Application Processor Lite *ApP Lite*

TZ1000 Series

Reference Manual

MCU SRAM Controller

Revision 1.1

2018-01

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Table of Contents

List of Figures

List of Tables

Preface

This specification document describes the specification of MCU SRAM Controller which has been developed for the TZ1000 Series.

Intended Audience

This document is intended for the following users.

Driver software developers.

System designers

Conventions in this document

- The numerical values are expressed as follows.
  Hexadecimal number: 0xABC
  Decimal number: 123 or 0d123 - Only when it needs to be explicitly shown that they are decimal numbers.
  Binary number: 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 [3:0].
  [3:0] shows four signal names 3, 2, 1, and 0 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], and [XYZ3] to [XYZn]
- The bit range of a register is written like as [3:0].
  Example: Bit[3:0] expresses the range of bits 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-bit
  Half word: 16-bit
  Word: 32-bit
  Double word: 64-bit
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, don't use the read value.
- Properties of each bit in a register are expressed as follows.
  R: Read only
  W: Write only
  W1C: Write 1 Clear - The corresponding bit is cleared (=0) when "1" is written to this bit.
  W1S: Write 1 Set - The corresponding bit is set (=1) when "1" is written to this bit.
  R/W: Read and Write are possible.
  R/W0C: Read/Write 0 Clear
  R/W1C: Read/Write 1 Clear
  R/W1S: Read/Write 1 Set
  RS/WC: Read Set/Write Clear - Set after read operation, cleared after write operation.
- 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.

1. Overview

This module is an SRAM controller. The feature of the controller is as follows.

- SRAM
  - Capacity: 288 KB
  - The SRAM space assignment can be changed for the following 3 ports which are used as the access interface.

	Code Area		SRAM Area
Mode	port0	port1	port2
Remap0	128 KB	128 KB	32 KB
Remap1	128 KB	96 KB	64 KB

    * The mode is changed by the boot configuration pin (BOOTMODE1).
  - The SRAM has three power domains. Each domain can be controlled separately to stop the power supply and to enter the retention state.
  - Retention function
    This function saves the power consumption and the data is protected. The following two modes are supported.
    - Retention mode: Smaller power consumption and longer returning time than the Wait mode.
    - Wait mode: Larger power consumption and shorter returning time than the Retention mode.

- SRAM access interface
  Each master accesses the SRAM using this interface through the main bus.
  - Protocol: AHB
  - Port count: 3 (port0, port1, and port2)
  - AHB protocol
    - HRESP does not support ERROR/RETRY/SPLIT. It supports only OKAY.
    - HBURST is not used.
    - HTRANS = BUSY is handled as the same as HTRANS = IDLE.
    - WRAP transfer is not supported.
    - HPROT signal is ignored.
    - HLOCK signal is ignored.

- Register setting interface
  The control of data transfer between memories and the control of the error detection, which are described later, are done using this interface.
  - Protocol: APB2

- Data transfer interface between the SPI Flash memory and the SRAM
  * This transfer is referred as S2M transfer.
  The SPIC can access directly the SRAM using this interface.

- Data transfer in the SRAM
  * This transfer is referred to as M2M transfer.
  - Data transfer can be done inside of the 288KB SRAM space.
  - The setting is done through the main bus.

- Arbiter function
  - The arbitration is done among the access by the main bus, M2M access, and S2M access.

- Error detection and notification
  - An error is detected and the error interrupt is requested for the corresponding access.

2. Block Diagram

The block diagram of this module is shown in Figure 2.1.

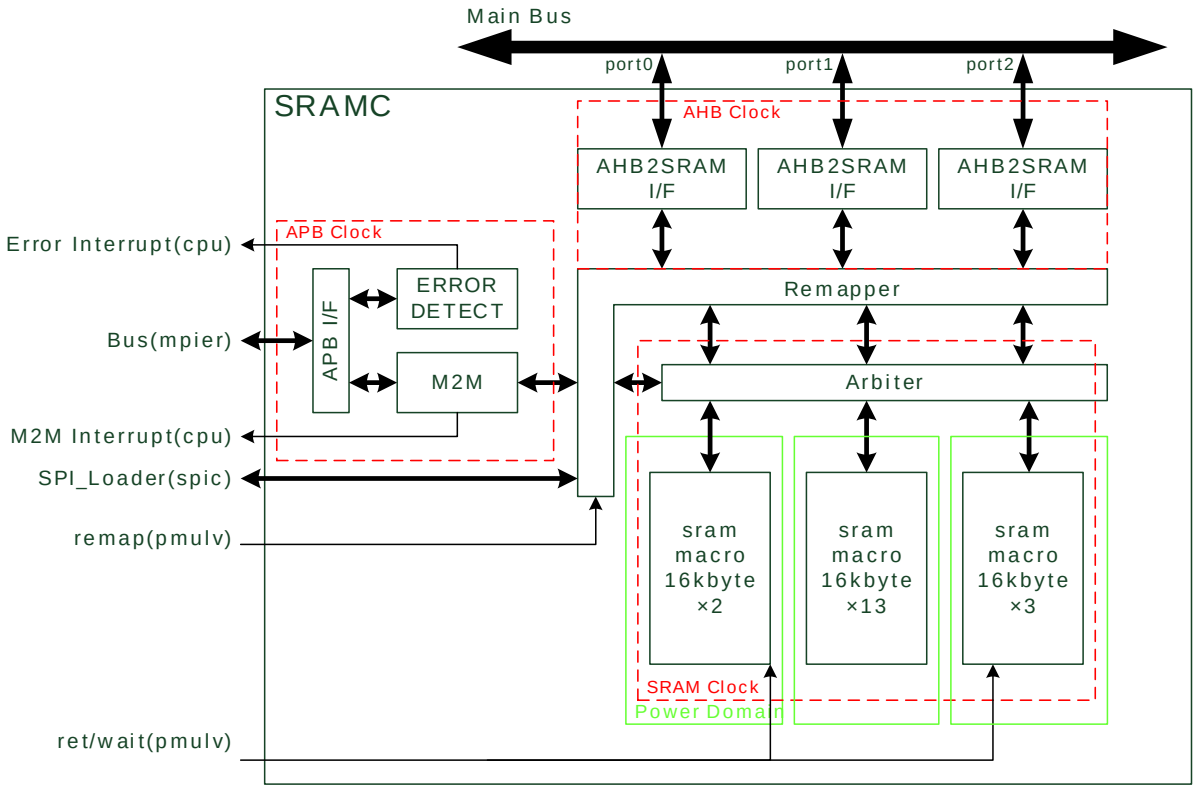

Figure 2.1 SRAMC block diagram

2.1. Block Function

- AHB2SRAM I/F
  - SRAM access interface
  - AHB protocol is converted to SRAM protocol.
- Remapper
  - Global address is converted to SRAM address.
  - Address conversion depends on the remap signal.
- Arbiter
  - The arbitration among the accesses by the main bus, the S2M access, and the M2M access is done by the 16 KB SRAM macro.
- M2M
  - Control for the data transfer inside of the SRAM.
- ERROR DETECT
  - Error detection control.
- APB I/F
  - The control for the M2M transfer and error detection is done using this interface.
- SRAM
  - 18 macros of 16KB SRAM blocks. 3 power domains are included.

2.2. Clock and Reset

7 clocks are supported.

- AHB clocks
  - 3 clocks (for each interface)
- SRAM clocks
  - 3 clocks (for each power domain)
- APB clock
  - 1 clock

2 reset signals are supported.

- AHB reset
  - For the blocks supplied with both the AHB clock and the APB clock
- APB reset
  - For the blocks supplied with the APB clock

The restrictions for the control of the clocks and reset signals are as follows.

- The stop of the clocks and the reset assertion are prohibited while the SRAM is accessed.
- When the AHB reset is asserted, the APB reset should be also asserted.
- The APB reset can be asserted solely.
- No order restrictions are applied to the clock stop and the reset assertion for all clocks and resets.

2.3. Address Hole

The SRAMC has the SRAM address space and the register address space. The address holes of each address space are handled as follows.

SRAM space: There are no address holes. The M2M transfer or the S2M transfer may access the space out of the SRAM space. If the access occurs, the write is ignored and the read receives 0 without the detection of any errors.

Register space: The write access is ignored. The read access receives 0 without the detection of any errors.

3. Register map

Table 3.1 MCU SRAM Controller Register map

Register Name	Type	Width	Reset Value	Address Offset
M2MSTART	RW	32	0x0000 0000	0x0000 0000
M2MSTOP	RW	32	0x0000 0000	0x0000 0004
M2MSTATUS	RO	32	0x0000 0000	0x0000 0008
M2MSRCADDR	RW	32	0x0000 0000	0x0000 0010
M2MDSTADDR	RW	32	0x0000 0000	0x0000 0020
M2MSIZE	RW	32	0x0000 0000	0x0000 0024
M2MINTEN	RW	32	0x0000 0000	0x0000 0030
M2MINTCLR	RW	32	0x0000 0000	0x0000 0034
ERRINTEN	RW	32	0x0000 0000	0x0000 0040
ERRSTATUS	RO	32	0x0000 0000	0x0000 0044
ERR1LOG	RO	32	0x0000 0000	0x0000 0048
ERR2LOG	RO	32	0x0000 0000	0x0000 004C
ERRCLR	RW	32	0x0000 0000	0x0000 0050

4. Function

4.1. Power Management

4.1.1. SLEEP0 Mode

Normal operation.

4.1.2. SLEEP1/2 Modes

All clocks in the SRAMC stop. The clocks can be stopped while the SRAMC is operating. But, when a master accesses the SRAMC, the SRAMC clocks may not stop because the master operation should be guaranteed.

4.1.3. WAIT Mode

In the WAIT mode, the PS0/1/2 power domains are controlled by the *[POWERDOMAIN_CTRL_MODE_FOR_WAIT]* register in the PMU.

The initial state is the WAIT mode. It can be set to OFF.

All clocks in the SRAMC stop. The clocks can be stopped while the SRAMC is operating. But, when a master accesses the SRAMC, the SRAMC clocks may not stop, because the master operation should be guaranteed.

4.1.4. WAIT-RETENTION Mode

In the WAIT-RETENTION mode, the PS0/1/2 power domains are controlled by the *[POWERDOMAIN_CTRL_MOD E_FOR_WRET]* register in the PMU. The initial state is the WAIT mode. It can be set to OFF.

All clocks in the SRAMC stop. The clocks can be stopped while the SRAMC is operating. But, when a master accesses the SRAMC, the SRAMC clocks may not stop, because the master operation should be guaranteed.

4.1.5. RETENTION Mode

In the RETENTION mode, the PS0/1/2 power domains are controlled by the *[POWERDOMAIN_CTRL_MODE_FO R_RET]* register in the PMU. The initial state is the RET mode. It can be set to OFF.

All clocks in the SRAMC stop. The clocks can be stopped while the SRAMC is operating. But, when a master accesses the SRAMC, the SRAMC clocks may not stop, because the master operation should be guaranteed.

4.1.6. RTC/STOP Modes

All clocks in the SRAMC stop. The clocks can be stopped while the SRAMC is operating. But, when a master accesses the SRAMC, the SRAMC clocks may not stop, because the master operation should be guaranteed.

The power is shut down. When returning to the normal operation, the registers are initialized and the re-setting is necessary.

4.2. Start-up and Stop Procedure

4.2.1. Clock and Reset Start-up Procedure

This procedure is supposed that the PS0/1/2 power domains are already supplied with the power. The power state can be confirmed by reading the ***[POWERDOMAIN_CTRL_STATUS]***.PDMODE_PS0/1/2 register in the PMU. If they are in OFF state, they should be in ON state before the following procedure is executed.

(1) The dynamic clock gating register should be set. The following bits should be set to 1.
[DCG_PM_0].DCG_mpierclk_sramc_pclk
[DCG_PM_0].DCG_mpierclk_sramc_s0hclk
[DCG_PM_0].DCG_mpierclk_sramc_s1hclk
[DCG_PM_0].DCG_mpierclk_sramc_s2hclk
[DCG_PM_0].DCG_mpierclk_sramc_hclk
[DCG_PM_0].DCG_mpierclk_mpier_hclk

(2) The clock gating should be cleared. The following bits should be set to 1.
[CG_OFF_PM_0].CG_mpierclk_sramc_pclk
[CG_OFF_PM_0].CG_mpierclk_sramc_s2hclk
[CG_OFF_PM_0].CG_mpierclk_sramc_s1hclk
[CG_OFF_PM_0].CG_mpierclk_sramc_s0hclk
[CG_OFF_PM_0].CG_mpierclk_sramc_hclk

(3) The resets should be deasserted. The following bits should be set to 1.
[SRST_OFF_PM_0].SRST_asyncrst_sramc_hrstn
[SRST_OFF_PM_0].SRST_asyncrst_sramc_prstn

4.2.2. Stop Procedure

The following procedure does not control the PS0/1/2 power domains.

4.2.2.1. Clock Stop Procedure

The following bits should be set to 1.
[CG_ON_PM_0].CG_mpierclk_sramc_pclk
[CG_ON_PM_0].CG_mpierclk_sramc_s2hclk
[CG_ON_PM_0].CG_mpierclk_sramc_s1hclk
[CG_ON_PM_0].CG_mpierclk_sramc_s0hclk
[CG_ON_PM_0].CG_mpierclk_sramc_hclk

4.2.2.1.1. Clock and Reset Stop procedure

The following bits should be set to 1.
[SRST_ON_PM_0].SRST_asyncrst_sramc_hrstn
[SRST_ON_PM_0].SRST_asyncrst_sramc_prstn
The reset assertion sets the clock gating automatically.

4.2.3. Frequency Setting

The SRAMC clocks belong to the CD_MAIN domain. Their frequencies can be changed. It is noted that the clocks are also used by other blocks. If the frequency of the clock is changed, the other blocks receive the changed clock. For detail, refer to the specification of the PMU.

4.3. SRAM Access I/F

The AHB protocol is converted to the SRAM protocol by this interface. Each master accesses the SRAM using this interface through the main bus.

4.3.1. Function

- Clock and reset
  - Before the access, the gating of the AHB/SRAM clocks should be released.
  - Before the access, the AHB reset should be deasserted.
- The access to the space out of the SRAM space
  - The write access is ignored. The read access receives undefined data without detecting any errors.

4.4. SRAM

The SRAMC includes 18 macros of 16KB SRAM (SMAA). The macros are assigned into 3 power domains (PSB). * AA represents 0 to 17 and B, 0 to 2.

- POWER_SRAM0 domain (PS0)
  - Capacity: 32 KB
  - SM00/SM16
- POWER_SRAM1 domain (PS1)
  - Capacity: 208 KB
  - SM03/SM04/SM05/SM06/SM07/SM08/SM09/SM10/SM11/SM12/SM13/SM14/SM15
- POWER_SRAM2 domain (PS2)
  - Capacity: 48 KB
  - SM01/SM02/SM17

4.5. Remap Function

The remap function can change the code space size and the data space size in the SRAM. And it can also change the usage of the SRAM macros in the power domain.

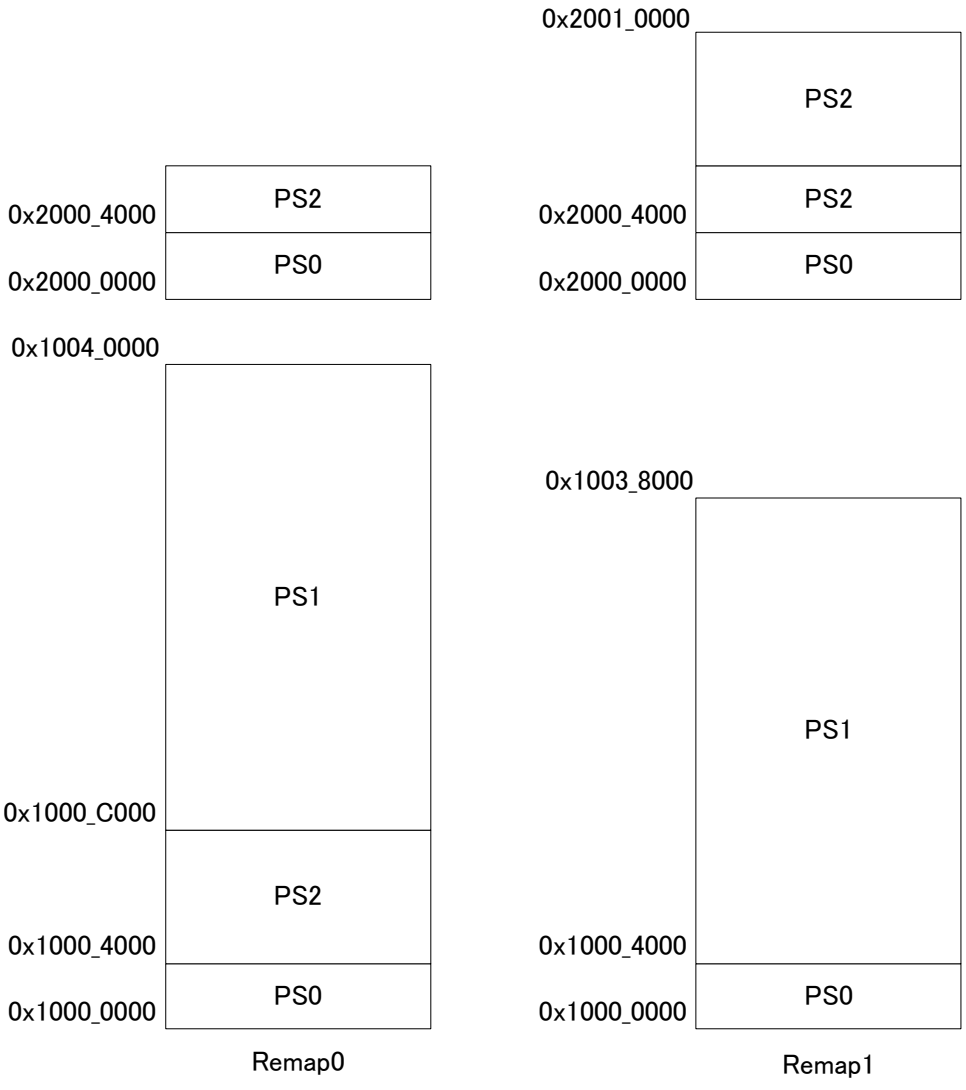

Figure 4.1 Address space remapping

4.6. Arbiter Function

The arbitration is done among the main bus access, the M2M access, and the S2M access, in units of 16 KB address spaces.

The arbitration priority order is as follows.

- Priority order
  - S2M access ≥ Main bus access > M2M access
  - When the competition between the S2M access and the main bus access occurs, the S2M is prioritized to access the SRAM. The write access of the main bus is ignored, and the read access receives 0. The error interrupt is generated by the error detection function.
  - The M2M access keeps waiting until the access becomes possible.

4.7. M2M Function

The data transfer can be done inside of the SRAMC without using DMAC.

4.7.1. Function

- Clock and reset
  - Before the access, the APB/SRAM clock gating should be released.
  - Before the access, the AHB reset and the APB reset should be deasserted.
  - The clock stop and the reset assertion are prohibited except in the Stop state.
- Access to the space out of the SRAM space
  - When the access occurs, the M2M does not execute and the error is detected.
- The M2M transfer repeats in units of Words to read data from the source address and to write the data to the destination address.
- M2M status
  - Stop
  - Active
  - Wait for Interrupt Clear
- The address is specified by the global address.
- The transfer address and the transfer data size are in units of Words.
- The transfer end interrupt can be generated.
- The transfer can be forced to stop in the Active state.

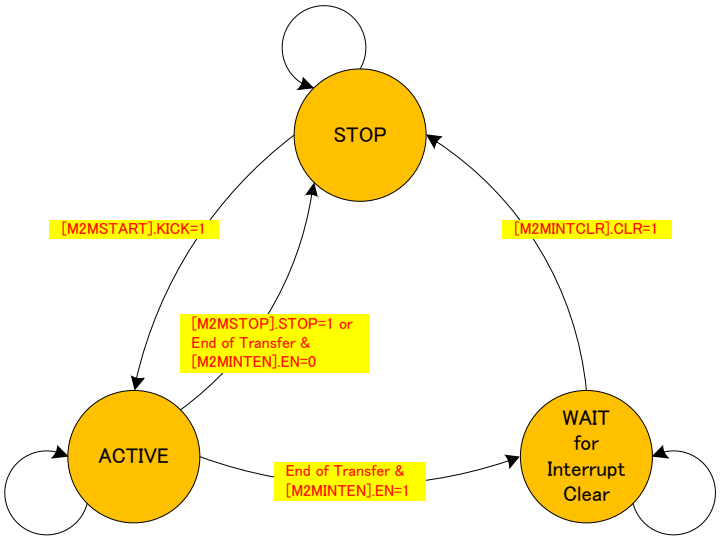

Figure 4.2 M2M transfer state transition

4.7.2. Interrupt Control

The interrupt request is generated by writing 1 to ***[M2MINTEN]***.EN. When the transfer completes, the M2M transfer transits to the state of Wait for Interrupt Clear and the interrupt request is issued. The interrupt request is cleared by writing 1 to ***[M2MINTCLR]***.CLR.

If ***[M2MINTEN]***.EN is 0, the interrupt request is not generated. When the transfer completes, the M2M transfer transits to the Stop state. This transition is confirmed by reading ***[M2MSTATUS]***.

* Note: The ***[M2MINTEN]*** register should be set in the Stop state only. The state can be confirmed by reading ***[M2MSTATUS]***.

4.7.3. M2M Transfer Start-up Procedure

The APB/SRAM clock gating should be released. And the AHB reset and the APB reset should be deasserted. The following procedure is supposed that ***[M2MINTEN]***.EN is 1.

(1) ***[M2MSTATUS]***.STATUS is read and the Stop state is confirmed.
(2) ***[M2MSRCADDR]***, ***[M2MDSTADDR]***, ***[M2MSIZE]***, and ***[M2MINTEN***] are set.
(3) ***[M2MSTART]***.START is written to 1. The operation starts up.
(4) M2M transfer starts.
(5) M2M transfer completes. Then it transits to the state of Wait for Interrupt Clear.M2M transfer end Interrupt is occurred.
(6) The interrupt is cleared. Then it enters the Stop state and the transfer finishes.

The start-up (***[M2MSTART]***.START = 1) is ignored in the state of Wait for Interrupt Clear. It stays in the state.

4.7.4. M2M Transfer Stop Procedure

The stop procedure forces to transit the state from the Active state to the stop one. This stop procedure is ignored in the state of Wait for Interrupt Clear. So, the interrupt is not cleared. It is not possible to resume from the stopped state. The start-up procedure should be done. again.

(1) ***[M2MSTOP]***.STOP is written to 1.
(2) The state transits from the Active state to the Stop one.

4.7.5. Error Detection

The illegal accesses and settings in the M2M transfer are detected and notified.

- If ***[M2MSIZE]***.SIZE is 0, the state transition from the Stop state to the Active state does not transit and does not occur at the start-up. The error is detected.
- When ***[M2MSRCADDR]***, ***[M2MDSTADDR]***, or ***[M2MSIZE]*** register is changed except in the Stop state, the error is detected.
- The access to the space out of the SRAM space may be done by the settings of ***[M2MSRCADDR]***, ***[M2MDSTADDR]***, and ***[M2MSIZE].*** But the state transition from the Stop state to the Active state does not transit and does not occur at the start-up. And the error is detected.
- When an error for the M2M transfer is detected, the state transits from the Active state to the Stop one. If it is in the state of Wait for Interrupt Clear, it does not transit to the Stop state.

4.8. Transfer Interface between SPI Flash Memory and SRAM

The SPIC uses this interface to access the SRAM. The transfer address is specified by the global address which is converted to the SRAM address in the SRAMC.

4.8.1. Function

- Clock and reset
  - Before the access, the SRAM clock gating should be released.
  - Before the access, the AHB reset should be deasserted.
- Access to the space out of the SRAM space
  - The write access is ignored and the read access receives 0 without the detection of any errors.

4.8.2. Precaution

The SPIC clock gating is supposed to be released, and the reset is supposed to be deasserted.

4.9. Error Detection and Notification

This function is served to facilitate the software debugging.

4 errors are defined as follows. When a cause of error occurs, the cause is set to ***[ERRSTATUS]***. The error information (the address and the master) of Error1 and Error2 can be checked in the ***[ERR1LOG]*** and ***[ERR2LOG]*** registers.

Error1: The access to the SRAM macro in the power shut-down or retention state.
- Cause address: The upper 22 bits are stored.
- Cause master: S2M, M2M, or the main bus (Port0/1/2)
- When pmu.***[DCG_PM_0]***.DCG_mpierclk_sramc_pclk is enabled, error detection by S2M and the main bus is no done. Therefore the cause of it is only M2M.

Error2: The access competition to the same SRAM macro caused by the main bus or the M2M while the S2M transfer is operating.
- Cause address: The upper 22 bits are stored.
- Cause master: M2M or the main bus (Port0/1/2)
- When pmu.***[DCG_PM_0]***.DCG_mpierclk_sramc_pclk is enabled, error detection by the main bus is no done. Therefore the cause of it is only M2M.

Error3: The register update while the M2M is in the Active state.
- Cause master: M2M

Error4: M2M illegal access to the SRAM (including the size 0 setting)
- Cause master: M2M

- If a cause of an error is already asserted, its cause is not detected again until the cause is cleared. A different cause of an error can be detected.
- When Error1 occurs, the write access is ignored. For the S2M and M2M, the read data is 0. For the main bus, the read data is undefined.
- When Error2 occurs, the write access is ignored. For the M2M, the read data is 0. For the main bus, the read data is undefined.
- When multiple Error1 errors occur, the detection priority order is Main bus (Port2) > Main bus (Port1) > Main bus (Port0) > M2M > S2M.
- When multiple Error2 errors occur, the priority order is Main bus > M2M. There is no priority order among the ports of the main bus because only one port can access the 16KB space by the S2M transfer.

- When the pmu.*[DCG_PM_0]*.DCG_mpierclk_sramc_pclk is enabled, the error detection mechanism with acesss from the main bus/S2M don't work.When disabled, it works,but the current consumption will increase.
- In the case of *[M2MSTATUS]* = START state,the error detection of all is possible without depending on the pmu.*[DCG_PM_0]*.DCG_mpierclk_sramc_pclk.

Please refer to the following table.

Table 4.1 pmu.*[DCG_PM_0]*.DCG_mpierclk_sramc_pclk is valid

cause of error	[M2MSTATUS] at the occurrence of error	Master that caused the error	[M2MSTATUS] after error detection	[ERRSTATUS] and 「ERR*LOG] after error detection	M2M Interrupt	Error Interrupt	Status
Error1	STOP	Bus	STOP	Error1, Bus	deassert	assert	The bus access is not successful.
		S2M	STOP	Error1, S2M	deassert	assert	The transfer of S2M is not successful.
		M2M	STOP	Error1, M2M	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
	ACTIVE	Bus	STOP	Error1, Bus	deassert	assert	The bus access is not successful. The transfer of M2M is stop.
		S2M	STOP	Error1, S2M	deassert	assert	The transfer of S2M is not successful. The transfer of M2M is stop.
		M2M	STOP	Error1, M2M	deassert	assert	transfer to the state of STOP by Error.
	WFIC	Bus	WFIC	Error1, Bus	assert	assert	The bus access is not successful.
		S2M	WFIC	Error1, S2M	assert	assert	The transfer of S2M is not successful.
		M2M	WFIC	Error1, M2M	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
Error2	STOP	Bus	STOP	Error2, Bus	deassert	assert	The bus access is not successful.
		M2M	STOP	Error2, M2M	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error2 does not occur.
	ACTIVE	Bus	STOP	Error2, Bus	deassert	assert	The bus access is not successful. The transfer of M2M is stop.
		M2M	STOP	Error2, M2M	deassert	assert	transfer to the state of STOP by Error.
	WFIC	Bus	WFIC	Error2, Bus	assert	assert	The bus access is not successful.
		M2M	WFIC	Error2, M2M	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error2 does not occur.
Error3	STOP	M2M	STOP	Error3, —	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error3 does not occur.
	ACTIVE	M2M	STOP	Error3, —	deassert	assert	Error4 is detected, the transfer of M2M is stopped
	WFIC	M2M	WFIC	Error3, —	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
Error4	STOP	M2M	STOP	Error4, —	deassert	assert	transfer to the state of ACTIVE by Error.
	ACTIVE	M2M	STOP	Error4, —	deassert	deassert	Error4 is detected when [M2MSTATUS] is transited from STOP to ACTIVE, so does not occur.
	WFIC	M2M	WFIC	Error4, —	assert	deassert	Since [M2MSTATUS] is not STOP, error4 does not occur.

*WFIC: WAIT for Interrupt Clear

Table 4.2 pmu.*[DCG_PM_0]*.DCG_mpierclk_sramc_pclk is not valid

cause of error	[M2MSTATUS] at the occurrence of error	Master that caused the error	[M2MSTATUS] after error detection	[ERRSTATUS] and [ERR*LOG] after error detection	M2M Interrupt	Error Interrupt	Status
Error1	STOP	Bus	STOP	not recorded	deassert	deassert	The bus access is not successful.
		S2M	STOP	not recorded	deassert	deassert	The transfer of S2M is not successful.
		M2M	STOP	Error1, M2M	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
	ACTIVE	Bus	STOP	Error1, Bus	deassert	assert	The bus access is not successful. The transfer of M2M is stop.
		S2M	STOP	Error1, S2M	deassert	assert	The transfer of S2M is not successful. The transfer of M2M is stop.
		M2M	STOP	Error1, M2M	deassert	assert	transfer to the state of STOP by Error.
	WFIC	Bus	WFIC	not recorded	assert	deassert	The bus access is not successful.
		S2M	WFIC	not recorded	assert	deassert	The transfer of S2M is not successful.
		M2M	WFIC	Error1, M2M	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
Error2	STOP	Bus	STOP	not recorded	deassert	deassert	The bus access is not successful.
		M2M	STOP	Error2, M2M	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error2 does not occur.
	ACTIVE	Bus	STOP	Error2, Bus	deassert	assert	The bus access is not successful. The transfer of M2M is stop.
		M2M	STOP	Error2, M2M	deassert	assert	transfer to the state of STOP by Error.
	WFIC	Bus	WFIC	not recorded	assert	deassert	The bus access is not successful.
		M2M	WFIC	Error2, M2M	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error2 does not occur.
Error3	STOP	M2M	STOP	Error3, —	deassert	deassert	Since [M2MSTATUS] is not ACTIVE, Error3 does not occur.
	ACTIVE	M2M	STOP	Error3, —	deassert	assert	Error4 is detected, the transfer of M2M is stopped
	WFIC	M2M	WFIC	Error3, —	assert	deassert	Since [M2MSTATUS] is not ACTIVE, Error1 does not occur.
Error4	STOP	M2M	STOP	Error4, —	deassert	assert	transfer to the state of ACTIVE by Error.
	ACTIVE	M2M	STOP	Error4, —	deassert	deassert	Error4 is detected when [M2MSTATUS] is transited from STOP to ACTIVE, so does not occur.
	WFIC	M2M	WFIC	Error4, —	assert	deassert	Since [M2MSTATUS] is not STOP, error4 does not occur.

4.9.1. Interrupt Control

- When ***[ERRINTEN]***.EN is written to 1, the interrupt is enabled.
- When an error is detected, the cause of the error should be confirmed by reading ***[ERRSTATUS]***. The cause is cleared by writing 1 to ***[ERRCLR]***.CLRn (n = 0 - 3).

5. Details of Registers

5.1. M2MSTART

M2MSTART			
Description	M2M activation signal		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0000		
Physical address View0	0x4002 2000		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:1	Reserved	-	-	-
0	START	1: Activation 0: Invalid * "0" is always returned when data is read	RW oneToSet	0

5.2. M2MSTOP

M2MSTOP			
Description	M2M stop		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0004		
Physical address View0	0x4002 2004		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:1	Reserved	-	-	-
0	STOP	1: Stop 0: Invalid * "0" is always returned when data is read	RW oneToSet	0

5.3. M2MSTATUS

M2MSTATUS			
Description	Status		
Address Region	sramc	**Type:**	RO
Offset	0x0000 0008		
Physical address View0	0x4002 2008		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:18	Reserved	-	-	-
17:2	SIZE	Size of data being currently transferred (amount of data that have been transferred so far)	RO	0x0000
1:0	STATUS	0b10: Waits for interrupt clearing 0b01: In operation 0b00: Under stop	RO	0x0

5.4. M2MSRCADDR

M2MSRCADDR			
Description	Source address		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0010		
Physical address View0	0x4002 2010		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:2	ADDR	Address from which to read data Set in units of words	RW modify	0x0000 0000
1:0	Reserved	-	-	-

5.5. M2MDSTADDR

M2MDSTADDR			
Description	Destination address		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0020		
Physical address View0	0x4002 2020		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:2	ADDR	Address to which to write data Set in units of words	RW modify	0x0000 0000
1:0	Reserved	-	-	-

5.6. M2MSIZE

M2MSIZE			
Description	Transfer size		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0024		
Physical address View0	0x4002 2024		
Physical address View1	-		

Bitfield Details

Bits	Name	Description	Access	Reset
31:18	Reserved	-	-	-
17:2	SIZE	Transfer size settings. The source and destination have the same transfer size. Set in units of words.	RW modify	0x0000
1:0	Reserved	-	-	-

5.7. M2MINTEN

M2MINTEN				
Description	Interrupt function ON/OFF			
Address Region	sramc	**Type:**	RW	
Offset	0x0000 0030			
Physical address View0	0x4002 2030			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	EN	1: Interrupt function ON 0: Interrupt function OFF	RW modify	0

5.8. M2MINTCLR

M2MINTCLR				
Description	Interrupt clearing			
Address Region	sramc	**Type:**	RW	
Offset	0x0000 0034			
Physical address View0	0x4002 2034			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	CLR	1: Activation 0: Invalid * "0" is always returned when data is read.	RW oneToClear	0

5.9. ERRINTEN

ERRINTEN				
Description	Error interrupt output function ON/OFF			
Address Region	sramc	**Type:**	RW	
Offset	0x0000 0040			
Physical address View0	0x4002 2040			
Physical address View1	-			
Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:1	Reserved	-	-	-
0	EN	1: Error interrupt output function ON 0: Error interrupt output function OFF	RW modify	0

5.10. ERRSTATUS

ERRSTATUS			
Description	Error cause		
Address Region	sramc	**Type:**	RO
Offset	0x0000 0044		
Physical address View0	0x4002 2044		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:4	Reserved	-	-	-
3:0	STATUS	Error cause indication [3]=1:Error4 occurs [2]=1:Error3 occurs [1]=1:Error2 occurs [0]=1:Error1 occurs	RO	0x0

5.11. ERR1LOG

ERR1LOG			
Description	Error1 cause		
Address Region	sramc	**Type:**	RO
Offset	0x0000 0048		
Physical address View0	0x4002 2048		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:10	ADDR	Error1 occurrence access globus address (in units of 1kB)	RO	0x00 0000
9:3	Reserved	-	-	-
2:0	STATUS	Error1 occurrence cause I/F 0b101:S2M 0b100:M2M 0b010:Bus(AHB Port2) 0b001:Bus(AHB Port1) 0b000:Bus(AHB Port0)	RO	0x0

5.12. ERR2LOG

ERR2LOG			
Description	Error2 cause		
Address Region	sramc	**Type:**	RO
Offset	0x0000 004C		
Physical address View0	0x4002 204C		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:10	ADDR	Error2 occurrence access globus address (in units of 1kB)	RO	0x00 0000
9:3	Reserved	-	-	-
2:0	STATUS	Error2 occurrence cause I/F 0b100:M2M 0b010:Bus(AHB Port2) 0b001:Bus(AHB Port1) 0b000:Bus(AHB Port0)	RO	0x0

5.13. ERRCLR

ERRCLR			
Description	Error cause detection clearing		
Address Region	sramc	**Type:**	RW
Offset	0x0000 0050		
Physical address View0	0x4002 2050		
Physical address View1	-		

Bitfield Details				
Bits	**Name**	**Description**	**Access**	**Reset**
31:4	Reserved	-	-	-
3:0	CLR	[3]: Clears detection of Error4 cause [2]: Clears detection of Error3 cause [1]: Clears detection of Error2 cause [0]: Clears detection of Error1 cause * "0" is always returned when data is read.	RW oneToClear	0x0

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
0.1	2014-03-31	Newly released
0.2	2014-04-08	Corrected some errors.
0.3	2014-10-07	4.2. Start-up and Stop Procedure Revised for rev.2.0.
0.4	2014-12-01	4.2.2. Stop Procedure Corrected CD_MAIN to CD_MPIER.??? 4.9. Error Detection and Notification Added information.
1.0	2015-01-22	Official version
1.1	2018-01-23	Changed header, footer and the last page. Changed corporate name. Modified Table in section 1.

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.**