TMPR3927 Specification Update

Revision History

23-May-2001	rev1.0	Initial release (up to ERT-TX3927-008)
04-Oct-2001	rev1.1	Added ERT-TX3927-009 and ERT-TX3927-010 Added CE* to ERT-TX3927-008. Added the table "Summary of the Differences Among Versions of the TX3927." Added descriptions to the Document Changes table.
30-Oct-2002	rev1.2	Deleted condition 3 of ERT-TX3927-008. Modified the description for WinCE in ERT-TX3927-009. Added descriptions to the Document Changes table.
22-Jan-2002	rev1.3	Added a description of TMPR3927CF. Added ERT-TX3927-011 to ERT-TX3927-014.
05-Jul-2003	rev1.4	Added ERT-TX3927-015 to ERT-TX3927-017. Modified a note showing product types in ERT-TX3927-009. Added descriptions to the Document Changes table. Changed Hard Hat Linux to Monta Vista Linux.
14-Mar-2003	rev1.5	Added ERT-TX3927-018. Added descriptions to the Document Changes table.
19-July-2006	rev1.6	Added ERT-TX3927-019. Modified some references to the related documents. Deleted the Document Changes table.

This specification update will be incorporated into the next revision of the documents.

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Related Documents: TX39 Family TMPR3927 DataBook (2003): Doc. No = BDE0016A TX39/H2 Processor Core Architecture (2000): Doc. No = 44124D-9908

RESTRICTIONS ON PRODUCT USE

The information contained herein is subject to change without notice.

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 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 may include products subject to the foreign exchange and foreign trade laws.

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.

If you have any technical questions regarding this document, contact us at:

TOSHIBA CORPORATION Semiconductor Company
Consumer System LSI Design Department
580-1, Horikawa-Cho, Saiwai-Ku, Kawasaki, 212-8520, Japan
Tel +81 -44 -548 -2222 Fax +81 -44 -548 -8320

ERT-TX3927-001

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: After opening moisture-proof packing

Effective usage period after opening moisture-proof packing

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

If more than 48 hours has elapsed after opening moisture-proof packing, bake the devices at 125°C for more than 20 hours before soldering them. After baking, store the devices in a cool, dry area at 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 Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Results of Electrostatic Discharge Testing

Results of electrostatic discharge testing

The following table shows the results of electrostatic discharge testing that was performed on this device.

When handling individual devices (which are yet to be mounted on a printed circuit board), be sure that the environment is protected against electrostatic electricity.

For more information, refer to the "General Safety Precautions and Usage Considerations" section of the databook.

Standard	Pins	Rated Voltage
Machine Model (MM) (EIAJ standard)	RXD[1:0], CTS[1:0]	200 V
	Other pins	250 V or more
Human Body Model (HBM) (MIL standard)	All pins	2000 V or more

ERT-TX3927-003

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note: This is already mentioned in the 2001 edition of the databook (Doc.No = 44150D-0111).
Condition: Using the TLB

Recommended operating supply voltage conditions

The following table shows the recommended operating supply voltage conditions for the TMPR3927F/AF/BF/CF. Keep in mind that the V_{DD2} rating is different, depending on whether the TLB is on or off.

When designing products that include this device, ensure that the recommended operating conditions for the device are always adhered to.

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

ERT-TX3927-004

Product Types: TMPR3927F, TMPR3927AF, (TMPR3927BF, TMPR3927CF) Note: Although this problem has been fixed in the TMPR3927BF/CF, there are some usage limitations.
Condition: Using the PCI Controller in Target mode

Outline

Under particular conditions, the PCI Controller in the TX3927 may assert the STOP* signal unnecessarily when the PCI bus is idle.

Symptoms

As shown in Figure 1 below, the PCI Controller in the TX3927 may, under particular conditions, assert the STOP* signal unnecessarily exactly when the PCI bus is idle. The external PCI master may be affected by this STOP* signal.

Conditions

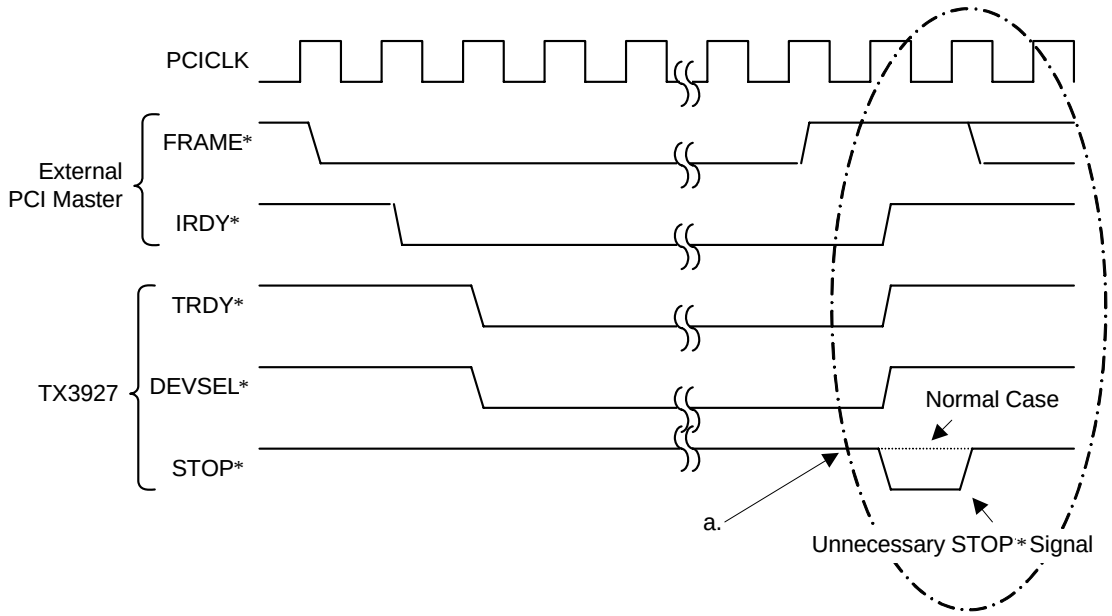


Figure 1 Unnecessary assertion of the STOP* signal

When the TX3927 PCI Controller is operating in Target mode and an external PCI master is bursting to the TX3927, the TX3927 PCI Controller asserts the STOP* signal unnecessarily when the completion of the burst cycle (a in the above figure) coincides with particular conditions of the TX3927.
There are the following three conditions:

1) When all of the following conditions a) to c) are true:
   a) Bit 3, or the OFIFO 8-Clock Rule Enable (OF8E) bit, in the Target Control (TC) register (at 0xFFFE_D090) of the PCI Controller is cleared.
   b) The current burst cycle is a read.
   c) The OFIFO becomes empty immediately before point a. in the above figure.

2) When all of the following conditions a) to c) are true:
   a) Bit 4, or the IFIFO 8-Clock Rule Enable (IF8E) bit, in the Target Control (TC) register of the PCI Controller is cleared.
   b) The current burst cycle is a write.
   c) The IFIFO becomes full immediately before point a. in the above figure.

3) When all of the following conditions a) to c) are true:
   a) Bit 11, or the PCI Snoop (PSBP) bit, in the Chip Configuration (CCFG) register at 0xFFFE_E000 is cleared.
   b) The current burst cycle is a write.
   c) The CPU core requests bus mastership at point a. in the above figure while another bus master on an internal bus is executing a bus cycle.

Workarounds

Burst reads and burst writes require separate workarounds. If both burst reads and burst writes are performed, use both of the following workarounds.

1) When the burst cycle is a read
   Set the OF8E bit in the Target Control (TC) register of the PCI Controller.

2) When the burst cycle is a write
   a) Set the IF8E bit in the Target Control (TC) register of the PCI Controller.
   b) Set the PSNP bit in the Chip Configuration (CCFG) register. In this case, however, it is prohibited to use the data cache in Write-back mode.

Status

This problem has been fixed in the TMPR3927BF as follows.

1) It is possible to clear the PSNP bit in the Chip Configuration register so as to use the data cache in Write-back mode.

2) There is still a usage limitation that the OF8E and IF8E bits in the Target Control register of the PCI Controller must be set when the PCI Controller is used in Target mode.

ERT-TX3927-005

Product Types: TMPR3927F, TMPR3927AF
Condition: Assertion of the RESET signal during operation

Outline

If the RESET signal is asserted under particular conditions, the SDCLK, SYSCLK and PCICLK outputs may assume the Hi-Z state.

Symptoms

If the RESET signal is asserted under particular conditions, the SDCLK, SYSCLK and PCICLK outputs may assume the Hi-Z state. Consequently, devices that operate with these clocks may be affected.
For example, when the SDCLK output enters the Hi-Z state, a constraint for the SDRAM clock is violated. As a result, the SDRAM might transition to an unexpected state.

Conditions

During a RESET sequence, the states of ADDR[4], [5] and [18] are used to determine whether or not to enable the SDCLK, SYSCLK and PCICLK outputs.

- SDCLK: When ADDR[4] is 1 during boot-up, SDCLK is generated. When 0, it is put in the Hi-Z state.
- SYSCLK: When ADDR[5] is 1 during boot-up, SYSCLK is generated. When 0, it is put in the Hi-Z state.
- PCICLK: When ADDR[18] is 1 during boot-up, PCICLK is generated. When 0, it is put in the Hi-Z state.

Since each ADDR pin has an internal pull-up resistor, these clocks are usually generated.
However, if the above ADDR pins are driven low while the RESET signal is asserted, SDCLK, SYSCLK, and PCICLK assume the Hi-Z state until these ADDR pins are pulled high by the pull-up resistor.

Workarounds

When the RESET* input is applied to the TX3927, all devices that use the SDCLK, SYSCLK or PCICLK output should be reset. With regard to SDRAM, remove the SDRAM power supply to bring it back to the initial state (power-on state) because it has no reset pin.

Status

This problem has been fixed in the TMPR3927BF as follows.

1) In the reset state, the SDCLK and SYCLK outputs remain enabled. The boot-up configurations of SDCLK and SYCLK by the ADDR[4] and ADDR[5] pins were deleted.
2) The SDCLK and SYSCLK outputs can still be disabled after boot-up via the Pin Configuration (PCFG) register at 0xFFFE_E008. These pins will assume the Hi-Z state when disabled.
3) The specification pertaining to PCICLK has not been modified.

ERT-TX3927-006

Product Types: TMPR3927F, TMPR3927AF Note: This problem occurs with the TX39/H2 core whose PRID is 0x0000_2240.
Condition: Using the TLB

Outline

The operation of a branch-likely instruction changes under particular conditions when the TLB is used.

Symptoms

The operation of a branch-likely instruction changes under particular conditions when the TLB is used.

Conditions

The branch-likely instruction nullifies the instruction in its delay slot when the branch condition is false. However, under the conditions described below, the instruction in the delay slot is executed, changing the program behavior.

The conditions that cause this problem are as follows:

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

Only when all these conditions are true at the same time, the address of the delay slot is stored into the EPC (or DEPC) register instead of the address of the branch-likely instruction, 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. However, NMI is an imprecise exception, and the return from an exception is not originally guaranteed. Since BUSERR is a fatal error, it is impossible to recover from this exception.

Note 2: The TX39/H2 Processor Core has a two-entry instruction TLB (ITLB) like a cache memory. An ITLB miss does not cause a TLB exception, since it is refilled from the actual TLB by hardware.

Workarounds

Include the following code in the interrupt handler.

```
-----------------------------------------------------------------------------------------------------------------------------
// If the EPC (or DEPC) register points to the end of a page and the preceding instruction is a branch-likely
// instruction, modify the EPC (or DEPC) register.
<Case of INT>
  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
-----------------------------------------------------------------------------------------------------------------------------
```

* This workaround assumes that no branch or jump instruction jumps into the delay slot of the branch-likely instruction. If that possibility can not be ignored, the above workaround can not be used.

Refer to the example procedure code of the workaround.

Status

This problem has been fixed in the TMPR3927BF.

Example procedure

Example of a patch procedure related to the branch-likely operation

```
#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: Modify the address of EPC_SAVE_AREA and the register number, depending on the OS used.

ERT-TX3927-007

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the PCI Controller

Outline

Under particular conditions, the processing of a bus error exception that occurs during a PCI configuration read doesn't work correctly.

Bus error exceptions due to other causes are processed properly.

Symptom

When the following conditions are true, the processing of a bus error exception doesn't work properly.

Conditions

This error occurs when all of the following conditions are true:

1) Timeout error is enabled. That is, the TOE bit of the Chip Configuration (CCFG) register is set to 1.

2) A PCI configuration read is executed in Direct mode. That is, it is executed via the Initiator Configuration Data Register (ICDR) at 0xFFFE_D13C and the Initiator Configuration Address Register (ICAR) at 0xFFFE_D138.
   A PCI configuration write causes no error.

3) A bus error exception occurs due to a G-Bus timeout. It occurs under the following conditions:
   a) The target PCI device repeats the "Retry" because its initialization is incomplete, etc. (i.e., repeats the "Retry" via the STOP* signal without asserting the TREADY signal)
   b) The PCI bus is deadlocked because the TX3927 and target PCI device repeat the "Retry" with each other. (This might occur in systems in which the TX3927 is configured to operate in both Target and Initiator modes. See page 12-84 of the TX3927 databook.)
   c) The TX3927 couldn't receive an acknowledge from the target PCI device within 512 G-Bus clock cycles because the PCI bus traffic is crowded.

Workarounds

There are two workarounds for this problem. Use either one of them.

1) Execute a PCI configuration read in Indirect mode.

   The Indirect mode can be executed by using the following registers:

   - Initiator Indirect Address register (IPCIADDR) at 0xFFFE_D150
   - Initiator Indirect Data register (IPCIDAT) at 0xFFFE_D154
   - Initiator Indirect Command/Byte Enable register (IPCICBE) at 0xFFFE_D158
   - Initiator Status register (ISTAT) at 0xFFFE_D044

In Direct mode, a PCI configuration read cycle is executed when the CPU reads the ICDR register. The TX3927 local bus cycle doesn't finish until the PCI bus cycle finishes.

On the other hand, in Indirect mode, PCI bus cycles start asynchronously to the TX3927 local bus when the CPU writes an address and command to the IPCIADDR and IPCIIBE registers. The result of a PCI configuration read is loaded into the IPCIDATA register. The completion of a PCI configuration read can be checked by polling the ISTAT register. (An interrupt can also be sent via the Initiator Interrupt Mask (IIM) register at 0xFFFE_D048.)

That is, in this mode, the local bus cycle finishes without waiting for a reply from the target PCI device. Therefore, a bus timeout error doesn't occur.

Note that the address and command are loaded into the IPCIADDR and IPCICBE registers. The contents of the IPCIADDR register will be placed on the PCIAD bus directly during the address phase, and the contents of the ICMD and IBE fields in the IPCICBE register will be executed as a PCI command and byte enables.

Refer to the following example code for Indirect mode. Keep in mind that this is just a sample code. Toshiba does not warrant its use in your system. You should always verify its operation in your system.

2) Disable timeout errors; that is, clear the TOE bit of the CCFG register.

In this case, the bus timeout error will never occur. However, there is a possibility that the TX3927 will be deadlocked when the local bus is waiting for an acknowledge from the target device. For instance, repeated retries by the target PCI device will cause a deadlock.

To avoid a deadlock, a system should be designed to assert a reset when it detects system failures such as a bus deadlock by using the WDT (Watchdog Timer), etc.

Example code

The following is an example code in which PCI configuration cycles are executed in Indirect mode.
This code is just an example. If an operating system is used, a routine that is called by an interrupt or RTOS might modify the ISTAT, IPCIADDR, IPCIDATA and IPCICBE registers. When you use this routine, exclusive control may be required so that the contents of these registers will not be altered.

```
---------------------------------------------------------------------------------------------------------------------------
/* This sample code is suitable for Type 0 configuration 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 Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the ROM Controller in Half-Speed mode

Outline

When a memory location is accessed via a ROM Controller channel configured for Half-Speed mode, the assertions of ADDR, ACE*, CE*, OE* and BE* are delayed by $t_{sysh}/2$ (i.e., a half of the reference clock for Half-Speed mode). Since these signals are deasserted as usual, the active periods of these signals become $t_{sysh}/2$ shorter than the usual.
No problem occurs when all channels are programmed in Full-Speed mode.

Symptom

Under particular conditions, the assertions of ADDR, ACE*, CE*, OE* and BE* are delayed by $t_{sysh}/2$ (i.e., a half of the reference clock for Half-Speed mode) when the TX3927 ROM Controller accesses the area mapped to a Half-Speed channel.
Consequently, the active periods of these signals become $t_{sysh}/2$ shorter than the usual.

Conditions

If bit 4, or the Half-Speed Bus (RHS) bit of the ROM Channel Control registers for one or more channels (RCCR0-7 at 0xFFFE_9000-901C) is set, this problem occurs when the following conditions become true simultaneously:

1) ADDR changes at the same time as the assertion of the GBSTART* on the TX3927 internal bus (G-Bus). This occurs in one of the following situations:
   a) The DMAC starts a bus cycle.
   b) The PCIC starts a bus cycle.
   c) The CPU starts a bus cycle immediately after GHPGGNT*, which is one of the bus grant signals for the G-bus, is deasserted. GHPGGNT* will be deasserted when the GHPGREQ* or GHAVEIT* is deasserted. GHPGREQ* is a G-Bus request without data cache snooping. GHAVEIT* is a signal to confirm G-Bus ownership. That is, when the DMAC and PCIC are programmed to use GHPGREQ* as a bus request signal and the CPU starts a bus cycle immediately after it has granted the bus to the DMAC or PCIC.

2) The above bus cycle is an access to the area mapped to a Half-Speed channel.

3) The beginning of the bus cycle is not aligned with the phase of the reference clock for Half-Speed mode. That is, when the bus cycle starts at the falling edge of the reference clock.

Under these conditions, ADDR, ACE*, CE*, OE* and BE* are delayed by $t_{sysh}/2$ (i.e., a half of the reference clock for Half-Sspeed mode). Since these signals are deasserted as usual, their active periods become $t_{sysh}/2$ shorter than the usual.

Note: GHPGGNT*, GHPGREQ* and GHAVEIT* are internal signals of the TX3927.

Workarounds

There are the following three workarounds for this problem. Use the one best suited for your application.

1) Use GSREQ* instead of GHPGREQ* as a G-Bus request signal for the PCIC and DMAC. In addition, program the PCIC and DMAC not to access any area mapped to the Half-Speed channels. For this workaround, it is necessary to program registers as follows.
   a) Set bit 11, or the PCI Snoop (PSNP) bit, of the Chip Configuration (CCFG) register (at 0xFFFE_E000).
   b) Set bit 7, or the Snoop (SNOP) bit, of the Channel Control Register (CCRn) (at 0xFFFE_B018, 0xFFFE_B038, 0xFFFE_B058 or 0xFFFE_B078).
   c) Clear bit 13, or the Write-Back Mode ON (WBON) bit, of the Config register (CP0: r3) to configure the cache for Write-Through (Non-Write-Allocate) mode. This is necessary because data cache snooping programmed by a) and b) above can not be used in Write-Back mode.
   d) Configure the DMAC not to access any area mapped to the Half-Speed channels.
   e) Configure the PCIC not to access any area mapped to the Half-Speed channels.

   Keep in mind that this workaround allows only Write-Back mode for the data cache.

   Note: GSREQ* is a bus request with data cache snooping. It is an internal signal of the TX3927.

2) When more than one ROM Controller channel is used in Half-Speed mode, design a system, based on the AC specification shown on the following page.

3) Use the ROM Controller only in Full-Speed mode.

SDRAMC and ROMC AC Characteristics

Difference

The Output Delay is defined separately for Full-Speed (td) and Half-Speed (tdh) modes.

In Half-Speed mode, the specs of ADDR, ACE*, CE*, OE* and BE* differ from those of the other signals.

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

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

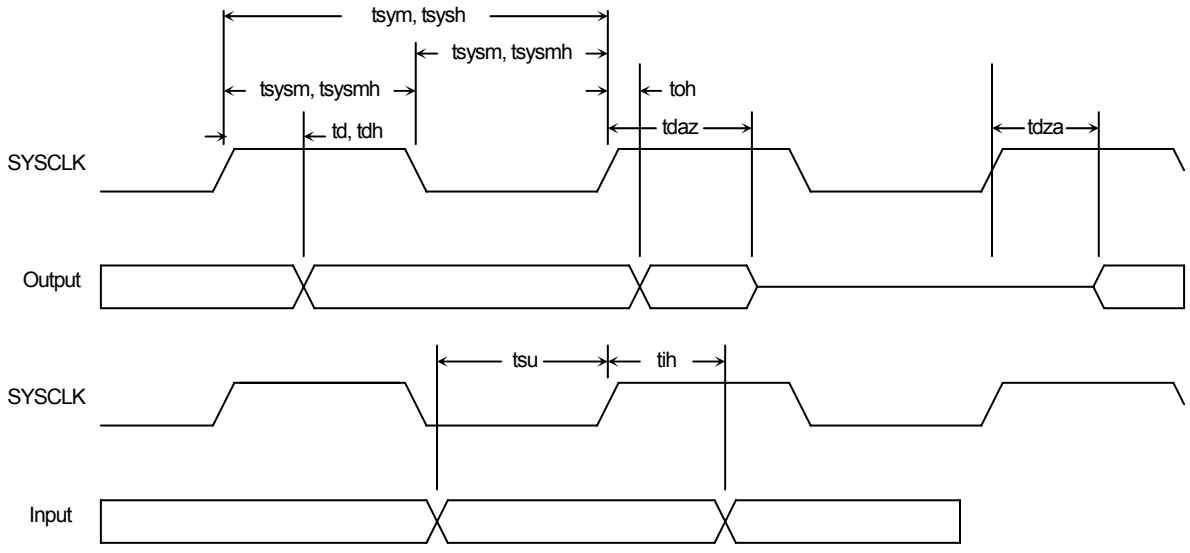
(1) ACK*, DATA[31:0], CE[7:0]*, OE*, ACE*, SWE*, BWE[3:0]*, ADDR[19:2], DMAACK[3:0], DMADONE*, PIO[15:0], TIMER[1:0]

(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 and ROMC Signals)



ERT-TX3927-009

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note: This problem occurs with the TX39/H2 cores whose PRID is 0x0000_2240, 0x0000_2241 or 0x0000_2242.
Condition: Using data cache snooping in Doze mode

Outline

In Doze mode, even if an external master owns the bus and performs data cache snooping, the corresponding data in the data cache is not invalidated.

The TX3927 enters Doze mode when the Doze bit (bit 9) of the Config register (CP0: r3) is set to 1. In Doze mode, the bus can be released in response to a bus request signal. If the snoop signal is sampled as asserted at the rising edge of a clock while an external master owns the bus, the cache line containing the data whose address is equal to that of the address bus is invalidated.
However, due to a bug, even if data cache snooping is performed in Doze mode, the data in the data cache is not invalidated.

Symptom

After the TX3927 returns from Doze mode, correct data may not be read.

Conditions

This problem occurs when data cache snooping is used in Doze mode.
In other modes, data cache snooping works normally.

Workarounds

There are the following three workarounds. Use the one best suited for your application.

1) Don't use data cache snooping.
2) Don't use Doze mode when data cache snooping is used.

   However, if UDEOS/r39 is used, when execution passes to an idle task, the Doze bit (bit 9) of the Config register (CP0: r3) is automatically set to 1 by the OS. Therefore, the source file (kidle.c) of the OS needs to be modified so as not to use Doze mode. However, this workaround cannot be used for a library package. Prepare a lowest-priority task. (Refer to "Workaroud Example for UDEOS/R39.")
3) When data cache snooping is used in Doze mode, invalidate the cache when the Doze mode exited.

Notes on Doze mode by OS

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

1) UDEOS/r39

In V3.3.0 or earlier versions of UDEOS/r39, when execution passes to an idle task, the Doze bit (bit 9) of the Config register (CP0: r3) is automatically set to 1 by the OS. Therefore, the source file (kidle.c) of the OS needs to be modified so as not to use Doze mode. However, this workaround cannot be used for a library package. Prepare a lowest-priority task.

Also, when you use Doze mode, avoid a problem according to the workaround.

2) VxWorks

VxWorks for the TX3927 uses the VxWorks kernel of Tornado 2.0/R3000, and only the cache library is replaced for TX39/H2 cores. That is, all functions are compatible with R3000 except the cache library. For this reason, Doze mode is not supported by the OS, but you can use it on your own responsibility. When you use Doze mode, avoid a problem according to the workaround.

3) WinCE

Ask your SI vender for details of this matter. When you use Doze mode in your system, avoid a problem according to the workaround.

4) Linux

Linux (Monta Vista Linux) of MontaVista Software, Inc. does not use Doze mode in its kernel. When you use Doze mode, avoid a problem according to the workaround.

Workaround Example for UDEOS/R39

The following example for UDEOS/R39 uses workdaround 2) described above.

1. Changing the source file of the OS

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

(kidle.c) before modification

```
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();
    }
}
```

Edit the source code as follows so as not to enter Doze mode and rebuild a kernel.

(kidle.c) after modification

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

The code for recording an idle log has been deleted so that buffers will not be filled up with an idle log. The kernel can still recognize when the process changes into the idle state.

2. Not changing the source file of the OS

Create a task with a priority lower than the lowest-priority task currently used in the application. Identify this 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(;;);
}
```

kidle.c is not contained in the library package. Thus, workaround 1, “Changing the source file of the OS,”cannot be used. Use workaround 2 instead.

ERT-TX3927-010

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF Note: This problem occurs in the TX39/H2 cores whose PRID is 0x0000_2240 or 0x0000_2241.
Condition: Using Write-back mode

Outline

In Write-back mode, the internal bus (G-Bus) and external buses such as the SDRAM bus may be locked. This problem may occur when the data cache is flushed by the CACHE instruction while the DMAC or PCIC owns the G-Bus through a non-snooping-capable bus request signal.

Due to this bug, even if the TOE bit (bit 14) of the Chip Configuration (CCFG) register is set and a timeout for bus errors is enabled, a bus error does not occur. In addition, when the watchdog timer of the TX3927 is used, although a reset is effective for the TX3927, whether or not it is effective for the entire system depends on your system design. If the reset is not effective for the entire system, you must turn off the power supply to your system.

Symptom

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

Conditions

This error may occur if both of the following conditions are true:

1) The Write-back mode is used.
2) The CACHE instruction is executed to flush the data cache while an external master owns the G-Bus.

   The CACHE instruction with the operation field (bits 20:16) of 0x01, 0x15 or 0x19 flushes the data cache.

This problem doesn't occur during a burst write operation that is triggered for a cache replacement due to a cache miss.

Workarounds

There are the following two workarounds for this problem. You can avoid this problem by using either one of them.

1) Use Write-through mode.
2) When using Write-back mode, don't use the CACHE instruction for flushing the data cache while an external master owns the G-Bus.

Status

This problem has been fixed in the TMPR3927CF.

Workaround Examples

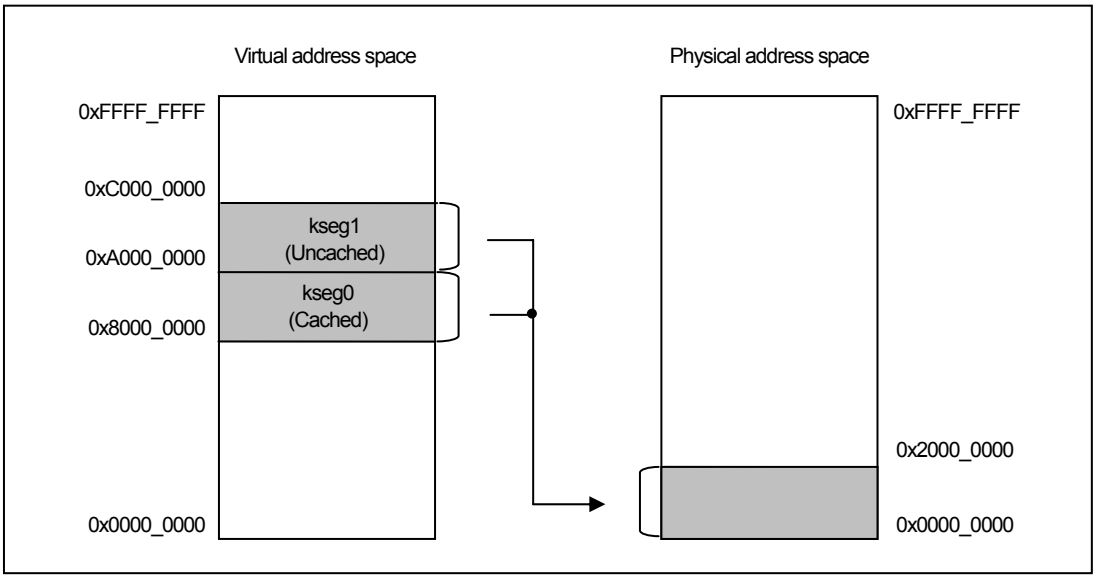
Here are examples of workaround 2 described above. In these examples, the cache is flushed without using the CACHE instruction.

< Example 1 >

0x8000_0000–0x9FFF_FFFF and 0xA000_0000–0xBFFF_FFFF in the virtual address space are mapped to the same range in the physical address space (0x0000_0000–0x1FFF_FFFF).

0x8000_0000–0x9FFF_FFFF is a cached area whereas 0xA000_0000–0xBFFF_FFFF is an uncached area.

Consequently, reading data from a cached area and writing that data to an uncached area whose physical address is the same as the read address has an effect equivalent to flushing the data cache with the CACHE instruction.



For example, if the data address that is the target of cache flushing is 0x8000_0000, perform the following steps:

1) Read out data from 0x8000_0000.

   Because 0x8000_0000–0x9FFF_FFFF is a cached area, a read request is first issued to the data cache. Thus, when address 0x8000_0000 hits in the data cache, data is read from it. In the case of a cache miss, data is read from main memory (physical address 0x0000_0000).

2) Write the data that was read in step 1) to 0xA000_0000.

   Because 0xA000_0000–0xBFFF_FFFF is an uncached area, data is always written to main memory (physical address 0x0000_0000).

3) Invalidate the data cache line containing data whose address is the same as that of step 1). This can be done with the CACHE instruction with the op field (bits 20:16) of 0x11.

In other words, when a read hits in the data cache, cache flushing (a read from the cache and a write to main memory) is done by the above operation.

However, in the case of a cache miss, a read and a write is done to same memory address; so the above operation is a waste of time.

< Example 2 >

When there is an unused cached data area of more than four Kbytes, the entire data cache area can be flushed by reading the area of more than four Kbytes in sequence (since cache replacements occur due to cache misses).

ERT-TX3927-011

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the PCI Controller in Target mode

Outline

Under specific conditions, a PCI read accesses the address following the one that was written immediately before the read instead of the specified read address.

Symptom

Under specific conditions, when the TX3927 PCIC is read as the target, the read access may be performed to a wrong address.

Conditions

Under these setup conditions, this problem may occur when PCI accesses occur in the following order.

< Setup >

1) When data in the OFIFO is cached after completion of a PCI transaction.

   That is, when the OFCAD bit (bit [19]) of the Target Control (TC) register is cleared to 0. The default value of this bit is 0.

2) When read streaming from the local memory to the OFIFO is enabled for a PCI read request.

   That is, when the OFPFO bit (bit [12]) of the TC register is cleared to 0. The default value of this bit is 0.

< The order of PCI accesses that causes this problem >

1) The PCI device reads from the TX3927 local memory.

   The data-in to the OFIFO is so late that the PCI read cycle ends with a Retry. (The TX3927 is reading from the local memory to the OFIFO.)

2) The PCI device writes to the TX3927 local memory.

   Actually, the PCI transaction is finished when data is written to the IFIFO.

   The write operation from the IFIFO to the main memory is made to wait until the read from the local memory to the OFIFO is completed.

3) The PCI device reads from the TX3927 local memory again.

   Because there is data in the OFIFO this time, data is returned and the PCI transaction is finished.

4) The PCIC completes the write operation from the IFIFO to the local memory before step 3 is finished, or the PCIC is still executing a write to the local memory when step 3 is finished.

5) The PCI device reads data from a location of the TX3927 local memory that follows the address of the previous read.

When the timing of step 4 occurs, the address of the local bus cycle being executed by the PCIC is latched into the address pointer of the local OFIFO bus. Because step 5 is an access to the address that follows the previous address, when the PCIC is programmed for the <Setup> conditions, the pointer of the local bus address isn't updated. Thus, the address of the next data is latched in step 4 (the address written in step 4). Consequently, data is read from a wrong address.

Workarounds

There are the following two workarounds for this problem. You can avoid this problem by using either one of them.

1) Discard unused OFIFO data after the PCI transaction is finished.

   In other words, set bit [19] of the TC register to 1. (default = 0)

2) For each PCI read request, read from the local memory to the OFIFO only once. (Don't stream data.)

   In other words, set bit [12] of the TC register to 1. (default = 0)

Notes on the use of an OS

Refer to the "Notes on the use of an OS for ERT-TX3927-011 to ERT-TX3927-014" shown in ERT-TX3927-014.

ERT-TX3927-012

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the “Never Time Out” feature of the PCI Controller

Outline

The PCI bus may be locked when the Never Time Out feature of the PCIC is enabled and the TBL_OFIFO field (bits [7:4]) of the Target Burst Length (TBL) register is programmed to 16 Dwords. The Never Time Out feature is enabled when the OFNTE bit (bit [18]), OF16E bit (bit [5]) and OF8E bit (bit [3]) of the Target Control (TC) register are all set to 1.The TBL_OFIFO field (bits [7:4]) of the TBL register is programmed to 16 Dwords when it is set to 01xx or 1x1x.

Symptom

If the Never Time Out feature of the PCIC is enabled, the PCI bus may be locked when the TX3927 PCIC is accessed as a target under particular conditions.

Conditions

Under these setup conditions, this problem may occur as follows:

< Setup >

1) When the Never Time Out feature is enabled.
   That is, when bit [18] of the TC register is set to1. The default is 0.
2) When bits [7:4] of the TBL register are set to 01xx or 1x1x.

< The process that causes this problem >

The following process causes this problem when an external bus master inserts a wait state by deasserting IRDY at the timing of the 16th Dword when it is reading 16 Dwords from the TX3927.

1) Because the transaction finishes at the next Dword, the TX3927 considers that the transaction will finish at the next clock when the 15th Dword is read, and stops there.
2) IRDY is set low again, and the external PCI bus master keeps waiting for the next data.

Because the Never Time Out feature is enabled, the TX3927 can't finish the transaction by asserting the STOP* signal.
Also, since bits [7:4] of the TBL register are programmed to 16 Dwords, which is the same size as the FIFO, the next data can't be put into the OFIFO because the final data (1 Dword) remains left in the OFIFO. Thus the output level (the number of words) for the PCI bus of the OFIFO is not satisfied.
Because the transaction isn't finished and the PCI bus master doesn't release the bus, the PCI bus is locked.

Workarounds

There are the following two workarounds for this problem. You can avoid this problem by using either one of them.

1) Disable the Never Time Out feature.
   In other words, clear bit [18] of the TC register to 0. (default = 0)
2) Set bits [7:4] of the TBL register to a value other than 01xx and 1x1x.

Notes on the use of an OS

Refer to the “Notes on the use of an OS for ERT-TX3927-011 to ERT-TX3927-014” shown in ERT-TX3927-014.

ERT-TX3927-013

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the PCI Controller in Target mode

Outline

The location of the PCIC address space is specified in the Target Memory Base Address Size (MBAS) and Target I/O Base Address Size (IOBAS) registers. The last four Dwords of this space are reserved for the PCIC. When the OFARD bit (bit [8]) or the IFARD bit (bit [7]) of the Target Control (TC) register is cleared to 0, the specification of the 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."

However, the PCIC actually operates as follows:

- Burst access

  The first three Dwords can be accessed. When the last Dword is accessed, the PCIC issues a target-abort.

- Single access

  The first one Dword can be accessed normally. The subsequent single access causes a master-abort instead of a target-abort.

Symptom

When the reserved area of the PCIC is accessed, the PCIC does not operate as specified.

Conditions

This problem occurs under the following conditions.

1) When the OFIFO address range checking is enabled.

   That is, when bit [8] of the TC register is cleared to 0. The default is 0.

2) When the IFIFO address range checking is enabled.

   That is, when bit [7] of the TC register is cleared to 0. The default is 0.

Workaround

Don't access the reserved area of the PCIC.

Notes on the use of an OS

Refer to the "Notes on the use of an OS for ERT-TX3927-011 to ERT-TX3927-014" shown in ERT-TX3927-014.

ERT-TX3927-014

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the PCI Controller in Initiator mode

Outline/Symptom

When a three-byte data access is done over the PCI bus in Initiator mode, IRDY is negated automatically at the fifth clock when the value of CBE becomes 0001b. Therefore, the following problems occur if TRDY isn't asserted by this time:

- Reads and writes aren't performed correctly.
- The PCI target device can't negate TRDY.

Conditions

This error may occur if both of the following conditions are true:

1) When a three-byte data access is done over the PCI bus in Initiator mode.

   There are two cases in which a three-byte data access is performed over the PCI bus in Initiator mode. This error occurs in both of the following cases:

   a) The CPU performs a three-byte access in Direct mode.

   b) The CPU performs a three-byte access in Indirect mode.

2) When the value of CBE is 0001b.

Workaround

Don't perform a three-byte data access over the PCI bus in Initiator mode.

Notes on the use of an OS for ERT-TX3927-011 to ERT-TX3927-014

The following paragraphs describe the influences that the problems discussed in ERT-TX3927-011 to ERT-TX3927-014 have on the OS. When you use an OS except the following, drivers, middleware and so on, verify such software because it has the possibility of causing problems.

1) UDEOS/r39

   UDEOS/r39 doesn't control the PCIC. When you use the PCIC in your system, avoid a problem according to the workaround.

2) VxWorks

   - ERT-TX3927-011, ERT-TX3927-012

     If you use the BSP for the JMR-TX3927 from Wind River without modification, this problem does not occur. If you change its settings, be sure to avoid a problem according to the above workaround. Additionally, earlier releases of BSPs provided by Toshiba as samples may also be affected by this problem. If you use a Toshiba-provided BSP, please check the values of the TC and TBL registers.

- ERT-TX3927-013

  The value of the TC register satisfies the conditions for an occurrence of this problem. Be careful not to access the reserved area of the PCIC.

- ERT-TX3927-014

  No problem occurs when you use the PCI drivers (TC35815, RTL8029 and Intel 82557/8/9) from Toshiba for JMR-TX3927 +VxWorks. If you include a driver other than those, avoid a problem according to the workaround.

3) WinCE

WinCE has the possibility of causing problems discussed in ERT-TX3927-11 to ERT-TX3927-14. Ask your SI vender for details of this matter.

4) Linux (Monta Vista Linux)

No problem occurs when the BSP for JMR-TX3927 from Monta Vista Software, Inc. is used as-is. When you build your system, avoid a problem according to the workaround.

ERT-TX3927-015

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF Note: This problem occurs with the TX39/H2 cores whose PRID is 0x0000_2240, 0x0000_2241 or 0x0000_2242.
Condition: Using the the bus timeout error function

Outline

The TX3927 generates a bus error if the internal bus cycle has timed out when the TOE bit (bit [14]) of the Chip Configuration (CCFG) register is set to1. If a bus error occurs, the CPU core may hang immediately due to a bag.

Symptom

When the bus error occurs, the CPU core may hang immediately.

Conditions

This error may occur if all of the following conditions are true:

1) The bus timeout error feature is enabled. That is, when the TOE bit of the CCFG register is set to1. The default is 0.
2) The bus error exception is vectored to a cacheable area. That is, when the BEV bit (bit [22]) of the Status register is cleared to 0. The default is 1.
3) No read operation occurs on the internal bus before all four lines of the write buffer are filled with write data after a bus error.
4) The CPU core requires a write operation when all four lines of the write buffer are filled with write data.

Workarounds

There are the following two workarounds. You can avoid this problem by using either one of them.

1) Load from the cache area at the top of the exception handler.

   Refer to Example 1 for a workaround program.

   This workaround must be executed before the first store operation after a bus error.

   This workaround is not effective if the data cache is disabled.

   This workaround is not effective if the load address is in the cache when loading from the cache area, because the read operation does not occur over the internal bus. To avoid this, execute cache-invalidate for the load address before a load operation.

   If the cache is programmed for Write-back mode, the necessary data may disappear as a result of cache-invalidate. To avoid this, the data in a cache line that is hit by a load address must be read-only data. (In the case of example 1, this is four words at 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 hits in r10 is invalidated.
          8c  lw      r11, 0(r10) ;read from r10
```

2) Execute the instruction fetch from an uncacheable area at the top of the exception handler.

Refer to Example 2 for a workaround program.

This workaround must be executed before the first store operation after a bus error.

In Example 2, suppose that the exception handler is located after 0x80000090. The program re-jumps to the exception handler after jumping to the uncacheable area.

< 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
```

Notes on the use of an OS

1) UDEOS/r39

Part of the source code of the exception vector is provided. If this problem occurs in your system, avoid it according to the workaround.

2) VxWorks

The code of the exception vector is set by the VxWorks kernel. If a bus error exception occurs, this problem occurs, depending on whether the exception vector code hits or misses in the cache. To avoid this problem, the exception vector code needs to be modified. Contact Toshiba if you want to know how to modify the exception vector code.

The bus error exception doesn't occur when the timeout feature is disabled. In the BSP of JMR-TX3927 from Wind River Systems, Inc. or Toshiba, the timeout feature is enabled.

3) WinCE

This problem may occur. Contact Toshiba for details.

4) Linux (Monta Vista Linux)

Part of the source code of the exception vector is provided. If this problem occurs in your system, avoid it according to the workaround.

ERT-TX3927-016

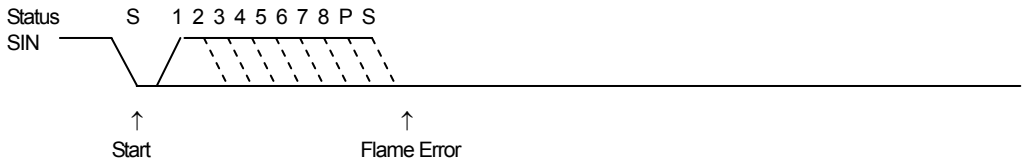
Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the break function of the SIO

Outline

When the transmitter sends a break condition in the middle of data, the TX3927 detects only the first framing error, but can't detect the break condition.

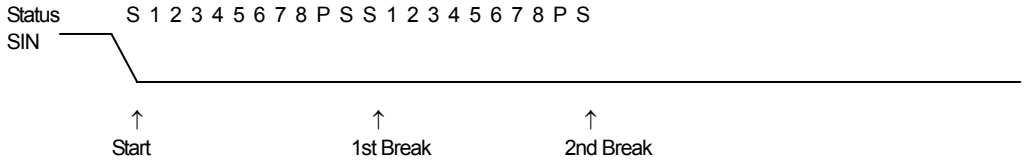
The TX3927 can detect a break condition normally when it is synchronized with a start bit (The received data remains low after the start bit).

< Abnormal operation >



- The receive status remains idle because the start bit can't be recognized after a framing error.

< Normal operation >



- The break condition can be detected normally when the received data remains low after the start bit.

Symptom

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

Conditions

This problem may occur when the transmitter sends a break condition in the middle of data.

Workaround

When sending a break condition to the TX3927, it must be synchronized with the start bit (The data must remain low after the start bit.).

Notes on Doze mode by OS

1) UDEOS/r39

The SIO driver is not contained in UDEOS/r39. If your application uses the break function of the SIO, avoid a problem according to the workaround.

2) VxWorks

The IO driver of VxWorks doesn't support the break function. Thus the OS driver and routines that use this driver are not affected by this problem. If your application uses the break function of the SIO, avoid a problem according to the workaround.

3) WinCE

This problem may occur, depending on your error handling. If the conditions of this problem apply to your system, the TX3927 can't detect a break condition. Handle the break condition as a framing error.

4) Linux (Monta Vista Linux)

The IO driver of Monta Vista Linux doesn't support the break function. Thus the OS driver and routines that use this driver are not affected by this problem. If your application uses the break function of the SIO, avoid a problem according to the workaround.

ERT-TX3927-017

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the TX3927 PCIC in Satellite mode

Outline

According to the specification, the R/WL bits in the PCI configuration registers of the TX3927 can't be written by the external PCI master. However, these bits are written by the PCI bus master in some cases.

This occurs when the configuration space of the TX3927 is written by the external PCI master while the CPU core of the TX3927 is writing to the internal bus. The data that is written by the PCI master is loaded into the register.

Here is a list of the R/WL bits in the PCI configuration registers:

- FBBCP and USPCP bits in the PCISTAT register (addr: 0xfffed006)
- 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)

Symptom

The R/WL bits in the PCI configuration registers of the TX3927 may be written by the PCI bus master.

Conditions

This error may occur if the following conditions are true at the same time:

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

Workaround

When the external PCI master writes to the R/WL bits of a PCI configuration register of the TX3927, it must perform a read-modify-write operation to write the same values as the read ones to the R/WL bits.

Notes on Doze mode by OS

If your system uses the TX3927 in Satellite mode, be careful because this problem may affect your system.

ERT-TX3927-018

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Condition: Using the “PCI broken master checking” feature

Outline and Symptom

If you enable the PCI broken master checking feature, a working master device may be regarded as a broken master device and detached from the PCI bus arbiter.

* The broken master checking feature of the PCI Controller identifies a PCI master that does not start a bus access even when it has gained bus mastership. It is treated as broken and excluded from the bus arbitration process.

When a given bus master is treated as broken, another PCI master might be treated as broken at the same time.

Conditions

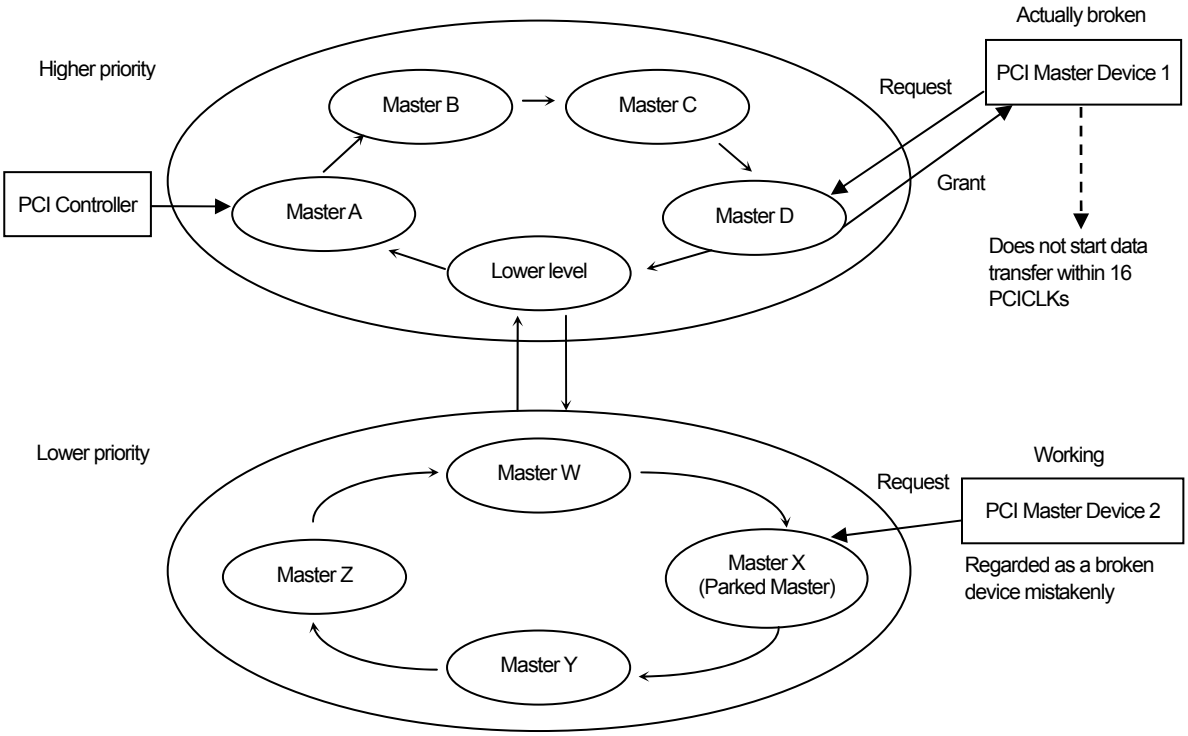
(1) The Broken Master Check Enable bit is set (BMCEN=1).

This bit is bit 0 of the PCI Bus Arbiter/Parked Master Control Register (PBAPMC). The default is 0.

(2) There are two PCI bus arbiter groups: higher-priority group and lower-priority group (see the following figure). The assignment of PCI bus masters to the bus arbiters is specified in the Request Trace Register (REQ_TRACE).

(3) When a master device in the higher-priority group is detected as broken, a master device in the lower-priority group activates a PCI bus request and is given the highest priority in that group. Among the lower-priority group, master W has the highest priority upon reset. Afterwards, a master which got the PCI bus last has the highest priority. (A master with the highest priority is called a parked master.)

If all the above conditions become true simultaneously, this problem occurs and the parked master device is detached from the PCI bus arbiter.



PCI Bus Arbitration Priority (an example when a problem occurred)

Workarounds

There are two workarounds for this problem:

1) Disable the the broken master checking feature by clearing the BMCEN bit of the PBAPMC register to 0.

2) If you need to use the broken master checking feature, then only use the higher-level arbiter (masters A, B, C and D).

Status

There is no plan to fix this bug in the TMPR3927F, TMPR3927AF, TMPR3927BF and TMPR3927C.

ERT-TX3927-019

Product Types: TMPR3927F, TMPR3927AF, TMPR3927BF, TMPR3927CF
Conditions When an SIO overrun error is detected by checking the OERS bit in the Status Change Interrupt Status Register (SISCISR) 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****** .

Outline

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.

Symptom

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

(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.

(Defect 2)

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

Workarounds

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.

Summary of the Differences Among Versions of the TX3927

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	×	×	×	×
002	Electrostatic discharge	×	×	×	×
003	Supply voltage when TLB is on	×	×	×	×
004	PCI STOP* signal	×	×	Δ	Δ
005	Clock output upon reset	×	×	○	○
006	Branch-likely instruction	×	×	○	○
007	PCI Configuration Read	×	×	×	×
008	ROMC in Half-Speed mode	×	×	×	×
009	Snooping in Doze mode	×	×	×	×
010	CACHE instruction in Write-back mode	×	×	×	○
011	PCI read access	×	×	×	×
012	Never Time Out of PCIC	×	×	×	×
013	Accesses to PCIC reserved area	×	×	×	×
014	PCIC 3-byte data access	×	×	×	×
015	CPU lock after a bus error	×	×	×	×
016	SIO break detection	×	×	×	×
017	Accesses to the PCIC R/WL bits	×	×	×	×
018	PCI broken master checking feature	×	×	×	×
019	Detection of SIO overrun errors	×	×	×	×

Note 1: Legned used in the summary table

×: Applies to this device.

○: Modified in this device.

Δ: Modified partly in this device.

Note 2: Refer to the respective sections for details.