TOSHIBA
Leading Innovation

# TX49/H4 Core Product Specification Update Rev. 1.6

TOSHIBA CORPORATION
Semiconductor Company

# Preface

Thank you for new or continued patronage of TOSHIBA semiconductor products.

Toshiba’s microprocessor and microcontroller offerings include discrete components suitable for a wide range of commercial and industrial systems.

This manual provides the usage considerations for both the TX49/H4 processor core of the 64-bit TX System RISC TX49 family and the applications incorporating the TX49/H4 core.

Document No. BFE0018E
January 2011

# Contents

**TX49/H4 Core Product Specification Update**

# Chapter 1 Introduction

## 1.1 Product Groups

This document covers the devices listed below:

1. Processor core
   - TX49/H4 core
2. General-purpose products
   - TMPR4955CFG-400/-450 (TX4955C)
   - TMPR4956CXBG-400/-450 (TX4956C)
   - TX4939XBG-400 (TX4939)

## 1.2 Related Documents

Documents related to this manual are listed below.
The contents of this document will be incorporated in the next release of these documents.

| Document Title | Document No. | Revision Date |
|---|---|---|
| 64-bit TX System RISC TX49/H2, H3, H4, W4 Core Architecture Rev. 2.1 | BDE0095C | Jan. 2007 |
| 64-bit TX System RISC TX49 Family TMPR4955B/TMPR4955C Rev. 1.1 | BDE0096A | Sep. 2006 |
| 64-bit TX System RISC TX49 Family TMPR4956C Rev. 1.1 | BDE0106B | Sep. 2006 |
| 64-bit TX System RISC TX49 Family TX4939 Rev. 3.3 | BDE0137C | May 2007 |

## 1.3 Usage Considerations (List of serial numbers and the corresponding products)

The following table lists shows applicable usage considerations for each product.
(X: Applies, -: Does not apply)

| Product Name (Product No.) / Serial Number | TX49/H4 Core (newest version) | TMPR4955CFG -400/-450 (TX4955C) | TMPR4956CXBG -400/-450 (TX4956C) | TX4939XBG -400 (TX4939) | |
|---|---|---|---|---|---|
| ERT-TX49H4-001 | X | X | X | X | |
| ERT-TX49H4-002 | – | – | X | – | |
| ERT-TX49H4-003 | X | X | X | X | |
| ERT-TX49H4-004 | X | X | X | X | |
| ERT-TX49H4-005 | X | X | X | X | |
| ERT-TX49H4-006 | X | X | X | X | |
| ERT-TX49H4-007 | X | X | X | X | |
| ERT-TX49H4-008 | X | X | X | X | |
| ERT-TX49H4-009 | – | – | – | X | |
| ERT-TX49H4-010 | X | – | – | X | |
| ERT-TX49H4-011 | – | – | – | X | |
| ERT-TX49H4-012 | – | – | – | X | |
| ERT-TX49H4-013 | – | – | – | X | |
| ERT-TX49H4-014 | – | – | – | X | |
| ERT-TX49H4-015 | – | – | – | X | |
| ERT-TX49H4-016 | – | – | – | X | |
| ERT-TX49H4-017 | – | – | – | X | |
| ERT-TX49H4-018 | – | – | – | X | |
| ERT-TX49H4-019 | – | – | – | X | |

## 1.4 Product Revision ID

The following table lists the Revision ID of each product.

| Product Name (Product No.) / Register | TX49/H4 Core (newest version) | TMPR4955CFG -400/-450 (TX4955C) | TMPR4956CXBG -400/-450 (TX4956C) | TX4939XBG -400 (TX4939) | |
|---|---|---|---|---|---|
| CP0 : PRId | 0x00002d40 | 0x00002d40 | 0x00002d40 | 0x00002d40 | |
| CP1 : FCR0 | 0x00002d40 | 0x00002d40 | 0x00002d40 | 0x00002d40 | |
| JTAGID | – | 0x0002C031 | 0x0002C031 | 0x20037031 | |
| REVID | – | – | – | 0x49390010 | |
| PCIID | – | – | – | 0x0184102f | |

# Chapter 2 Usage Considerations

Issue No.: ERT-TX49H4-001

| |
|---|
| Applicable Products:<br>TX49/H4 Core, TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
| Conditions:<br>When a data cycle generated by a preceding load instruction resulted in an error and an exception with a higher priority than the bus error exception of the following instruction occurs, Bus errors are no longer detected. |

**[Overview]**

When a Bus error exception (DBE) occurs during a data Read cycle generated by a preceding load instruction and an exception with a higher priority than the Bus error exception (DBE) occurs in a subsequently executed instruction, the exception of the subsequent instruction is processed first and Bus error exceptions (DBE) are no longer detected.

**[Problem Description]**

The TX49/H4 Core has a non-blocking load function. With this function, the instruction that follows the preceding load instruction is executed without stalling if it is not dependent on the preceding load instruction.

When reading the data from the preceding load instruction and a Bus Error exception (DBE) occurs and an exception (see Table 2.1 for the priority order when consecutive instructions issue multiple exceptions at the same timing) with a higher priority than the Bus Error exception (DBE) of the subsequently executed instruction occurs, the exception that the subsequent instruction issued is processed before the Bus Error exception (DBE) and Bus Error exceptions (DBE) can no longer be detected.

Table 2.1 Priority Order for Exceptions Issued at the Same Timing

| Priority Sequence (High) | | Detected PipeStage | Instruction synchronous or asynchronous |
|---|---|---|---|
| Cold Reset | | M | Async |
| Soft Reset | | M | Async |
| NMI | | M | Async |
| Bus Error (IBE) | Instruction Fetch | M | Async |
| Ov, Tr, Sys, Bp, RI, CpU, FPE | | M | Sync |
| Address Error (AdEL/AdES) | Data Access | M | Sync |
| TLB Refill (TLBL/TLBS) | Data Access | M | Sync |
| TLB Invalid (TLBL/TLBS) | Data Access | M | Sync |
| TLB Modify (TLBL/TLBS) | Data Access | M | Sync |
| **Bus Error (DBE)** | **Data Access** | **M** | **Async** |
| Interrupt | | M | Async |
| Address Error (AdEL) | Instruction Fetch | E | Sync |
| TLB Refill (TLBL) | Instruction Fetch | E | Sync |
| TLB Invalid (TLBL) | Instruction Fetch | E | Sync |

Note: Table 2.1 above differs from Table 11-3 (Priority Order when the Same Instruction Issues Multiple Exceptions at the Same Timing) on page 11-2 of the “TX49/H2, H3, H4, W4 Core Architecture”.

Bus errors occur under different conditions with each product as follows:
In the case of the TX4955C and TX4956C, in one of the following situations:

1. The SysCmd Bus received notification of “incorrect data” during Read operation from an external agent
2. A Parity error occurred in which a mismatch occurred between the SysAD Bus and the SysADC Bus during a Read operation from an external agent
3. In an R4300 type protocol, the Timeout function of the G2SConfig Register was used and a timeout was detected

There is no problem for ColdReset or SoftReset exceptions because initialization processing is performed after the exception occurs. Also, there is no problem for NMI exceptions if the process after the exception occurs is similar to the above reset process.

**[Work-around]**

1. There is no problem if error notification to the TX49/H4 Core using Bus errors is not enabled in the above Problem Content.
2. Executing a SYNC instruction immediately after the preceding load instruction allows you to avoid this problem because the next instruction will not be executed until the Load data arrives.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-002

| |
|---|
| Applicable Products:<br>TMPR4956CXBG-400/-450(TX4956C) |
| Conditions:<br>When accessing the G2SConfig Register in the 64-bit Bus Width mode in Little Endian bit order, normal access is not possible. |

**[Overview]**

When using the TX4956C in the 64-bit Bus Width mode in Little Endian bit order, it locks up when performing 64-bit (double word) access to the G2SConfig Register (0xFF10_0000).

**[Problem Description]**

The G2SConfig Register (0xFF10_0000) of the TX4956C is allocated into the upper 32 bits of the 64-bit Internal Bus (G-Bus). Therefore, when performing 64-bit (double word) access to 0xFF10_0000 in the Little Endian bit order, the internal decoder does not respond, the TX49 core enters the Stand By state, then it locks up.

**[Work-around]**

The above problem will not occur if you can access the data using one of the following methods:

1. Perform 32-bit (word) access on the G2SConfig Register
2. When performing 64-bit (double word) access on the G2SConfig Register, specify address 0xFF10_0000 when using Big Endian bit order or address 0xFF10_0004 when using Little Endian bit order.

**[Correction]**

The applicable products will not be corrected.

This problem does not apply when using the TX4955C or the TX4956C in the 32-bit Bus Width mode (BSIZE64*=1).

Issue No.: ERT-TX49H4-003

| Applicable Products:<br>TX49/H4 Core,TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
|---|
| Conditions:<br>When an instruction that affects the TLB itself in TLB-mapped address space is executed |

**[Overview]**

If an instruction that works on the TLB is executed in TLB-mapped address space, the target instruction of a branch or jump might not be executed correctly.

**[Content of Specification Cautions]**

If the mtc0 instruction in TLB-mapped address space is executed on the EntryHi, EntryLo0 or EntryLo1 register and a branch or jump instruction in the same or next cache line specifies a target address with the low-order four bits being 0x4 or 0xc, then operation of the target instruction is not guaranteed.

**[Limitation]**

Execute an instruction that affects the TLB itself (i.e., a write to EntryHi, EntryLo0 or EntryLo1) in unmapped address space, not in TLB-mapped address space.

If it is necessary to write to these registers from TLB-mapped address space, a sync instruction must be inserted as shown below.

```
        mtc0    r4,  r2         # Write into EntryLo0
        sync                    # Wait for execution of Load/Store
        beq     r0, r0, 1f      # Refill ITLB
        nop
 1:
```

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-004

| |
|---|
| Applicable Products:<br>TX49/H4 Core,TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
| Conditions:<br>When an SC or SCD instruction is followed by a special sequence of instructions |

**[Overview]**

If an SC or SCD instruction is followed by a load or cache instruction or if a register modified by an SC or SCD instruction is referenced by the next instruction, then the result of the latter instruction might be different from what is expected.

**[Problem Description]**

The success or failure of the SC(D) instruction is indicated by the contents of general register rt after execution of the instruction (1 = success, 0 = fail). Generally, the SC(D) instruction is used in combination with a branch instruction. In this case, a problem never occurs.

If the SC(D) instruction is followed by any of the instructions described below, the success or failure of the SC(D) instruction will not be forwarded to that instruction properly. Consequently, it uses stale data, causing the result to be different from what is expected.

If the SC(D) instruction is followed by a load or cache instruction with the LL bit cleared, the SC(D) instruction will not be nullified completely and might corrupt the data in the cache.

**[Conditions]**

1. Sequence of instructions that might cause a problem:

   SC or SCD
   instruction A
   instruction B

   If all of the following are true, instruction B generates an incorrect result:

   (1) Instruction B references a register modified by SC(D) and instruction A.
   (2) Instruction A does not reference the register modified by SC(D) as an operand.
   (3) SC(D) results in a DTLB miss.
   (4) The fetches of SC(D), instruction A and instruction B hit in the instruction cache, and a memory reference by instruction A hits in the data cache.

2. Sequence of instructions that might cause a problem:

   SC or SCD
   Load or cache instruction

   If all of the following are true, part of the data in the data cache will be corrupted:

   (1) The LL bit is cleared when the SC(D) instruction is executed.
   (2) The SC(D) instruction hits in the DTLB.
   (3) The SC(D) instruction hits in the Level-1 data cache.
   (4) The SC(D) instruction is followed by “a non-load instruction that accesses the same index of the Level-1 data cache,” or “a cache instruction.”

**[Work-around]**

No problem occurs if the SC(D) instruction is immediately followed by a branch instruction. Otherwise, if either of the above conditions is true, a nop or sync instruction must be inserted after the SC(D) instruction.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-005

| |
|---|
| Applicable Products:<br>TX49/H4 Core,TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
| Conditions:<br>When using an EJTAG ICE/Debugger |

**[Overview]**

The TX49 core might not be able to return to User mode from Debug mode during PC tracing using an EJTAG ICE.

**[Problem Description]**

No problem occurs when a commercially available EJTAG ICE is used. This problem might occur when an EJTAG ICE is created by the user.

If the EJTAG is put in PC Trace mode and then Debug mode is entered by applying a debug interrupt via the TDI pin, the TX49 core can not be brought back to User mode by clearing the JtagBrk bit (bit 12 in the JTAG Control register); this causes another debug interrupt to be generated.

**[Conditions]**

This problem occurs if it is attempted to return to User mode after a debug interrupt is applied to the TDI input in PC Trace mode and a 0 is written to the JtagBrk bit. However, no problem occurs if after the debug interrupt is applied, a 1 is written to the JtagBrk bit before a 0 is written to it.

**[Work-around]**

When you create an EJTAG ICE or a debugger, a 1 must be written to the JtagBrk after a debug interrupt is applied to the TDI pin. Then, a 0 must be written to the JtagBrk bit to bring the TX49 core back to User mode.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-006

| Applicable Products:<br>TX49/H4 Core, TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
|---|
| Conditions:<br>When an unimplemented FPU instruction is written |

**[Overview]**

When an unimplemented FPU instruction is executed, operation of the following instruction might not be guaranteed.

**[Specification Restrictions]**

Unimplemented instructions are not supposed to be used. The following symptoms occur if an FPU instruction that might write to the FCR31 register is followed by an unimplemented FPU instruction:

1. The E bit (bit 17) in the FCR31 register might be cleared even if the unimplemented instruction causes an E (Unimplemented) exception. (It appears as if the E exception did not occur.)
2. If the FCR31 register is written by the ctc1 instruction, the TX49 core might hang.

**[Conditions]**

Here is a sequence of instructions that might cause a problem:

FPU instruction (that might write to FCR31) (e.g., c.ngle.d, ctc1)
(integer instruction)
Unimplemented FPU instruction (e.g., .word 0x4624eb6a)

A problem might also occur if two or more integer instructions are placed between the two FPU instructions.
The number of integer instructions depends on the first FPU instruction. If it is no greater than the following, a problem might occur.

| First instruction | |
|---|---|
| div.d, sqrt.d | 28 integer instructions or less |
| div.s, sqrt.s | 14 integer instructions or less |
| mul.d | 1 or zero integer instruction |
| add.d,sub.d,cvt group,c.cond.d,ctc1 | 1 or zero integer instruction |

Note 1: The cvt instruction group includes ceil.l.fmt, ceil.w.fmt, cvt.d.fmt, cvt.l.fmt, cvt.s.fmt, cvt.w.fmt, floor.l.fmt, floor.w.fmt, round.l.fmt, round.w.fmt, trunc.l.fmt, and trunc.w.fmt.

Note 2: This problem does not occurr with mov.fmt, neg.fmt, abs.fmt, mul.s and c.cond.s.

**[Limitation]**

It is prohibited to use unimplemented FPU instructions. If you have no choice but to use unimplemented FPU instructions, place the cfc1 instruction to read FCR31 before any unimplemented FPU instruction.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-007

| |
|---|
| Applicable Products:<br>TX49/H4 Core, TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
| Conditions:<br>When store and load instructions are executed in write-through mode in special cases |

**[Overview]**

When store/load instructions are sequentially executed, an unnecessary write-back might occur.

**[Problem Description]**

When store and load instructions in the instruction cache are sequentially executed under the condition shown below, the data written by a single-write is written again by a burst-write. This is an unnecessary re-write, but the data is correct.

**[Conditions]**

Here is a sequence of instructions that might cause an unnecessary re-write:

store
load

If all of the following are true, an unnecessary re-write occurs:

a) The store instruction hits in the data cache.
b) The store and load addresses are different, and the indexes d are the same.
c) The store and load addresses are different doublewords.
d) The way hit by store and the way pointed by FIFO (i.e., the way to be replaced next) are the same.
e) The load instruction misses in the data cache.

The sequence of instructions shown below might also cause an unnecessary re-write. Only a load instruction(s) exist between store and load, “and it has no index relation with store and hits in the cache.

store
load 1 or more load, no index relation with store
load

When there are instruction(s) other than load between store and load, no unnecessary re-write occurs.

**[Work-around]**

Write-back mode does not cause this unnecessary re-write.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-008

| Applicable Products:<br>TX49/H4 Core, TMPR4955CFG-400/-450(TX4955C), TMPR4956CXBG-400/-450(TX4956C), TX4939XBG-400(TX4939) |
|---|
| Conditions:<br>When a TLB exception occurs on an instruction fetch under a special condition |

**[Overview]**

An mfc0 instruction immediately before a TLB exception caused by an instruction fetch might read incorrect data from a register under a special condition.

**[Problem Description]**

When a TLB exception occurs on an instruction fetch, the mfc0 instruction that is executed immediately before the exception might read the Status or Cause register incorrectly. Although the mfc0 instruction should read these registers before the exception, it reads the EXL bit of the Status register or the BD and ExcCode bits of the Cause register after the exception and writes incorrect data into a GPR.

**[Conditions]**

Here is a sequence of instructions that might cause a problem.

```
load/store instruction
load/store/sync instruction
MFC0      rt, rd      ; when rd is the Status or Cause register
--- page boundary ---
nop                   ; TLB-mapped area
```

If all of the following are true, incorrect data is read:

(1) The mfc0 instruction is immediately before a page boundary.
(2) The load, store or sync instruction is immediately before the mfc0 instruction.
(3) The execution of the load or store instruction immediately before (2) is not finished yet.
(4) A TLB exception occurs on an instruction fetch that crosses a page boundary in a TLB-mapped area.

**[Work-around]**

A nop instruction must be inserted before the mfc0 instruction that reads the Status or Cause register. This problem can also be avoided by masking the EXL bit in the Status register or the BD and ExcCode bits in the Cause register outside an exception handler.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-009

| Applicable Products:<br>TX4939XBG-400(TX4939) |
| --- |
| Conditions:<br>When an SIO overrun error is detected by checking the OERS bit in the Status Change Interrupt Status Register (SISCISR)<br>There are two way to detect this error. One is by checking the OERS bit by software. The other is by setting the STIE field of the DMA/Interrupt Control Register (SIDICR) to 1***** . |

**[Overview]**

Writing a 0 to the OERS bit in the Status Change Interrupt Status Register (SISCISR) does not clear it. The OERS bit can be cleared by writing a 0 to the UBRKD bit in the SISCISR register.

**[Problem Description]**

After initializing the SIO, the OERS bit is set to 1 upon the first overrun error.

1. Defect 1
   Writing a 0 to the OERS bit does not clear it. Thus, once an overrun error occurs, the state of overrun detection is not correctly reflected to the OERS bit. Consequently, subsequent overrun errors can't be detected by checking the OERS bit.

2. Defect 2
   The OERS bit is cleared by a writing 0 to the UBRKD bit in the SISCISR register.

**[Work-around]**

Use the UOER bit in the DMA/Interrupt Status Register (SIDISR) to detect overrun errors. (Even in the presence of this bug, overrun errors can be handled through the UOER bit.)

According to the SIO specification, a software reset should be performed when an overrun error occurs. A software reset can be performed by writing a 1 to the SWRST bit in the FIFO Control Register (SIFCR).

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-010

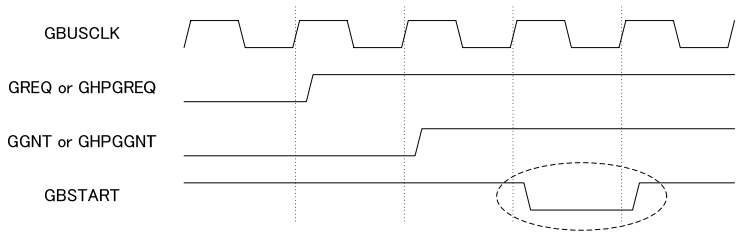
| Applicable Products:<br>TX49/H4 Core, TX4939XBG-400(TX4939) |
|---|
| Conditions:<br>When the EJTAG ICE executes the CFC0 or CTC0 instruction while an external bus master using the GREQ/GGNT or GHPGREQ/GHPGGNT signals owns the G-Bus for ET concurrency operations |

**[Overview]**

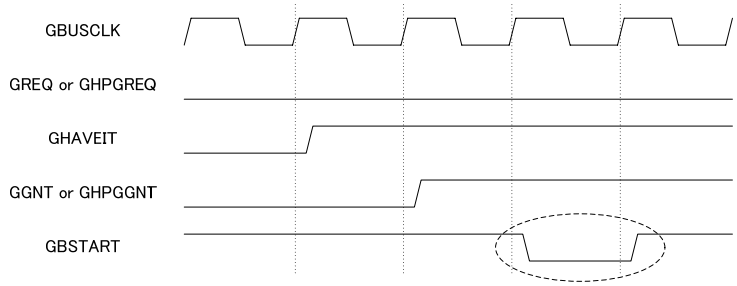
The EJTAG ICE should not execute the CFC0 and CTC0 instructions while an external bus master using the GREQ/GGNT or GHPGREQ/GHPGGNT signals owns the G-Bus for ET concurrency operations. Otherwise, the GBSTART signal is asserted improperly.

**[Problem Description]**

For dumping of a general-purpose register (GPR) by the EJTAG ICE, a coprocessor bus cycle of the TX49 core is used to transfer the GPR value to the integrated Debug Support Unit (DSU). A coprocessor bus cycle should not be requested while an external bus master assumes ownership of the G-Bus using the GREQ/GGNT or GHPGREQ/GHPGGNT signals for ET concurrency operations. Otherwise, one cycle after the TX49 core regains G-Bus ownership, GBSTART is asserted for one cycle improperly without an assertion of GRD or GWR.




This problem occurs when the external bus master gives up the G-Bus not only by deasserting the GREQ or GHPGREQ signal but also by deasserting the GHAVEIT signal.




When this problem happens, the address and data buses have values according to the protocol of the coprocessor bus cycle in progress. Therefore, the opcode of the CTC0 or CFC0 instruction (0x0_40**_****) is placed on the address bus (GAFM35-GAFM0), and the value of the GPR specified by the CTC0 instruction is placed on the data bus.

The coprocessor bus cycle initiated by the EJTAG ICE completes within the TX49 core; neither the GCPRD nor GCPWR signal is driven out to indicate a coprocessor bus cycle.

**[Conditions]**

This problem occurs when both of the following conditions are true:

1. An ET concurrency operation using the GREQ/GGNT or GHPGREQ/GHPGGNT signals is in progress.
2. The firmware in the EJTAG ICE issues a coprocessor bus cycle with the CTC0 or CFC0 instruction.

This problem does not occur in the following cases:

1. Systems in which the G-Bus is never granted to an external bus master or systems in which ST concurrency operations using the GSREQ/GSGNT or GHPSREQ/GHPSGNT signals are utilized.
2. Normally, the EJTAG ICE uses the CTC0 and CFC0 instructions to write and read a GPR. This problem can be avoided if the EJTAG ICE uses the DESAVE register in CP0 to achieve the same result.

This problem has no effect in the following cases:

1. If external bus masters on the G-Bus are designed to sample the assertion of the GRD or GWR signal simultaneously with the assertion of the GBSTART signal to recognize the beginning of a bus cycle
2. If all external bus masters on the G-Bus are mapped to an address region outside 0x_0_40_**_**** that is always placed onto the address bus when this problem occurred

Possibilities That This Problem Will Occur for Different Design Practices of External Bus Masters

| Designing of External Bus Masters | | | | Possibility of a Problem Occurring |
|---|---|---|---|---|
| GBSTART | GRD/GWR | GAFM35-GAFM0 | Address Mapping | |
| Sampled | Sampled | – | – | No |
| Sampled | Not sampled | Decoded | Outside the 0x0_40**_**** range | No |
| Sampled | Not sampled | Decoded | In the 0x0_40**_**** range | Yes |
| Sampled | Not sampled | Not decoded | – | Yes |

Note: Each asterisk character, *, represents a hexadecimal character.

**[Work-around]**

Since coprocessor bus cycles can not be used in User mode, this problem only occurs when the EJTAG ICE uses the CTC0 or CFC0 instruction. Therefore, this problem can be avoided if it is possible to temporarily disable the ET-concurrency bus request from external bus masters when the EJTAG ICE executes these instructions.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-011

| |
|---|
| Applicable Products:<br>TX4939XBG-400 (TX4939) |
| Conditions:<br>When using the RTCTBC register (time base correction register) |

**[Overview]**

According to the TX4939 databook, the RTCTBC register is assumed to retain its contents as far as the VDDRTC power is supplied. However, the RTCTBC register is reset and loses its contents if the VDDC power is removed.

**[Problem Description]**

The RTCTBC is assumed to retain its contents as far as the VDDRTC power is supplied. However, when the VDDC power is removed, the RTCTBC register is reset and loses its contents. Without the VDDC power, the RTC counter is not properly corrected. Therefore, the RTC counter value may not indicate the correct time upon system recovery because of a displacement of the RTC counter value.

**[Work-around]**

Constantly supply the VDDC power to the system.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-012

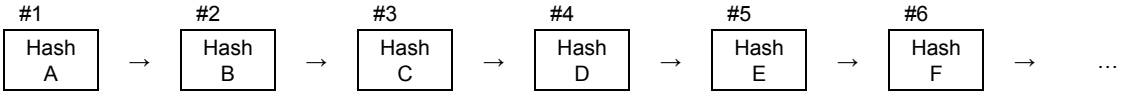
| Applicable Products:<br>TX4939XBG-400 (TX4939) |
|---|
| Conditions:<br>After performing a Hash calculation using chained descriptors in the CRYPT engine |

**[Overview]**

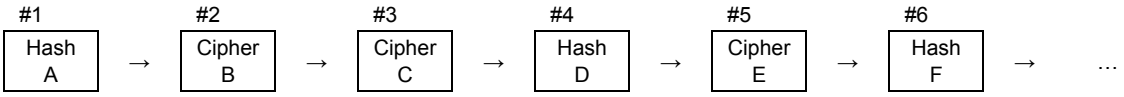
The process after a Hash calculation may not correctly proceed when using chained descriptors.

This problem is caused by chained descriptors configured as illustrated below. As shown below, each of the descriptors manipulates unique data and provides the result separately.

Example 1




Example 2




**[Problem Description]**

In the above configuration, a problem occurs at the descriptor immediately following a descriptor that performs a Hash calculation. This is because the value of the Context Index Pointer (Cptr [2:0]) is not correctly loaded into the CRYPT engine.

In Example 1, descriptors other than #1 may output wrong results upon Hash calculations. In Example 2, descriptors #2 and #5 may output wrong results upon data encryptions.

**[Work-around]**

The following conditions should be noted while using the CRYPT engine circuitry.

1. In descriptor chaining operation, a descriptor that handles a Hash calculation must not be linked to a descriptor that handles any process other than a Hash calculation.
   Configurations like examples 1 and 2 above are prohibited.
2. When a hash value of data is handled using chained descriptors, the number of descriptors that comprise the chain is limited to two.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-013

| |
|---|
| Applicable Products:<br>TX4939XBG-400 (TX4939) |
| Conditions:<br>When using the Additional Control register (ATA0/1 0x3C28/0x4C28) in the ATA100 controller |

**[Overview]**

Upon Ultra DMA reads, the ATA Host Interface Controller does not work according to the ATA/ATAPI-6 specification if bit 6 (Transfer Pause bit) of the Additional control register is set before receiving the first beat of data of a burst.

**[Problem Description]**

Pause commands must not normally be issued until the reception of the first data once the ATA Host Interface Controller responds to the DMACK* signal from the device that indicates the beginning of initialization.
However, pause commands are wrongly issued when the above condition is met.

**[Work-around]**

Do not set bit 6 of the Additional control register (Transfer Pause bit) to one.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-014

| |
|---|
| Applicable Products:<br>TX4939XBG-400(TX4939) |
| Conditions:<br>When an access from the External Bus Interface to an external device crosses a page boundary in Page mode |

**[Overview]**

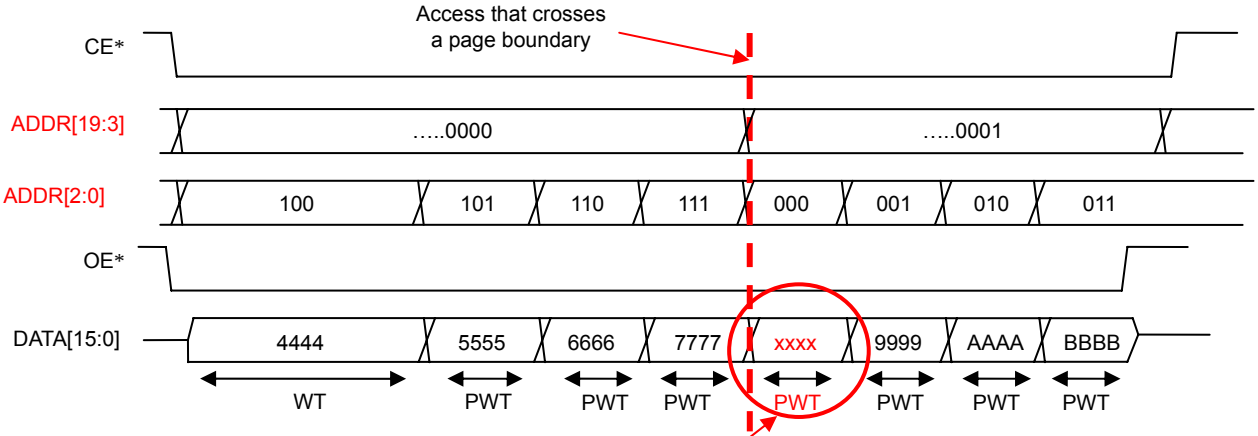
When the External Bus Interface accesses an external device in Page mode (except when EBCCRn.PM = 00) and the access crosses a page boundary, proper wait cycles are not inserted into the access cycle for the data whose address crosses a page boundary.

**[Problem Description]**

In Page mode, the wait cycle count for the first data of a page is determined by the EBCCRn.WT value. The wait cycle count for the subsequent data is determined by the EBCCRn.PWT value. However, if an access crosses a page boundary, the wait cycle count for the data that crosses the boundary is not determined by EBCCRn.WT, but by EBCCRn.PWT.

If the External Bus Controller (EBUSC) is set to Page mode, the EBUSC operates in Page mode when a bus master (the CPU, DMA Controller or PCI Controller) performs a burst access to the External Bus. The number of pages can be selected from 4, 8 and 16. The EBUSC accesses the data of the specified page size, automatically incrementing the starting address. The EBUSC repeats this operation till the accessed data size matches the burst size.

Therefore, depending on the starting address, the wait cycle count of the access that crosses a page boundary might be determined by the EBCCRn.PWT value.




Though an access to the NOR Flash crosses a page boundary, the wait cycles are not properly inserted as programmed in EBCCRn.WT. (The wait cycles are inserted as programmed in EBCCRn.PWT.) Therefore, the access cycle does not meet the required NOR Flash access time, and the first data of a new page cannot be read properly.

Note: Signals shown in the above figure are those of the TX4937.

**[Work-around]**

1. Usage Restrictions on the TX49 Core
   When the TX49 core is programmed for critical-word-first transfers, the External Bus Controller must not be configured to any of the following.

   (1) EBCCRn.BSZ = 1 (8-bit bus width) and EBCCRn.PM = 11 (16-page mode)
   (2) EBCCRn.BSZ = 0 (16-bit bus width) and EBCCRn.PM = 10 (8-page mode)

   TX49 core always performs four-beat burst accesses with the address of each beat aligned on an 8-byte boundary. Therefore, a problem occurs when the page size is 16 bytes, that is, when any of the above three settings is selected.

2. Usage Restrictions on the PCI Controller
   When the PCI controller is the bus master and performs a burst access to the External Bus (that is, when the initiator on the PCI Bus performs a PCI target access to the External Bus through Memory Space n (n = 0, 1, 2)), the External Bus Controller must not be configured to any of the following.

   (1) EBCCRn.BSZ = 1 (8-bit bus width) and EBCCRn.PM = 11 (16-page mode)
   (2) EBCCRn.BSZ = 0 (16-bit bus width) and EBCCRn.PM = 10 (8-page mode)
   (3) EBCCRn.BSZ = 0 (16-bit bus width) and EBCCRn.PM = 11 (16-page mode)

3. Usage Restrictions on the DMA Controller
   When using the DMA controller to perform a burst transfer to the External Bus, the External Bus Controller must be configured to one of the following. These restrictions can be ignored when no burst transfer is performed to the External Bus.

   (1) EBCCRn.BSZ = 1 (8-bit bus width) and EBCCRn.PM = 11 (16-page mode),
   (2) EBCCRn.BSZ = 0 (16-bit bus width) and EBCCRn.PM = 10 (8-page mode)

   When any of the above settings is selected, DMA registers must also be set as follows:
   - i) Set the DMA Source Address Register (DMSARn) to an address on a 16-byte boundary
   - ii) Set the DMA Destination Address Register (DMDARn) to an address on a 16-byte boundary.
   - iii) Set the DMA Chained Address Register (DMCHARn) to an address on a 16-byte boundary.
   - iv) Set the transfer byte count to a multiple of a doubleword. (DMACNTRn[2:0] = 000)

   (3) EBCCRn.BSZ = 0 (16-bit bus width) and EBCCRn.PM = 11 (16-page mode)

   When any of the above settings is selected, DMA registers must also be set as follows:
   - i) Set the DMA Source Address Register (DMSARn) to an address on a 32-byte boundary.
   - ii) Set the DMA Destination Address Register (DMDARn) to an address on a 32-byte boundary.
   - iii) Set the DMA Chained Address Register (DMCHARn) to an address on a 32-byte boundary.
   - iv) Set the transfer byte count to a multiple of a doubleword. (DMACNTRn[2:0] = 000)

**[Status]**

There is no plan to fix this problem.

Issue No.: ERT-TX49H4-015

| Applicable Products:<br>TX4939XBG-400 (TX4939) |
|---|
| Conditions:<br>When the DMA controller performs a dual-address burst transfer with Address register value not on 8-byte boundary and with Count register value of other than a multiple of 8 bytes. |

**[Overview]**

There are restrictions for the DMA controller transfers performed with Address register value not on 8-byte boundary and with Count register value of other than a multiple of 8 bytes, which are not described in the data sheet.

**[Problem Description]**

If the DMA controller carries out a dual-address burst transfer with Address register value not on 8-byte boundary and with Count register value of other than a multiple of 8 bytes, the following conditions must be satisfied.

1. The DMCCRn.USEXFSZ value must be 0.
2. One DMA controller can perform only one dual-address burst transfer.

Example: When a channel of the DMAC0 controller is used to for a dual-address burst transfer, other channels of the DMAC0 controller are never allowed to carry out any dual-address burst transfer. (However, the DMAC0 controller is allowed to carry out dual-address single transfers and single-address transfers using its channels other than the one being used for a dual-address burst transfer.)
The DMAC1 has no special restriction.
(The DMAC0 and DMAC1 are the separate controllers.)

Channel Registers Programming Restriction for Dual-Address Burst Transfers

| Programmed transfer size (DMCCRn.XFSZ) | DMSARn[2:0] | | DMDARn[2:0] | | DMSAIRn | DMDAIRn | DMCNTRn | DMCCRn. REVBYTE | DMCCRn. USEXFSZ |
|---|---|---|---|---|---|---|---|---|---|
| | If the DMSAIRn value is programmed more than 0 | If the DMSAIRn value is programmed a negative number | If the DMDAIRn value is programmed more than 0 | If the DMDAIRn value is programmed a negative number | | | | | |
| 4/8 Double Word (DMMCR.FIFUM [n]=1) | 000 | 111 | 000 | 111 | 8/0/-8 | 8/-8 ‡ | 000 | 0/1 | 0/1 |
| | *** | – | *** | – | 8 | 8 | *** | 0 | 0 |
| | – | *** | – | *** | -8 | -8 | | 0 | 0 |

‡: 8, 0, or -8 can be specified when the Destination Burst Inhibit bit (DMCCRn.DBINH) is set.

**[Work-around]**

No workaround. It is one of the DMA controller specifications.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-016

| Applicable Products:<br>TX4939XBG-400 (TX4939) |
|---|
| Conditions:<br>When the DMA controller performs a dual-address burst transfer with Address register value not on 8-byte boundary and with Count register value of other than a multiple of 8 bytes. |

**[Overview]**

Where the DMA controller carries out a dual-address burst transfer with Address register value not on 8-byte boundary and with Count register value of other than a multiple of 8 bytes, the DMA controller will not operate normally if the register values are programmed to satisfy the following specific conditions.

**[Problem Description]**

If all the following conditions are met, the DMA controller malfunctions.

1. If DMCNTRn is more than 0x100 and DMCNTRn[7:0] is in the range of 0xfa to 0xff.

2. If the addition of DMSARn[2:0] and DMCNTRn[2:0] is more than 9.

3. - If DMSARn[4:0] is less than 0x17 where DMCCRn.XFSZ is four double words and DMSAIRn and DMDAIRn is 8.
   - If DMSARn[5:0] is less than 0x37 where DMCCRn.XFSZ is eight double words and DMSAIRn and DMDAIRn is 8.
   - If DMSARn[4:0] is more than 0x08 where DMCCRn.XFSZ is four double words and DMSAIRn and DMDAIRn is -8.
   - If DMSARn[5:0] is more than 0x08 where DMCCRn.XFSZ is eight double words and DMSAIRn and DMDAIRn is -8.

Table for Condition 2.

| | | DMSARn[2:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| DMCNTRn[2:0] | 000 | – | – | – | – | – | – | – | – |
| | 001 | – | – | – | – | – | – | – | – |
| | 010 | – | – | – | – | – | – | – | NG |
| | 011 | – | – | – | – | – | – | NG | NG |
| | 100 | – | – | – | – | – | NG | NG | NG |
| | 101 | – | – | – | – | NG | NG | NG | NG |
| | 110 | – | – | – | NG | NG | NG | NG | NG |
| | 111 | – | – | NG | NG | NG | NG | NG | NG |

Table for Condition 3.

| DMCCRn.XFSZ | DMSAIRn/DMDAIRn | |
|---|---|---|
| | 8 | -8 |
| 4DW | DMSARn[4:0] ≤ 0x17 | DMSARn[4:0] ≥ 0x08 |
| 8DW | DMSARn[5:0] ≤ 0x37 | DMSARn[5:0] ≥ 0x08 |

**[Work-around]**

Divide the DMA transfer so as not to satisfy any error condition.

Example: If any error condition is satisfied when the Count register set value is 0x2ff

- Divide the DMA transfer into two sizes of 0x2f7 and 0x8.
- The two transfers will be carried out; the size 0x2f7 is transferred by the DMA and the size 0x8 by the CPU.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-017

| Applicable Products:<br>TX4939XBG-400 (TX4939) |
|---|
| Conditions:<br>Using Open-Drain output of the SIO (UART) while break detection for serial communication is available |

**[Overview]**

If Open-Drain output of the SIO (UART) is used, the applicable products can not send the break correctly. So the other party can not detect the correct status.

**[Conditions]**

Following conditions must apply at the same time:

- Break detection is used for serial communication
- Open Drain enable bit (UODE: bit13) in the line control register (SILCR) is set to 1'b1 (Open Drain output)

Open Drain is used for the following cases:

1. Multi controller system
2. Output signal(s) is pulled-up in single controller system

**[Work-around]**

Open Drain enable bit (UDOE) must be disabled at the time of transmitting break.

[Start of break]

```
[SIFLCR].TBRK = 1'b1        ; enable break transmittance
[SILCR].UODE = 1'b0         ; disable Open Drain
```

[End of break]

```
[SILCR].UODE = 1'b1         ; enable Open Drain
[SIFLCR].TBRK = 1'b0        ; disable break transmittance
```

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-018

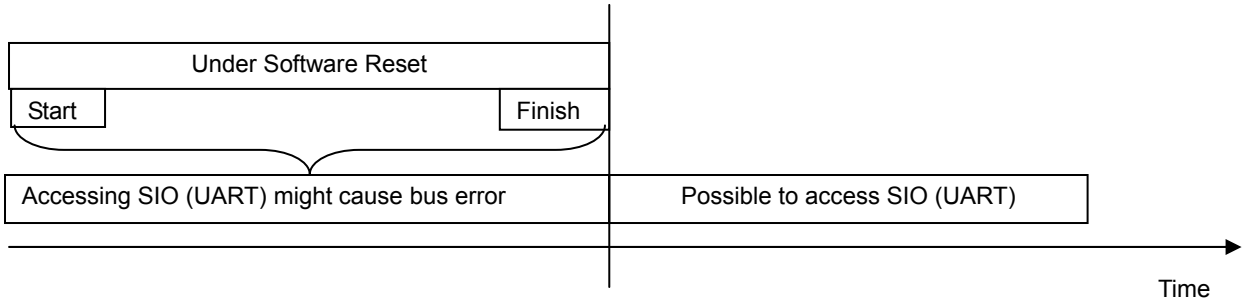
| Applicable Products:<br>TX4939XBG-400 (TX4939) |
|---|
| Conditions:<br>Executing software reset with Software Reset bit (SWRST) in the FIFO control register (SIFCR) of the SIO (UART) |

**[Overview]**

SIO (UART) can be initialized (Software Reset) by writing 1'b1 into the SWRST (bit 15) of the FIFO control register (SIFCR). If any register of SIO is accessed before completion of Software Reset, bus cycle is hung up and bus error is caused. A polling SWRST causes the same problem.

**[Problem Description]**

When Software Reset is executed by writing 1'b1 into the SWRST (bit 15) of the FIFO control register (SIFCR), it might cause bus hung up and bus error.




**[Work-around]**

The following sequence of actions must be performed for Software Reset and accessing SIO (UART).

1. Write 1’b1 to [SIFCR].SWRST
2. Access SIO (UART) register(s) after elapsing six system clock cycles or more

Any way to make interval of six system clocks or more is available to avoid this problem. e.g. accessing any register except SIO, memory, etc.

**[Status]**

The applicable products will not be corrected.

Issue No.: ERT-TX49H4-019

| |
|---|
| Applicable Products:<br>TX4939XBG-400 (TX4939) |
| Conditions:<br>Using CTS of the SIO (UART) |

**[Overview]**

During the serial data transmission in the particular condition, the 1 byte data, which is 0x80 and is not written in the shift register, may be mixed in the transmission data.

**[Conditions]**

Case 1. The data, which will be transmitted in the next transmission phase, is set in the shift register of the data transmission and the CTS is asserted in the time of transmitting the STOP bit.

Case 2. After the setting the data in the shift register for data transmission, the CTS is asserted.

**[Work-around]**

Set the FIFO control register “SIFCR.TDIL[4:3]” as 2'b00. By setting such, the data transmission FIFO will be full by writing 1byte data. After the 1byte data transmission, the data transmission FIFO will be empty.

Or

After confirming the CTS signal assertion, set the transmission data to the data transmission FIFO.

**[Status]**

The applicable products will not be corrected.

# Appendix A Revision History

| Date | Revision | Revision History |
|---|---|---|
| 09/08/2004 | Rev.1.0 | – |
| 12/26/2005 | Rev.1.1 | – |
| 01/19/2007 | Rev.1.2 | – |
| 03/01/2007 | Rev.1.3 | Added ERT-TX49H4-011, ERT-TX49H4-012 and ERT-TX49H4-013. |
| 09/06/2007 | Rev.1.4 | Added ERT-TX49H4-014 |
| 06/19/2008 | Rev.1.5 | Added ERT-TX49H4-015 and ERT-TX49H4-016. |
| 01/31/2011 | Rev.1.6 | Added ERT-TX49H4-017, ERT-TX49H4-018 and ERT-TX49H4-019. |

## RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively “TOSHIBA”), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively “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 intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment 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 or serious public impact (“Unintended Use”). 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. Do not use Product for Unintended Use unless specifically permitted in this document.
- 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 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.
- Product is subject to foreign exchange and foreign trade control laws.
- 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.