Connect Tech Inc.
Industrial Strength Communications

USER MANUAL

PCI and PCI Express
Dump Switch

CTIM-00031 Rev.0.03 – October 16, 2009

Copyright Notice

Trademark Acknowledgement

Revision History

Revision 0.03	**October 16, 2009** Added Setup Using the Connect Tech Dump Switch Driver
Revision 0.02	**August 8, 2009** Added PCI Express Dump Switch
Revision 0.01	**March 2006** Original Document

Table of Contents

Introduction

Connect Tech's Dump Switch cards are designed to enable developers to debug a crash situation even when a system freeze dictates a hardware reset. The Dump Switch increases developer productivity by eliminating the guess work normally required when debugging the cause of a system hang by preserving valuable system information normally lost during reset. By pushing the button located on the exterior of your Dump Switch card, a Non-Maskable Interrupt (NMI) will be triggered, enabling a crash dump or an execution drop into your operating system's debugger.

Connect Tech's Dump Switch cards can troubleshoot events even when the bug causes a complete lock of the operating system. (The dump does not reset the host computer).

PCI Dump Switch Features

- Universal 32-bit PCI card (PCI 2.3 compliant)
- Operating temperature range 0°C to 65°C
- Push button generation of a Non-Maskable Interrupt
- Transparent PCI to PCI bridge
- Standard MD1 format

PCI Express Dump Switch Features

- PCI Express 1.0 compliant
- Operating temperature range 0°C to 65°C
- Push button generation of a Non-Maskable Interrupt
- x1 lane PCI Express
- Small PCB footprint

General Usage Notes

Most operating systems will require you to set up the following to operate the Dump Switch Card:

- Enable NMI support using a few IO instructions
- Set the SERR Enable bit on any PCI bridge chips between the Dump Switch card and processor

These instructions are intended as guidelines only. Every operating system manages the implementation of NMI support in a different manner, so setups may vary slightly. Please refer to the **QNX**, **Windows** and **Linux** General Usage Notes for set up variations pertaining to your operating system.

Enabling the NMI

Generally a few I/O registers need to be set to allow the NMI to pass through the hardware to the processor. In a typical system, the following I/O instructions will enable the NMI:

```
outp( 0x61, ( inp( 0x61 ) & ~( 3 << 2 ) ) );
outp( 0x70, ( inp( 0x70 ) & ~( 1 << 7 ) ) );
inp( 0x71, );  // dummy read
```

Enabling the PCI Bridge SERR Option

The SERR option in the command registers of the PCI bridge chips between the Dump Switch card and the ISA bus (if present) or bus 0 PCI bridge may need to be altered. This will enable the SERR signal from the PCI card to be passed onto the main CPU.

In a typical system, the following will turn on the SERR Enable bit:

- Read in the Command register value for a given PCI bridge device
- OR this value with 0x100 to turn on the SERR Enable bit
- Store this value back to the Command register
- Read in the Bridge Control register value for the PCI bridge device
- OR this value with 0x2 to enable the SERR forwarding bit
- Store this value back to the Bridge Control register

To ensure the SERR PCI status is passed back to the CPU:

- Look for the Dump Switch PCI bridge chip and find its connected bus
- Find the PCI bridge chip that supplied that bus and perform the same enable SERR bit change as described above
- Repeat this sequence until you reach bus 0
- If present, set the SERR bit for the PCI to ISA bridge chip since some computers implement NMI on the ISA bus

Some computers do not require alteration of this SERR Enable bit (also known as SEE), but it is best if implementing a general solution, that the PCI buses are scanned and have the bit enabled.

Enabling PCI Express Error Reporting and Forwarding

In order for the PCI Express Dump Switch to function in your system, some additional PCI Express specific registers must be configured as part of the PCI traversal and modification process. To ensure that Error Reporting and Forwarding is performed:

- Check the device PCI capabilities list for PCI Express capability
- Note the device capability flags and device type
- Activate Fatal Error Reporting in the Device Control capability register
- If this is an Express Root Port, also activate the Fatal Error Forwarding in the Root Control capability register

Using the NMI

Connect Tech's Dump Switch cards are typically used one of two ways:

- Use the native system debugger and configure it to initialize when the button on the Dump Switch card is pushed
  OR
- Hook the NMI interrupt into a program and have it take specific action, such as output data to a serial or parallel port, or save the status to some form of non-volatile memory

QNX4 General Usage Notes

QNX4 systems rely on the computer's BIOS to manage the implementation of NMI support and PCI resource, so setups may vary. Contact support@connecttech.com if you require assistance.

Sample source code (nmi_example.c) is located on the CD that shipped with your Dump Switch card and demonstrates how to perform the necessary initialization.

Most QNX4 systems will require the following two steps:

- Enable NMI support
- Set the SERR Enable bit on any PCI bridge chips between the Dump Switch card and processor

Using the NMI in a QNX4 System

There are two suggested ways of using the Dump Switch card:

- Build a system debugger into the OS image and configure it to initialize when the button on the Dump Switch card is pushed
- Hook into the NMI interrupt in a program and have it take specific action

Debugger Method

To have the system debugger initialize, add the following to your QNX4 operating system build file and then rebuild the image:

```
Sys/Debugger32
$ Debugger32
```

When you boot the image created above, you may need to press `g<enter>` a few times to completely load QNX4.

User Action Method

The sample codes **main.c** and **example.c** provided on your Dump Switch CD provide examples of how the NMI can be set to trigger other actions, such as:

- Store parameters into non-volatile memory
- Output information directly to a device such as a serial or parallel port

Windows NT, 2000, XP General Usage Notes

Ensure the system is set to enable system crash dumps via the Advanced Settings tab of:
Control Panel>System>Startup and Recovery Settings.

Follow this link http://www.microsoft.com/whdc/system/CEC/dmpsw.mspx for more information on Dump Switch support for Microsoft Windows.

Setup Using the Connect Tech Dump Switch Driver

Once installed, the Windows driver and application included on the Dump Switch CD will automatically configure your system to generate a crash dump when requested by the card. For installation instructions, please refer to the readme file found with the driver files.

Manual Setup

In addition to the Windows driver, the settings required to generate a crash dump can also be set manually. To do so, create the following registry value:
HKLM\System\CurrentControlSet\Control\CrashControl\NMICrashDump.
Set the DWORD entry value to 1 to enable the card and 0 to disable (default).When the Dump Switch button is pressed, it will bring the user to the Windows debugger (WinDbg) if connected, otherwise it will produce a crash dump. If the debugger is running, you can choose to use WinDbg to debug the system immediately or begin a crash dump.

1. DP (disable-parity-check) bit is not cleared in I/O location 0x61 (NMI Status Register)
2. NMI is masked by APIC or is not set properly
3. PCI bridges are not SERR enabled on command register at offset 0x04
4. NMI is not enabled on the NMI enable\disable I/O location 0x70 (NMI Mask Register)

NOTE: Some motherboards/operating system combinations may result in a delay. This is due to a time service that checks the validity of the NMI when the application runs during testing.

1. DP (disable-parity-check) bit is not reset in I/O location 0x61

I/O location 0x61 is known as the NMI status register. If bit 2 is set, the Dump Switch card will not operate. Type `ib 61` in WinDbg to see the value. If bit 2 is set, clear it using the WinDbg command `ob`. Do not alter any other bit fields.2. NMI is masked by APIC or is not set properly

Newer motherboards ship with APIC. Type **!apic** in WindDbg to see whether the NMI is masked. The following is a sample output from WinDbg:

```
kd> !apic
Apic @ fffe0000  ID:0 (50014)  LogDesc:01000000  DestFmt:ffffffff  TPR D1
TimeCnt: 05f14d20clk  SpurVec:1f  FaultVec:e3  error:0
Ipi Cmd: 00040041  Vec:41  FixedDel  Dest=Self  edg  high
Timer..: 000300fd  Vec:FD  FixedDel  Dest=Self  edg  high  masked
Linti0.: 0001001f  Vec:1F  FixedDel  Dest=Self  edg  high  masked
Linti1.: 000184ff  Vec:FF  NMI  Dest=Self  lvl  high  masked
TMR: 63, 83, B1, B4
IRR: 41, 63, B1, D1
ISR: D1
```

In this example, Liniti1 NMI is masked. The status location is at offset 0xfee00360 and the status value is 0x000184ff. In this case, bit 16 is set and therefore it is shown as masked.

Unmask the NMI in the APIC by !ed [uc] fee00360 000084ff. To check, use the !apic command. The following output should appear:

```
kd> !ed [uc] fee00360 000084ff
kd> !apic

Apic @ fffe0000  ID:0 (50014)  LogDesc:01000000  DestFmt:ffffffff  TPR D1
TimeCnt: 05f14d20clk  SpurVec:1f  FaultVec:e3  error:0
Ipi Cmd: 00040041  Vec:41  FixedDel    Dest=Self     edg  high
Timer..: 000300fd  Vec:FD  FixedDel    Dest=Self     edg  high      masked
Linti0.: 0001001f  Vec:1F  FixedDel    Dest=Self     edg  high      masked
Linti1.: 000084ff  Vec:FF  NMI         Dest=Self     lvl  high
TMR: 63, 83, B1, B4
IRR: 41, 63, B1, D1, E3
ISR: D1
```

3. PCI bridges are not SERR enabled on command register at offset 0x04

Sometimes SERR are not enabled on PCI bridges on the bus route of the Dump Switch card. Use the !pci command to see the values set for the command register.

```
kd> !pci 2 f

PCI Bus 0
00:0  8086:2530.04  Cmd[0106:.mb..s]  Sts[2090:c....]  Intel Host Bridge  SubID:1028:010d
01:0  8086:2532.04  Cmd[0107:imb..s]  Sts[00a0:.6...]  Intel PCI-PCI Bridge 0->0x1-0x1
1e:0  8086:244e.04  Cmd[0107:imb..s]  Sts[0080:.....]  Intel PCI-PCI Bridge 0->0x2-0x3
1f:0  8086:2440.04  Cmd[000f:imb...]  Sts[0280:.....]  Intel ISA Bridge
1f:1  8086:244b.04  Cmd[0005:i.b...]  Sts[0280:.....]  Intel IDE Controller  SubID:1028:010d
1f:2  8086:2442.04  Cmd[0005:i.b...]  Sts[0280:.....]  Intel USB Controller  SubID:1028:010d
1f:3  8086:2443.04  Cmd[0001:i.....]  Sts[0280:.....]  Intel SMBus Controller  SubID:1028:010d
1f:4  8086:2444.04  Cmd[0005:i.b...]  Sts[0280:.....]  Intel USB Controller  SubID:1028:010d
PCI Bus 0x1
00:0  1002:5446.00  Cmd[0087:imb...]  Sts[02b0:c6...]  ATI VGA Compatible Controller  SubID:1002:0409
PCI Bus 0x2
0a:0  8086:b152.00  Cmd[0107:imb..s]  Sts[0290:c....]  Intel PCI-PCI Bridge 0x2->0x3-0x3
0c:0  10b7:9200.78  Cmd[0117:imb..s]  Sts[0210:c....]  3Com Ethernet Controller  SubID:1028:010d
1f
```

The above shows the output for bus 0 to 2. The *Cmd* column is the value set for any PCI device in the command register. If bit 8 is set then SERR is enabled. In this case, all the bridges have SERR enabled with the exception of the ISA bridge. You can view the details of the configuration registers value:

kd> **!pci 100 0**

PCI Bus 0

00:0 8086:2530.04 Cmd[0106:.mb..s] Sts[2090:c....] Intel Host Bridge SubID:1028:010d

Config Space: (Bus: 0 Device: 0 Func: 0)

00: VendorID 8086 Intel Corporation

02: DeviceID 2530

04: Command 0106 MemSpaceEnable BusInitiate SERREnable

The bridge is SERR enabled as shown:

04: Command 0106 MemSpaceEnable BusInitiate SERREnable

All the bridges should be SERR enabled. In this case the ISA bridge is not SERR enabled:

1f:0 8086:2440.04 Cmd[000f:imb...] Sts[0280:.....] Intel ISA Bridge

If the NMI is on the route of this ISA bridge, then the card will not produce an NMI interrupt. To SERR enable the ISA bridge, use the command **!ecw 0.1f.0 4 10f**.

Check the value again. In the following example, the ISA bridge is now SERR enabled.

kd> **!ecw 0.1f.0 4 10f**

kd> **!pci 2 f**

PCI Bus 0

00:0 8086:2530.04 Cmd[0106:.mb..s] Sts[2090:c....] Intel Host Bridge SubID:1028:010d

01:0 8086:2532.04 Cmd[0107:imb..s] Sts[00a0:.6...] Intel PCI-PCI Bridge 0->0x1-0x1

1e:0 8086:244e.04 Cmd[0107:imb..s] Sts[0080:.....] Intel PCI-PCI Bridge 0->0x2-0x3

1f:0 8086:2440.04 Cmd[010f:imb..s] Sts[0280:.....] Intel ISA Bridge

1f:1 8086:244b.04 Cmd[0005:i.b...] Sts[0280:.....] Intel IDE Controller SubID:1028:010d

1f:2 8086:2442.04 Cmd[0005:i.b...] Sts[0280:.....] Intel USB Controller SubID:1028:010d

1f:3 8086:2443.04 Cmd[0001:i.....] Sts[0280:.....] Intel SMBus Controller SubID:1028:010d

1f:4 8086:2444.04 Cmd[0005:i.b...] Sts[0280:.....] Intel USB Controller SubID:1028:010d

PCI Bus 0x1

00:0 1002:5446.00 Cmd[0087:imb...] Sts[02b0:c6...] ATI VGA Compatible Controller SubID:1002:0409

PCI Bus 0x2

0a:0 8086:b152.00 Cmd[0107:imb..s] Sts[0290:c....] Intel PCI-PCI Bridge 0x2->0x3-0x3

0c:0 10b7:9200.78 Cmd[0117:imb..s] Sts[0210:c....] 3Com Ethernet Controller SubID:1028:010d

4. NMI is not enabled on the NMI I/O location 0x70 (NMI Mask Register)

If after performing the first three steps, the card does not produce a crash dump, then the NMI may not be enabled at offset location 0x70. Bit 7 is responsible for enabling NMI. Use the **ob** command to clear it, and then a dummy read on offset 0x71 by **ib 71**.

Linux 2.4 and 2.6 General Usage Notes

Setup

The Linux usage notes assume a Linux 2.4 or 2.6 kernel with kernel debugger (KDB) is installed and complete with kernel source code. Please note that some steps may differ based on individual configurations.

To begin, ensure that PCI access is enabled (**CONFIG_PCI=y** and **CONFIG_PCI_GOANY=y**) and that the KDB is on (**CONFIG_KDB=y** and **#CONFIG_KDB_OFF is not set**). These options are located in your kernel configuration file.

Modify the file (source_tree)/arch/i386/kernel/traps.c, adding the following to the function do_nmi(...)

```
#ifdef CONFIG_KDB
        kdb_diemsg = "AD005 nmi";
        kdb(KDB_REASON_OOPS, 0, regs)
#endif
```

This code will start the kernel debugger when the Dump Switch button on the card is pressed. Rebuild the kernel and copy it to the appropriate location. Copy the system map file as the KDB needs to provide up-to-date information. Reboot the computer and select the modified kernel for the changes to take effect.

Press the button on the Dump Switch card. A KDB prompt should appear: kdb>

If the prompt does not appear, please verify that you have followed the instructions above correctly.

Newer motherboards will require PCI SERR Enable to be configured. Please see the sample code included on the CD for a working example of how to do this.

If the prompt appears, type in the g command and press enter:
kdb>**g**

Ignore any warnings and enter the g command again.
kdb>**g**

This will return you to the Linux command prompt.

Using the Dump Switch Card in Linux

The information below illustrates how you can use the Dump Switch card to debug your own software. This assumes that you have setup your kernel with KDB, and compiled and run the sample application which is included on the CD.

To enable proper debugging of your application you will need to compile it with debugging symbols added. For gcc this is done with the '-g' flag.
e.g. **gcc -g <filename>.c -o <filename>**

Run your program from a local terminal and when the lock-up you are trying to debug occurs, press the button on the Dump Switch card. The kdb> prompt should appear after a few seconds.

From the kernel debugger:
kdb>**btc <enter>**

This will present information and a stack trace of the current running process.

To see the registers of the processor, type:
kdb>rd <enter>

Record the value found in the eip register (e.g. 0x080483BE).

For assembly language instructions (beginning at the address specified), type:
kdb>id 0x080483BE (the eip register value) <enter>

If you can read the assembly language, you can jump out of infinite loops using the kdb go command with an address of an instruction beyond the end of the loop. For example, kdb>g 0x080483FE <enter>
will start the program at address 0x080483FE.

Once the problem address is located, return to the Linux command prompt. (This may require a reboot).

Enter #objdump -dS <executable file name> | less
This will pipe your program to the less program, where you will see the address values. Finding the address 0x080483BE reveals your code in mixed assembler and C code. This will lead to where the bug occurred. You can now fix the bug.

Limited Lifetime Warranty

Connect Tech Inc. provides a Lifetime Warranty for all Connect Tech Inc. products. Should this product, in Connect Tech Inc.'s opinion, fail to be in good working order during the warranty period, Connect Tech Inc. will, at its option, repair or replace this product at no charge, provided that the product has not been subjected to abuse, misuse, accident, disaster or non Connect Tech Inc. authorized modification or repair.

You may obtain warranty service by delivering this product to an authorized Connect Tech Inc. business partner or to Connect Tech Inc. along with proof of purchase. Product returned to Connect Tech Inc. must be pre-authorized by Connect Tech Inc. with an RMA (Return Material Authorization) number marked on the outside of the package and sent prepaid, insured and packaged for safe shipment. Connect Tech Inc. will return this product by prepaid shipment service.

The Connect Tech Inc. lifetime warranty is defined as the serviceable life of the product. This is defined as the period during which all components are available. Should the product prove to be irreparable, Connect Tech Inc. reserves the right to substitute an equivalent product if available or to retract lifetime warranty if no replacement is available.

The above warranty is the only warranty authorized by Connect Tech Inc. Under no circumstances will Connect Tech Inc. be liable in any way for any damages, including any lost profits, lost savings or other incidental or consequential damages arising out of the use of, or inability to use, such product.

Customer Support Overview

If you experience difficulties after reading the manual and/or using the product, contact the Connect Tech reseller from which you purchased the product. In most cases the reseller can help you with product installation and difficulties.

In the event that the reseller is unable to resolve your problem, our highly qualified support staff can assist you. Our online Support Center is available 24 hours a day, seven days a week on our website at: www.connecttech.com/sub/support/support.asp. Please go to the Download Zone or the Knowledge Database for product manuals, installation guides, device driver software and technical tips. Submit your questions to our technical support engineers at support@connecttech.com. Our technical support is always free.

Contact Information

Telephone/Facsimile
Technical Support representatives are ready to answer your call Monday through Friday, from 8:30 a.m. to 5:00 p.m. Eastern Standard Time. Our numbers for calls are:
Toll: 800-426-8979 (North America only) | Tel: 519-836-1291 | Fax: 519-836-4878 (online 24 hours)

Email/Internet
You may contact us through the Internet. Our email and URL addresses are:
sales@connecttech.com | support@connecttech.com | www.connecttech.com

Mail/Courier
Connect Tech Inc.
42 Arrow Road
Guelph, Ontario, N1K 1S6, Canada