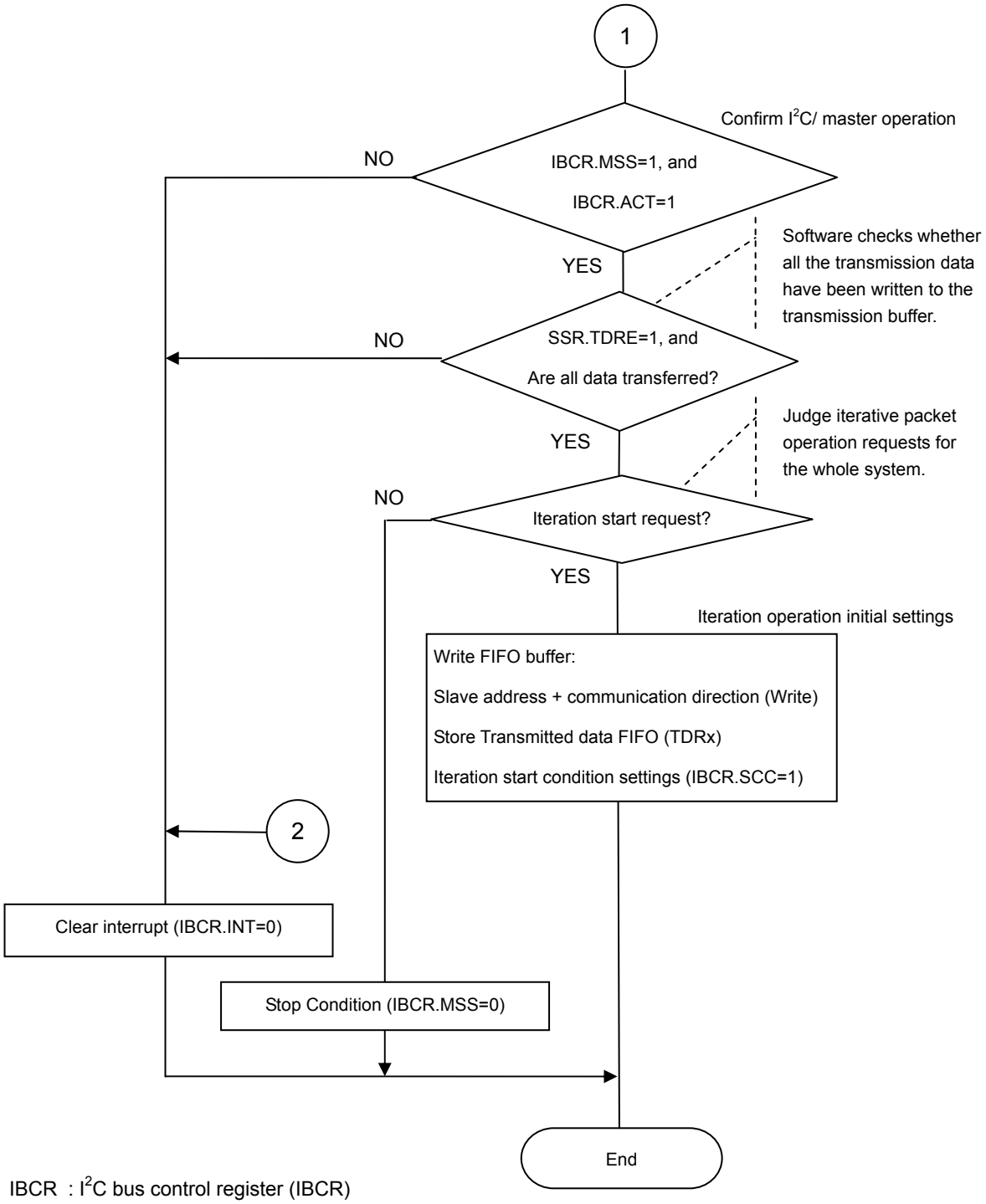
    1. Write the first byte (slave address) to the transmission data register by software.
    2. Set to 8-bit for wait selection (IBCR:WSEL="1" write) at the same time that master is started (IBCR:MSS="1" write).
    3. After sending the first byte, the interrupt flag (IBCR:INT) becomes "1". Write the second byte to transmission data register (TDR) by software after confirming ACK response (IBSR:RACK="0"). Set the DMAC, and activate DMA transfer, then write "0" to interrupt flag (IBCR:INT).
    4. After transmission and reception are completed, terminate the master (IBCR:MSS="0" write) or reboot (IBCR:SCC="1" write).

< Attached document 5-1 > Master Reception Interrupt Process



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

< Attached document 5-2 > 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