32-bit RISC Microcontroller

TXZ Family

Reference Manual
Serial Memory Interface
(SMIF-A)

Revision 1.1

2018-07

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document name
Clock Control and Operation Mode
Memory Map
Exception
Input/Output Ports
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1], [XYZ2], [XYZ3] → [XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C . . .
  Example: ***[ADACR0], [ADBCR0], [ADCCR0] → [ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 . . .
  Example: [***T32A0RUNA], [T32A1RUNA], [T32A2RUNA] → [T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

The flash memory uses the Super Flash® technology under license from Silicon Storage Technology, Inc.
Super Flash® is registered trademark of Silicon Storage Technology, Inc.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviation

Some of abbreviations used in this document are as follows:

SI	Serial Input
SO	Serial Output
SPI	Serial Peripheral Interface
SMIF	Serial Memory Interface

1. Outlines

The serial memory interface (SMIF) connects to the memory which has serial I/O's (SPI Flash memory and others).

The list of the functions of SMIF is shown in the following table.

Function category	Function	Description
Connection to a serial memory	Connection	- 2 serial memories at maximum can be connected.
	Memory capacity	- 64 KB to 16 MB
	Transfer clock	- 20 MHz at maximum
	Communication Mode	- SPI compatible SPI Mode 0 support Single I/O, Dual I/O read and Quad I/O read are supported. - MSB first
	Memory mapping	- Mapping is possible to addresses: 0xA0000000 to 0xA0FFFFFF.
	Access mode	- Direct access - Program register access
	Command transfer count	- 264 Bytes at maximum can be transferred through a register.
	Chip select	- Selection from Serial memory 0 and Serial memory 1. - Deassertion times of SMIxCS0_N and SMIxCS1_N can be set.
	Other functions	- Read after Write/Erase completion using the polling of the status of SPI Flash memory.

2. Configuration

The block diagram and the list of the signals of SMIIF are shown as follows:

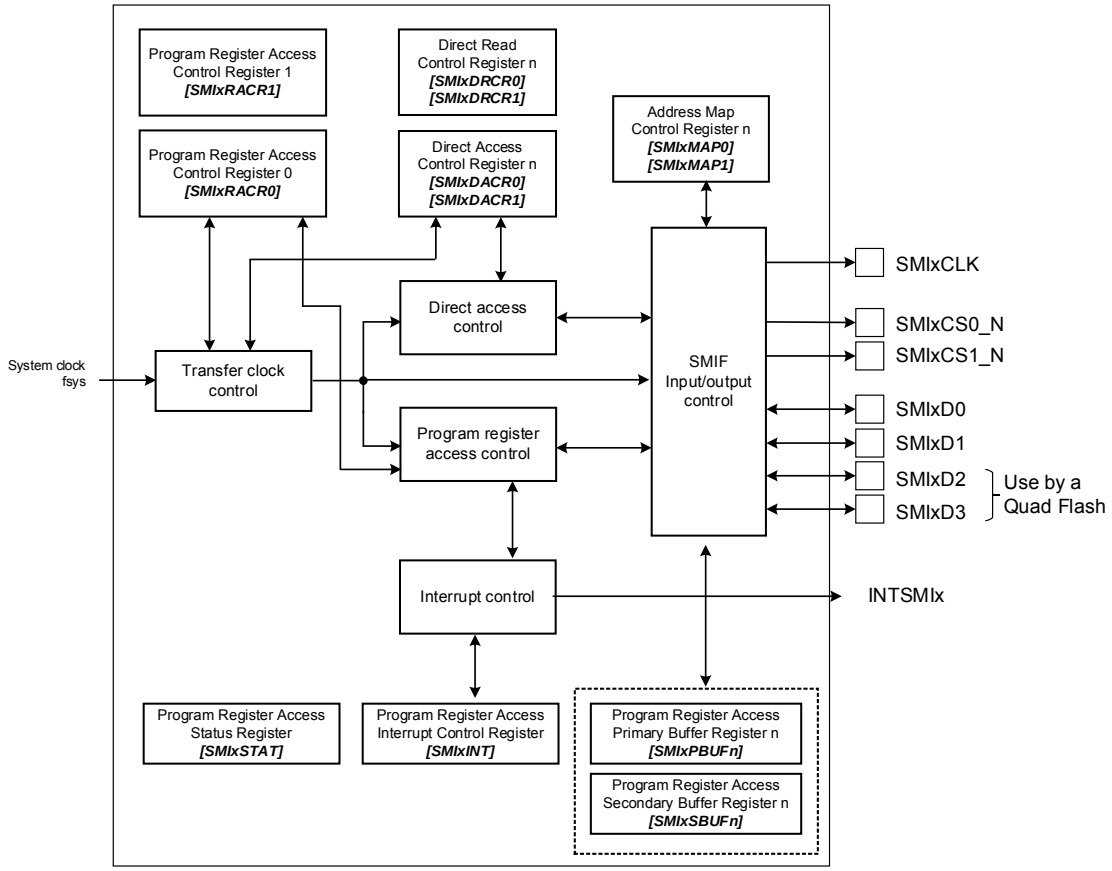

Figure 2.1 Block diagram of SMIF

Table 2.1 List of Signals

No	symbol	Signal name	I/O	Reference manual
1	f_{sys}	System clock	Input	Clock Control and Operation Mode
2	SMIxCLK	Access clock	Output	Product Information, Input/Output ports
3	SMIxCS0_N	Chip select 0	Output	Product Information, Input/Output ports
4	SMIxCS1_N	Chip select 1	Output	Product Information, Input/Output ports
5	SMIxD0	Data input and output	I/O	Product Information, Input/Output ports
6	SMIxD1	Data input and output	I/O	Product Information, Input/Output ports
7	SMIxD2	Data input and output	I/O	Product Information, Input/Output ports
8	SMIxD3	Data input and output	I/O	Product Information, Input/Output ports
9	INTSMIx	Interrupt	Output	Exception

3. Function and Operation

SMIF can connect to two serial memories at maximum. Each memory capacity of 64 KB to 16MB is available. The memory can be read by “Direct access” using addresses setting and also accessed by “Program register access” which issues a command using a program register.

3.1. Clock Supply

When SMIF is used, the corresponding clock enable bits should be set to “1” (Clock supply) in fsys supply stop register A (***[CGFSYSENA], [CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB], [CGFSYSMENB]***), and fc supply stop register (***[CGFCEN]***). The corresponding registers and the bit location depend on a product. Some products do not have all registers. For the details, refer to “Clock control and operation mode” in Reference manual.
When the clock supply is stopped or the status is transited to STOP1/STOP2 mode, it should be checked that SMIF stops.

3.2. Communication Mode

The communication format between SMIF and a serial memory is SPI-compatible. Dual I/O Read and Quad I/O Read are supported.

The serial memory should satisfy the following conditions.

- Capacity: 64 KB to 16 MB
- Fast Read is supported.
- SPI Mode 0 is supported.
- The unused upper addresses are “don’t care”.
- The bit 0 of the status register is the bit of Write In Progress (WIP).
- MSB first

When SPI is connected, the control is done in the following 4 phases.

1. Command (Output)
2. Address (Output)
3. Dummy Byte (Output)
4. Data (Input)

The I/O bit width in each phase should be set to ***[SMIxDRCRn]***.

3.3. Memory Mapping

The addresses to access a serial memory can be mapped to any region in “0xA0000000” to “0xA0FFFFFF” (16 MB). When write to this region, bus error does not occur. Also, no operation is performed on the serial memory. If the area which is not mapped is read, an indefinite value returns. After the reset assertion, the serial memory 0 is memory-mapped, but the serial memory 1 is not. If the serial memory whose capacity is smaller than the mapped region is implemented, a mirror region is read in the serial memory when the unimplemented region is accessed.

Figure 3.1 shows an example that the mapping region of the serial memory 0 is “0xA0000000” to “0xA0BFFFFF” (12 MB) and the mapping region of the serial memory 1 is “0xA0C00000” to “0xA0FFFFFF” (4 MB), and a 4-MB serial memory is connected to each region, respectively.

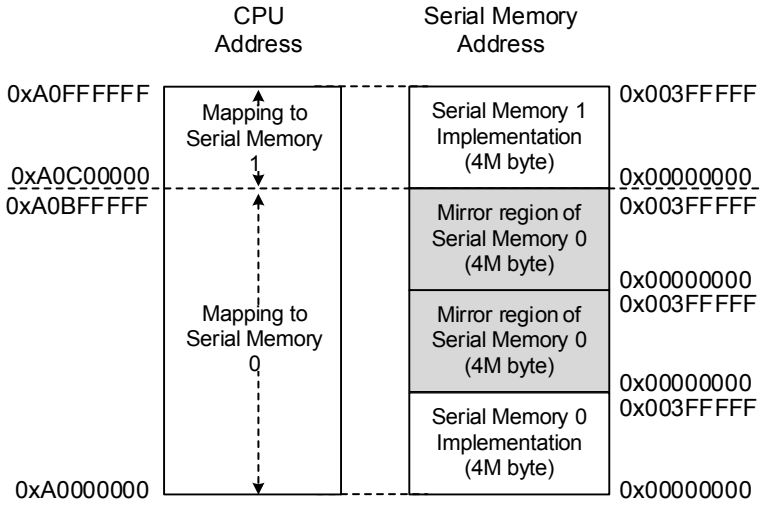

Figure 3.1 Example of memory mapping

3.4. Access Mode

There are two access modes to access a serial memory, “Direct access mode” which reads the memory using addresses settings and “Program register access mode” which issues a command using the control of a program register.

3.4.1. Direct Access Mode

The serial memory can be directly read by accessing addresses “0xA0000000” to “0xA0FFFFFF”. This is “Direct access”. When a read to an address in the range of “0xA0000000” to “0xA0FFFFFF” is detected, a read command is issued to SPI Flash memory.

3.4.1.1. SPI Flash Command

The initial command which is issued in the direct access mode is Fast Read (Op-code=0x0B). The command can be replaced to one of the following multi-I/O commands. But the supported commands depend on the connected SPI Flash memory.

- Fast Read Dual Output
- Fast Read Dual I/O
- Fast Read Quad Output
- Fast Read Quad I/O

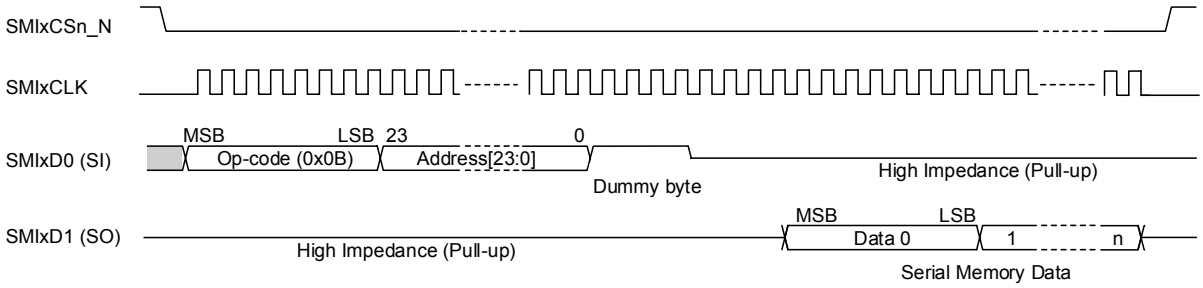

Figure 3.2 Fast Read sequence

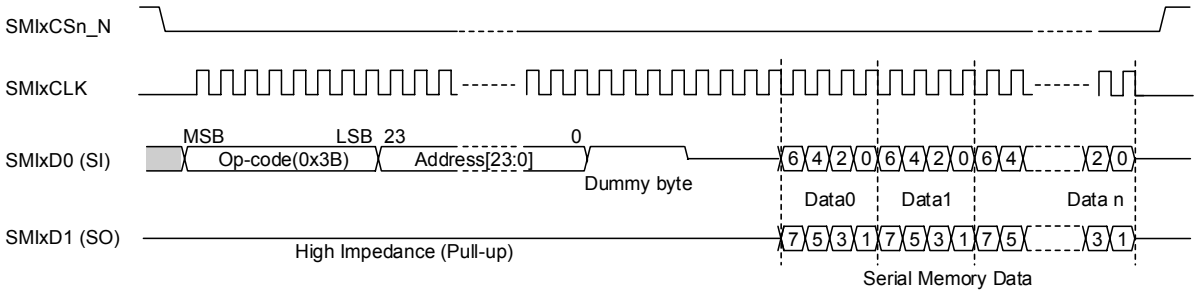

Figure 3.3 Fast Read dual output sequence

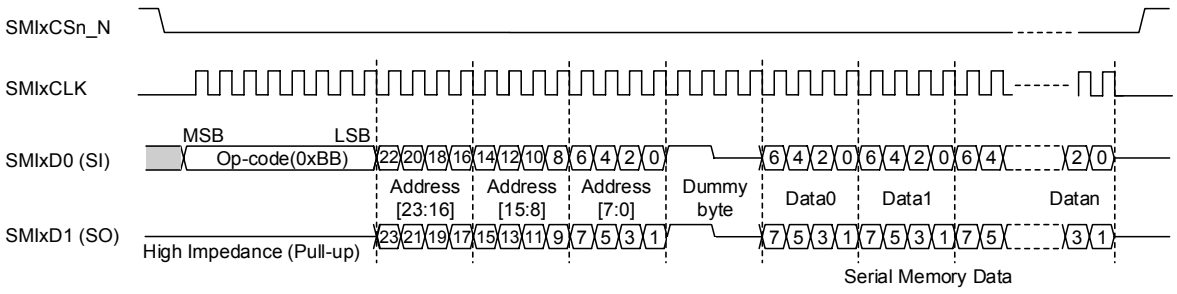

Figure 3.4 Fast Read Dual I/O sequence

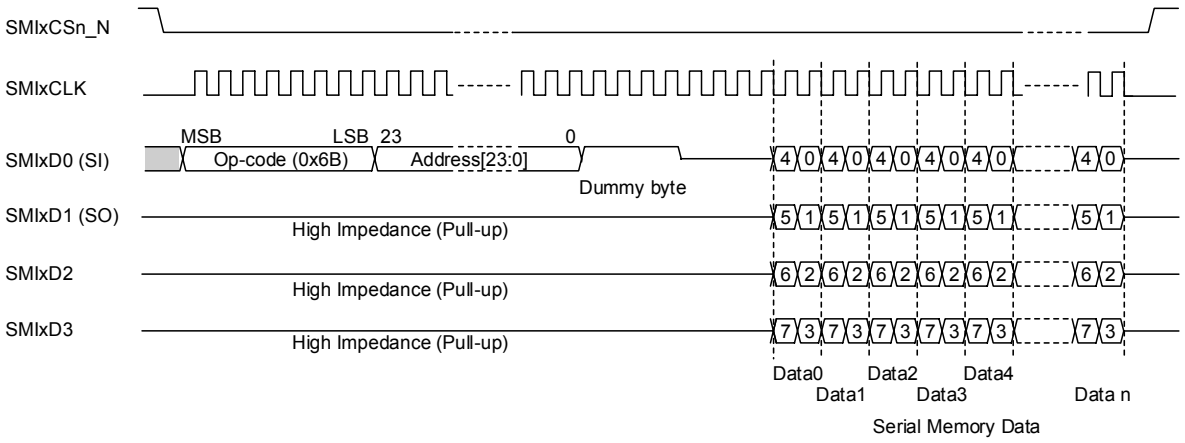

Figure 3.5 Fast Read quad output sequence

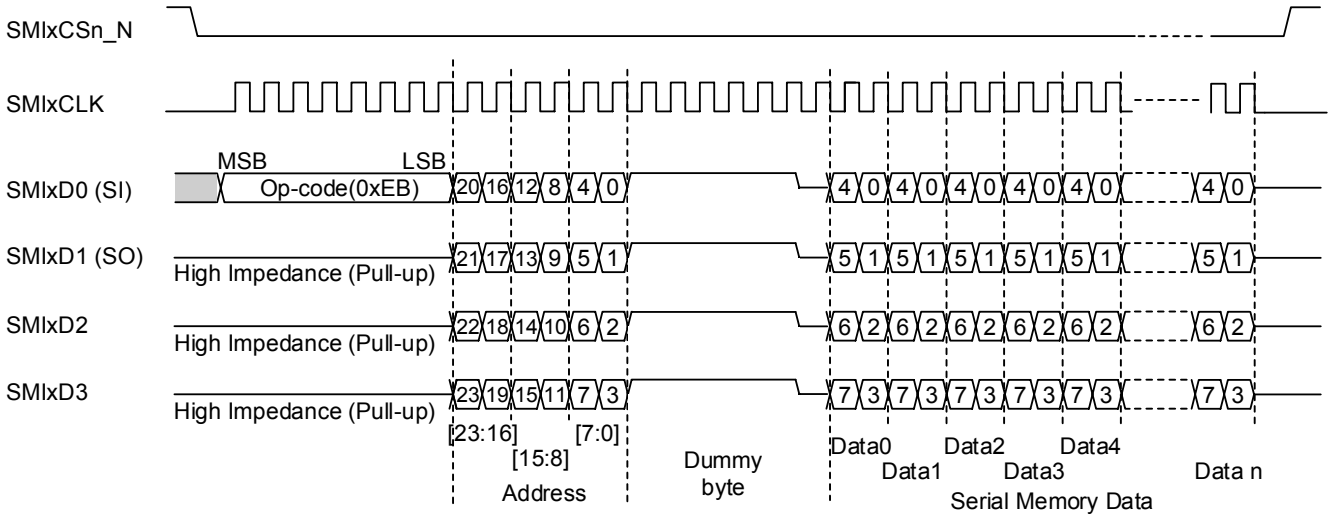

Figure 3.6 Fast Read quad I/O sequence

3.4.1.2. Polling of WIP bit

This function is mainly used for debugging.

Generally, SPI Flash memory cannot receive Read command during Write or Erase operation. So, when SMIF issues Read command to SPI Flash memory, the memory does not respond during Write or Erase operation. As a result, invalid data returns. Software should control this access not to be done.

This function enables to receive a correct data even when SPI Flash memory is read during Write or Erase operation. When a read is detected during enable of this function, SMIF polls WIP bit in Status register in SPI Flash memory until it becomes “0”. Then, the Read command is issued. This makes the correct data received. ***[SMIxDACRn]***<PollWIP> should be set to “1” to enable this function.

When the operation starts after the reset deassertion, WIP bit in Status register in SPI Flash memory should be polled to confirm that the state is neither Write nor Erase one. The operation is controlled by the value in ***[SMIxDACRn]***<PollWIP>.

When this function is enabled, the overhead of a read becomes bigger because of the polling per read. And, if the polling result does not return or a timeout error occurs, the operation of SMIF is not guaranteed.

3.4.1.3. Setting Procedure of Direct Access Mode

1. The base address and the capacity are set to ***[SMIxMAPn]*** according to the implemented serial memory.
2. The transfer clock, CS deassertion time, and WIP polling operation are set to ***[SMIxDACRn]***.
3. The command op-code, the dummy byte count, and the input/output control are set to ***[SMIxDRCRn]***.
4. Address in the mapping region is read in the serial memory.

Note: Immediately after the reset deassertion, the serial memory 0 is mapped to addresses “0xA0000000” to “0xA0FFFFFF” (16 MB space).

3.4.2. Program Register Access

The program register access issues such commands as Page Program, Erase, Full Chip Erase, Status Read, and Read to SPI Flash memory through registers. 264-byte commands at maximum can be issued to SPI Flash memory using 8-byte Primary buffer and 256-byte secondary buffer. The setting of the program register access mode is set to ***[SMIxRACR0]*** and ***[SMIxRACR1]*** for the serial memory 0 and the serial memory 1, respectively.

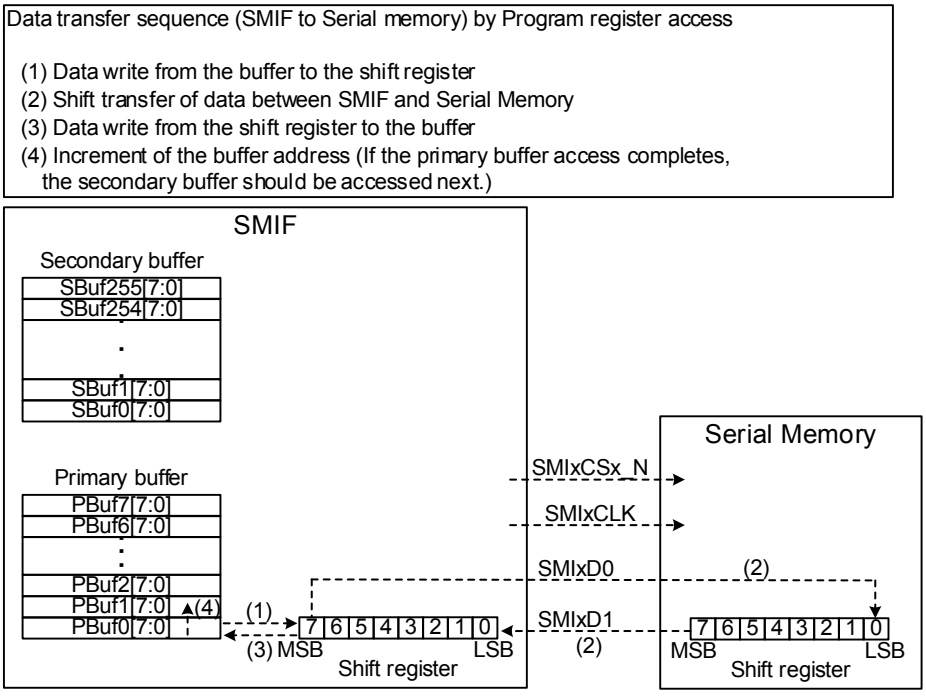

Figure 3.7 Program register access

3.4.2.1. Setting Procedure of Program Register Access Mode

1. No execution of the program register access should be checked by reading ***[SMIxSTAT]***<CycProg>.
2. The serial memory which should be accessed, the transfer byte count, and others should be set to ***[SMIxMAPn]***, ***[SMIxRACR0]***, ***[SMIxRACR1]***, and ***[SMIxINT]***.
3. The command data should be set to the buffers ***[SMIxPBUFn]*** and ***[SMIxSBUFn]***.
4. When ***[SMIxRACR1]***<CycGo> is set to "1", the commands are issued successively.

Note1: The upper 4 bits (bits 7 to 4) in the dummy byte should be set to "1".
Note2: An example of this access is shown in Section "5.1 Example of Program Register Access".

3.5. Transfer Clock

The frequency of the transfer clock (SMIxCLK) is determined by the value in ***[SMIxDACR0]***<SPR[4:0]> or ***[SMIxRACR0]***<SPR[4:0]>. <SPR[4:0]> + 1 is each dividing value.

Transfer clock = fsys frequency / (<SPR[4:0]> + 1)

The frequency of the transfer clock is determined by a dividing value as shown in the following table.

Table 3.1 Transfer clock

<SPR[4:0]>	1	2	3	4	5	6	7	8	9	10	11	12	...	28	29	30	31
Dividing value / fsys [MHz]	2	3	4	5	6	7	8	9	10	11	12	13	...	29	30	31	32
160	80	53.3	40.0	32.0	26.7	22.9	20.0	17.8	16.0	14.5	13.3	12.3		5.52	5.33	5.16	5.00
140	70	46.7	35.0	28.0	23.3	20.0	17.5	15.6	14.0	12.7	11.7	10.8		4.83	4.67	4.52	4.38
120	60	40.0	30.0	24.0	20.0	17.1	15.0	13.3	12.0	10.9	10.0	9.2		4.14	4.00	3.87	3.75
100	50	33.3	25.0	20.0	16.7	14.3	12.5	11.1	10.0	9.1	8.3	7.7		3.45	3.33	3.23	3.13
80	40	26.7	20.0	16.0	13.3	11.4	10.0	8.9	8.0	7.3	6.7	6.2	...	2.76	2.67	2.58	2.50
60	30	20.0	15.0	12.0	10.0	8.6	7.5	6.7	6.0	5.5	5.0	4.6		2.07	2.00	1.94	1.88
40	20	13.3	10.0	8.0	6.7	5.7	5.0	4.4	4.0	3.6	3.3	3.1		1.38	1.33	1.29	1.25
20	10	6.7	5.0	4.0	3.3	2.9	2.5	2.2	2.0	1.8	1.7	1.5		0.69	0.67	0.65	0.63

Note1: The transfer clock should be 20 MHz or less. The combinations of the fsys frequency and the division value in gray-colored of Table 3.1 should not be set.
Note2: When the dividing value is an odd number, the duty of the transfer clock is not 50 %.
Low width = (Dividing value)/2 + 0.5, and High width = (Dividing value)/2 - 0.5.

3.6. Data Input and Output Timing

The data input or output is synchronous with the falling edge of SMIxCLK.

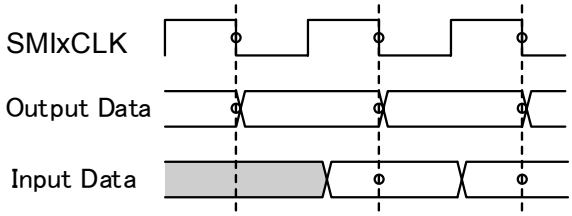

Figure 3.8 Data output and input timing

3.7. Interrupt

When the SPI cycle controlled by the program register access completes, ***[SMIxSTAT]***<CycDone> is set to “1”. At the same time, the interrupt (INTSMIx) can be generated. The setting of ***[SMIxINT]***<IntEn>=1 enables the interrupt generation.

In order to clear the generated interrupt, ***[SMIxSTAT]***<CycDone> should be set to “0”.

4. Registers

4.1. List of Registers

The control registers and their addresses are shown in the following tables.

Peripheral function		Channel/Unit	Base address (Base)	
			TYPE1	TYPE2
Serial memory interface	SMIF	ch 0	-	0x4000C000

Note: The channel/unit and base address type are different by products. Please refer to "Product Information" of the reference manual for the details.

Register name		Address (Base+)
Address Map Control Register 0	***[SMIxMAP0]***	0x0000
Address Map Control Register 1	***[SMIxMAP1]***	0x0004
Direct Access Control Register 0	***[SMIxDACR0]***	0x0008
Direct Access Control Register 1	***[SMIxDACR1]***	0x000C
Direct Read Control Register 0	***[SMIxDRCR0]***	0x0010
Direct Read Control Register 1	***[SMIxDRCR1]***	0x0014
Reserved	-	0x0018 to 0x03FF
Program Register Access Control Register 0	***[SMIxRACR0]***	0x0400
Program Register Access Control Register 1	***[SMIxRACR1]***	0x0404
Program Register Access Interrupt Control Register	***[SMIxINT]***	0x0408
Program Register Access Status Register	***[SMIxSTAT]***	0x040C
Reserved	-	0x0410 to 0x04FF
Program Register Access Primary Buffer Register 0	***[SMIxPBUF0]***	0x0500
Program Register Access Primary Buffer Register 1	***[SMIxPBUF1]***	0x0504
Reserved	-	0x0508 to 0x05FF
Program Register Access Secondary Buffer Register 0 to 63	***[SMIxSBUF0]*** to ***[SMIxSBUF63]***	0x0600 to 0x06FC

4.2. Details of Registers

4.2.1. *[SMIxMAP0]* (Address Map Control Register 0)

Bit	Bit Symbol	After Reset	Type	Description
31:28	FBA[15:12]	0xA	R	Read as “0xA”
27:16	FBA[11:0]	0x000	R/W	Mapping base address of Serial memory 0. (Note1)(Note2) 0x000 to 0x0FF: Base address <FBA[15:0]> det upper base address of Serial memory 0.The lower address is set to “0x0000”. (“0xA0000000” to “0xA0FF0000”)
15:6	-	0	R	Read as “0"
5:2	FDEN[3:0]	1000	R/W	Capacity of Serial memory 0 (Note2) 0000: 64KB 0101: 2MB 0001: 128KB 0110: 4MB 0010: 256KB 0111: 8MB 0011: 512KB 1000: 16MB 0100: 1MB 1001 to 1111: Reserved This field sets the capacity of Serial memory 0.
1	-	0	R/W	Write as "0"
0	RE	1	R/W	Mapping of Serial memory 0. 0: No mapping 1: Serial memory 0 is mapped When this bit is “0”, the memory is not mapped.

Note1: The value of <FBA[15:0]> should be aligned with the value set by <FDEN[3:0]>.

Note2: The address regions of Serial memory 0 and Serial memory 1 cannot overlap.

4.2.2. *[SMIxMAP1]* (Address Map Control Register 1)

Bit	Bit Symbol	After Reset	Type	Description
31:28	FBA[15:12]	0xA	R	Read as “0xA”
27:16	FBA[11:0]	0x000	R/W	Mapping base address of Serial memory 1. (Note1) 0x000 to 0x0FF: Base address <FBA[15:0]> det upper base address of Serial memory 1.The lower address is set to “0x0000”. (“0xA0000000” to “0xA0FF0000”)
15:6	-	0	R	Read as “0”
5:2	FDEN[3:0]	1000	R/W	Capacity of Serial memory 1 0000: 64KB 0101: 2MB 0001: 128KB 0110: 4MB 0010: 256KB 0111: 8MB 0011: 512KB 1000: 16MB 0100: 1MB 1001 to 1111: Reserved This field sets the capacity of Serial memory 1.
1	-	0	R/W	Write as "0"
0	RE	0	R/W	Mapping of Serial memory 1. 0: No mapping 1: Serial memory 0 is mapped When this bit is “0”, the memory is not mapped.

Note1: The value of <FBA[15:0]> should be aligned with the value set by <FDEN[3:0]>.

Note2: The address regions of Serial memory 0 and Serial memory 1 cannot overlap.

4.2.3. *[SMIxDACRn]* (Direct Access Control Register n) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:21	-	0	R	Read as “0”
20:16	SPR[4:0]	11111	R/W	Transfer clock of Serial memory n 0: Reserved 1 to 11111: Division setting of the transfer clock
15:8	SCSD[7:0]	0x00	R/W	SMIxCSn_N deassertion time of Serial memory n 0x00 to 0xFF: Deassertion time Deassertion time = fsys cycle time x <SCSD[7:0]>
7	-	0	R	Read as “0”
6	PollWIP	0	R/W	WIP polling before Read command is issued to Serial memory n 0: Disabled 1: Enabled For the details, refer to Section “3.4.1.2 Polling of WIP bit”.
5:4	-	00	R/W	Write as "01" (Note 2)
3	-	0	R	Read as “0”
2:0	-	000	R/W	Write as "000"

Note1: The deassertion time should be set from 0 ns to (fsys cycle time x 255 ns)ns.
Note2: The bit[5:4] field becomes “00” after the reset. The field should be written to “01” in the initialization setting.

4.2.4. *[SMIxDRCRn]* (Direct Read Control Register n) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:24	CmdOp[7:0]	0x0B	R/W	Op-code of SPI command of Serial memory n Op-code of SPI command should be set.
23:16	-	0	R	Read as “0”
15:12	DmyBc[3:0]	0001	R/W	SPI dummy byte count of Serial memory n 0x0 to 0xF: Dummy byte count SPI dummy byte count (0 to 15) should be set.
11	-	0	R/W	Write as "0"
10:8	-	0	R	Read as “0”
7:6	DatIO[1:0]	00	R/W	SPI data input and output control of Serial memory n 00: Single 10: Quad 01: Dual 11: Reserved
5:4	DmyIO[1:0]	00	R/W	SPI dummy input and output control of Serial memory n 00: Single 10: Quad 01: Dual 11: Reserved
3:2	AdrIO[1:0]	00	R/W	SPI address input and output control of Serial memory n 00: Single 10: Quad 01: Dual 11: Reserved
1:0	CmdIO[1:0]	00	R/W	SPI command input and output control of Serial memory n 00: Single 10: Quad 01: Dual 11: Reserved

Note1: When a multi-I/O is used, the value in ***[SMIxDRCRn]*** should be set to the corresponding value in “ Table 4.1 ***[SMIxDRCRn]*** setting value when using a multi-I/O”.

Table 4.1 *[SMIxDRCRn]* setting value when using a multi-I/O

Command	*[SMIxDRCRn]* setting value
Fast Read Dual Output	0x3B001040
Fast Read Dual I/O	0xBB001054
Fast Read Quad Output	0x6B001080
Fast Read Quad I/O	0xEB0030A8

4.2.5. *[SMIxRACR0]* (Program Register Access Control Register 0)

Bit	Bit Symbol	After Reset	Type	Description
31:21	-	0	R	Read as “0”
20:16	SPR[4:0]	11111	R/W	Transfer clock of Serial Memory 0: Reserved 1 to 11111: Division setting of the transfer clock
15:8	SCSD[7:0]	0xFF	R/W	SMIxCSn_N deassertion time 0x00 to 0xFF: Deassertion time Deassertion time = fsys frequency x <SCSD[7:0]> (Note1)
7:6	-	0	R	Read as “0”
5:4	-	00	R/W	Write as "01" (Note2)
3	-	0	R	Read as “0”
2:0	-	000	R/W	Write as "000"

Note1: The deassertion time should be set from 0 ns to (fsys cycle time x 255 ns)ns.
Note2: The bit[5:4] field becomes “00” after the reset. The field should be written to “01” in the initialization setting.

4.2.6. *[SMIxRACR1]* (Program Register Access Control Register 1)

Bit	Bit Symbol	After Reset	Type	Description
31:24	SBufBc[7:0]	0x00	R/W	Transfer Byte count in the secondary buffer 0x00 to 0xFF: Transfer byte count The transfer byte count is (the set value +1) bytes.
23:19	-	0	R	Read as “0”
18:16	PBufBc[2:0]	0x0	R/W	Transfer Byte count in the primary buffer 0x0 to 0x7: Transfer byte count The transfer byte count is (the set value +1) bytes.
15:6	-	0	R	Read as “0”
5	SBufEn	0	R/W	Secondary buffer enable 0: Secondary buffer is not used. 1: Secondary buffer is used.
4	PBufEn	0	R/W	Primary buffer enable 0: Primary buffer is not used. 1: Primary buffer is used.
3:2	-	0	R	Read as “0”
1	CSNum	0	R/W	Asserted CS 0: SMIxCS0_N 1: SMIxCS1_N
0	CycGo	0	R/W	Program register access control 0: don't care 1: Access start by the program register access (Note) Read as “0”

Note: <CycGo> should not be set to “1” while ***[SMIxSTAT]***<CycProg> is “1” (during SPI cycle).

4.2.7. *[SMIxINT]* (Program Register Access Interrupt Control Register)

Bit	Bit Symbol	After Reset	Type	Description
31:1	-	0	R	Read as “0”
0	IntEn	0	R/W	Enable or disable of SMIF interrupt 0: Interrupt generation disable 1: Interrupt generation enable

4.2.8. *[SMIxSTAT]* (Program Register Access Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	0	R	Read as “0”
1	CycProg	0	R	SPI cycle status 0: SPI cycle completion 1: During SPI cycle When ***[SMIxRACR1]***<CycGo> is set to “1”, this bit becomes “1”.
0	CycDone	0	R	SPI cycle completion status 0: - 1: SPI cycle completion
			W	<CycDone> clear 0: <CycDone> is cleared to “0” 1: don't care

4.2.9. *[SMIxPBUFn]* (Program Register Access Secondary Buffer Register n) (n=0 to 1)

The primary buffer consists of 8-bit registers, which is assigned to ***[SMIxPBUF0]*** and ***[SMIxPBUF1]***.

[SMIxPBUF0]

Bit	Bit Symbol	After Reset	Type	Description
31:24	PBuf3[7:0]	Undefined	R/W	Primary buffer 3
23:16	PBuf2[7:0]	Undefined	R/W	Primary buffer 2
15:8	PBuf1[7:0]	Undefined	R/W	Primary buffer 1
7:0	PBuf0[7:0]	Undefined	R/W	Primary buffer 0

[SMIxPBUF1]

Bit	Bit Symbol	After Reset	Type	Description
31:24	PBuf7[7:0]	Undefined	R/W	Primary buffer 7
23:16	PBuf6[7:0]	Undefined	R/W	Primary buffer 6
15:8	PBuf5[7:0]	Undefined	R/W	Primary buffer 5
7:0	PBuf4[7:0]	Undefined	R/W	Primary buffer 4

4.2.10. *[SMIxSBUFn]* (Program Register Access Secondary Buffer Register n) (n=00 to 63)

The secondary buffer consists of 256 8-bit registers, <SBuf0[7:0]> to <Sbuf255[7:0]>. It is assigned to ***[SMIxSBUF00]*** to ***[SMIxSBUF63]***.

The example of ***[SMIxSBUF00]*** and ***[SMIxSBUF63]*** is shown as follows. The <Sbuf4[7:0]> to <Sbuf251[7:0]> is also same constitution. It is assigned to ***[SMIxSBUF01]*** to ***[SMIxSBUF62]***.

[SMIxSBUF00]

Bit	Bit Symbol	After Reset	Type	Description
31:24	SBuf3[7:0]	Undefined	R/W	Secondary buffer 3
23:16	SBuf2[7:0]	Undefined	R/W	Secondary buffer 2
15:8	SBuf1[7:0]	Undefined	R/W	Secondary buffer 1
7:0	SBuf0[7:0]	Undefined	R/W	Secondary buffer 0

[SMIxSBUF63]

Bit	Bit Symbol	After Reset	Type	Description
31:24	SBuf255[7:0]	Undefined	R/W	Secondary buffer 255
23:16	SBuf254[7:0]	Undefined	R/W	Secondary buffer 254
15:8	SBuf253[7:0]	Undefined	R/W	Secondary buffer 253
7:0	SBuf252[7:0]	Undefined	R/W	Secondary buffer 252

5. Example of Usage

5.1. Example of Program Register Access

Some examples of the program register access of SPI Flash memory are shown in this section. The op-code, the command issuing procedure, the input and output timings, and others depend on SPI Flash memory. The specification of the used SPI Flash memory should be checked in advance.

5.1.1. Fast Read

256-Byte data is read from SPI Flash memory in Fast Read mode (0x0B).

1. Primary buffer setting
   [SMIxPBUF0]<PBuf0> // 0x0B Fast Read op-code
   [SMIxPBUF0]<PBuf1> // Serial memory address (bits 23 to 16)
   [SMIxPBUF0]<PBuf2> // Serial memory address (bits 15 to 8)
   [SMIxPBUF0]<PBuf3> // Serial memory address (bit 7 to 0)
   [SMIxPBUF1]<PBuf4> // Dummy data

2. ***[SMIxRACR1]*** setting
   <PBufBc>=4 // (1-byte op-code, 3-byte address and 1-byte dummy) - 1
   <SBufBc>=255 // 256 Secondary buffers are used.
   <PBufEn>=1 // Primary buffer is enabled.
   <SBufEn>=1 // Secondary buffer is enabled.
   <CSNum>= The setting value depends on the implemented Serial memory.

3. Start of the program register access
   When ***[SMIxRACR1]***<CycGo> is set to “1”, the command is transferred and executed. The data in the Serial memory is transferred to Secondary buffer.

5.1.2. Status Read

Status Read command (0x05) reads the value of the status register in SPI Flash memory.

1. Primary buffer setting
   [SMIxPBUF0]<PBuf0>: 0x05 // Status Read op-code
   [SMIxPBUF0]<PBuf1>: 0xFF // Dummy Byte (Note)

2. ***[SMIxRACR1]*** setting
   <PBufBc>=1 // (1-byte ope-code and 1-byte dummy) - 1
   <SBufBc>=0 // Secondary buffer is not used.
   <PBufEn>=1 // Primary buffer is enabled
   <SBufEn>=0 // Secondary buffer is disabled.
   <CSNum>= The setting value depends on the implemented serial memory.

3. Start of the program register access

   When ***[SMIxRACR1]***<CycGo> is set to “1”, the command is transferred and executed. The data of the status register in the Serial memory is transferred to Primary buffer ***[SMIxPBUF0]***<PBuf1>.

Note: After the op-code is transmitted, SMIF is in Read cycle. And the dummy data is transmitted at the same time.

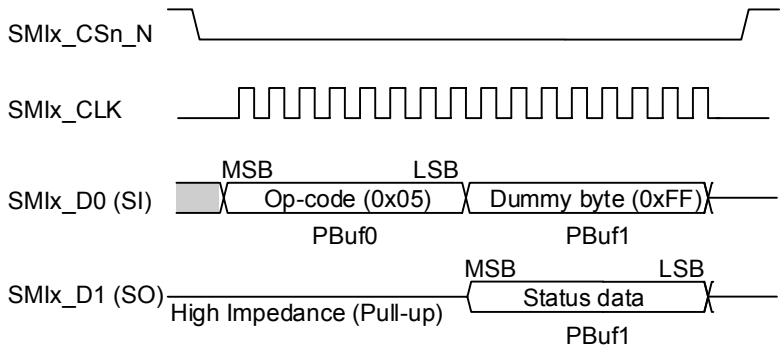

Figure 5.1 Example of Status Read

5.1.3. Page Program

Page Program command (0x02) writes 256-byte data to SPI Flash memory.

1. This setting specifies that SPI Flash memory is not accessed until Write Enable completes.

2. Primary buffer setting
   [SMIxPBUF0]<PBuf0>: 0x06 // Write Enable op-code

3. ***[SMIxRACR1]*** setting
   <PBufBc>=0 // (1-byte op-code) - 1
   <SBufBc>=0 // Secondary buffer is not used.
   <PBufEn>=1 // Primary buffer is enabled.
   <SBufEn>=0 // Secondary buffer is disabled.
   <CSNum>= The setting value depends on the implemented serial memory.

4. Start of the program register access
   When ***[SMIxRACR1]***<CycGo> is set to "1", the command is transferred and executed. The write to the serial memory is enabled.

5. This setting specifies that SPI Flash memory is not accessed until Page Program completes.

6. Primary buffer setting
   [SMIxPBUF0]<PBuf0> // 0x02 Page Program op-code
   [SMIxPBUF0]<PBuf1> // Erased sector address (bit 23 to 16) in the Serial memory
   [SMIxPBUF0]<PBuf2> // Erased sector address (bit 15 to 8) in the Serial memory
   [SMIxPBUF0]<PBuf3> // Erased sector address (bit 7 to 0) in the Serial memory

7. Secondary buffer setting
   [SMIxSBUF0]<SBuf0> // Data [7:0] which is written to the address [23:0] + 0x00.
   [SMIxSBUF0]<SBuf1> // Data [15:8] which is written to the address [23:0] + 0x01
   [SMIxSBUF0]<SBuf2> // Data [23:16] which is written to the address [23:0] + 0x02.
   :
   [SMIxSBUF63]<SBuf254> // Data [2039:2032] which is written to the address [23:0] + 0xFE
   [SMIxSBUF63]<SBuf255> // Data [2047:2040] which is written to the address [23:0] + 0xFF

8. ***[SMIxRACR1]*** setting
   <PBufBc>=3 // (1-byte op-code, 3-byte address) - 1
   <SBufBc>=255 // 256 secondary buffer registers are used.
   <PBufEn>=1 // Primary buffer is enabled.
   <SBufEn>=1 // Secondary buffer is enabled.
   <CSNum>= The setting value depends on the implemented serial memory.

9. Start of the program register access

   When ***[SMIxRACR1]***<CycGo> is set to "1", the command is transferred and executed. The 256-byte data is written to the Serial memory.

5.1.4. Full Chip Erase

Full Chip Erase command (0xC7) erases all the regions of SPI Flash memory.

1. This setting specifies that SPI Flash memory is not accessed until Full Chip Erase completes.

2. Primary buffer setting
   [SMIxPBUF0]<PBuf0> // 0xC7 Full Chip Erase op-code

3. ***[SMIxRACR1]*** setting
   <PBufBc>=0 // (1-byte op-code) - 1
   <SBufBc>=0 // Secondary buffer is not used.
   <PBufEn>=1 // Primary buffer is enabled.
   <SBufEn>=0 // Secondary buffer is disabled.
   <CSNum>= The setting value depends on the implemented serial memory.

4. Start of the program register access

   When ***[SMIxRACR1]***<CycGo> is set to "1", the command is transferred and executed. All the regions of the Serial memory are erased.

5.1.5. Sector Erase

Sector Erase command (0x20) erases a specified sector in SPI Flash memory.

1. This setting specifies that SPI Flash memory is not accessed until Sector Erase completes.

2. Primary buffer setting
   [SMIxPBUF0]<PBuf0> // 0x20 Sector Erase op-code
   [SMIxPBUF0]<PBuf1> // Erased sector address (bit 23 to 16) in the serial memory
   [SMIxPBUF0]<PBuf2> // Erased sector address (bit 15 to 8) in the serial memory
   [SMIxPBUF0]<PBuf3> // Erased sector address (bit 7 to 0) in the serial memory

3. ***[SMIxRACR1]*** setting
   <PBufBc>=3 // (1-byte op-code, 3-byte address) - 1
   <SBufBc>=0 // Secondary buffer is not used.
   <PBufEn>=1 // Primary buffer is enabled.
   <SBufEn>=0 // Secondary buffer is disabled.
   <CSNum>= The setting value depends on the implemented serial memory.

4. Start of the program register access

   When ***[SMIxRACR1]***<CycGo> is set to "1", the command is transferred and executed. The specified sector of the Serial memory is erased.

5.2. Example of Serial Memory Connection

Examples of connection to the Serial memory are shown in this section.
Each pin is supposed to be connected as follows. Actually, the connection should be done according to the specification of the connected memory devices or external devices.

SMIxCS0_N and SMIxCS1_N: Pull-up
SMIxCLK: Pull-down
SMIxD0 to 3: Pull-up (Note)

Note: When SMIxD2 and SMIxD3 can be used as normal ports when they are not connected to Serial memory devices.

5.2.1. Example of Single/Dual Connection

1) Only Serial memory 0

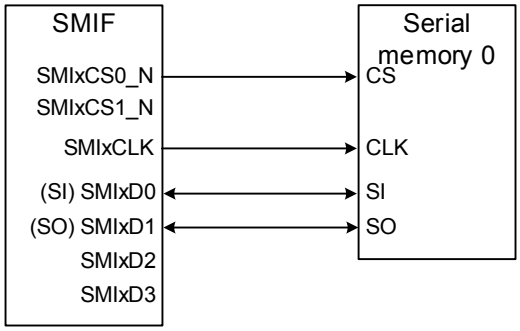

Figure 5.2 Example of Single/Dual connection (Serial memory 0)

2) Serial memory 0 and Serial memory 1

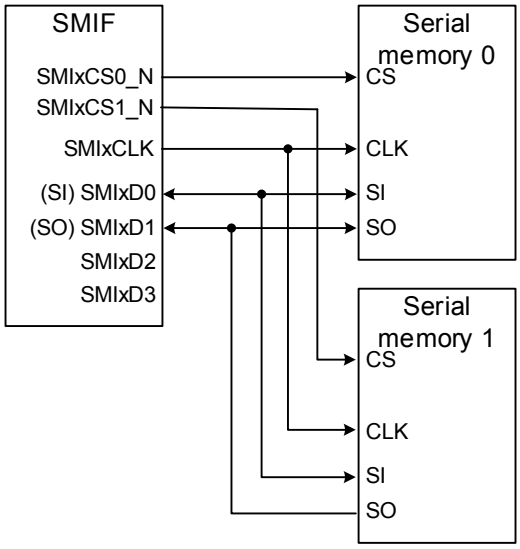

Figure 5.3 Example of Single/Dual connection (Serial memory 0 and 1)

5.2.2. Example of Quad Connection

1) Only Serial memory 0

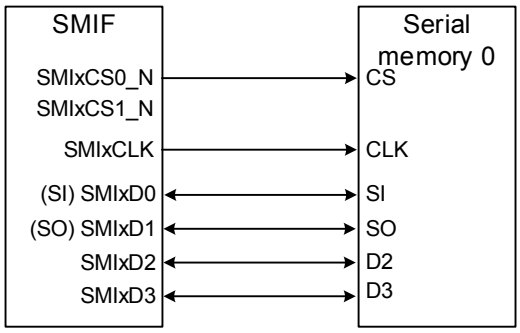

Figure 5.4 Example of Quad connection (Serial memory 0)

2) Serial memory 0 and Serial memory 1

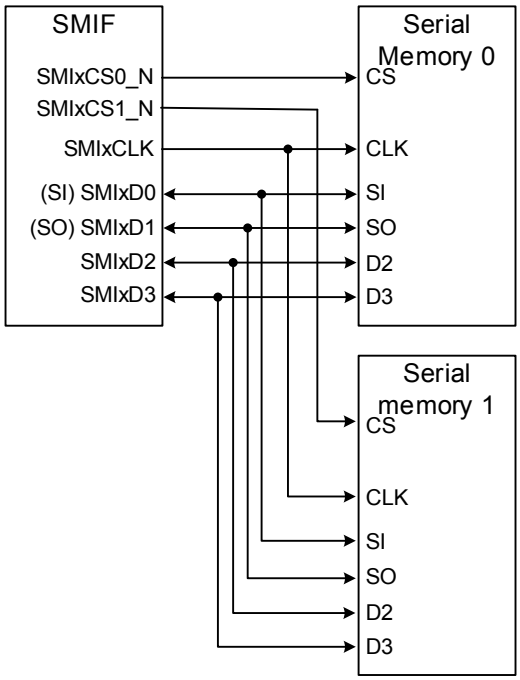

Figure 5.5 Example of Quad connection (Serial memory 0 and 1)

6. Precaution for Usage

- The addresses to which the registers are not assigned should not be accessed.
- Some products do not have SMIxCSn_N pin and SMIxDn pin.

7. Revision History

Table 7.1 Revision History

Revision	Date	Description
1.0	2018-01-23	First release
1.1	2018-07-26	-Conventions Modified explanation of trademark - 3.4.1.1 SPI Flash Command Modified waveform of Dummy byte in Figure 3.6 - 4.2.10 *[SMIxSBUFn]* (Program Register Access Secondary Buffer Register n) (n=00 to 63) Modified title “n=0” to “n=00” Modified register name ***[SMIxSBUF0] to [SMIxSBUF00], [SMIxSBUF1]*** to ***[SMIxSBUF01]***

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as "TOSHIBA".
Hardware, software and systems described in this document are collectively referred to as "Product".

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**