Memory Stick Information for Developers

Memory Stick ▶▶ Command Control

6. Command Control

6.1. Command Overview

The two types of commands (CMD), flash control command and function control command, are executed by Flash Memory Controller on Memory Stick, with issuing of SET_CMD TPC. The operation flow of these commands on Memory Stick is shown below.

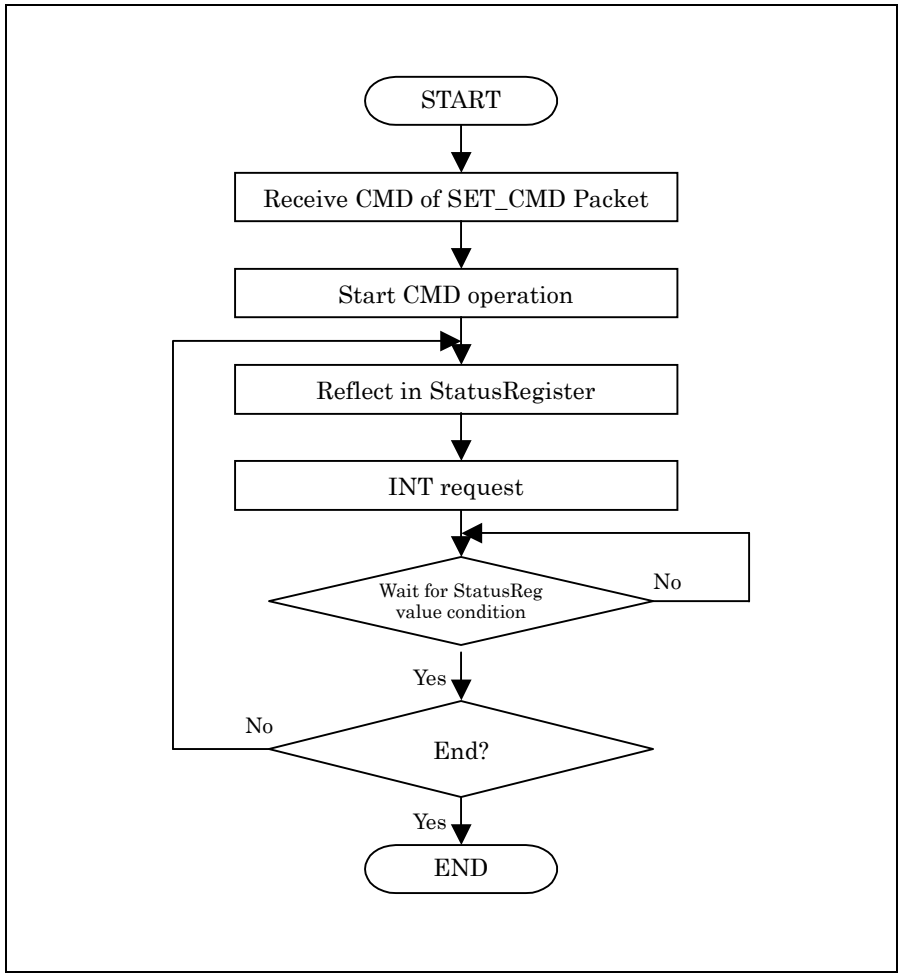

Fig. 6.1.1 CMD Flowchart

6.1.1. Flash CMD

Flash control commands which access Flash Memory Chip on Memory Stick.

Table 6.1.1 Flash CMD

Function		Description
BLOCK_READ		Read data from the designated Block (page).
BLOCK_WRITE		Write data to the designated Block (page).
BLOCK_END		When in BLOCK_READ/BLOCK WRITE Block Access Mode, ends in a Page halfway through a block.
BLOCK_ERASE		Erase all data in the designated Block.
FLASH_STOP		Force to stop all actions of Flash Memory.

While above command is executing, bit 7 and 6 (MB, FB0) in StatusRegister0 shown below become 1.

Table 6.1.2 StatusRegister0 Value

	D7	D6	D5	D4	D3	D2	D1	D0
StatusRegister0	**1** **MB**	**1** **FB0**	X BE	X BF	X –	X –	X SL	X WP

6.1.2. Function CMD

Function control command.

Table 6.1.3 Function CMD

Function		Description
SLEEP		Suspend the clock oscillation inside Memory Stick.
CLEAR_BUF		Clear the Page Buffer (BE=1, BF=0).
RESET		Reset the Flash Memory Controller, and initialize the Register.

While above command is executing, bit 7 (MB) in StatusRegister0 shown below become 1.

Table 6.1.4 StatusRegister0 Value

	D7	D6	D5	D4	D3	D2	D1	D0
StatusRegister0	**1** **MB**	0 FB0	X BE	X BF	X –	X –	X SL	X WP

6.1.3. Factors of INT

During or after command execution, the execution result is reflected to INT register, and INT occurs at BS0. The relation between command and each bit of INT register showing the factor of INT is shown below.

Table 6.1.5 INT Register Bit Change

	INT Register			
SET_CMD CMD	b7 : CED	b6 : ERR	b5 : BREQ	b0 : CMDNK
BLOCK_READ	Operating 0 / End 1	FlashReadError	Data request:1	Operation condition error
BLOCK_WRITE	Operating 0 / End 1	FlashWriteError	Data request:1	
BLOCK_END	Operating 0 / End 1	—	X	
BLOCK_ERASE	Operating 0 / End 1	FlashEraseError	—	
FLASH_STOP	Operating 0 / End 1	—	—	
SLEEP	Operating 0 / End 1	—	X	
CLEAR_BUF	Operating 0 / End 1	—	0	
RESET	0	0	0	0

- CED

  Occurs at the end of command execution transferred by SET_CMD.

  CED bit becomes 1 in concurrence with ERR bit in the following three cases:

  - When FlashWriteError occurs at BLOCK_WRITE.
  - When uncorrectable FlashReadError occurs at BLOCK_READ.
  - When FlashEraseError occurs at BLOCK_ERASE.

- ERR

  Occurs at FlashReadError, FlashWriteError and FlashEraseError.

  - When FlashWriteError occurs at BLOCK_WRITE. (In concurrence with CED.)
  - When FlashReadError occurs at BLOCK_READ. (In concurrence with CED if uncorrectable.)
  - When FlashEraseError occurs at BLOCK_ERASE. (In concurrence with CED.)

- BREQ
  - Occurs upon request for Read (READ_PAGE_DATA TPC) from PageBuffer (Full) at BLOCK_READ.
  - Occurs upon request for Write (WRITE_PAGE_DATA TPC) to PageBuffer (Empty) at BLOCK_WRITE.

- CMDNK
  - ・Occurs during command execution (when MB=1 of StatusRegister0/1).
  - ・Occurs when ParameterRegister setting is false.
- Commands executable while other command is being executed(MB=1 of Status Register 0/1).
  - ・BLOCK_END (Only when BLOCK_READ or BLOCK_WRITE in block access mode is executed.)
  - ・RESET (for all commands)

6.1.4. INT Timeout

When waiting for INT after CMD is executed by SET_CMD TPC, and INT does not occur due to a malfunction of Memory Stick, the timeout shall be as below:

Table 6.1.6 INT Timeout Specification

CMD	Timeout (ms)
BLOCK_READ	5
BLOCK_WRITE	10
BLOCK_ERASE	100
FLASH_STOP	5
RESET	-
SLEEP	1
CLEAR_BUF	1

6.1.5. INT Clear factor

INT is cleared by reading INT Register with executing READ_REG_TPC or GET_INT_TPC, it returns from High level to Low during the BS0 period.

If a new INT factor occurs while the READ_REG_TPC or GET_INT_TPC are executing, it needs to pay attention to the system with selecting a low frequency as SLCK, because of being no guaranteed whether an INT signal outputs or not, at the BS0 period after processing TPC.

6.2. Flash CMD

Parameter Register setting shows the parameter value to be set by WRITE_REG TPC, just before the command transfer of SET_CMD TPC.

Status Register shows the status value at INT occurrence by command execution.

6.2.1. BLOCK_READ Operation in Single Page Access Mode

DataArea and ExtraDataArea of single page are read from PageBuffer and Registers.

6.2.1.1. Parameter Register Setting

Table 6.2.1 Parameter Registers for Single Page Access BLOCK_READ

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0 –
Command Parameter	**0** **CP2**	**0** **CP1**	**1** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	Page Address							

6.2.1.2. Status Register Value

6.2.1.2.1. INT at Normal End

Table 6.2.2 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller already read							

6.2.1.2.2. INT at FlashReadError

Table 6.2.3 Status Registers with FlashReadError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**0/1** **DTER**	**0/1** **UCDT**	**0/1** **EXER**	**0/1** **UCEX**	**0/1** **FGER**	**0/1** **UCFG**
Page Address	Page Address which Memory Stick Controllerread and error occurred							

6.2.2. BLOCK_READ Operation in Block Access Mode

DataArea and ExtraDataArea from the target page to the last page of that block are read consecutively.

6.2.2.1. Parameter Register Setting

Table 6.2.4 Parameter Registers for Block Access BLOCK_READ

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0 –
Command Parameter	**0** **CP2**	**0** **CP1**	**0** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	First Page Address							

6.2.2.2. Status Register Value

6.2.2.2.1. INT at Normal FlashRead

Table 6.2.5 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	0 CED	0 ERR	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
Status Register0	1 MB	X FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
Status Register1	1 MB	X FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address							

6.2.2.2.2. INT at Correctable FlashReadError

Table 6.2.6 Status Registers with Correctable FlashReadError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	0 CED	**1** **ERR**	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
Status Register0	1 MB	X FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
Status Register1	1 MB	X FB1	**0/1** **DTER**	0 UCDT	**0/1** **EXER**	0 UCEX	**0/1** **FGER**	0 UCFG
Page Address	Page Address which Memory Stick Controller read and error occurred							

6.2.2.2.3. INT at Uncorrectable FlashReadError

Table 6.2.7 Status Registers with Uncorrectable FlashReadError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**0/1** **DTER**	**0/1** **UCDT**	**0/1** **EXER**	**0/1** **UCEX**	**0/1** **FGER**	**0/1** **UCFG**
Page Address	Page Address which Memory Stick Controller read and error occurred							

6.2.2.2.4. INT at Last Page or BLOCK_END

Table 6.2.8 Status Registers with Command EnD INT at Last Page or BLOCK_END

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
Status Register0	**0** **MB**	0 FB0	0 BE	**1** **BF**	X –	X –	0 SL	X WP
Status Register1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller read							

6.2.3. BLOCK_READ Operation in ExtraDataAccessMode

Only ExtraDataArea of single page is read.

6.2.3.1. Parameter Register Setting

Table 6.2.9 Parameter Registers for ExtraDataAccess BLOCK_READ

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0 –
Command Parameter	**0** **CP2**	**1** **CP1**	**X** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	Page Address							

6.2.3.2. Status Register Value

6.2.3.2.1. INT at Normal End

Table 6.2.10 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller read							

6.2.3.2.2. INT at FlashReadError

Table 6.2.11 Status Registers with FlashReadError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	**0/1** **EXER**	**0/1** **UCEX**	**0/1** **FGER**	**0/1** **UCFG**
Page Address	Page Address which Memory Stick Controller read and error occurred							

6.2.4. BLOCK_WRITE Operation in Single Page Access Mode ~ WritePage ~

DataArea and ExtraDataArea of single page are written.

6.2.4.1. Parameter Register Setting

Table 6.2.12 Paramter Registers for Single Page BLOCK_WRITE

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0
Command Parameter	**0** **CP2**	**0** **CP1**	**1** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	Page Address							

6.2.4.2. Extra Data Register Value

Table 6.2.13 Extra Data Registers

	D7	D6	D5	D4	D3	D2	D1	D0
ExtraData Registers	Extra area data to write							

6.2.4.3. Status Register Value

6.2.4.3.1. INT at PageBufferEmpty

Table 6.2.14 Status Registers with Buffer REQuest INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	0 CED	0 ERR	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
StatusRegister0	1 MB	0 FB0	1 BE	0 BF	X –	X –	0 SL	X WP
StatusRegister1	1 MB	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Don't care.							

6.2.4.3.2. INT at Normal End

Table 6.2.15 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	0 BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	1 BE	0 BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote							

6.2.4.3.3. INT at FlashWriteError

Table 6.2.16 Status Registers with FlashWriteError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	0 BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	1 BE	0 BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**1** **DTER**	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote and failed							

6.2.5. BLOCK_WRITE Operation in BlockAccessMode ~ WriteBlock ~

DataArea and ExtraDataArea from the target page to the last page of that block are consecutively written.

6.2.5.1. Parameter Register Setting

Table 6.2.17 Parameter Registers for Block Access BLOCK_WRITE

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0 –
Command Parameter	**0** **CP2**	**0** **CP1**	**0** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	First Page Address							

6.2.5.2. Extra Data Register Value

Table 6.2.18 Extra Data Registers

	D7	D6	D5	D4	D3	D2	D1	D0
Extra Data Registers	Extra area data to write							

6.2.5.3. Status Register Value

6.2.5.3.1. INT at Data Transfer Request to PageBuffer (BE=1)

Table 6.2.19 Status Registers with Buffer REQuest INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	0 CED	0 ERR	**1** **BREQ**	X –	X –	X –	X –	0 CMDNK
StatusRegister0	1 MB	X FB0	1 BE	0 BF	X –	X –	0 SL	X WP
StatusRegister1	1 MB	X FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote							

6.2.5.3.2. End INT at Last Page or BLOCK_END

Table 6.2.20 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	0 BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	1 BE	0 BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote							

6.2.5.3.3. INT Occurrence by FlashWrite Error

Table 6.2.21 Status Registers with FlashWriteError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	0 BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	1 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote and failed							

6.2.6. BLOCK_WRITE Operation in ExtraDataAccessMode

Only ExtraDataArea of single page is written.

6.2.6.1. Parameter Register Setting

Table 6.2.22 Parameter Registers for Extra Data Access BLOCK_WRITE

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0
Command Parameter	**0** **CP2**	**1** **CP1**	**X** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	Page Address							

6.2.6.2. ExtraDataRegister Setting

Table 6.2.23 Extra Data Registers

	D7	D6	D5	D4	D3	D2	D1	D0
ExtraDataRegisters	Extra area data to write							

6.2.6.3. Status Register Value

6.2.6.3.1. INT at Normal End

Table 6.2.24 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote							

6.2.6.3.2. INT at FlashWriteError

Table 6.2.25 Status Registers with FlashWriteError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	X BREQ	X –	X –	X –	X	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**1** **DTER**	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote and failed							

6.2.7. BLOCK_WRITE Operation in OverWriteMode

Overwrite the OverwriteFlag of single page.

6.2.7.1. Parameter Register Setting

Table 6.2.26 Parameter Registers for Overwrite BLOCK_WRITE

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0
Command Parameter	**1** **CP2**	**X** **CP1**	**X** **CP0**	X –	X –	X –	X –	X –
Block Address	Block Address							
Page Address	Page Address							

6.2.7.2. OverwriteFlag Setting

Table 6.2.27 OverwriteFlag

	D7	D6	D5	D4	D3	D2	D1	D0
OverwriteFlag	Flag Mask Data to write							

6.2.7.3. Status Register Value

6.2.7.3.1. INT at Normal End

Table 6.2.28 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote							

6.2.7.3.2. INT at FlashWriteError

Table 6.2.29 Status Registers with FlashWriteError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**1** **DTER**	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG
Page Address	Page Address which Memory Stick Controller wrote and failed							

6.2.8. BLOCK_END

- BLOCK_READ CMD in Block Access Mode is terminated by the BLOCK_END CMD when reading of a page halfway through a block is completed. Upon occurrence of INT by BREQ=1, BLOCK_END shall be executed, and then the reading is terminated at that page (MB=0, FB=0 of Status Register0).
- BLOCK_WRITE CMD in Block Access Mode is terminated by the BLOCK_END when writing to a page halfway through a block is completed. Upon occurrence of INT by BREQ=1, BLOCK_END CMD shall be executed, and then the writing is finished.

6.2.8.1. Status Register Value

See the section on BLOCK_WRITE/BLOCK_READ action in Block Access Mode.

6.2.9. BLOCK_ERASE

All pages in a block are erased.

6.2.9.1. Parameter Register Setting

Table 6.2.30 Parameter Registers for BLOCK_ERASE

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	1 BAMD	0 ATEN	X –	X –	X –	X –	X –	0
Command Parameter	X CP2	X CP1	X CP0	X –	X –	X –	X –	X –
Block Address	Block Address							

6.2.9.2. Status Register Value

6.2.9.2.1. INT at Normal End

Table 6.2.31 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	0 –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG

6.2.9.2.2. INT at FlashEraseError

Table 6.2.32 Status Registers with FlashEraseError INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	**1** **ERR**	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	0 –	0 SL	X WP
StatusRegister1	**0** **MB**	0 FB1	**1** **DTER**	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG

6.2.10. FLASH_STOP

To send a reset command to Flash Memory chip.

Used to recover from the abnormal state of Flash Memory chip (e.g. MB=0, FB0=1 of StatusRegister0).

6.2.10.1. Status Register Value (at End INT)

Table 6.2.33 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	**0** **FB0**	X BE	X BF	X –	X –	0 SL	X WP
StatusRegister1	**0** **MB**	**0** **FB1**	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG

6.3. Function CMD

6.3.1. SLEEP

To suspend the oscillation for the internal clock of Memory Stick.

6.3.1.1. Status Register Value (at End INT)

Table 6.3.1 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	X BREQ	X –	X –	X –	X –	0 CMDNK
StatusRegister0	**0** **MB**	0 FB0	X BE	X BF	X –	0 –	**1** **SL**	X WP
StatusRegister1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG

6.3.1.2. WakeUp Processing

Memory Stick wakes up and performs a packet processing, when a packet transfer by SET_CMD TPC or WRITE_REG TPC occurs at SL=1 in Status Register 0.

When the above TPC is not performed correctly, Memory Stick will be interrupted to wake up and will become sleep state again. When CMD cannot be executed after transferring SET_CMD TPC, CMDNK will be output and Memory Stick will become sleep state again.

For other TPC, Memory Stick will perform a packet processing, but will not wake up. (See Table 5.9.1 in Section 5)

BS3 BSY of Serial I/F continues until waking up. (See Section 6)

6.3.2. CLEAR_BUF

To clear the data in PageBuffer. (BF=0,BE=1 of Status Register0)

6.3.2.1. Status Register Value (at End INT)

Table 6.3.2 Status Registers with Command EnD INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**1** **CED**	0 ERR	**0** **BREQ**	X –	X –	X –	X –	0 CMDNK
Status Register0	**0** **MB**	0 FB0	**1** **BE**	**0** **BF**	X –	X –	0 SL	X WP
Status Register1	**0** **MB**	0 FB1	0 DTER	0 UCDT	0 EXER	0 UCEX	0 FGER	0 UCFG

6.3.3. RESET

Flash Memory Controller is forced to reset. State after reset will be as follows:

- Registers are initial value (Only the registers accessed by READ_REG/WRITE_REG)
- PageBuffer status is clear (BF=0,BE=1 of Status Register 0).
- BusState is BS0. INT has not generated.

Flash Memory Chip will not be reset during the execution of BLOCK_ERASE, BLOCK_READ or BLOCK_WRITE CMD of SET_CMD TPC (remains as FB0=1 of StatusRegister0).

→ See section on FLASH_STOP for reset of Flash Memory chip.

6.3.3.1. Status Register Value (INT does not occur at End)

Table 6.3.3 Status Registers after RESET CMD

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	**0** **CED**	**0** **ERR**	**0** **BREQ**	X –	X –	X –	X –	**0** **CMDNK**
StatusRegister0	**0** **MB**	X FB0	**1** **BE**	**0** **BF**	X –	X –	**0** **SL**	X WP
StatusRegister1	**0** **MB**	X FB1	**0** **DTER**	**0** **UCDT**	**0** **EXER**	**0** **UCEX**	**0** **FGER**	**0** **UCFG**

6.4. Access to Attribute ROM

To read ATTRIB ROM information.

Access to ATTRIB ROM is prohibited except at the time of boot making.

6.4.1. Parameter Register Setting

Table 6.4.1 Parameter Registers for Read Attribute ROM

	D7	D6	D5	D4	D3	D2	D1	D0
System Parameter	X BAMD	**1** **ATEN**	X –	X –	X –	X –	X –	0 –
Command Parameter	X CP2	X CP1	X CP0	X –	X –	X –	X –	X –

Access by READ_PAGE_DATA TPC is switched from usual PageBuffer to ATTRIB ROM with above setting.

During data transfer of 512Bytes, data after ATTRIB Code becomes 0xFF.

6.5. Acceptable CMD

To execute each CMD of SET_CMD TPC, Status Register0 and Parameter Register shall be as follows.

Table 6.5.1 Acceptable CMD

	Status Register0						Parameter Registers			
CMD	MB	FB	BE	BF	SL	WP	ATEN	CP 2-0	Block Address	Page Address
BLOCK_READ	0	0	X	X	X	X	0	0XX	Note 1	Note 2
BLOCK_WRITE	0	0	X	X	X	X	0	X	Note 1	Note 2
BLOCK_END	1Note3	X	X	X	X	X	X	X	X	X
BLOCK_ERASE	0	0	X	X	X	X	X	X	Note 1	X
SLEEP	0	X	X	X	0	X	X	X	X	X
CLEAR_BUF	0	X	X	X	X	X	X	X	X	X
FLASH_STOP	0	X	X	X	X	X	X	X	X	X
RESET	X	X	X	X	X	X	X	X	X	X

Note 1) BlockAddress range implemented on Memory Stick (Invalid BlockAddress is not accepted).
Note 2) PageAddress range implemented on Memory Stick (Invalid PageAddress is not accepted).
Note 3) Only when BLOCK_READ, BLOCK_WRITE CMD is being executed.

CMD cannot be executed by Status Register value other than above. It is also necessary to set above Parameter Register value correctly (by WRITE_REG TPC) before executing CMD by SET_CMD TPC.

If CMD is executed by SET_CMD TPC without meeting above conditions, Memory Stick will set the CMDNK bit of INT Register to 1 and INT will be generated.

Status Register value when INT is generating by CMDNK is as in the list below.

Table 6.5.2 Status Registers with CoMmanD NacK INT

	D7	D6	D5	D4	D3	D2	D1	D0
INT Register	X CED	X ERR	X BREQ	X –	X –	X –	X –	1 CMDNK
StatusRegister0	X MB	X FB0	X BE	X BF	X –	X –	X SL	X WP
StatusRegister1	X MB	X FB1	X DTER	X UCDT	X EXER	X UCEX	X FGER	X UCFG

6.6. Command Control

The control procedure, which needed to access to Memory Stick by CMD from Host, is given below.

6.6.1. Attribute of Memory Stick

Attribute needed for physical access to Memory Stick is as below.

- Total number of blocks (Total capacity of Memory Stick)
- Block size (Number of pages per block) : 16 or 32 pages
- Page size : 512Byte fixed value
- ExtraDataArea size : 9Byte fixed value

These attributes can be obtained from the Boot Block information. (See Section 7, Physical Format.)

6.6.2. Basic Flow

The basic flow of Memory Stick access viewed from Serial Protocol is as follows:

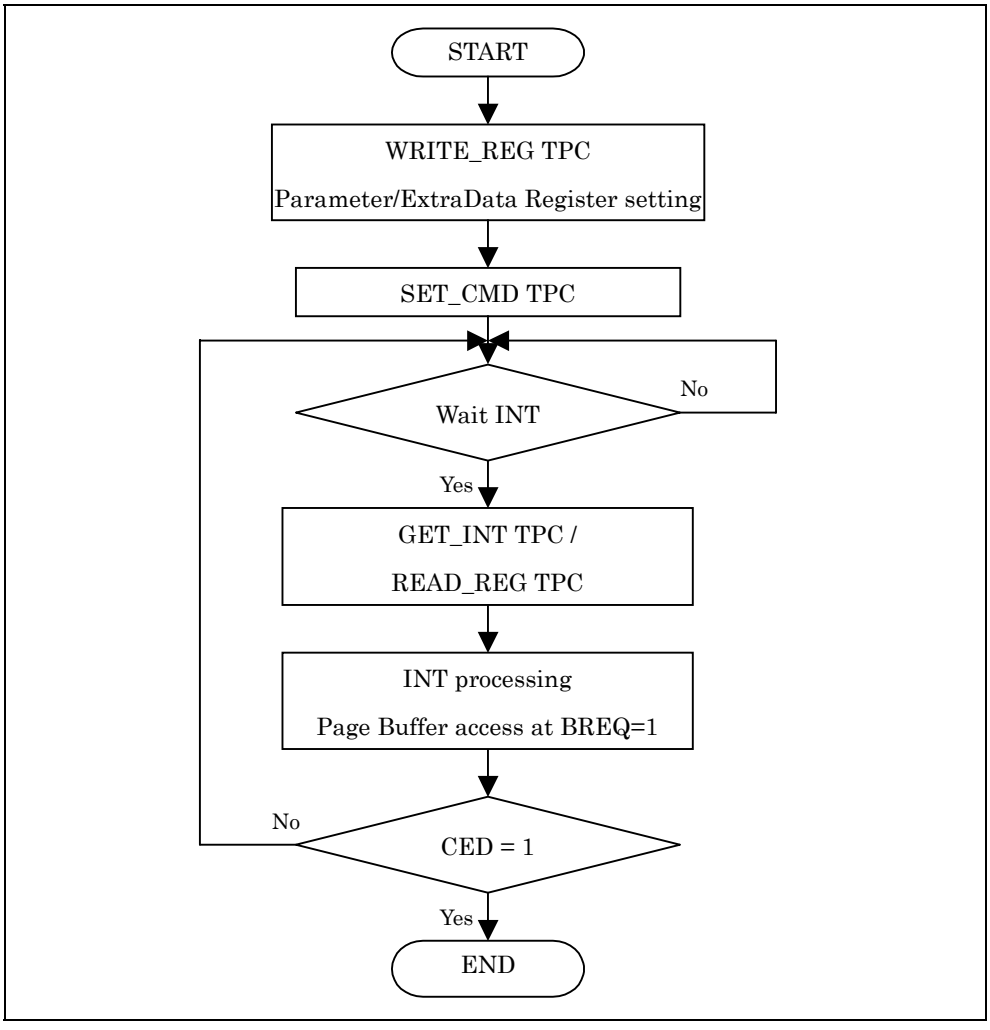

Fig 6.6.1 Flowchart for General Access Control

6.6.3. PageAccess

6.6.3.1. ReadPage

To read DataArea and ExtraDataArea of single page.

6.6.3.1.1. ParameterRegister Setting

SystemParameterReg	=0b10XXXXX0
BlockAdrReg2	=BlockAdr[23:16]
BlockAdrReg1	=BlockAdr[15:8]
BlockAdrReg0	=BlockAdr[7:0]
CommandParamReg	=0b001XXXXX
PageAdrReg	=PageNo.

6.6.3.1.2. Status Register Value

INT Register	: SET_CMD TPC end result(CED,ERR,CMDNK) and data transfer request (BREQ)
StatusRegister1	: Error content of BLOCK_READ (Only when INT Register ERR = 1)
OverwriteFlag	: Physical information flag (ExtraDataArea)
ManagementFlag	: Logical information flag (ExtraDataArea)
LogicalAddress1~0	: Logical address (ExtraDataArea)
ReserveArea4~0	: Reserved area (ExtraDataArea)

6.6.3.1.3. TPC Flow

Table 6.6.1 TPC Flow Model for ReadPage

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Parameter Register value setting
③	SET_CMD[BLOCK_READ]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT(READ_REG) Note	Read INT Register and clear INT CMDNK=1→Error end CED=1→End CED&ERR&BREQ=1→FlashReadError →⑥ CED&BREQ=1→⑦
⑥	READ_REG(Note)	Read StatusRegister 1 Verify FlashReadError content UCDT/UCEX/UCFG=1→Uncorrectable error→Error end Others→Correctable→⑦
⑦	Read_REG(Note)	Read ExtraDataRegister
⑧	READ_PAGE_DATA	Read PageData from PageBuffer
⑨	End	

Note) If READ_REG TPC is used instead of GET_INT TPC, Status Registers and ExtraDataRegisters are read simultaneously. In such case, READ_REG TPC in ⑥ and ⑦ are not necessary.

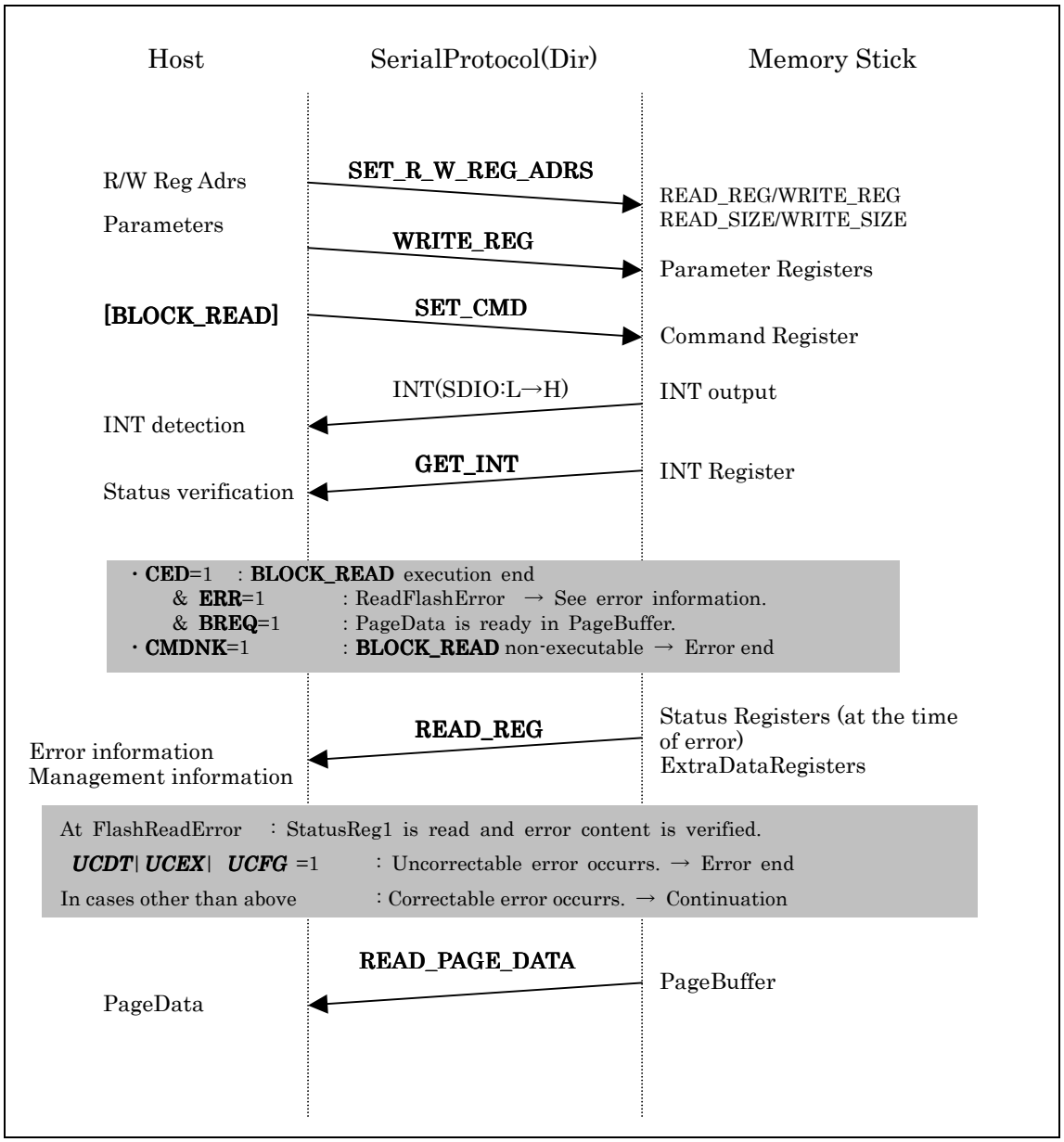

Fig. 6.6.2 Procedure for ReadPage Control

6.6.3.2. WritePage

To write DataArea and ExtraDataArea of single page.

6.6.3.2.1. ParameterRegister Setting

SystemParameterReg =0b10XXXXX0
BlockAdrReg2 =BlockAdr[23:16]
BlockAdrReg1 =BlockAdr[15:8]
BlockAdrReg0 =BlockAdr[7:0]
CommandParamReg =0b001XXXXX
PageAdrReg =PageNo.
OverwriteFlag =Physical information flag (ExtraDataArea)
ManagementFlag =Logical information flag (ExtraDataArea)
LogicalAddress1~0 =Logical address (ExtraDataArea)
ReserveArea4~0 =Reserved area (ExtraDataArea)

6.6.3.2.2. Status Register Value

INT Register : SET_CMD end result(CED,ERR,CMDNK) and data transfer request (BREQ)

6.6.3.2.3. TPC Flow (Pattern 1)

Table 6.6.2 TPC Flow Model-1 for WritePage

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set ParameterRegister and ExtraDataRegisters
③	WRITE_PAGE_DATA	Write PageData to PageBuffer
④	SET_CMD[BLOCK_WRITE]	Command transmission
⑤	Waiting for INT	INT detection→⑥
⑥	GET_INT	Read INT Register CMDNK=1→Error end CED=1→ End CED&ERR=1→FlashWriteError →Error end
⑦	End	

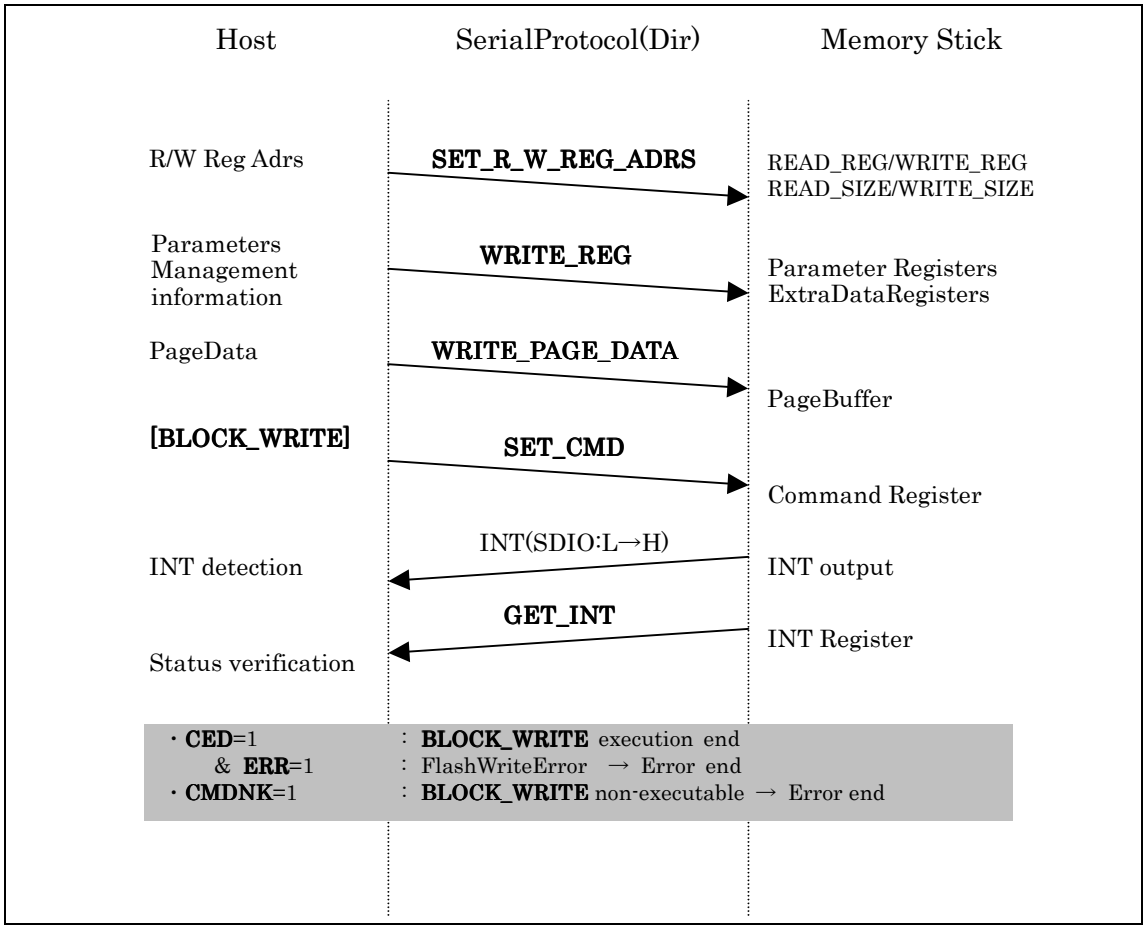

Fig. 6.6.3 Procedure for WritePage Control-1

6.6.3.2.4. TPC Flow (Pattern 2)

Table 6.6.3 TPC Flow Model-2 for WritePage

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register and ExtraDataRegisters
③	SET_CMD[BLOCK_WRITE]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT	Read INT Register CMDNK=1→Error end BREQ=1→⑥
⑥	WRITE_PAGE_DATA	Write PageData to PageBuffer
⑦	Waiting for INT	INT detection→⑧
⑧	GET_INT	Read INT Register. CED=1→ End CED&ERR= FlashWriteError →Error end
⑨	End	

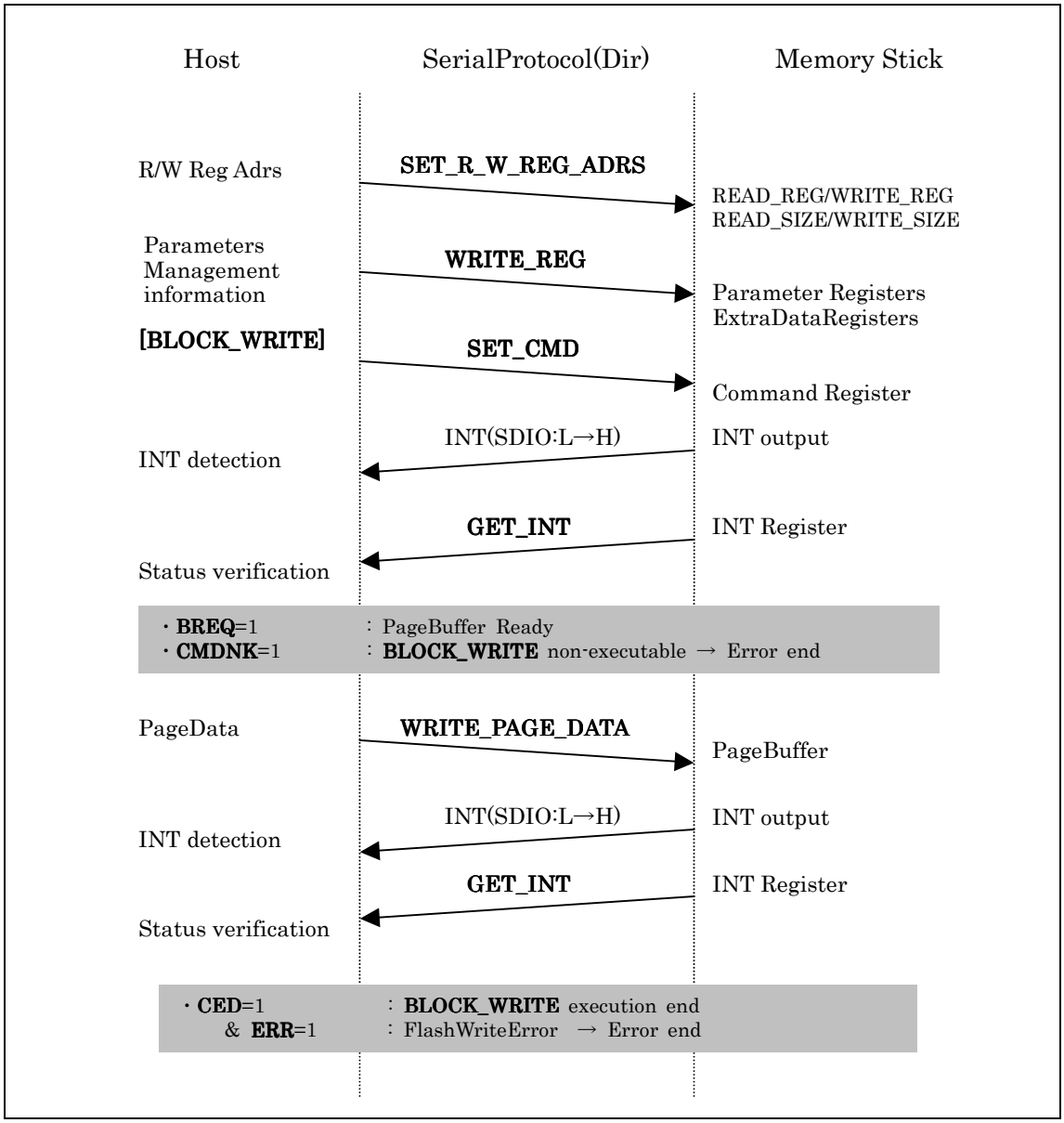

Fig. 6.6.4 Procedure for WritePage Control-2

6.6.3.3. CopyPage

DataArea and ExtraDataArea of single page are copied to the other page by the BLOCK_WRITE command without execution of READ_PAGE_DATA command after the BLOCK_READ command.

6.6.3.3.1. Parameter Register Setting

6.6.3.3.1.1. When Read (Page to copy from)

SystemParameterReg	=0b10XXXXX0
BlockAdrReg2	= BlockAdr [23:16] to copy from
BlockAdrReg1	= BlockAdr [15:8] to copy from
BlockAdrReg0	= BlockAdr [7:0] to copy from
CommandParamReg	= 0b001XXXXX
PageAdrReg	= Page No. to copy from
Page Buffer status	= Empty

6.6.3.3.1.2. When Write (Page to copy to)

SystemParameterReg	= 0b10XXXXXX0
BlockAdrReg2	= BlockAdr [23:16] to copy to
BlockAdrReg1	= BlockAdr [15:8] to copy to
BlockAdrReg0	= BlockAdr [7:0] to copy to
CommandParamReg	= 0b001XXXXX
PageAdrReg	= Page No. to copy to
OverwriteFlag	= Physical information flag (ExtraDataArea)
ManagementFlag	= Logical information flag (ExtraDataArea)
LogicalAddress1~0	= Logical address (ExtraDataArea)
ReserveArea4~0	= Reserved area (ExtraDataArea)

6.6.3.3.1.3. Status Register Value

INT Register	: SET_CMD end result (CED,ERR,CMDNK) and data transfer request (BREQ)
Status Register1	: BLOCK_READ error contents (only when INT Register ERR=1)
OverwriteFlag	= Physical information flag (ExtraDataArea)
ManagementFlag	= Logical information flag (ExtraDataArea)
LogicalAddress1~0	= Logical address (ExtraDataArea)
ReserveArea4~0	= Reserved area (ExtraDataArea)

6.6.3.3.2. TPC Flow

Table 6.6.4 TPC Flow Model for CopyPage

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set ParameterRegister for page to copy from
③	SET_CMD[BLOCK_READ]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT(READ_REG)Note	Read INT Register CMDNK=1→Error end CED＆BREQ=1→⑥ CED＆ERR=1→FlashReadError →⑦
⑥	READ_REG(Note)	Read ExtraDataRegisters→⑩
⑦	READ_REG(Note)	Read ExtraDataRegisters Read StatusRegister1 UCDT|UCEX|UCFG=1→Error end Others→⑧
⑧	READ_PAGE_DATA	Read PageData corrected by PageBuffer
⑨	WRITE_PAGE_DATA	Write PageData to PageBuffer
⑩	WRITE_REG	Set ParameterRegister and ExtraDataRegisters for the page to be copied to
⑪	SET_CMD[BLOCK_WRITE]	Command transmission
⑫	Waiting for INT	INT detection→⑬
⑬	GET_INT	Read INT Register CMDNK=1→Error end CED=1→ End CED&ERR=1→FlashWriteError →Error end
⑭	End	

Note) If READ_REG TPC is used instead of GET_INT TPC, Status Registers and ExtraDataRegisters are read simultaneously. In such case, READ_REG TPC in ⑥ and ⑦ are not necessary.

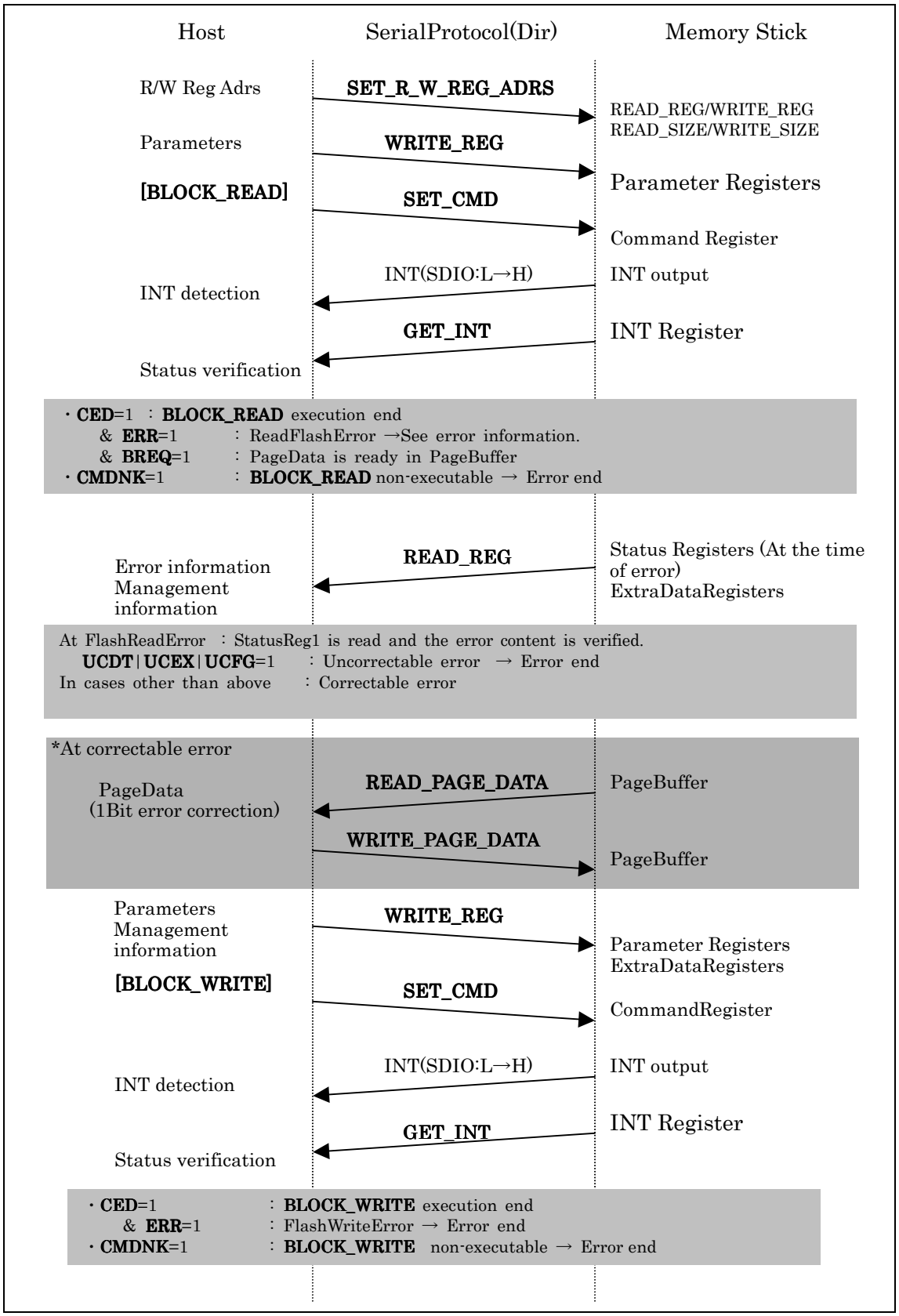

Fig. 6.6.5 Procedure for CopyPage Control

6.6.4. Block Access

6.6.4.1. Read Block

To read DataArea and ExtraDataArea of the pages in single block consecutively.

6.6.4.1.1. ParameterRegister Setting

SystemParameterReg	=0b10XXXXX0
BlockAdrReg2	=BlockAdr[23:16]
BlockAdrReg1	=BlockAdr[15:8]
BlockAdrReg0	=BlockAdr[7:0]
CommandParamReg	=0b000XXXXX
PageAdrReg	=Starting page no.
Number of pages to be read	=PageNum

6.6.4.1.2. StatusRegister Value

INT Register	: SET_CMD end results(CED,ERR,CMDNK) and data transfer request (BREQ)
Status Register1	: Error content of BLOCK_READ (Only when INT Register ERR=1)
OverwriteFlag	=Physical information flag (ExtraDataArea)
ManagementFlag	=Logical information flag (ExtraDataArea)
LogicalAddress1~0	=Logical address (ExtraDataArea)
ReserveArea4~0	=Reserved area (ExtraDataArea)

6.6.4.1.3. TPC Flow

Table 6.6.5 TPC Flow Model for ReadBlock

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register
③	SET_CMD[BLOCK_READ]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT(READ_REG)Note	Read INT Register CMDNK=1→Error end BREQ=1→⑥ ERR&BREQ=1→Correctable FlashReadError→⑥ CED＆BREQ=1→⑩ CED＆ERR&BREQ=1→FlashReadError →Error end
⑥	READ_REG(Note)	Read ExtraDataRegisters
⑦		(--PageNum>0) →⑨
⑧	SET_CMD[BLOCK_END]	Command transmission
⑨	READ_PAGE_DATA	Read PageData from PageBuffer→④
⑩	READ_REG	Read ExtraDataRegisters
⑪	READ_PAGE_DATA	Read Page Data from PageBuffer
⑫	End	

Note) If READ_REG TPC is used instead of GET_INT TPC, Status Registers and Extra DataRegisters are read simultaneously. In such case, READ_REG TPC in ⑥ is not necessary.

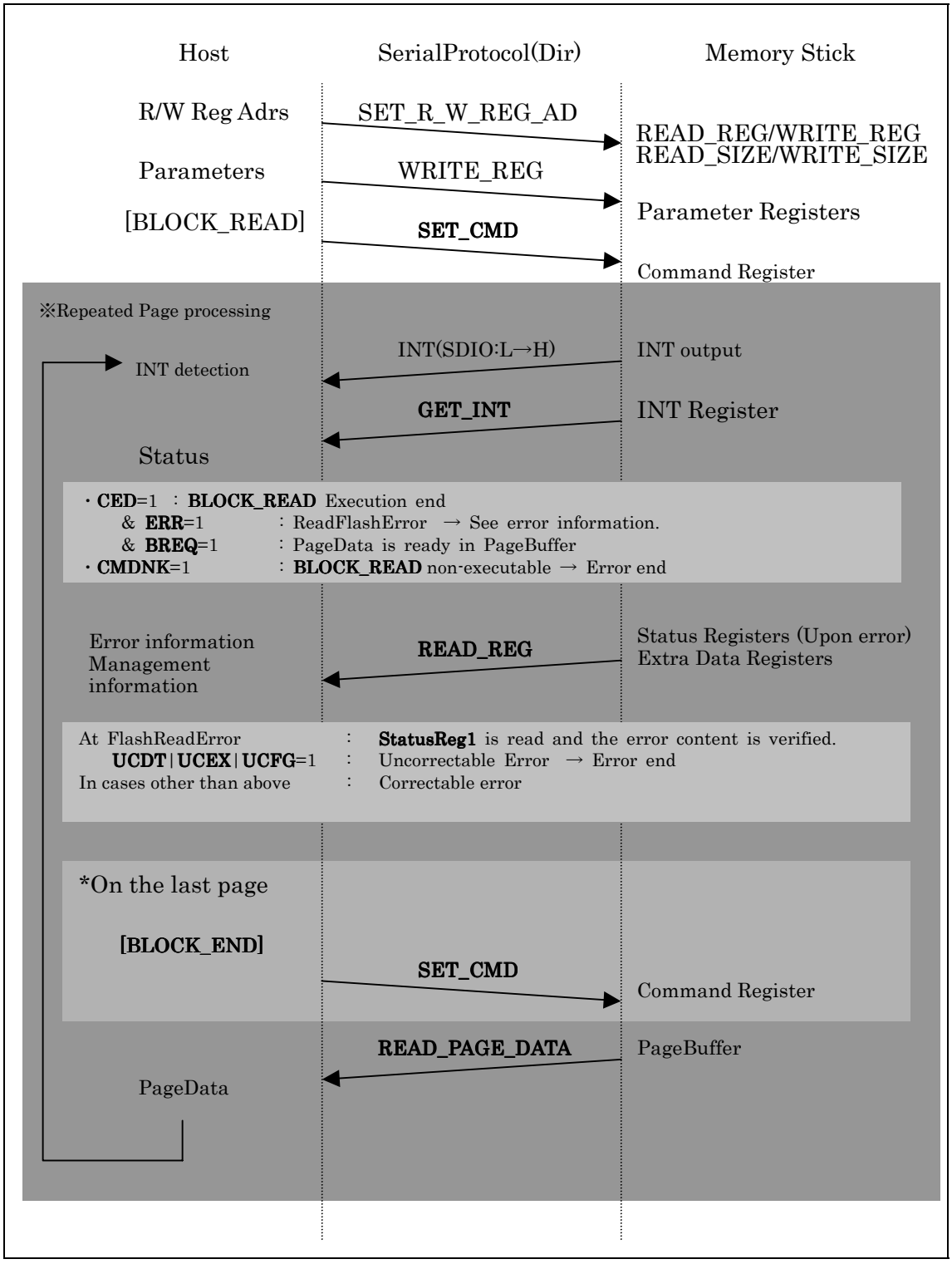

Fig. 6.6.6 Procedure for ReadBlock Control

6.6.4.2. WriteBlock

To write DataArea and ExtraDataArea of the pages in single block consecutively.

ExtraDataRegister value is common for all pages.

6.6.4.2.1. ParameterRegister Setting

SystemParameterReg	=0b10XXXXX0
BlockAdrReg2	=BlockAdr[23:16]
BlockAdrReg1	=BlockAdr[15:8]
BlockAdrReg0	=BlockAdr[7:0]
CommandParamReg	=0b000XXXXX
PageAdrReg	=Starting page no.
OverwriteFlag	=Physical information flag(ExtraDataArea)
ManagementFlag	=Logical information flag (ExtraDataArea)
LogicalAddress1~0	=Logical address (ExtraDataArea)
ReserveArea4~0	=Reserved area (ExtraDataArea)
Number of pages to be written	=PageNum

6.6.4.2.2. Status Register Value

INT Register	: SET_CMD end result (CED,ERR,CMDNK)
	Data transfer request (BREQ)

6.6.4.2.3. TPC Flow

Table 6.6.6 TPC Flow for WriteBlock

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set ParameterRegister and ExtraDataRegisters
③	SET_CMD[BLOCK_WRITE]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT	Read INT Register CMDNK=1→Error end BREQ=1→⑥ CED=1→End CED＆ERR=1→FlashWriteError →Error end
⑥		(PageNum-->0) →⑧
⑦	SET_CMD[BLOCK_END]	Command transmission→④
⑧	WRITE_PAGE_DATA	Write PageData to PageBuffer→④
⑨	End	

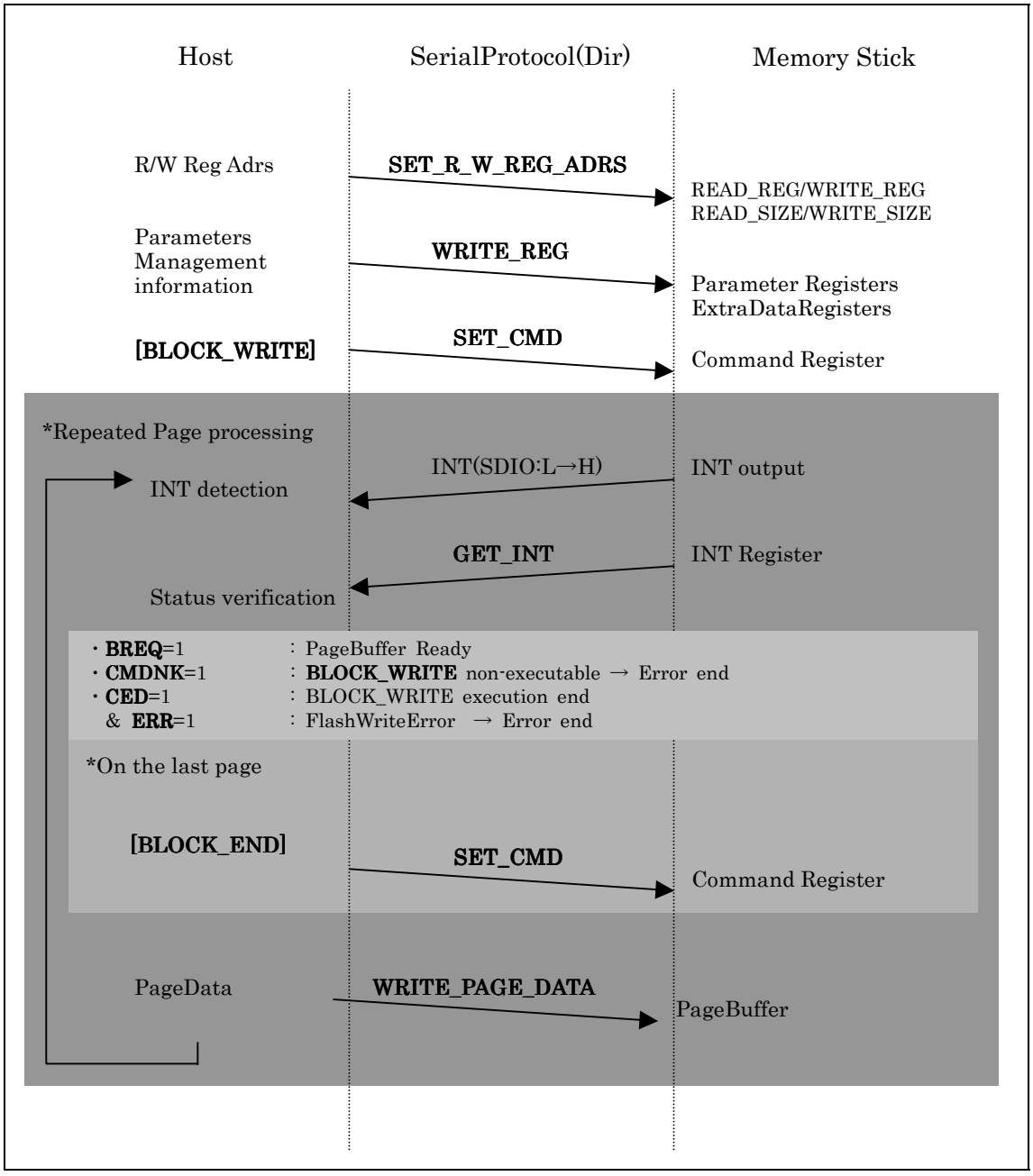

Fig. 6.6.7 Procedure for WriteBlock Control

6.6.4.3. EraseBlock

To erase all pages in single block.

6.6.4.3.1. Parameter Register Setting

SystemParameterReg =0b10XXXXX0
BlockAdrReg2 =BlockAdr[23:16]
BlockAdrReg1 =BlockAdr[15:8]
BlockAdrReg0 =BlockAdr[7:0]
CommandParamReg =0xXX
PageAdrReg =0xXX

6.6.4.3.2. Status Register Value

INT Register : SET_CMD end result (CED,ERR,CMDNK)

6.6.4.3.3. TPC Flow

Table 6.6.7 TPC Flow Model for WriteBlock

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register
③	SET_CMD[BLOCK_ERASE]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT	Read INT Register CMDNK=1→Error end CED=1→End CED＆ERR=1→FlashWriteError →Error end
⑥	End	

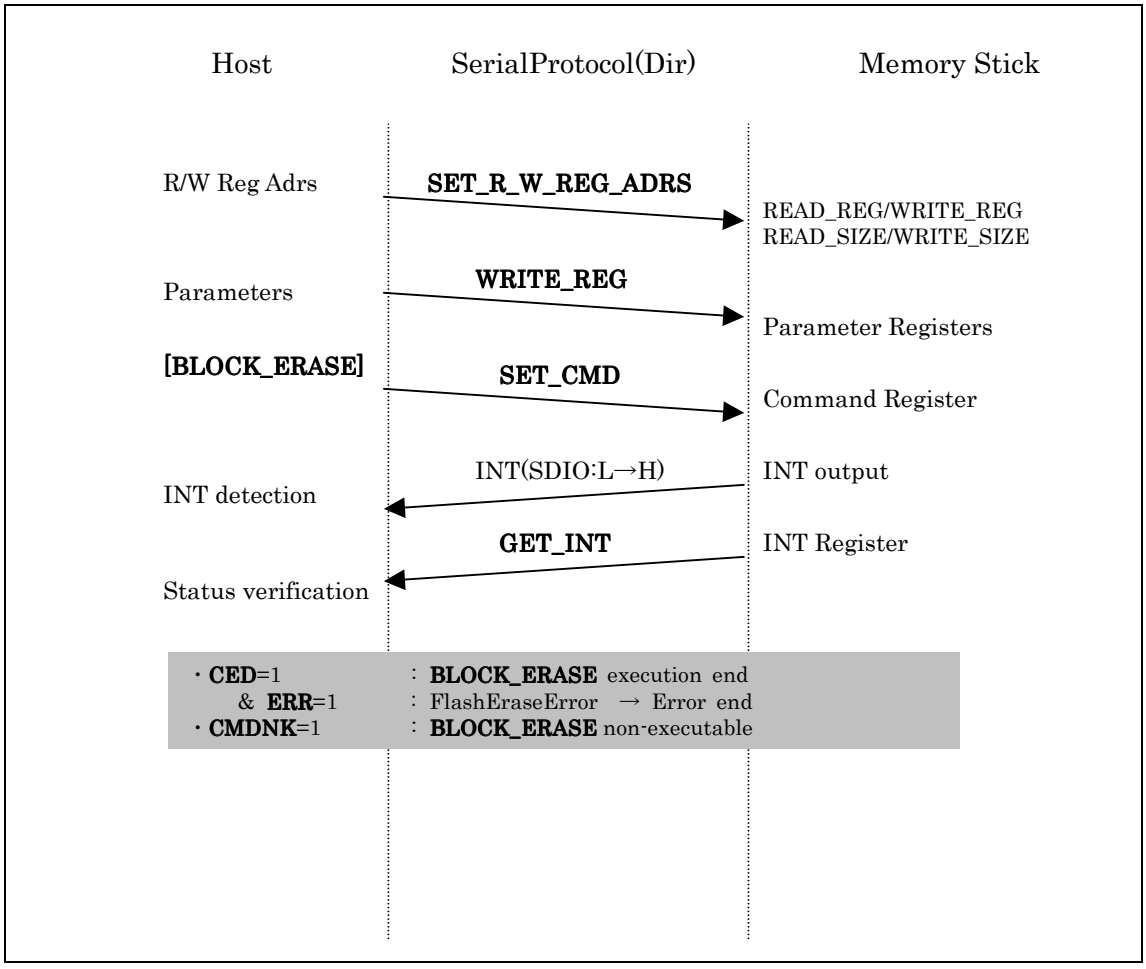

Fig. 6.6.8 Procedure for EraseBlock Control

6.6.5. ExtraDataAccess

6.6.5.1. ReadExtraData

To read ExtraDataArea of single page.

6.6.5.1.1. Parameter Register Setting

SystemParameterReg = 0b10XXXXX0
BlockAdrReg2 = BlockAdr[23:16]
BlockAdrReg1 = BlockAdr[15:8]
BlockAdrReg0 = BlockAdr[7:0]
CommandParamReg = 0b01XXXXXX
PageAdrReg = PageNo.

6.6.5.1.2. Status Register Value

INT Register : SET_CMD end result (CED,ERR,CMDNK)
Status Register1 : BLOCK_READ error content (only when INT Register ERR=1)
OverwriteFlag = Physical information flag (ExtraDataArea)
ManagementFlag = Logical information flag (ExtraDataArea)
LogicalAddress1~0 = Logical address (ExtraDataArea)
ReserveArea4~0 = Reserved area (ExtraDataArea)

6.6.5.1.3. TPC Flow

Table 6.6.8 TPC Flow Model for ReadExtraData

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register
③	SET_CMD[BLOCK_READ]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT(READ_REG)Note	Read INT Register CMDNK=1→Error end CED=1→⑦ CED＆ERR=1→FlashReadError →⑥
⑥	READ_REG(Note)	Read StatusReg1 UCEX|UCFG=1→Uncorrectable FlashReadError→Error end
⑦	Read_REG(Note)	Read ExtraDataRegisters
⑧	End	

Note) If READ_REG TPC is used instead of GET_INT TPC, StatusRegisters and ExtraDataRegisters are read simultaneously. In such case, READ_REG TPC in ⑥ and ⑦ are not necessary.

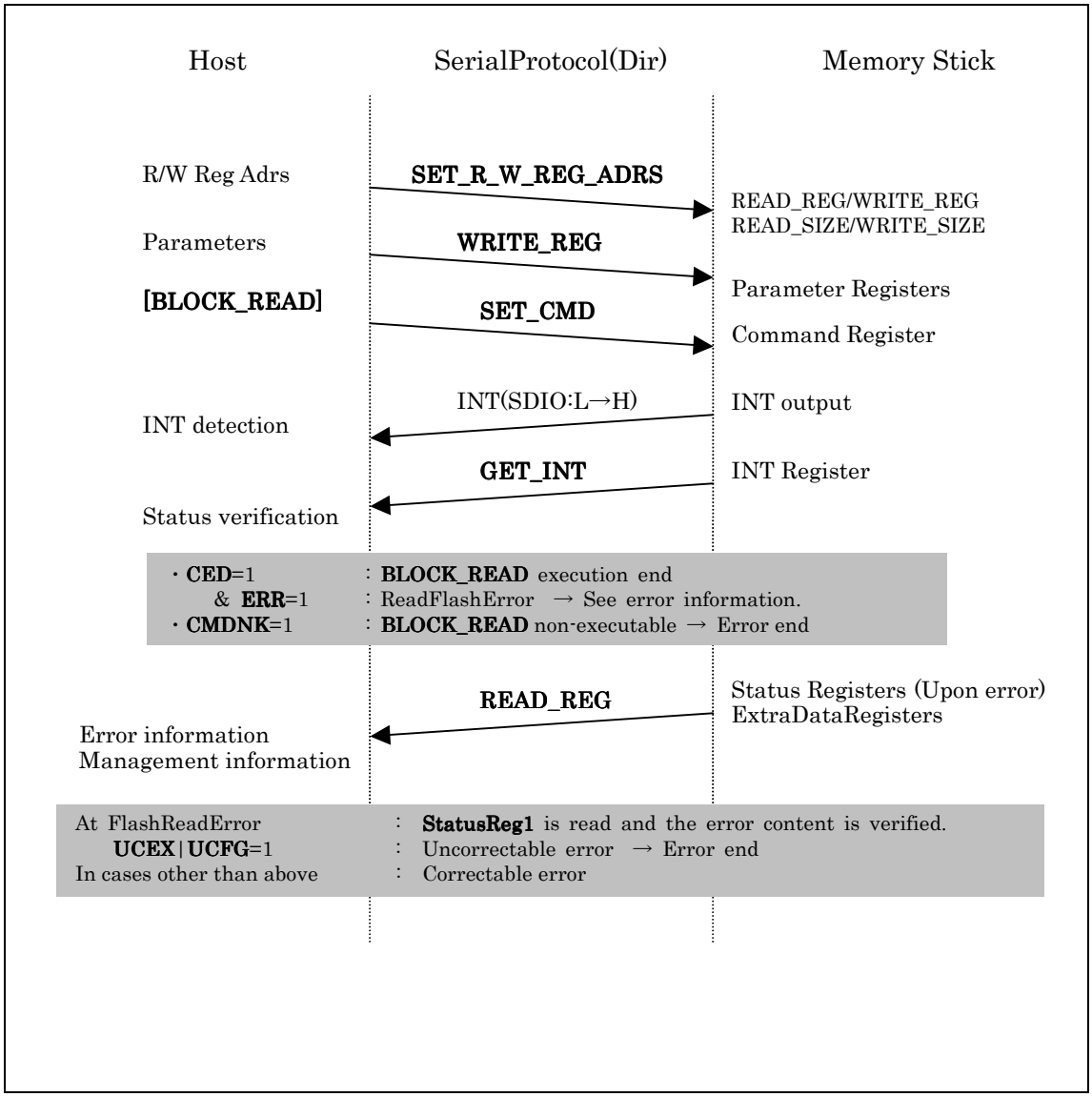

Fig. 6.6.9 Procedure for ReadExtraData Control

6.6.5.2. WriteExtraData

To write ExtraDataArea of single page.

6.6.5.2.1. Parameter Register Value

SystemParameterReg = 0b10XXXXX0
BlockAdrReg2 = BlockAdr[23:16]
BlockAdrReg1 = BlockAdr[15:8]
BlockAdrReg0 = BlockAdr[7:0]
CommandParamReg = 0b01XXXXXX
PageAdrReg = PageNo.
OverwriteFlag = Physical information flag (ExtraDataArea)
ManagementFlag = Logical information flag (ExtraDataArea)
LogicalAddress1~0 = Logical address (ExtraDataArea)
ReserveArea4~0 = Reserved area (ExtraDataArea)

6.6.5.2.2. Status Register value

INT Register : SET_CMD end result (CED,ERR,CMDNK)

6.6.5.2.3. TPC Flow

Table 6.6.9 TPC Flow Model for WriteExtraData

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register and ExtraDataRegisters
③	SET_CMD[BLOCK_WRITE]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT	Read INT Register CMDNK=1→Error end CED=1→ End CED＆ERR=1→FlashWriteError →Error end
⑥	End	

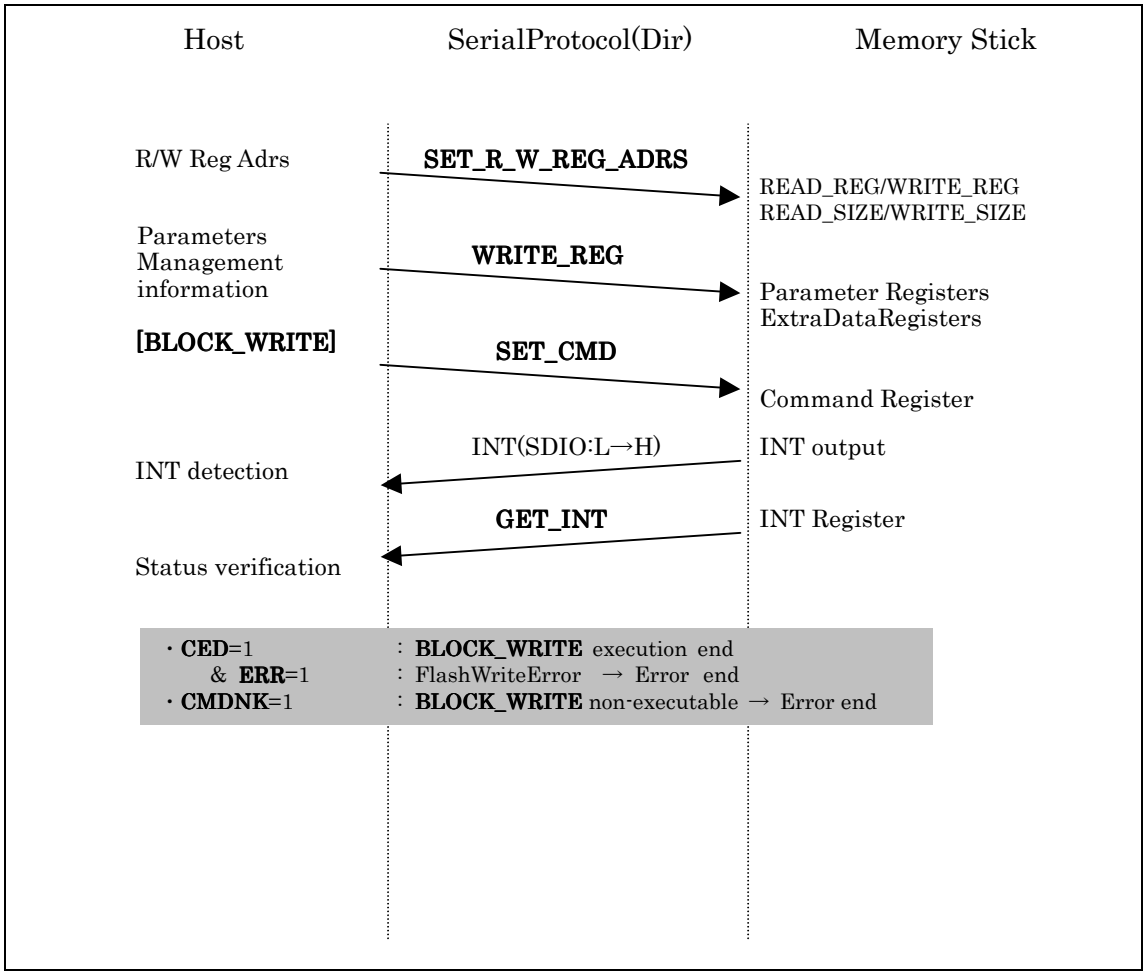

Fig. 6.6.10 Procedure for WriteExtraData Control

6.6.5.3. OverwriteExtraData

OverwriteFlag for ExtraDataArea of single page is overwritten with MaskData.

6.6.5.3.1. ParameterRegister Setting

SystemParameterReg = 0b10XXXXX0
BlockAdrReg2 = BlockAdr[23:16]
BlockAdrReg1 = BlockAdr[15:8]
BlockAdrReg0 = BlockAdr[7:0]
CommandParamReg = 0b100XXXXX
PageAdrReg = PageNo.
OverwriteFlag = MaskData (Set 0 only to the bit to overwrite from 1 to 0.) (See Note 2 under Table 5.1.1)

6.6.5.3.2. StatusRegister Value

INT Register : SET_CMD end result (CED,ERR,CMDNK)

6.6.5.3.3. TPC Flow

Table 6.6.10 TPC Flow Model for OverwriteExtraData

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register and OverwriteFlag
③	SET_CMD[BLOCK_WRITE]	Command transmission
④	Waiting for INT	INT detection→⑤
⑤	GET_INT	Read INT Register CMDNK=1→Error end CED=1→ End CED&ERR=1→FlashWriteError →Error end
⑥	End	

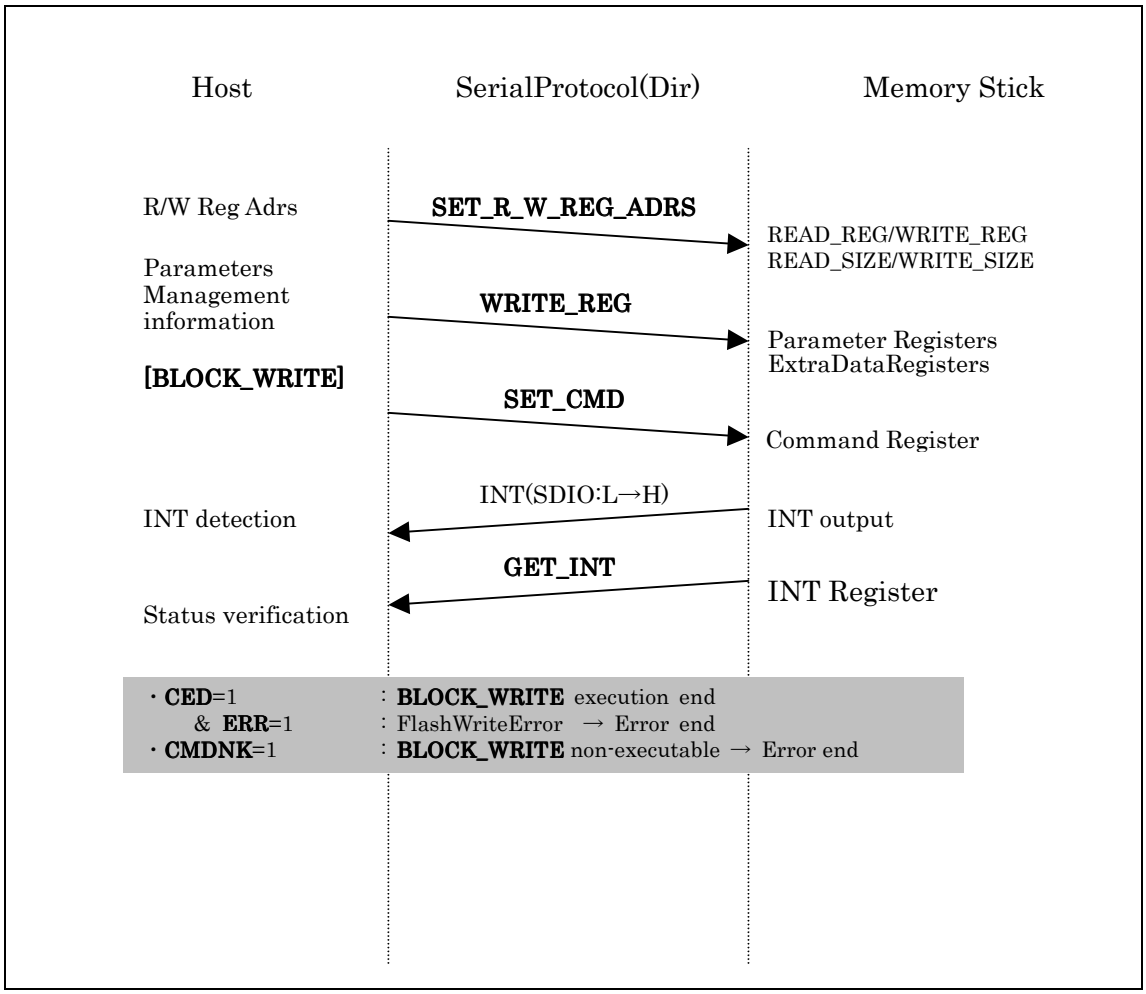

Fig. 6.6.11 Procedure for OverwriteExtraData Control

6.6.5.4. Sleep

Flash Memory Controller on Memory Stick is put to sleep state.

It is recovered by normal execution of WRITE_REG or SET_CMD protocol. In this case, busy period in BS3 lasts 1ms.

6.6.5.4.1. Parameter Register Setting

None

6.6.5.4.2. Status Register Value

INT Register : SET_CMD end result (CED, CMDNK)

6.6.5.4.3. TPC Flow

Table 6.6.11 TPC Flow Model for Sleep

No	TPC	Description
①	SET_CMD[BLOCK_SLEEP]	Command transmission
②	Waiting for INT	INT detection→③
③	GET_INT	Read INT Register CMDNK=1→Error End CED=1→ End
④	End	

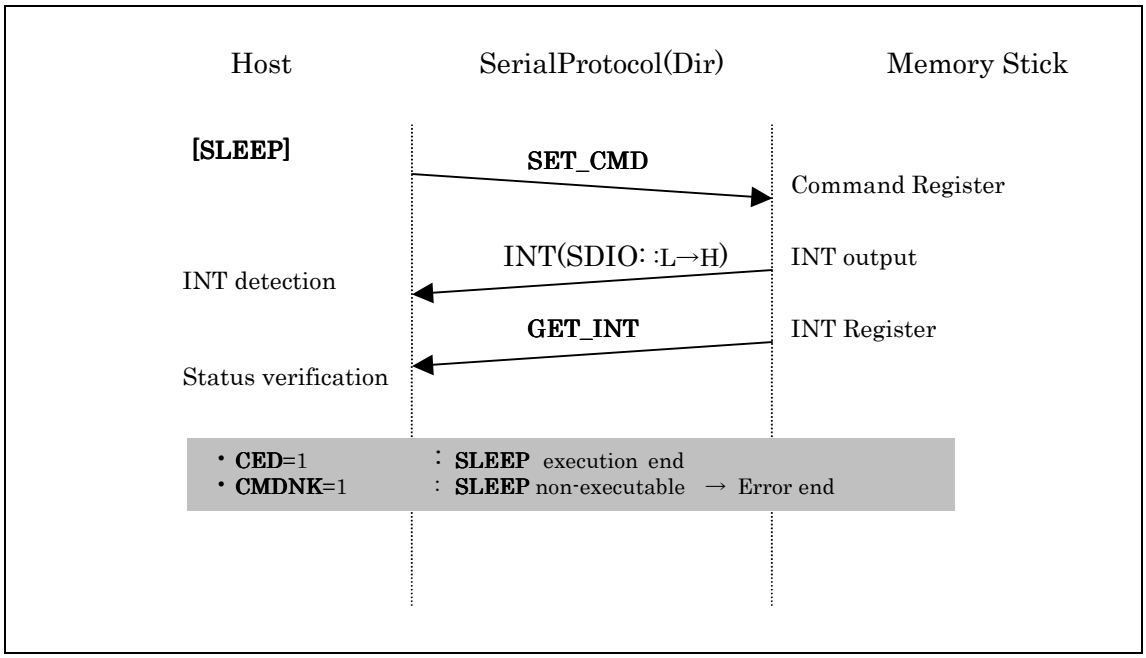

Fig. 6.6.12 Procedure for Sleep Control

6.6.5.5. ClearBuffer

PageBuffer in Memory Stick is cleared.

6.6.5.5.1. Parameter Register Setting

None

6.6.5.5.2. StatusRegister Value

INT Register : SET_CMD end result (CED, CMDNK)

6.6.5.5.3. TPC Flow

Table 6.6.12 TPC Flow Model for ClearBuffer

No	TPC	Description
①	SET_CMD[CLEAR_BUF]	Command transmission
②	Waiting for INT	INT detection→③
③	GET_INT	Read INT Register CMDNK=1→Error End CED=1→ End
④	End	

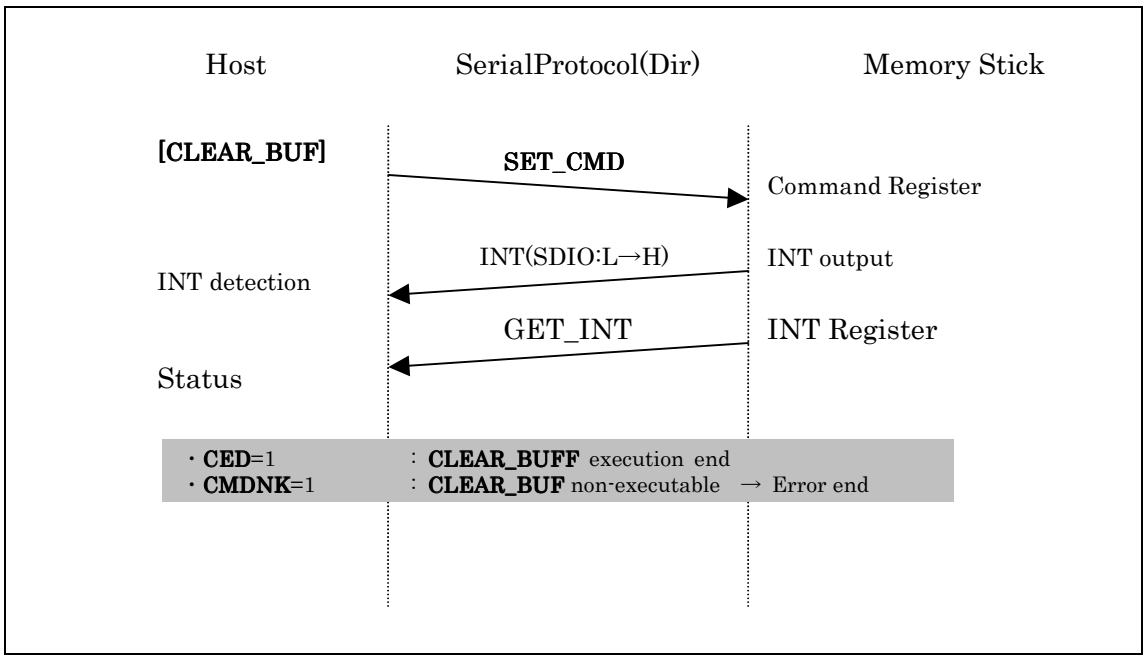

Fig. 6.6.13 Procedure for ClearBuffer Control

6.6.5.6. FlashStop

FlashBusy in Memory Stick is forced to end.

All data in block which is being accessed at that time, become indefinite value.

6.6.5.6.1. Parameter Register Setting

None

6.6.5.6.2. Status Register Value

INT Register : SET_CMD end result (CED, CMDNK)

6.6.5.6.3. TPC Flow

Table 6.6.13 TPC Flow Model for FlashStop

No	TPC	Description
①	SET_CMD[FLASH_STOP]	Command transmission
②	Waiting for INT	INT detection→③
③	GET_INT	Read INT Register CMDNK=1→Error end CED=1→ End
④	End	

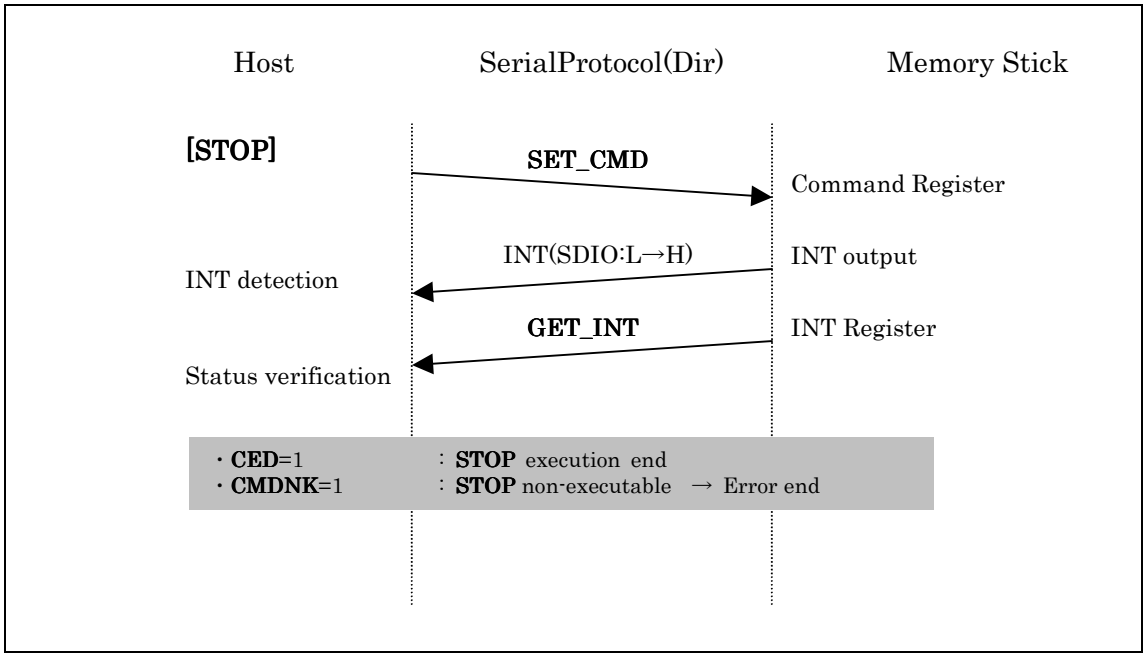

Fig. 6.6.14 Procedure for FlashStop Control

6.6.5.7. Reset

Controller on Memory Stick is reset.

When reset, PageBuffer becomes empty and all registers accessed by READ_REG, WRITE_REG shall return to the default settings.

6.6.5.7.1. Parameter Register Setting

None

6.6.5.7.2. Status Register Value

INT does not occur.

6.6.5.7.3. TPC Flow

Table 6.6.14 TPC Flow Model for Reset

No	TPC	Description
①	SET_CMD[RESET]	Command transmission
②	End	

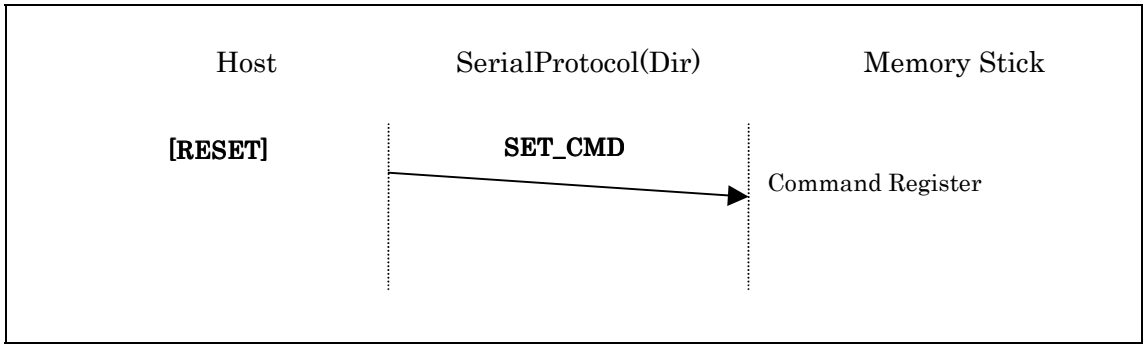

Fig. 6.6.15 Control Model for Reset

6.6.5.8. ReadAttribute

Attribute information is read from ATTRIB ROM on Memory Stick.

6.6.5.8.1. Parameter Register Setting

SystemParameterReg =0bX1XXXXX0
BlockAdrReg2 =0xXX
BlockAdrReg1 =0xXX
BlockAdrReg0 =0xXX
CommandParamReg =0xXX
PageAdrReg =0xXX

6.6.5.8.2. Status Register Value

None

6.6.5.8.3. TPC Flow

Table 6.6.15 TPC Flow Model for ReadAttribute

No	TPC	Description
①	SET_R_W_REG_ADRS	May be omitted if used with default value.
②	WRITE_REG	Set Parameter Register
③	READ_PAGE_DATA	Read ATTRIB ROM
④	End	

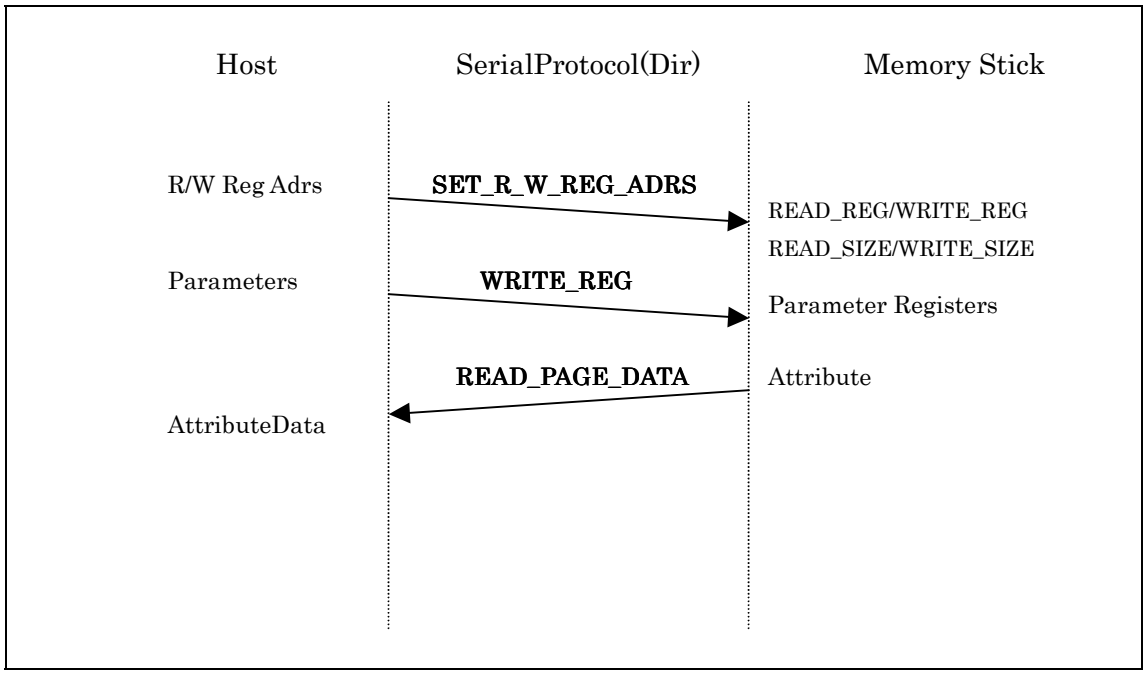

Fig. 6.6.16 Procedure for ReadAttribute Control

6.6.5.9. Retry

It is shown below, whether the protocol is capable of retry when timeout for Busy or CRC error occurs.

6.6.5.9.1. In ReadPacket

- READ_PAGE_DATA : Retry is possible except for CRC error.
- READ_REG : Retry is possible.
- GET_INT : Retry is possible.

6.6.5.9.2. In WritePacket

- WRITE_PAGE_DATA : Retry is possible.
- WRITE_REG : Retry is possible.
- SET_R/W_REG_ADRS : Retry is possible.
- SET_CMD : Retry is possible.