32-bit RISC Microcontroller
Reference Manual

Secure Access Memory (SAM-A)

Revision 1.0

2026-01

Toshiba Electronic Devices & Storage Corporation

Contents

List of Figures

List of Tables

Preface

Related Document

文書名
Clock Control and Operation Mode
Exception
Security Function
Flash Memory

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 bits 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.

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

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

MPU Memory Protection Unit
FPB Flash Patch and Breakpoint Unit
DWT Data Watchpoint and Trace Unit

1. Outlines

SAM is a security function that monitors memory access from the bus managers.
When using SAM, it is recommended to also enable the Chip Protect function to block illegal access from outside.
Refer to the reference manual "Security Function" for the Chip Protect function.

Function classification	Function	Operation explanation
Monitoring memory access	Bus manager to be monitored	Monitor access from the following bus managers: - CPU - DMAC
	Memory to be accessed	Monitor access to the following memories: - Flash memory - RAM
	Access control by state	SAM has the following three states, and only the area defined for each state can be accessed. - Boot state - Lib state - App state
	Occurrence of reset	Any access that violates the definition will cause a SAM reset.
Starup mode	Restriction of startup mode	Prohibits starting in the Single boot mode.

2. Function and Operation

2.1. States

SAM has three states, each with user-defined accessible areas. If the bus managers access an area that is not allowed in the current state, a SAM reset occurs.

If the bus manager is the CPU, the definition of each state applies.

If the bus manager is the DMAC, the App state definition is applied in all states.

- Boot state
  This is the state in which the startup program is executed.
  A reset puts SAM in the Boot state.
  The area definition for each state is set in the Boot state.
  In the Boot state, non-maskable interrupt (NMI) is treated as SAM reset.
  Executing a branch instruction to the App state area causes a transition to the App state.

- App state
  State in which the application program is executed.
  Executing a branch instruction to a specific address (JMPLADD) in the Lib state area causes a transition to the Lib state.

- Lib state
  The Lib state is temporarily isolated from the App state.
  Part of the Lib state area is shared with the Boot state. Transitioning to the Lib state allows access to data shared with the Boot state in isolation from the application program.
  In the Lib state, non-maskable interrupt (NMI) is treated as SAM reset.
  Executing a branch instruction to the App state area causes a transition to the App state.

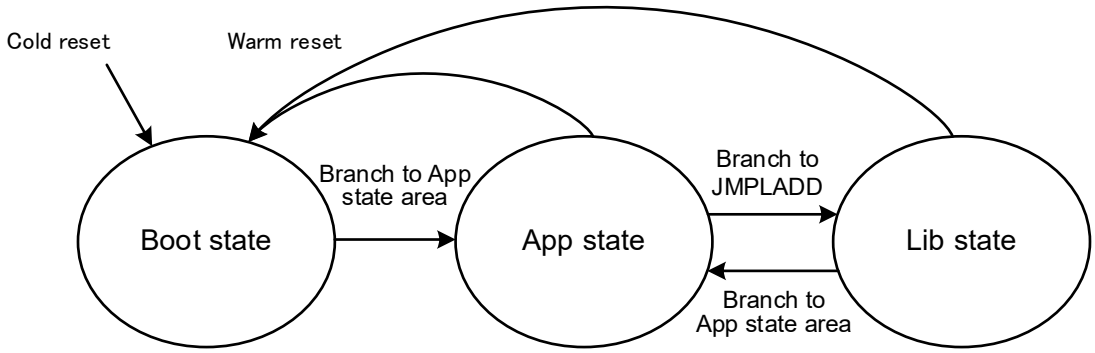

Figure 2.1 State Transitions

2.2. Area Definition

Define accessible areas for each state in the Flash memory and RAM, respectively. The Flash memory can be divided into SAM code area and SAM data area, and each area can be configured. In addition to the area definition, define the address (JMPLADD) at which to branch when transitioning from the App state to the Lib state.

The areas are placed consecutively in the order shown in Figure 2.2. Therefore, the size of the area is determined by the end address.

The start address of the Flash memory is 0x00000000 and the start address of RAM is 0x20000000. For the end address of each memory, refer to "Memory Map" in the reference manual "Clock Control and Operation Mode".

In the SAM code area of the Flash memory, the end address of CVP (CVPEND), the end address of CBP (CBPEND), and the end address of CLP (CLPEND) are set to define the Vector table area (CVP), Boot state area (CBP), and Lib state area (CLP). The remaining area is the App state area (CAP). Also, JMPLADD is set to transition from the App state to the Lib state.

In the SAM data area of flash memory, the start address (DBPSTART) and end address (DBPEND) of DBP and end address (DLPEND) of DLP are set to define the Boot state area (DBP), and Lib state area (DLP). The remaining area is the App state area (DAP).

The mirror area of the Flash memory has the same area definition with the start address as 0x5E000000.

In RAM, the end address (SBPEND) of SBP and the end address (SLPEND) of SLP are set to define the Boot state area (SBP), and Lib state area (SLP). The remaining area is the App state area (SAP).

For bit-band alias region, the definition is the same as the address of the corresponding bit-band region.

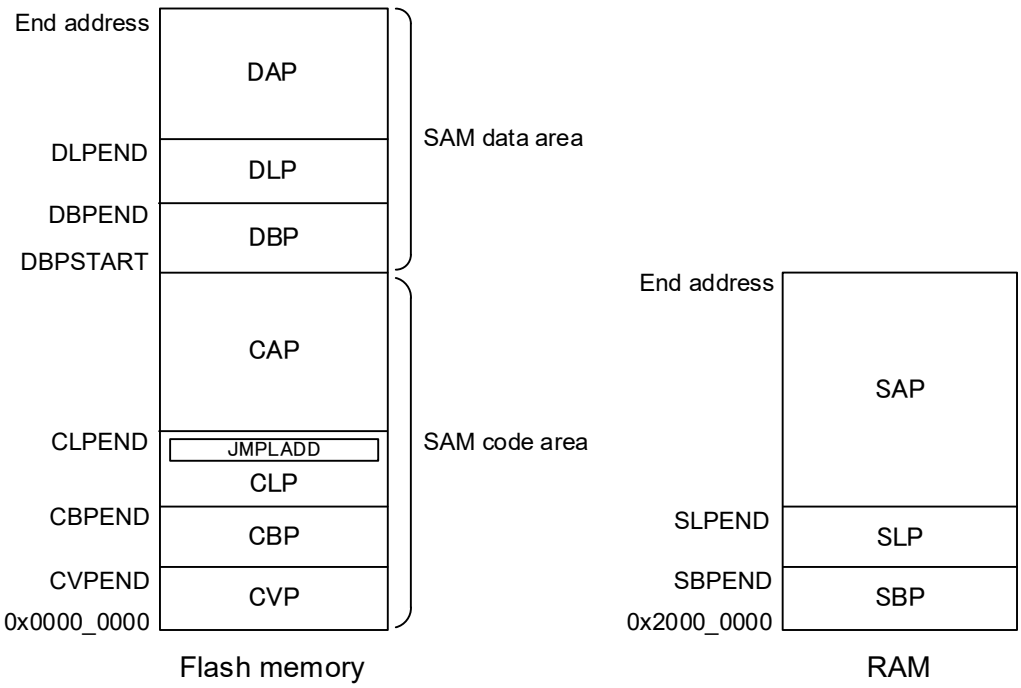

Figure 2.2 Definition of Areas

2.3. States and Accessible Areas

Table 2.1 and Table 2.2 for each state whether access to each area is allowed from the CPU and DMAC when SAM is enabled.

The meanings of the notations in the table are as follows:

R: Data read
W: Data write or bus write cycles by Flash commands
E: Instruction execution
✓: Accessible
- : Occurrence of SAM reset
(1): Transition to the App state
(2): Transition to the Lib state by executing branch instruction to JMPLADD
Otherwise, occurrence of SAM reset

Table 2.1 Accessible Areas (CPU)

Area	Boot state			App state			Lib state		
	R	W	E	R	W	E	R	W	E
CAP	✓	✓	(1)	✓	-	✓	✓	✓	(1)
CLP	✓	✓	✓	-	-	(2)	✓	✓	✓
CBP	✓	✓	✓	-	-	-	-	-	-
CVP	✓	✓	✓	✓	-	-	✓	-	-
DAP	✓	✓	-	✓	-	-	✓	✓	-
DLP	✓	✓	-	-	-	-	✓	✓	-
DBP	✓	✓	-	-	-	-	-	-	-
SAP	-	-	-	✓	✓	✓	✓	✓	(1)
SLP	-	-	-	-	-	-	✓	✓	✓
SBP	✓	✓	✓	-	-	-	-	-	-

Table 2.2 Accessible Areas (DMAC)

Area	Boot state		App state		Lib state	
	R	W	R	W	R	W
CAP	✓	-	✓	-	✓	-
CLP	-	-	-	-	-	-
CBP	-	-	-	-	-	-
CVP	✓	-	✓	-	✓	-
DAP	✓	-	✓	-	✓	-
DLP	-	-	-	-	-	-
DBP	-	-	-	-	-	-
SAP	✓	✓	✓	✓	✓	✓
SLP	-	-	-	-	-	-
SBP	-	-	-	-	-	-

2.4. Setting Method

2.4.1. SAM Permission and CBPEND Setting

SAM permission and CBPEND are written to a nonvolatile area of the Flash memory, so the values are retained. Setting is done by the Flash command Automatic SAMNRFP Programming and releasing is done by Automatic SAMNRFP Releasing. Refer to the reference manual "Flash Memory" for the Flash commands.

After the command is executed, a occurrence of a reset loads the SAM permission state and CBPEND to ***[SAMPROTECT]***<SAMPROECT> and ***[SAMCBPEND]***<CBPEND> and the product starts operation in the Boot state. Thereafter, the retained values are loaded each time a reset occurs.

2.4.2. Address Settings Other than CBPEND

The following procedure is performed to set the end address other than the CBP area, the start address of the SAM data area, and JMPLADD.

(a) Set "0x87" to ***[SAMPROTECT]***<SAMREGKEY>.
(b) Set a value to the target address register.

When changing settings, be sure that the change does not affect the area currently under executing. Also, do not access the changed area immediately after the change operation.

After setting, set ***[SAMPROTECT]***<SAMREGKEY> to a value other than "0x87" to prevent incorrect writing.

The registers of the end addresses other than the CBP area, the start address of the SAM data area, and the JMPLADD registers are initialized by a reset, so they need to be set at each reset.

2.4.3. Configurable Address Registers by State

In the Boot state, all address registers can be set.

In the Lib state, only ***[SAMDLPEND]*** and ***[SAMSLPEND]*** can be set.

In the App state, all address registers cannot be set.

2.4.4. Lock Setting

The Automatic SAMNRFP Releasing command is disabled by the lock setting of the Automatic SAMNRFP Programming command. This makes it impossible to disable SAM and the Non-releasable Flash Protect.

Refer to the reference manual "Security Function" for the Non-releasable Flash Protect.

2.5. Restrictions when SAM Is Enabled

When SAM is enabled, the following functions are restricted.

(1) Single Boot Mode
If SAM is enabled, the Single Boot Mode does not start.
In addition, if the Chip Protect is also enabled, the Forced CPROTCON Release is performed by the Single Boot Mode startup operation.
Refer to the reference manual "Flash Memory" for the Single Boot Mode and "2.6 Forced CPROTCON Release" for the Forced CPROTCON Release.

(2) Flash Writer Mode
If SAM is enabled and the Chip Protect is also enabled, the Flash Writer Mode does not start.

(3) Chip Protect Mask Register (***[FCCPROMR]***)
The Chip Protect Mask Register (***[FCCPROMR]***) cannot be accessed in the App state, so the Chip Protect cannot be temporarily released.
Refer to the reference manual "Flash Memory" for the ***[FCCPROMR]***.

2.6. Forced CPROTCON Release

When the Chip Protect and SAM are enabled and the Non-releasable Flash Protect is disabled, the Forced CPROTCON release is executed by Single Boot Mode startup operation (reset by RESET_N pin with BOOT_N input "Low").

The Forced CPROTCON Release erases all data in the Flash memory, releases the Chip Protect bit, and releases the TOSHIBA Test Mode Control. Refer to the "Electrical Characteristics" in the "Datasheet" for the time required for erasing.

After the erase time has elapsed, turn off the power supply before performing the next operation.

3. Registers

3.1. Registers List

The control registers and the addresses are as follows.

Peripheral function		Channel/unit	Base address
			TYPE1
Secure Access Memory	SAM	-	0x40043400

Register name		Address (Base+)
SAM Protect Register	***[SAMPROTECT]***	0x0000
CBP Area End Address Register	***[SAMCBPEND]***	0x0004
CLP Area End Address Register	***[SAMCLPEND]***	0x0008
CVP Area End Address Register	***[SAMCVPEND]***	0x000C
DBP Area Start Address Register	***[SAMDBPSTART]***	0x0010
DBP Area End Address Register	***[SAMDBPEND]***	0x0014
DLP Area End Address Register	***[SAMDLPEND]***	0x0018
SBP Area End Address Register	***[SAMSBPEND]***	0x001C
SLP Area End Address Register	***[SAMSLPEND]***	0x0020
Lib State Transition Address Register	***[SAMJMPLADD]***	0x0024
SAM State Register	***[SAMSTATE]***	0x0028

3.2. Detail of Registers

3.2.1. *[SAMPROTECT]* (SAM Protect Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as "0".
15:8	SAMREGKEY[7:0]	0x00	R/W	Key code for writing registers 0x87: Target register can be rewritten. Other than 0x87: Target register cannot be rewritten. Target registers are as follows. ***[SAMCLPEND]***, ***[SAMCVPEND]***, ***[SAMDBPSTART]***, ***[SAMDBPEND]***, ***[SAMDLPEND]***, ***[SAMSBPEND]***, ***[SAMSLPEND]***, ***[SAMJMPLADD]***
7:1	-	0	R	Read as "0".
0	SAMPROTECT	(Note)	R	The status of SAM 0: Disabled 1: Enabled Note: The status of SAM is loaded by a reset.

3.2.2. *[SAMCBPEND]* (CBP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	CBPENDU[9:0]	(Note)	R	CBP area end address (upper) The upper 10 bits of the CBP area end address Note: The value is loaded by a reset.
7:0	CBPENDL[7:0]	0xFF	R	CBP area end address (lower) The lower 8 bits of the CBP area end address, which is fixed at “0xFF”.

3.2.3. *[SAMCLPEND]* (CLP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	CLPENDU[9:0]	0x000	R/W	CLP area end address (upper) Sets the upper 10 bits of the end address in the CLP area. When the value is “0x000”, there is no CLP area. It is possible to write in the Boot state.
7:0	CLPENDL[7:0]	0xFF	R	CLP area end address (lower) The lower 8 bits of the CLP area end address, which is fixed at “0xFF”.

3.2.4. *[SAMCVPEND]* (CVP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	CVPENDU[9:0]	0x000	R/W	CVP area end address (upper) Sets the upper 10 bits of the end address in the CVP area. When the value is “0x000”, there is no CVP area. It is possible to write in the Boot state.
7:0	CVPENDL[7:0]	0xFF	R	CVP area end address (lower) The lower 8 bits of the CVP area end address, which is fixed at “0xFF”.

3.2.5. *[SAMDBPSTART]* (DBP Area Start Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	DBPSTARTU[9:0]	0x000	R/W	DBP area start address (upper) Sets the upper 10 bits of the start address in the DBP area. When the value is “0x000”, there is no SAM data area. It is possible to write in the Boot state.
7:0	DBPSTARTL[7:0]	0x00	R	DBP area start address (lower) The lower 8 bits of the DBP area start address, which is fixed at "0x00".

3.2.6. *[SAMDBPEND]* (DBP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	DBPENDU[9:0]	0x000	R/W	DBP area end address (upper) Sets the upper 10 bits of the end address in the DBP area. When the value is “0x000”, there is no DBP area. It is possible to write in the Boot state.
7:0	DBPENDL[7:0]	0xFF	R	DBP area end address (lower) The lower 8 bits of the DBP area end address, which is fixed at “0xFF”.

3.2.7. *[SAMDLPEND]* (DLP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:8	DLPENDU[9:0]	0x000	R/W	DLP area end address (upper) Sets the upper 10 bits of the end address in the DLP area. When the value is “0x000”, there is no DLP area. It is possible to write in the Boot state and the Lib state.
7:0	DLPENDL[7:0]	0xFF	R	DLP area end address (lower) The lower 8 bits of the DLP area end address, which is fixed at “0xFF”.

3.2.8. *[SAMSBPEND]* (SBP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as "0".
15:8	SBPENDU[7:0]	0x00	R/W	SBP area end address (upper) Sets the upper 8 bits of the end address in the SBP area. When the value is “0x00”, there is no SBP area. It is possible to write in the Boot state.
7:0	SBPENDL[7:0]	0xFF	R	SBP area end address (lower) The lower 8 bits of the SBP area end address, which is fixed at “0xFF”.

3.2.9. *[SAMSLPEND]* (SLP Area End Address Register)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as "0".
15:8	SLPENDU[7:0]	0x00	R/W	SLP area end address (upper) Sets the upper 8 bits of the end address in the SLP area. When the value is “0x00”, there is no SLP area. It is possible to write in the Boot state and the Lib state.
7:0	SLPENDL[7:0]	0xFF	R	SLP area end address (lower) The lower 8 bits of the SBP area end address, which is fixed at “0xFF”.

3.2.10. *[SAMJMPLADD]* (Lib State Transition Address Register)

Bit	Bit symbol	After reset	Type	Function
31:18	-	0	R	Read as "0".
17:2	JMPLADDU[15:0]	0x0000	R/W	Lib state transition address (upper) Sets the upper 16 bits of the JMPLADD. Sets an address within the CLP area. When the value is “0x0000”, JMPLADD is invalid. It is possible to write in the Boot state.
1:0	JMPLADDL[1:0]	00	R	Lib state transition address (lower) The lower 2 bits of the JMPLADD, which is fixed at "00".

3.2.11. *[SAMSTATE]* (SAM State Register)

Bit	Bit symbol	After reset	Type	Function
31:3	-	0	R	Read as "0".
2	SAMSTL	0	R	Lib state flag 0: Not in Lib state 1: In Lib state
1	SAMSTA	0	R	App state flag 0: Not in App state 1: In App state
0	SAMSTB	1	R	Boot state flag 0: Not in Boot state 1: In Boot state

4. Example of Usage

Here is an example of operation using SAM.

- Boot state

On reset release, CBPEND is loaded to set the CBP area and boot state program is started. The RAM is all set to the SAP area.

- Disabling of FPB
  To prevent illegal operations by the FPB, a function of the CPU, it is recommended that all RAM be set to the SAP area and disable FPB.

- Disabling exceptions/interrupts
  When an exception or interrupt occurs, the CPU registers are stacked in RAM. To prevent this content from becoming a security risk, exceptions and interrupts should be disabled.
  It is recommended that any CPU functions that may cause exceptions, such as the MPU, the SysTick, the FPB, and the DWT, be disabled.

- Initialization of RAM
  Initialize all RAM areas after setting them as SBP areas, considering the possibility of illegal FPB patches being placed.
  After initialization, clear the CPU registers.

- Checking Flash memory data
  Performs an integrity check of all Flash memory areas.

- Settings of areas
  Set the respective addresses for the App state and Lib state area settings.
  When using the Lib state, also set JMPLADD.

- Execution of Boot program
  Performs the processing necessary for boot-up.
  After execution, erase used data as necessary.

- Branch to App state area
  A transition to the App state is made by executing a branch instruction to the App state area.

- App state

- Enabling exceptions/interrupts
  Vector tables can be placed in the CVP area. It can also be moved to the CAP area by rewriting the Vector Table Offset Register.

- Execution of App program
  Execute the application program.

- Transition to Lib state
  A transition to the Lib state is made by executing a branch instruction to the JMPLADD.

- Lib state

  – Disabling exceptions/interrupts
    It is recommended that exceptions/interrupts be disabled as well as Boot state.
    It is recommended that any CPU functions that may cause exceptions, such as the MPU, the SysTick, the FPB, and the DWT, be disabled.

  – Setting of areas
    The DLP and the SLP area can be configured as needed.

  – Execution of Lib program
    It can handle data shared with the Boot state while being isolated from App state.

  – Return to the App state
    Performs the following processing as necessary and transitions to the App state by executing a branch instruction to the App state area.

    ▪ Erase used data.
    ▪ Restore settings of the DLP and the SLP.

5. Revision History

Table 5.1 Revision History

Revision	Date	Description
1.0	2026-01-30	- First release

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, lifesaving and/or life supporting 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, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- 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.**