TMPR3927 Specification Update

Revision History

11-Jly-2001	rev1.0	Initial Release
29-Oct-2001	rev1.1	Add ERT-TX3927-009, ERT-TX3927-010
		Delete condition 3 of ERT-TX3927-008
		Add contents of Documentation Changes
05-Feb-2002	rev1.2	Add ERT-TX3927-011 - ERT-TX3927-014
		Add TMPR3927CF version
		Delete Documentation Changes No. (44150D-9909)
		Add Documentation Changes No. (44150D-0112)
08-Jul-2002	rev1.3	Add ERT-TX3927-015, ERT-TX3927-017
		Modify Product Type Notes of ERT-TX3927-009
		Add contents of Documentation Changes
20-Mar-2003	rev1.4	Add ERT-TX3927-018
		Add contents of Documentation Changes

These contents will be incorporated in the next release of the documents.

Product Type: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Related Documents: TX39Family TMPR3927 DataBook (2002):Doc.No= 44150D-0112 TX39/H2Core Architecture (2000):Doc.No= 44124D-9908

The information contained herein is subject to change without notice.

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others.

The products described in this document contain components made in the United States and subject to export control of the U.S. authorities. Diversion contrary to the U.S. law is prohibited.

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.
It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications.
Also, please keep in mind the precautions and conditions set forth in the “Handling Guide for Semiconductor Devices,” or “TOSHIBA Semiconductor Reliability Handbook” etc..

The Toshiba products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).
These Toshiba products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of Toshiba products listed in this document shall be made at the customer’s own risk.

The products described in this document may include products subject to the foreign exchange and foreign trade laws.

ERT-TX3927-001

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : After opening moisture-proof packing

■Effective usage period after opening moisture-proof packing

Please store the devices in a cool, dry area with a temperature of 30°C or less and a humidity of 60% or less. Please be sure to solder the devices within 48 hours after opening its moisture-proof packing.

In case the devices are stored for more than 48 hours after opening moisture-proof packing, then please bake the devices at 125°C for more than 20 hours before soldering them. After baking, please store the devices in a cool, dry area with a temperature of 30°C or less and a humidity of 60% or less and be sure to solder the devices within 48 hours.

ERT-TX3927-002

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Electrostatic Discharge Testing Results

■Electrostatic Discharge Testing Results

The following table shows the results of Electrostatic Discharge testing that were performed on this device. When handling individual devices (which are not yet mounted on a printed circuit board), be sure that the environment is protected against electrostatic electricity

Please refer to “General Safety Precautions and Usage Considerations” section of the data book for more information.

Standard	Pin(s)	Rated Voltage
Machine Model (0[ohm],200[pF],1time)	RXD[1:0], CTS[1:0]	200 V
	Other pins	250 V or more
Human Body Model (1.5[kohm],100[pF],3times) (MIL standard)	All pins	2000 V or more

ERT-TX3927-003

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note) This is mentioned from the databook in 2002 edition (Doc.No= 44150D-0112).
Application : Using TLB

■Recommended Operating Conditions of Supply Voltage

The following table shows the recommended operating conditions of supply voltage for TMPR3927F/AF/BF/CF. Please make sure that the supply voltage is different when TLB ON or TLB OFF. When designing products that include this device, please ensure that the recommended operating conditions for the devices are always adhered to.

Parameter		Symbol	Condition	Min.	Max.	Unit
Supply voltage	I/O	V_{DDS}		3.0	3.6	V
	Internal logics	V_{DD2}	TLB OFF	2.3	2.7	V
			TLB ON	2.4	2.7	

ERT-TX3927-004

Product Type : TMPR3927F, TMPR3927AF, (TMPR3927BF, TMPR3927CF) Note)This is modified by TMPR3927BF/CF, however, there is some usage limitation.
Application : Using PCI controller in the target mode

(Outline)

Under particular conditions, the PCI controller in the TX3927 may assert unnecessary STOP* signals.

(Phenomenon/Conditions)

As illustrated in Figure 1 below, the PCI controller in the TX3927 may, under particular conditions, assert unnecessary STOP* signals when the external PCI master is performing burst access to the TX3927 local memory, exactly when the burst cycle ends (a. in Figure 1).

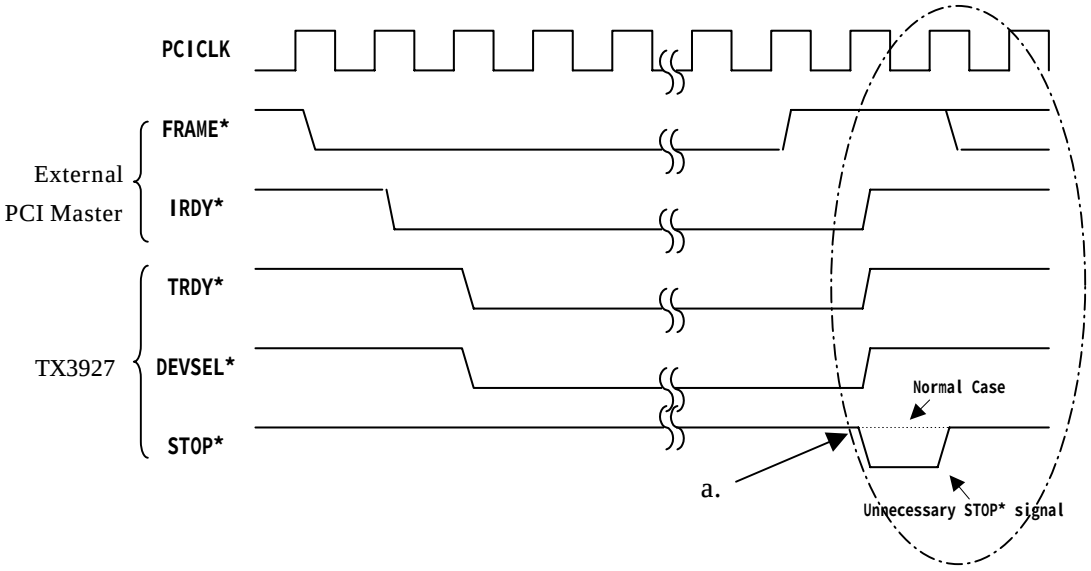


Figure 1 Unnecessary STOP* signal assertion operation

This phenomenon occurs under the following conditions:

(1) When the burst cycle is in the Read state
If OFIFO becomes empty just before timing a. (in the above figure) when the target control register (TC : address 0xFFFE_D090) in the PCI controller has set OFIFO8 Clock Rule Enable (OF8E : bit 3) to "Disable."

(2) When the burst cycle is in the Write state
(a) If IFIFO becomes full just before timing a. (in the above figure) when the target control register (TC) in the PCI controller has set IFIFO8 Clock Rule Enable (IF8E : bit 4) to

"Disable."

(b) If the CPU core requests the bus at timing a. (in the above figure) while an other bus master of an internal bus is executing a bus cycle and PSNP (PCI snoop : bit 11) of the chip configuration register (CCFG) has been set to "Disable."

(Work-around)

(1) When the burst cycle is in the Read state
Set OF8E of the target control register (TC) in the PCI controller to "Enable."

(2) When the burst cycle is in the Write state
Set IF8E of the target control register (TC) in the PCI controller to "Enable," and also set PSNP of the chip configuration register (CCFG) to "Enable." In this situation however, it is prohibited to use the Data cache in the Write-back mode.

(Status)

This is modified at TMPR3927BF as follows.

(1) It is possible to set PSNP of Chip Configuration Register to "disable" so as to use the Data cache in the Write-back mode.

(2) There is still the usage limitation to set OF8E and IF8E of the target control register in the PCI controller to "Enable" whenever it is possible to execute PCI burst cycles.

ERT-TX3927-005

Product Type : TMPR3927F, TMPR3927AF Note) This is modified at TMPR3927BF/CF,however, there are some usage limitations remained.
Application : Assertion of the RESET signal during operation.

(Outline)

If the RESET signal is asserted under particular conditions, the output of SDCLK, SYSCLK, and PCICLK may be set in the Hi-Z state.

(Phenomenon)

During RESET sequence, the state of ADDR[4],[5] and [18] are used to set outputs of SDCLK,SYSCLK and PCICLK to enable or not. If setting to disable, these CLK turn into the Hi-Z state. The TX3927 usually selects from either on-chip pull-up resistance or external pull-down resistance. Under the influence of these function, if the TX3927 outputs "0" to each of the above ADDR pins at the timing of RESET signal assertion, then SDCLK, SYSCLK, and PCICLK turn in the Hi-Z state until these ADDR pins are set to "1" by the pull-up resistance.

It violates the SDRAM specification that the SDCLK output of the TX3927 is in the Hi-Z state. Therefore, if this phenomenon is caused by the RESET* signal assertion while SDRAM access, SDRAM may turn into unexpected state.

(Work-around)

When the RESET* signal inputs to the TX3927, please design to reset all devices which use the SDCLK, SYSCLK and PCICLK. As regards SDRAM, please turn off the SDRAM power supply to return to the initial state (power-on state) because it has no reset pin.

(Status)

This is modified at TMPR3927BF as follows.

(1) In the RESET state, SDCLK and SYCLK output are always enabled, and the initial settings for SDCLK and SYCLK by the ADDR pins are omitted.

(2) Retain the function to disable SDCLK and SYSCLK output by settings in the pin configuration register after boot up. These pins will turn into Hi-Z signals after the clock output is disabled.

(3) Specification pertaining to PCICLK is not modified.

ERT-TX3927-006

Product Type : TMPR3927F, TMPR3927AF Note) This malfunction is on TX39/H2 core whose PRID is 0x0000_2240.
Application : Using TLB

(Outline)

Program behavior may change if the Branch Likely instruction is executed under particular conditions when using the TLB.

(Phenomenon/Conditions)

The branch likely instruction nullifies the instruction in its delay slot in the case that the branch condition is false. But under the conditions described below, the instruction in its delay slot of the branch likely instruction is executed and the program behavior is changed.

The conditions that cause this phenomenon are as follows.

(1) TLB is used for instruction.
(2) The last two instructions on the page boundary are a branch likely instruction and an instruction for its delay slot.
(3) Above branch condition is false.
(4) INT exception or DINT exception occurs at the delay slot. (Note 1)
(5) The next instruction of the delay slot causes ITLB miss. (Note 2)

Only when above all conditions meet at the same time, the address of the delay slot instead of the branch likely instruction is stored into EPC(or DEPC) register and BD(or DBD) bit in the Cause(or Debug) Register is not set. Therefore after returning from the exception handler the instruction in the delay slot is executed instead of being nullified.

(Note 1)
Both NMI and BUSERR cause the same problem. But NMI is the imprecise exception and the return of exception is not guaranteed. And BUSERR is the fatal error and it is impossible to recover from this exception.
(Note 2)
TX39/49 Core has 2-entry instruction TLB (ITLB) like a cache memory. The miss of ITLB does not cause TLB exception, since it is refilled from actual TLB by hardware.

(Workaround)

We would like to show you its workaround as follows.

(1) In the exception handler check whether EPC(or DEPC) Register is on a page boundary or not.
(2) If it is on a page boundary, check whether the instruction followed by the instruction pointed to by EPC(or DEPC) is a branch likely instruction or not.
(3) If it is a branch likely instruction, decrement the EPC(or DEPC) value by four.

```
<INT procedure>
   if ((EPC & 0xffc == 0xffc) &&
                ((* (unsigned long *) (EPC - 4) & 0xf0000000 == 0x50000000) ||/*1*/
                  (* (unsigned long *) (EPC - 4) & 0xfc0e0000 == 0x04020000) ||/*2*/
                  (* (unsigned long *) (EPC - 4) & 0xf3fe0000 == 0x41020000))) /*3*/
                       EPC -= 4;

   /*1*/ --> beql, bnel, blezl, bgtzl
   /*2*/ --> bltzl, bgezl, bltzall, bgezall
   /*3*/ --> bczfl, bcztl
```

In the workaround above, we assume no branch/jump instruction jump into the delay slot of the branch likely instruction.

And please refer to the example program that shows the concrete example for workaround.

(Status)
This is modified at TMPR3927BF.

(Example program)

Example of patch procedure related Branch Likely operation

[Detail Code of the patch]

```
---------------------------------------------------------------------------------------------------------------
#define BADADDR_OFF             0xffc

        mfc0    a0, EPC_SAVE_AREA               // Save EPC -> a0 (Note)

        li      a1, BADADDR_OFF
        and     a2, a0, a1
        bne     a2, a1, patch_exit
        nop

        lw      a1, -4(a0)

        li      a2, 0xf0000000
        li      a3, 0x50000000
        and     a2, a1, a2
        beq     a2, a3, err_intr
        nop

        li      a2, 0xfc0e0000
        li      a3, 0x04020000
        and     a2, a1, a2
        beq     a2, a3, err_intr
        nop

        li      a2, 0xf3fe0000          // li a2, 0xf01e0000
        li      a3, 0x41020000          // li 0x40020000

        and     a2, a1, a2
        beq     a2, a3, err_intr
        nop

        j       patch_exit
        nop

err_intr:
        addiu   a0, a0, -4
        sw      a0, EPC_SAVE_AREA

patch_exit:
```

<Note>
Please modify the address of EPC_SAVE_AREA and the register number for the applied OS.

ERT-TX3927-007

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using PCI controller

(Outline)

Under particular conditions, the bus error exception process that occurs during the execution of "PCI Configuration Read", doesn't work correctly.

(Phenomenon)

Under particular conditions, the bus error exception process that occurs during the execution of "PCI Configuration Read", doesn't work correctly. The bus error will occur when the TOE bit (Timeout Enable for Bus Error: bit14) of the CCFG register (Chip Configuration Register: address=0xFFFE_E000) is set to "1".

(Conditions)

If the following (1), (2), (3) conditions are existed, this error will occur.

(1) Timeout error is enabled. It means that TOE bit of the CCFG register is set to "1".

(2) The "PCI Configuration Read" is executed in the direct mode. It means that it is executed by using ICDR (Initiator Configuration Data Register: address=0xFFFE_D13C) and ICAR (Initiator Configuration Address Register: address=0xFFFE_D138).
There is no problem regarding PCI Configuration Write.

(3) Bus error (Timeout error) occurs. It will be occurred on condition as follows.

a) The target PCI device repeats the "Retry", because its initialization is incomplete or it is out of order, etc.

b) PCI bus turns into deadlock, because TX3927 and target PCI device repeats the "Retry" each other.

c) The TX3927 couldn't receive acknowledge from the target PCI device within 512 G-bus clock because PCI bus traffic is crowded.

Only when above all conditions (1)-(3) meet at the same time, this phenomenon occurs.

(Work-around)

There are two work-around ways described below. Please take one of them for work-around.

(1) Execute "PCI Configuration Read" in the indirect mode.

The indirect mode can be executed by using IPCIADDR, IPCIDAT, IPCICBE and ISTAT.

In the direct mode, "PCI Configuration Read" cycle for PCI bus is caused when CPU reads ICDR register. The TX3927 local bus cycle doesn't finish until the PCI bus cycle finishes.

On the other hand, in the indirect mode, PCI cycles start asynchronously to the TX3927 local bus. For read cycle, address and command are set into IPCIADDR and IPCICBE registers .The value set in IPCIADDR register will be put on PCIAD bus directly during the address phase and the value set in ICMD and IBE of IPCICBE register will be executed as PCI command and byte enable. Then CPU polls a status bit of ISTAT register (or interrupt). The PCIC performs PCI cycle and puts the data into the IPCIDATA register and sets the status bit. In this mode, the local bus cycle finishes without waiting a reply from target PCI device. Therefore bus error by timeout error doesn't occur.

As regards sample program for the indirect mode, please refer to the example program.

(2) Disable the timeout error, that is to set TOE bit of CCFG register to disable.

In this case, the bus error by timeout will never occur. But TX3927 has a possibility of deadlock when the local bus is waiting for acknowledge from the target device. For instance, repeating RETRY by the target PCI device would cause deadlock.

To avoid deadlock, system should be designed to assert reset when it detects something wrong with itself such as deadlock by WDT (WatchDog Timer) etc.

(Example program)

```
---------------------------------------------------------------------------------------------------------
/* This sample program is suitable for Type 0 configration cycle                    */

#define WAITTIME 0x1000

void dummyloop(void){
        int i;
        for( i=0; i< WAITTIME;i++);
}

unsigned int
indirect_config_read( unsigned int dev, unsigned int func, unsigned int reg)
{
        /* dev : target device number : 0x00 -- 0x14( AD[11] -- AD[31] )    */
        /* func : target device function number : 0x0 -- 0x7                        */
        /* reg : terget device configration space address offset
                                        : 0x00 -- 0x3f                          */
        unsigned int address;    /* ad[31:0] during the address phase              */
        unsigned int read_data;  /* the value of configration read data    */


        /* ISTAT register IDICC bit == 1 , write clear */
        if( *(unsigned int *)(0xfffed044) & 0x00001000   ){
                        *(unsigned int *)(0xfffed044) = 0x00001000;
        }

        /* make address value */
        address = 0x00000000 |((0x1) << (11 + (dev & 0x1f)))|((func & 0x7) << 8) | ((reg & 0x3f)<<2);
        *(unsigned int *)(0xfffed150) = address;

        /* execute indirect configration read      */
        *(unsigned int *)(0xfffed158) = 0x000000a0;

        /* status polling configration access      */
        while(1){
                if( *(unsigned int *)(0xfffed044) & 0x00001000   ){
                        /* ISTAT register IDICC bit == 1 , indirect initiator command terminates */
                        break;
```

```
                        }
                        dummyloop();
                }

                /* read cofigration register value from internal register */
                read_data = *(unsigned int *)(0xfffed154);

                /* clear IDICC bit(ISTAT register's all bit are R/WC)    */
                *(unsigned int *)(0xfffed044) = 0x00001000;

        return read_data;
}

void indirect_config_wirte(unsigned int dev, unsigned int func, unsigned int reg, unsigned int data)
{
                unsigned int address;        /* ad[31:0] during the address phase */

                /* ISTAT register IDICC bit == 1 , write clear */
                if( *(unsigned int *)(0xfffed044) & 0x00001000    ){
                                *(unsigned int *)(0xfffed044) = 0x00001000;
                }

                /* make address value */
        address = 0x00000000 |((0x1) << (11 + (dev & 0x1f)))|((func&0x7) << 8) | ((reg & 0x3f)<<2);
        *(unsigned int *)(0xfffed150) = address;

                /* write value set internal register            */
                *(unsigned int *)(0xfffed154) = data;

        /* execute indirect configration write*/
        *(unsigned int *)(0xfffed158) = 0x000000b0;

                /* status polling configration access        */
                while(1){
                        if( *(unsigned int *)(0xfffed044) & 0x00001000    ){
                                /* ISTAT register IDICC bit == 1 , indirect initiator command terminates */
                                break;
                        }
                        dummyloop();
                }

                /* clear IDICC bit(ISTAT register's all bit are R/WC)    */
                *(unsigned int *)(0xfffed044) = 0x00001000;
}
```

ERT-TX3927-008

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using ROM controller in the half-speed mode

(Outline)

Under particular conditions, some ROM controller signals will be delayed and be shortened at the half-speed mode.

(Phenomenon)

Under particular conditions, the assert timing of ADDR, ACE*, CE*, OE* and BE* will be delayed for $t_{sysh}/2$(1/2 clock of the half speed mode reference clock) when the TX3927 ROM controller access the area of half speed mode channel.

Then the assert period is $t_{sysh}/2$ shorter than the usual because the deassert timing of these signal are the same as usual.

In the full speed mode, this phenomenon will not occur.

(Conditions)

In case that the Half speed bus bit(RHS:bit4) of the ROM channel control register (RCCR0-7:0xFFFE_9000-901C) is set to "1" in one or more channel, this problem happen when the following conditions occur simultaneously.

(1) ADDR changes at the same timing as the GBSTART* assert on TX3927 internal bus(G-bus). This occurs in one of the following situations a)-c).
   a) DMAC starts bus cycle.
   b) PCIC starts bus cycle.
   c) CPU starts bus cycle immediately after the GHPGGNT*, which is one of bus grant signal of the G-bus, is deasserted. The GHPGGNT* will be deasserted when the GHPGREQ * or the GHAVEIT* is deasserted. The GHPGREQ* is G-bus request signal without snoop function which will be used by PCIC or DMAC, and the GHAVEIT* is a signal for confirming the bus ownership of the G-bus.

(2) A bus cycle is an access to the area of half speed mode channel.

(3) The bus cycle starts at the falling edge of half-speed mode reference clock.

As a result of above conditions, ADDR, ACE*, CE*, OE* and BE * are delayed for $t_{sysh}/2$(1/2 clock of the half speed mode reference clock). As the deassertion timing is the same as usual , the assertion period is also $t_{sysh}/2$ shorter than usual.

Note) GHPGGNT*, GHPGREQ, and GHAVEIT* are internal signals of TX3927

(Work-around)

There are three work-around (1)-(3). Use the best fit your application.

(1) Use the GSREQ instead of the GHPGRE as G-bus request signal for PCI controller and DMAC. In addition to it, set PCIC and DMAC not to access the half-speed mode channel area. For this work-around, it is necessary to set some register as follows.

a) Set the PSNP bit(PCI snoop bit : bit11) of Chip Configuration Register (CCFG:0xFFFE_E000) to enable(="1").

b) Set the SNOP bit(snoop bit:bit7) of the Channel Control Register(CCRn: 0xFFFE_B018,0xFFFE_B038, 0xFFFE_B058, 0xFFFE_B078) to enable(="1").

c) Set the WBON bit(Write Back Mode ON:bit13) of the configuration register(Register number3) to Write Trough Mode(="0").

d) Set the DMAC not to access the half-speed mode channel area.

e) Set the PCIC not to access the half-speed mode channel area.

Note) The GSREQ is bus request signal with snoop , and is internal signal of TX3927.

(2) Design a system based on new AC specification.

(3) Use the ROM controller only in the full speed mode.

【Table of AC Characteristics(SDRAMC, ROMC signals)】

■Difference

The spec of Output Delay is defined respectively in the full-speed mode(td) and the half-speed mode(tdh).

In the half-speed mode, the spec of ADDR, ACE*, CE*, OE* and BE* differ from these of the other signals.

(Tc = 0 ~ 70°C, V_{DDS} = 3.3V ± 0.3V, V_{DD2} = 2.5V ± 0.2V, V_{SS} = 0V, CL = 50pF)

Parameter	Signals	Description	Min	Max	Unit
t_{sys}	SYSCLK/SDCLK[4 : 0]	Cycle Time (Full-speed bus mode)	15		ns
t_{sysh}	SYSCLK	Cycle Time (Half-speed bus mode)	30		ns
t_{sysm}	SYSCLK/SDCLK[4 : 0]	Min High/Low Level	5		ns
t_{sysmh}	SYSCLK	Min Half-Speed High/Low Level	12		ns
t_d	(1)	Output Delay(Full-speed bus mode)		7	ns
t_{dh}	**(3)**	**Output Delay(Half-speed bus mode)**		**7**	**ns**
t_{dh}	**(4)**	**Output Delay(Half-speed bus mode)**		**t_{sysh}/2 + 7**	**ns**
t_{oh}	(1)	Output Hold	1		ns
t_{su}	(2)	Input Setup	7		ns
t_{ih}	(2)	Input Hold	0		ns
t_{daz}	DATA[31 : 0], ACK*	Data Active to Hi-Z		7	ns
t_{dza}	DATA[31 : 0], ACK*	Data Hi-Z to Active	1		ns

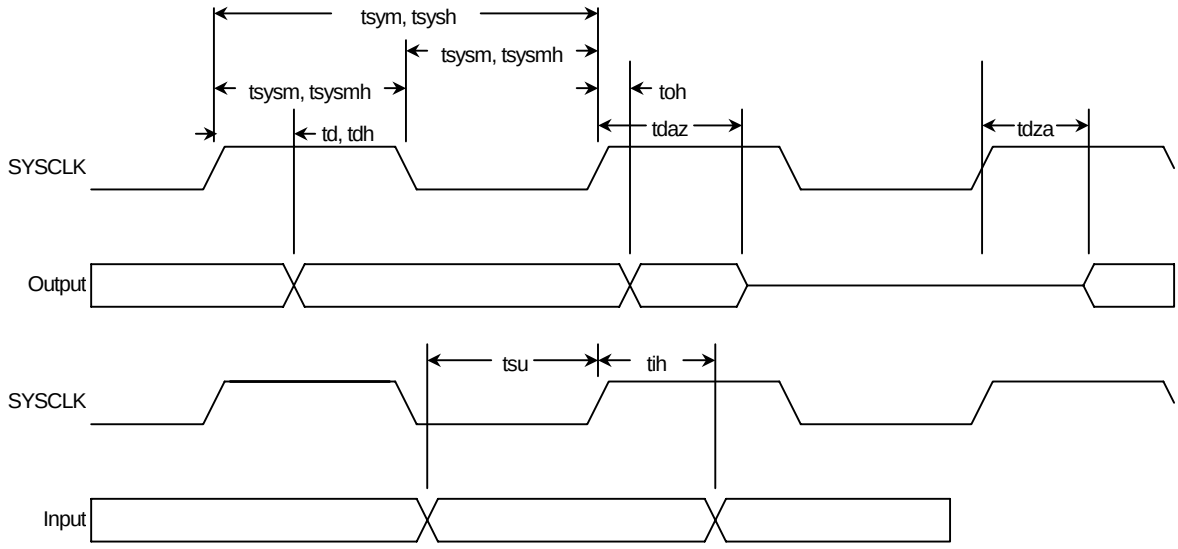
(1) ACK*, DATA[31 : 0], CE[7:0]*, OE*, ACE*, SWE*, BWE[3 : 0]*, ADDR[19 : 2], DMAACK[3 : 0], DMADONE*, PIO[15 : 0], TIMER[1 : 0]

(2) ACK*, DATA[31 : 0], NMI*, INT[5 : 0], DMAREQ[3 : 0], DMADONE*, PIO[15 : 0]

(3) ACK*, DATA[31 : 0], BWE[3:0]*,SWE*, DMAACK[3 : 0], DMADONE*, PIO[15 : 0], TIMER[1 : 0]

(4) CE[7:0]*, BE*[3:0], OE*, ACE*, ADDR[19 : 2]

【Timing Diagram (SDRAMC, ROMC signals)】



ERT-TX3927-009

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note) This malfunction is on TX39/H2 core whose PRID is 0x0000_2240, 0x0000_2241 or 0x0000_2242.
Application : Using Snoop function in Doze mode

(Outline)

In Doze mode, even if TX3927 core releases the bus possession and snoop function is used, the corresponding data that is in data cashe does not become invalid.

TX3927 becomes Doze mode by setting Doze bit (bit 9) of Config register (Cp0:r3) as 1. During Doze mode, the bus possession can be released according to a bus demand signal. If assert of snoop signal is recognized at the rising edge of a clock while having released the bus possession, the cache line containing the data whose address is equal to an address bus is invalidated by the snoop function.
By this malfunction, even if snoop function is used in Doze mode, data in data cashe does not become invalid.

(Phenomenon)

When TX3927 returns from Doze mode, right data may not be read.

(Conditions)

When snoop function is used in Doze mode, this malfunction may occur.
In modes other than Doze mode, snoop function operates normally.

(Work-around)

There are three work-around (1) – (3). Use the best fit your application.

(1) Not to use snoop function.

(2) When using snoop function, not to use Doze mode.
However, in use of UDEOS/r39, when it changes to an idle task, Doze bit (bit 9) of Config register (Cp0:r3) is automatically set to 1 by OS. Therefore, the source file (kidle.c) of OS needs to be changed so as not to use Doze mode. But, the above-mentioned evasion cannot be used in a library package. Please prepare the task of the minimum priority. (Please refer to 'The example of work-aroud in use of UDEOS/R39'.)

(3) When using snoop function in Doze mode, invalidate the cash at the Doze mode end.

(Notice on Doze mode by OS)

1	UDEOS/r39	Doze is used.
2	VxWorks	No problem(Doze is not used.)
3	WinCE	Please refer to following item 3).
4	Linux	No problem(Doze is not used.)

1) UDEOS/r39
In UDEOS/r39 V3.3.0 or earlier version, when it changes to an idle task, Doze bit (bit 9) of Config register (Cp0:r3) is automatically set to 1 by OS. Therefore, the source file (kidle.c) of OS needs to be changed so as not to use Doze mode. However, the above-mentioned work-around cannot be used in a library package. Please prepare the task of the minimum priority.

Moreover, when you use Doze mode, according to the work-around of notes, please avoid this malfunction.

2) VxWorks

VxWorks for TX3927 uses VxWorks kernel of Tornado2.0/R3000, and only cache library is replaced for TX39/H2 cores. That is, all functions are compatible with R3000 except cache library. For this reason, Doze mode is not supported by the OS, but you can use it in your responsibility. When you use Doze mode, according to the work-around of notes, please avoid this malfunction.

3) WinCE

Please inquire of your SI vender about detail of this matter. When you use Doze mode in your system, according to the work-around of notes, please avoid this malfunction.

4) Linux

Linux(Monta Vista Linux) of MontaVista Software, Inc. does not use Doze mode in kernel. When you use Doze mode, according to the work-around of notes, please avoid this malfunction.

(The example of work-aroud in use of UDEOS/R39)

The example of work-aroud in use of UDEOS/R39 indicated in this text evasion 2) is given to below.

1. Changing the source file of OS

In UDEOS/r39 V3.3.0 or earlier version, when it changes to an idle task, Doze bit of Config register is set in ./src/kidle.c.

(kidle.c) before change

```
TASK TR_Eidl(void)
{
        for(;;){
#if defined(TR_KNL_LOG) || defined(TR_MTD_LOG)
            /* record idle log */
            TR_writeLog(TR_LOG_IDL, 0, (ID)0, (ID)0, 0);
#endif
            /* power-saving mode */
            TR_lowPower();
        }
}
```

Please change source program as follows so as not to set in Doze mode and reconstruct a kernel after correction.

(kidle.c) after change

```
TASK TR_Eidl(void)
{
        for(;;);
}
```

The program for recording idle log is deleted not to filled up with an idle log in all buffers. In this case, you can also recognize if the process changes into the idle state.

2. Not changing the source file of OS

Prepare the task of a priority lower than the task of the minimum priority currently used with application. Please register the task in a configuration macro.

```
CRE_TSK(id, exinf, TA_HLNG | TA_START, task, pri, stk)
```

The content of the task is as follows.

```
TASK
task( )
{
        for(;;);
}
```

In library package, kidle.c is not attached. So the above work-around 1(changing the source file of OS) cannot be used. Please use work-around 2.

ERT-TX3927-010

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF Note) This malfunction is on TX39/H2 core whose PRID is 0x0000_2240 or 0x0000_2241.
Application : Using write-back mode

(Outline)

In the write-back mode, the internal bus (G-Bus) and external buses such as SDRAM may be locked. This phenomenon may occur when flash of data cache by CACHE instruction is operated while CPU releases the G-bus possession by non-snoop-capable bus request signal that is asserted by DMAC or PCIC.

In this malfunction, even if TOE bit (bit 14) of Chip Configuration Register (CCFG) is set "1" and a timeout for bus error is enable, a bus error does not occur. In addition, when Watchdog timer of TX3927 is used, although reset is effective for TX3927, whether reset is effective for whole system depends on your system. If reset is not effective for your whole system, you have to make power supply of your system turn off.

(Phenomenon)

G-Bus and external buses such as SDRAM may be locked.

(Conditions)

If both of the following (1) and (2) conditions are existed, this error may occur.

(1) The write-back mode is used.

(2) CACHE instruction is operated while CPU releases the G-bus possession.
CACHE instruction which cause flash of data cash is one whose operation field (bit20:16) is 0x01, 0x15 or 0x19.

In the burst write operation that is caused by cache replacement from cache miss, this malfunction doesn't occur.

(Work-around)

There are two work-around (1), (2). You can evade this malfunction by using either one method.

(1) Use the write-through mode.

(2) When using the write-back mode, don't use CACHE instruction for flash of data cache during the CPU releases the G-bus possession.

(Status)

This is modified at TMPR3927CF.

(Example of evasion)

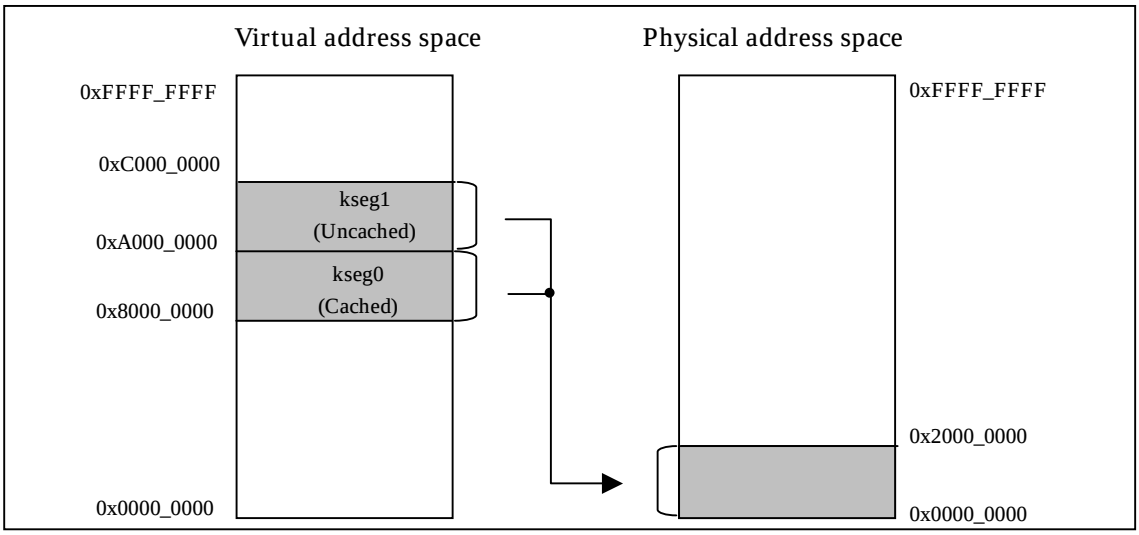
The following sentences are evasion examples of evasion plan 2 of the text. These are examples that operate flash of data cash without CACHE instruction.

Example 1

0x8000_0000~0x9FFF_FFFF and 0xA000_0000~0xBFFF_FFFF in virtual address is allocated in same physical address (0x0000_0000~0x1FFF_FFFF). 0x8000_0000~0x9FFF_FFFF is cached area, and 0xA000_0000~0xBFFF_FFFF is uncached area.

So reading data from cached area and writing the data to uncached area whose physical address is the same as

reading address is equal to flash of data cache by CACHE instruction.



For example, if data address that is target of cache flash is 0x8000_0000, please operate following 1) - 3).

1) Read out data from 0x8000_0000.
   Because 0x8000_0000~0x9FFF_FFFF is cached area, read operation is done for data cash at first. So when address 0x8000_0000 hit data cash, data is read from data cash. In case of miss hit, it is read from memory (physical address 0x0000_0000).
2) Write data that was read in phase 1) in 0xA000_0000.
   Because 0xA000_0000~0xBFFF_FFFF is uncached area, data is always wrote in memory (physical address 0x0000_0000).
3) Invalidate data cash for an address of phase 1). It is possible by setting op-field (bit20:16) of CACHE instruction with 0x11.

In other words, when data cache is hit in reading operation, cash flash (read from cash and write to memory) is done by the operation mentioned above.
But, in case of miss hit, reading and writing is done for same memory address, so the operation becomes useless.

Example 2

When there is unused cached data area more than 4K bytes in succession, automatic flash of all area of data cache is possible by reading out the area more than 4K bytes in succession.

ERT-TX3927-011

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using PCI controller in the target mode

(Outline)

When the PCI access of a certain specific condition happens, in the read cycle, the read access may be done for not an address to read but the next address that has be written right before the front.

(Phenomenon)

Under specific conditions, when the PCIC of TX3927 is read as the target, the read access may be done for wrong address.

(Conditions)

Under the following (1), (2) <Setup> conditions, this malfunction may occur when the PCI access happens in the following turn.

<Setup>

(1) When data in OFIFO is cached after the PCI transaction end.
That is to say, when OFCAD bit (bit [19]) of Target Control Register (TC) is set in "0". The value of the default is "0".

(2) When data is read from the local memory to OFIFO for one demand of PCI read, when the setup which continues to read more data into OFIFO until it is full is enable.
That is to say, when OFPFO bit (bit [12]) of TC register is set in "0". The value of the default is "0".

<The turn of the PCI access which this malfunction occurs>

(1) The PCI device reads from the local memory of TX3927.
The data inn to OFIFO is late, so the PCI read cycle ends by Retry. (TX3927 is executing data intake from the local memory to OFIFO.)

(2) The PCI device writes to the local memory of TX3927.
Actually, the transaction of PCI is finished at the moment when data is written in IFIFO.
It is made to wait for writing from IFIFO to the memory until the data intake from the local memory to OFIFO ends in above.

(3) The PCI device reads from the local memory of TX3927 again.
Because there is data in OFIFO this time, data is returned, and the PCI transaction is finished.

(4) The PCIC finishes writing from IFIFO to the local memory before the finishing timing of 3, or the PCIC is executing writing to the local memory in the finishing timing of 3.

(5) The PCI device reads data from the local memory of TX3927 whose address is the next address of the address that is read in the front.

When the timing of the above 4 occurs, the address of the local bus that is under execution by PCI now is latched to the address pointer of the local bus of OFIFO. Because above 5 is the access to the address that continues from the previous address, when the setup of the PCIC is as <Setup> conditions, the pointer of the local bus address isn't updated. So the address of the next data becomes the address which is latched in 4 (the address which has written in 4).
Because of this, data is read from the wrong address.

(Work-around)

There are two work-around (1), (2). You can evade this malfunction by using either one method.

(1) Discard unused OFIFO data after the PCI transaction is finished.
In other words, set bit [19] of TC register in "1". (default "0")

(2) For one demand of PCI read, read from the local memory to OFIFO only once. (Don't stream data.)
In other words, set bit [12] of TC register in "1". (default "0").

(Notice on use of OS)

Please refer to "**ERT-TX3927-011～ERT-TX3927-014 Notice on use of OS**" of ERT-TX3927-014.

ERT-TX3927-012

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using Never Time Out function of PCI controller

(Outline)

The PCI bus may lock when Never Time Out function of the PCIC is made enable and TBL_OFIFO bit (bit [7:4]) of Target Burst Length Register (TBL) is set in 16D word. Never Time Out function becomes enable when OFNTE bit (bit [18]), OF16E bit (bit [5]) and OF8E bit (bit [3]) of target control register (TC) are all set in "1". And, bit [7:4] of TBL register becomes set 16D word when it is set in "01XX" or "1X1X".

(Phenomenon)

If Never Time Out function of the PCIC is made enable, when the PCIC of TX3927 is accessed as the target, the PCI bus may lock.

(Conditions)

Under the following (1), (2) <Setup> conditions, this malfunction may occur in the following process.

<Setup>

(1) When Never Time Out function is enable.
That is to say, when bit [18] of TC register is set in "1". The value of the default is "0".

(2) When bit [7:4] of TBL register is set in "01XX" or "1X1X".

<The process which this malfunction occurs>

This malfunction occurs in the following process, when external bus master executes Wait by deassert IRDY at the timing of 16D word when external bus master reads 16D word data from TX3927.

(1) Because the transaction finishes at the next, TX3927 thinks that the transaction finishes at the next clock when 15D word is read, and it stops there.

(2) IRDY is set low again, and the external PCI bus master keeps waiting for the next data.

Because Never Time Out function is enable at this time, TX3927 can't finish the transaction by asserting STOP signal.

And bit [7:4] of TBL register is set in 16D word and it is same size as FIFO, the next data can't be taken in OFIFO because the final data (1D word) is left in OFIFO. So the output level (the number of words) to the PCI bus of OFIFO can't be satisfied.

Because the transaction isn't finished by the above and PCI bus master doesn't release the bus possession, the PCI bus locks.

(Work-around)

There are two work-around (1), (2). You can evade this malfunction by using either one method.

(1) Disable Never Time Out function.
In other words, set bit [18] of TC register as "0". (default "0")

(2) Set bit [7:4] of TBL register as the one except for "01XX" and "1X1X".

(Notice on use of OS)

Please refer to "**ERT-TX3927-011～ERT-TX3927-014 Notice on use of OS**" of ERT-TX3927-014.

ERT-TX3927-013

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using PCI controller in the target mode

(Outline)

The location of the PCIC address space is specified in the target memory base address (MBA) and the target I/O base address (IOBA). The last 4D words of this space are reserved for the PCIC. When OFARD bit (bit [8]) or IFARD bit (bit [7]) of Target Control register (TC) is set in "0", the specification of TX3927 is as follows.
"The PCIC will issue a target-abort if an attempt is made by an external PCI bus master to access addresses outside the defined PCI address space or to access the reserved area".
But, an actual movement becomes the following.

- Burst access
  3D words of the beginning can be accessed. And when the last D-word is accessed, the PCIC issues a target-abort.
- Single access
  1D word of the beginnings can be responded normally. And single access after that becomes not target-abort but master-abort.

(Phenomenon)

When the reservation area of the PCIC is accessed, the movement is different from specification.

(Conditions)

Under the following (1) or (2) conditions, this malfunction occurs.

(1) When OFIFO address range checking is enable.
That is to say, when bit [8] of TC register is set in "0". The value of the default is "0".

(2) When IFIFO address range checking is enable.
That is to say, when bit [7] of TC register is set in "0". The value of the default is "0".

(Work-around)

Don't access the reservation area for the PCIC.

(Notice on use of OS)

Please refer to "**ERT-TX3927-011～ERT-TX3927-014 Notice on use of OS**" of ERT-TX3927-014.

ERT-TX3927-014

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using PCI controller in the initiator mode

(Outline / Phenomenon)

IRDY is negated automatically at the fifth clock when the value of CBE becomes 0001b when three bytes data access is done in the PCI bus in the initiator mode. Therefore the following problems occur if TRDY isn't asserted by this time.

- Reading data and writing data can't be operated correctly.
- PCI target device can't negate TRDY.

(Conditions)

If both of the following (1) and (2) conditions are existed, this error may occur.

(1) When three bytes data access is done in the PCI bus in the initiator mode.
There are two cases that operate three bytes data access in the PCI bus in the initiator mode. This malfunction occurs in both cases.
a) The CPU operates three bytes access in the direct mode.
b) The CPU operates three bytes access in the indirect mode.

(2) When the value of CBE is 0001b.

(Work-around)

Don't operate three bytes data access in the PCI bus in the initiator mode.

(ERT-TX3927-011～ERT-TX3927-014 Notice on use of OS)

The influence on the OS of these malfunctions (ERT-TX3927-011～ERT-TX3927-014) is as mentioned in the following. And when an OS except for the following, a driver, a middleware, and so on are used, please confirm those software because it has the possibility to occur those malfunctions.

1) UDEOS/r39

UDEOS/r39 doesn't control the PCIC. When you use the PCIC in your system, according to the work-around of notes, please avoid those malfunctions.

2) VxWorks

- ERT-TX3927-011, ERT-TX3927-012
  The workaround regarding errata "ERT-TX3927-011" and "ERT-TX3927-012" has been provided in the BSP Revision 1.2/014 released by Toshiba as a reference. This revision has been certified by Wind River Systems, Inc., and is release by Wind River Systems, Inc. as Revision 1.2/000. The value of TC register and TBL register of the PCI controller have been changed to TC = 0x00001038 and TBL = 0x000009a6 to avoid these problems in the BSP Revision 1.2/014 released by Toshiba. If you are using earlier BSP release than Revision 1.2/014, please check the value of TC register and TBL register.
- ERT-TX3927-013
  The set value of TC register satisfies conditions of occurrence of this malfunction. Be careful not to access the reservation area for the PCIC.

- ERT-TX3927-014
  The problem doesn't occur when the PCI drivers (TC35815, RTL8029 and Intel82557/8/9) that are prepared for RBHMA3100(CE) +VxWorks by our company are used. If you include a driver except for these, according to the work-around of notes, please avoid this malfunction.

3) WinCE

It has the possibility to occur those malfunctions. Please inquire of your SI vender about detail of this matter.

4) Linux(Monta Vista Linux)

This problem doesn't occur when the BSP for RBHMA3100(CE) released by MontaVista Software, Inc. is used as it is. When you build your system, according to the work-around of notes, please avoid those malfunctions.

ERT-TX3927-015

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note) This malfunction is on TX39/H2 core whose PRID is 0x0000_2240, 0x0000_2241 or 0x0000_2242.
Application : Using the Timeout function for the bus error

(Outline)

The TX3927 can generate a bus error if the internal bus cycle becomes timeout when the TOE bit (bit [14]) of Chip Configuration Register (CCFG) is set in "1". In this malfunction, if the bus error occurs, the CPU core may be hung-up immediately after the bus error.

(Phenomenon)

When the bus error occurs, the CPU core may be hung-up immediately after the bus error.

(Conditions)

If all of the following (1) - (4) conditions are existed, this error may occur.

(1) The Timeout function for the bus error is enable. That is to say, when the TOE bit of CCFG register is set in "1". The value of the default is "0".

(2) The vector address of the bus error exception is set in cacheable area. That is to say, when the BEV bit (bit [22]) of Status register is set in "0". The value of the default is "1".

(3) The read operation doesn't occur on the internal bus before all 4 lines of write buffer are filled with write data after the bus error.

(4) The requirement of write operation from CPU core occurs when all 4 lines of the write buffer are filled with write data.

(Work-around)

There are two work-around (1), (2). You can evade this malfunction by using either one method.

(1) Assign the load operation for loading from cache area at the top of the exception handler.
Please refer to the work-around program example 1.
This work-around must be operated before the first store operation after the bus error.
This work-around is not effective if the data cache is disable.
This work-around is not effective if the load address is in cache when loading from the cache area, because the read operation is not occurred on the internal bus. To avoid this, please operate cache invalidate for the load address before load operation.
And, in write back mode of cache, the necessary data may be disappeared by cache invalidate. To avoid this, the data that is in cache line that is hit by load address must be read-only data. (In case of example 1, this is 4 word of 0x80000000 - 0x8000000c.)

[example 1]

```
0x80000080   lui    r10, 0x8000
       84    ori    r10, r10, 0x0000  ;r10 <- 0x80000000(cache area)
       88    cache 17, 0(r10) ;cache line that is hit r10 is invalidated.
       8c    lw     r11, 0(r10) ;read from r10
```

(2) Operate the instruction fetch from uncacheable area at the top of the exception handler.
Please refer to example 2 of work-around program.
This work-around must be operated before the first store operation after the bus error.
In example 2, supporting that the exception handler is assigned after 0x80000090, after jumping to the uncachealbe area, the program re-jump to the operation of the exception handler.

[example 2]

```
0x80000080   lui    r10, 0xbfc0       ;cache area
        84   ori    r10, r10, 0x1000
        88   jr     r10               ;jump to uncacheable area
        8c   nop
           ..                         ;operation of the exception handler

0xbfc01000   lui    r11, 0x8000        ;uncacheable area
        04   ori    r11, r11, 0x0090
        08   jr     r11                ;jump to the exception handler
        0c   nop
```

(Notice on use of OS)

1) UDEOS/r39
The source code of the exception vector is provided. If your system comes under this malfunction, according to the work-around of notes, please avoid the malfunction.

2) VxWorks
The code of the exception vector is set by the VxWorks kernel. If the bus error exception occurs, this malfunction may occur depends on the conditions of hit/ miss hit of cache of the execution of the exception vector code. To avoid this malfunction, the exception vector code needs to be modified. If you want to know how to modify the exception vector, please let us know.
The bus error exception doesn't occur when timeout function is disable. In the BSP of RBHMA3100(CE) of Wind River systems, Inc. or Toshiba, the timeout function is set enable.

3) WinCE
The malfunction may be occurred. Please inquire of us about detail of this matter.

4) Linux (Monta Vista Linux)
The source code of the exception vector is provided. If your system comes under this malfunction, according to the work-around of notes, please avoid the malfunction.

ERT-TX3927-016

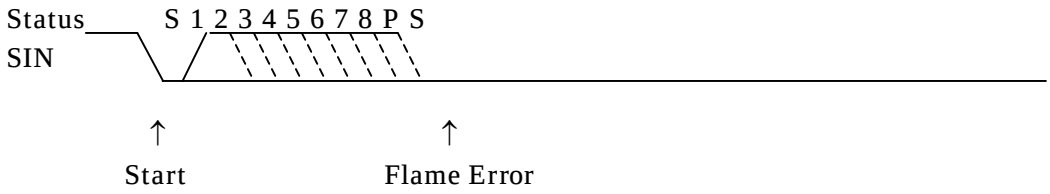
Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using the break function of the SIO

(Outline)

When the transmitter sends break signal in the middle of sending data, the TX3927 detects only the first framing error, and it can't detect a break signal

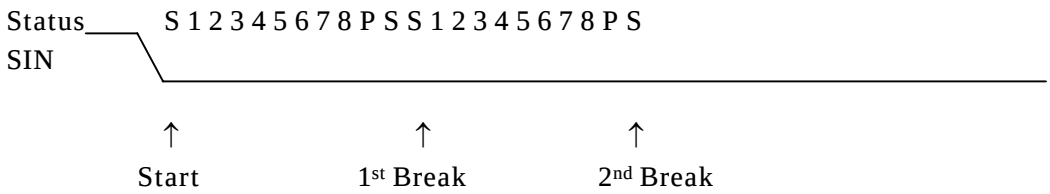
The TX3927 can detect break signal normally, when break signal is synchronized with start bit (The received data is low continuously right after the start bit recognition.).

(abnormal movement)



- The receive status stops as Idle, because the start bit after flaming error can't be recognized.

(normal movement)



- The break signal can be detected normally, when the received data is low continuously right after the start bit recognition.

(Phenomenon)

The TX3927 may not be able to detect a break signal.

(Conditions)

This malfunction may occur when the TX3927 is sent break signal in the middle of sending data.

(Work-around)

In case of sending a break signal to the TX3927, the break signal must be synchronized with start bit (The data must be low continuously right after the start bit.).

(Notice on Doze mode by OS)

1) UDEOS/r39

The SIO driver is not provided in UDEOS/r39. If your application uses the break function of the SIO, according to the work-around of notes, please avoid the malfunction.

2) VxWorks

The IO driver of VxWorks doesn't support the break function. So the OS driver and the customer who uses this driver doesn't take influence by this malfunction. If your application uses break function of the SIO, according to the work-around of notes, please avoid the malfunction.

3) WinCE

This malfunction may occur, depending on your error handling. If the conditions of this malfunction apply to your system, the TX3927 can't detect the break signal. So please handle the break signal with the flame error.

4) Linux (Monta Vista Linux)

The IO driver of Monta Vista Linux doesn't support the break function. So the OS driver and the customer who uses this driver doesn't take influence by this malfunction. If your application uses break function of the SIO, according to the work-around of notes, please avoid the malfunction.

ERT-TX3927-017

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using the TX3927 PCIC in satellite mode

(Outline)

In the specification, the R/WL bit of the PCI configuration register of the TX3927 can't be written by the external PCI master. But the bit is written by the PCI bus master in some cases.

The timing is when the configuration area of the TX3927 is written by the external PCI master when the CPU core of the TX3927 has written to the internal bus. The data that is written by the PCI master is set in the register.

The R/WL bit of the TX3927 is as follows.

- PCISTAT register(addr:0xfffed006) FBBCP and USPCP
- CC register(addr:0xfffed008)
- SCC register(addr:0xfffed009)
- RID register(addr:0xfffed00b)
- SVID register(addr:0xfffed02c)
- SSVID register(addr:0xfffed02e)
- ML register(addr:0xfffed03c)
- MG register(addr:0xfffed03d)
- IP register(addr:0xfffed03e)

(Phenomenon)

The R/WL bit of the PCI configuration register of the TX3927 may be written by the PCI bus master.

(Conditions)

If the following (1) - (2) conditions are existed at the same time, this error may occur.

(1) The external PCI master writes to the R/WL bit of the PCI configuration register of the TX3927.
(2) The CPU core of the TX3927 writes to the internal bus.

(Work-around)

When the external PCI master writes to the R/WL bit of the PCI configuration register of the TX3927, please operate read-modify-write and writes same value as read data in R/WL bit.

(Notice on Doze mode by OS)

If your system uses the TX3927 in satellite mode, please be careful because this malfunction may have influence on your system.

ERT-TX3927-018

Product Type : TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Application : Using “PCI broken master check function”

(Outline and phenomenon)

If you enable “PCI broken master check function”, a working master device may be regarded as a broken master device and detached from PCI bus arbiter.

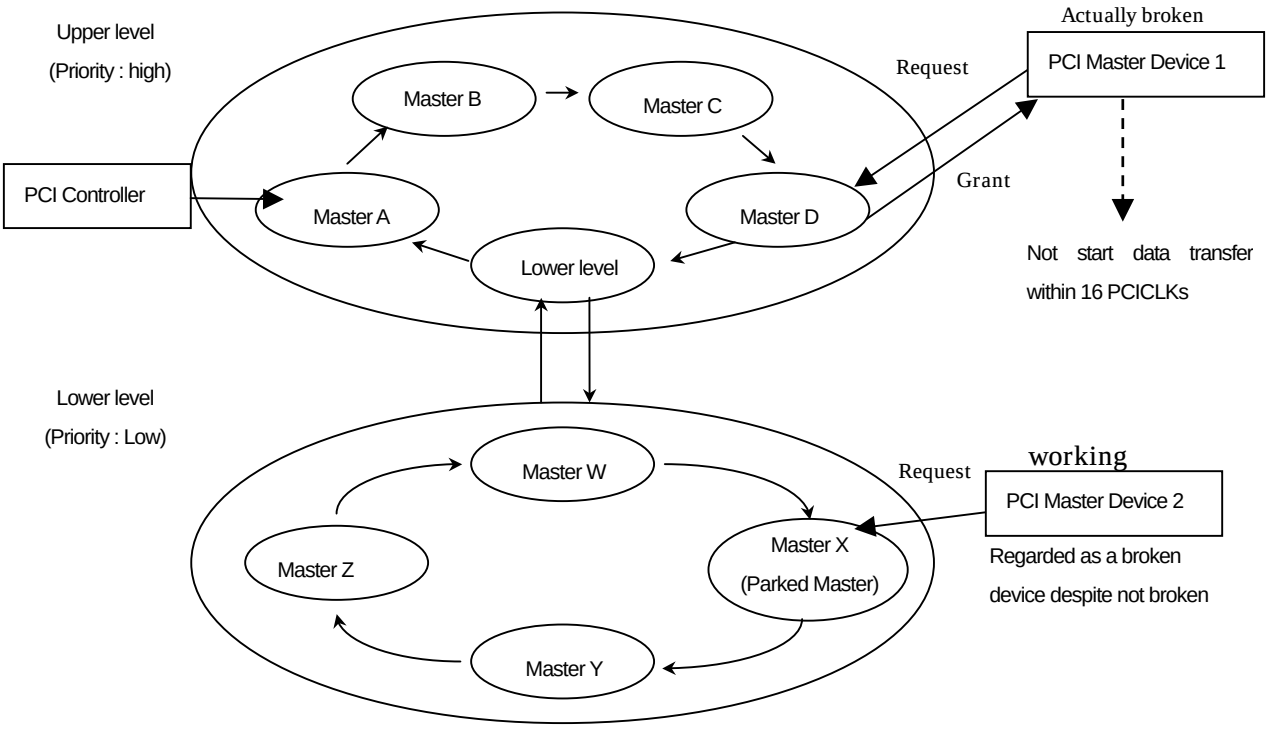
(Condition)

(1) Set Broken Master Check Enable bit. (BMCEN=1)
This bit is bit 0 of PCI Bus Arbiter /Parked Master Control Register (PBAPMC). Default is 0.

(2)There are two PCI bus arbiter groups, one is upper level and the other is lower level. (Please see the following figure) The assignment of PCI bus master devices to the bus arbiters is specified in Request Trace Register (REQ_TRACE)

(3)When a master device assigned to the upper level arbiter is detected as a broken device, another master device assigned to the lower level arbiter activates a PCI bus request and has the highest priority among other devices assigned to the lower arbiter. In the lower arbiter, master W right after reset or a master which got PCI bus at last has the highest priority. (A master with highest priority is called a parked master.)

If all the conditions above are taken simultaneously, the problem will come up and the parked master device will be detached from the PCI bus arbiter.



State of PCI bus arbiter (an example when the problem occurred)

(Work-around)

There are two solutions to this problem.

(1)Do not use "Broken Master Check Function" by not setting BMCEN bit of PBAPMC register to 1.

(2)If you must use "Broken Master Check Function" then only use the upper level arbiter.
(Master A,B,C, and D)

Documentation Changes

date: 05-Jul.-2002

Document : TX39 Family TMPR3927 data book Doc.No : (44150D-0112)

These contents will be incorporated in the next release of the documentations.

No.	Page	Contents
1	3-3	SYSCLK Description (delete) SYSCLK is not output when boot signal SYSCLKEN (ADDR[5] pin) is set to 0.
2	3-4	SDCLK[4:0] Description (delete) SDCLK is not output when boot signal SDRCLKEN (ADDR[4] pin) is set to 0.
3	3-4	SDCLK[4:0] Description (addition) When SDRAM is used, SDCLK[0] must be enable.
4	3-6	PCICLK[3:0] Description (addition) When internal PCICLK is used, PCICLK[0] must be enable.
5	3-6	FRAME, TRDY, IRDY, DEVSEL, STOP, SERR, PERR Description (addition) In the PCI specification, pull-up is needed.
6	3-12	SYSCLKEN (delete line)
7	3-12	SDRCLKEN (delete line)
8	3-13 Table 3.4.1	SYSCLKEN (delete line)
9	3-13 Table 3.4.1	SDRCLKEN (delete line)
10	5-4	SYSCLKEN Description (delete) Latched from ADDR[5] at the rising edge of RESET.
11	5-4	SYSCLKEN Description (addition) When output is set in disable, it becomes Hi-Z.
12	5-4	SDRCLKEN[4:0] Description (delete) Latched from ADDR[4] at the rising edge of RESET.
13	5-4	SDRCLKEN[4:0] Description (addition)Be sure to make SDRCLKEN[0] enable when you use SDRAM. When each output is set in disable, it becomes Hi-Z.
14	5-4	PCICLKEN [3:0] Description (addition)Be sure to make PCICLKEN[0] enable when you use internal PCICLK. When each output is set in disable, it becomes Hi-Z.
15	7-4	7.2.1 Bus Error (addition) Note: When CCFG.TOE is set to “1”, PCI configuration read must be operated in indirect mode.
16	8-11	Note 2: (wrong) t_{CAS} (right) t_{CASL}
17	8-28	8.5.6 Notes on Programming (addition) SDCLK output can be disable with bit[26:22] of the PCI Pin Configuration Register(PCFG). But, when SDRAM is used, SDCLK[0] must be enable, because SDCLK[0] is used for internal feedback.
18	9-4, 5	9.3.2 ROM Channel Control Registers Initial value of RWT, RCS, 16BUS, RBC, RHS, RME (wrong) Channels 2 to 7 (right) Channels 1 to 7
19	9-5	RHS Description (addition) Under particular conditions, the assert timing of ADDR, ACE*, CE*, OE* and BE* will be delayed for $t_{sysh}/2$(1/2 clock of the half speed mode reference clock) when the TX3927 ROM controller access the area of half speed mode channel.
20	10-2	Figure 10.2.1 ch.3 (wrong) (Receive) (right) (Transmit) (wrong) DMAREQ0, DMAACK0 (right) DMAREQ3, DMAACK3

No.	Page	Contents
21	12-10	Note (Addition) In the specification, bit23, 21 of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
22	12-12	Note (Addition) In the specification, bit[31:24] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
23	12-13	Note (Addition) In the specification, bit[23:16] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
24	12-15	Note (Addition) In the specification, bit[7:0] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
25	12-18	12.3.2.11 Target I/O Base Address Register (IOBA) (addtion) note: Set up Target I/O Base Address Size Register (IOBAS) before setting up this register.
26	12-19	12.3.2.12 Target Memory Base Address Register (MBA) (addtion) note: Set up Target Memory Base Address Size Register (MBAS) before setting up this register.
27	12-20	Note (Addition) In the specification, bit[31:16] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
28	12-21	Note (Addition) In the specification, bit[15:0] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
29	12-23	Note (Addition) In the specification, bit[31:24] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
30	12-24	Note (Addition) In the specification, bit[23:16] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
31	12-25	Note (Addition) In the specification, bit[15:8] of the PCI configuration register of the TX3927 can't be written by the external PCI master. But these bits are written by the PCI bus master in some cases.
32	12-34	Figure 12.3.27 Target Control Register(1/3) Bit 19 Field Name (wrong) OFIFO Caching Enable (right) OFIFO Caching Disable
33	12-34	Figure 12.3.27 Target Control Register(1/3) OFCAD Description (addition) In the target mode, when Single Burst Enable(OFPFO=0) is set, this bit must be set to "1".
34	12-34	Figure 12.3.27 Target Control Register(1/3) OFNTE Description (addition) In the target mode, when this bit is set to "1", the bit[7:4] of Target Burst Length Register(TBL) must not be set to "01xx" and "1x1x".
35	12-35	Figure 12.3.27 Target Control Register(2/3) OFPFO Field Name (wrong) Single Burst Enable (right) Single Burst Disable
36	12-35	Figure 12.3.27 Target Control Register(2/3) OFPFO Description (addition) In the target mode, when OFIFO Caching Enable(OFCAD=0) is set, this bit must be set to "1".
37	12-35	OFARD, IFARD Description (wrong) The last four D-words of this space are reserved for the PCIC. If this bit is cleared, the PCIC will issue a target-abort ... (right) The last four D-words of this space are reserved for the PCIC. So don't access to this area. If this bit is cleared, the PCIC may issue a target-abort or the initiator may issue a master-abort ...
38	12-35	OF16E Description (wrong) 1: ... , the PCIC target state machine disconnects the PCI bus master. (right) 1: ... , the PCIC target state machine issues a retry to the PCI bus master. (wrong) 0: If the OFIFO is not ready to send out the first data of a burst in response to a master read request within 16 PCI clock cycles, the PCIC target modules issues a retry to the PCI bus master. (right) 0:Target modules issues a retry to the PCI bus Master immediately when the OFIFO is not ready.
39	12-35	IF8E, OF8E Description (addtion) Set up this bit in "1" in the target mode. Otherwise, TX3927 may assert unnecessary STOP* signals.

No.	Page	Contents
40	12-52	TBL_OFIFO Description (addition) If the function is 16D-words (“01XX” or “1X1X”), don’t set bit[18] of Target Control Register(TC) as “1”
41	12-65	IOBAS Description line2 (wrong) Target I/O Base Address (IOBAS) (right) Target I/O Base Address (IOBA) IOBAS Description line8 (wrong) TLBMMAR and MBA (right) TLBIOMAR and IOBA
42	12-69	ILMDE Description line3 (wrong) the IPBMMAR register (right) the ILBMMAR register
43	12-69	ILIDE Description line3 (wrong) the IPBIOMAR register (right) the ILBIOMAR register
44	12-81	12.4.1 Transfer Modes Initiator PIO mode (addition) Note 2:In initiator mode, you can’t use the burst access of the local bus. Not to allocate to the cacheable area when you use in the direct mode. Note 3: When PCI configuration read is operated in direct mode, CCFG.TOE must be set as “0”.
45	12-82	12.4.2 Configuration Cycles (addition) In the configuration cycle, the device whose ID_SEL is high respond. If the TX3927 has never become target in configuration cycle, ID_SEL should be set low.
46	12-83	12.4.3 Address translation (addition) Note: The last four D-words of the space that is specified by the MBA register and the IOBA register is reserved for the PCIC. So don’t access to this area.
47	13-3	Figure 13.2.2 (wrong) SIRXDREQ (right) SIRXDREQ*
48	13-6	USBL Description (wrong) 1:1 stop bit 0:2 stop bits (right) 0:1 stop bit 1:2 stop bits
49	13-8	STIE Description (wrong) Status Change Interrupt Enable Channel1 (right) Status Change Interrupt Enable Channel
50	13-11	RBRKD Description (addition) Monitors break of RXD signal.
51	13-20	13.4.5 Receive Controller line1 (wrong) SIFLCR.RSDE bit is set. (right) SIFLCR.RSDE bit is low.
52	13-20	13.4.5 Receive Controller line3 (wrong) Therefore, if RXD is low when the SIFLCR.RSDE bit is set, the start bit is invalid. (right) Therefore, if RXD is low when the SIFLCR.RSDE bit is low the start bit is invalid.
53	13-23	13.4.14 Break Indication (addition) Notes: When the transmitter sends break signal in the middle of sending data, the TX3927 detects only the first framing error, and it can't detect a break signal. The TX3927 can detect break signal normally, when break signal is synchronized with start bit.
54	13-26	Figure 13.5.2, 13.5.3 (wrong) SIRXDREQ (right) SIRXDREQ*
55	17-5	17.6.1 AC Characteristics (wrong) (1) ACK*, DATA[31:0], ROMCE* (right) (1) ACK*, DATA[31:0], CE*
56	A-11	struct INTR (wrong) volatile int IRDER; /* *** Interrupt Detect Enable Register *** */ volatile int IRDMR[2]; /* *** Interrupt Detect Mode Register *** */ (right) volatile int IRCER; /* *** Interrupt Control Enable Register *** */ volatile int IRCR[2]; /* *** Interrupt Control Mode Register *** */

No.	Page	Contents
57	A-22	function intInt (wrong) CPUReg->IREG.IRDER = 1; (right) CPUReg->IREG.IRCER = 1;
58	A-29	function test_timerInt0 (wrong) CPUReg->IREG.IRDER = 1; (right) CPUReg->IREG.IRCER = 1;

Notice) The hatching parts are added to rev.1.3.

Summary Table of TX3927 Difference among F, AF, BF and CF

Errata No.	Description	TX3927F	TX3927AF	TX3927BF	TX3927CF
-	PRID Value		0x0000_2240	0x0000_2241	0x0000_2242
-	CRIR Value		0x3927_0032	0x3927_0040	0x3927_0040
-	DDRAD bit of LBC register in PCI controller	No	Yes	Yes	Yes
001	Moisture-proof packing	x	x	x	x
002	Electrostatic Discharge	x	x	x	x
003	Supply Voltage at TLB ON	x	x	x	x
004	PCI Stop* signal anomaly	x	x	△	△
005	Clock output at reset	x	x	○	○
006	Branchlikely Instruction anomaly	x	x	○	○
007	PCI Configuration Read anomaly	x	x	x	x
008	ROMC in the half-speed mode anomaly	x	x	x	x
009	Snoop function in Doze mode anomaly	x	x	x	x
010	CACHE instruction use in the write-back mode anomaly	x	x	x	○
011	PCI read access anomaly	x	x	x	x
012	Never Time Out function of PCIC anomaly	x	x	x	x
013	Access to PCIC reservation area anomaly	x	x	x	x
014	3 bytes data access of PCIC anomaly	x	x	x	x
015	The CPU lock after the bus error anomaly	x	x	x	x
016	Detection of the break signal of the SIO anomaly	x	x	x	x
017	Access to the R/WL bit of PCIC anomaly	x	x	x	x
018	PCI broken master check function	x	x	x	x

Note1) Codes used in summary table

x : Errata apply to this stepping.

○ : Errata is modified in this stepping.

△ :Errata is modified partly in this stepping.

Note2) Please refer the respectively section for more information about 001-010.

Note3) If you are using documentation of No. 44150D-9909, as regards “DDRAD bit of LBC register”, please refer to “No.122 of Documentation Change”.