TMS320C62x EVM Interrupt (IRQ) Conflict Resolution

Description   If a PC will not boot properly after an EVM is installed, or an EVM application locks up the PC or causes a Windows exception screen to be displayed, there is probably an interrupt conflict between the EVM and another board in the system. The EVM's plug-and-play functionality provides problem-free interrupt assignments in most cases, but conflicts can arise.

The PC's plug-and-play BIOS automatically determines the appropriate interrupt assignment for each board on the PCI bus. Sometimes this is a difficult task for a computer to do without human intervention. As boards are added and removed, the BIOS can get confused and may end up making inappropriate decisions that can result in an interrupt being shared between the EVM and another board.

The EVM's support software properly handles shared interrupts. However, other boards or their drivers may not be able to share interrupts. This is the root of the IRQ conflict which is quite common with PCs. PCI boards are supposed to be able to share interrupts, so many times the PC's BIOS ends up assigning multiple devices to the same interrupt, such as IRQ9, even if there are other IRQs available. When a board that does not support shared interrupts is assigned the same IRQ as the EVM, the system will not operate properly. In most cases the PC will not boot at all. This problem is a system-level problem, not a problem with your EVM.

PCI network interface cards (NICs) and video adapters tend to be the source of interrupt conflict problems. Two popular boards that have been found to cause this problem are the 3COM 3C905 Etherlink NIC and the STB Velocity 128 (PCI and AGP) video adapter. STB explicitly states on their support web page that this board cannot share an IRQ.

Action   Unfortunately, there is not one solution that applies to all PCs since there are different types of BIOS and releases of the Windows operating systems. The following actions can be taken to address the problem, but every one may not work for your particular system. It is recommended that you read all the items before you take action so that you understand all the potential actions that may apply to your situation.

- It may be helpful to check your system's current interrupt assignments to determine exactly which device is conflicting with the EVM's interrupt.

  To observe the interrupt (IRQ) assignments, perform the following steps within the applicable Windows operating system:

TMS320C62x EVM Interrupt (IRQ) Conflict Resolution

Windows 95

1) From the Windows 95 desktop, right click on the *My Computer* icon.

2) Select the *Properties* menu item.

3) Select the *Device Manager* tab.

4) Double-click on the *Computer* menu item.

5) Select the *Interrupt request (IRQ)* radio button in the *View Resources* tab to display the interrupt assignments.

Windows NT 4.0

1) Select the *Run....* menu item from the Windows NT Start menu.

2) At the Open: prompt type **winmsd** and press Enter.

3) Select the *Resources* tab of the Windows NT Diagnostics window.

4) Clock on the *IRQ* button to display the interrupt assignments.

- The easiest way to resolve an interrupt conflict that works in some cases is to simply move the EVM to another PCI slot. The system's plug-and-play BIOS sometimes assigns a different interrupt to the board when you move it. If the new assignment does not conflict with any other boards, the problem is solved.

- If moving the EVM to another PCI slot does not solve the problem, or there is not another full-size PCI slot available, either remove the other board or have its interrupt physically changed to an unused IRQ if possible.

- Sometimes IRQs are set to *Reserved* in the system's BIOS setup. If these IRQs are not required to be reserved, then it is recommended that they be changed to *Available* to make IRQs available. This feature may not be available with your particular BIOS.

- If all IRQs are already being used in the system, it is recommended that any unused IDE and USB interfaces in the system be disabled in the system BIOS, or in the Windows Device Manager, to free up

interrupts. This may allow the BIOS to assign the EVM to another available interrupt.

- If your system BIOS allows you to manually assign interrupts to specific PCI slots, the problem can be easily solved in this manner by assigning the EVM's slot to an unused IRQ in the BIOS setup.

- Sometimes if you remove and reinstall all of the PCI boards one at a time (rebooting in between each addition), the interrupt conflict may disappear. This procedure allows the system BIOS to start from scratch and reallocate interrupts for each board.

    1) Power off the computer and remove all PCI cards.

    2) Power on the computer momentarily (do not worry about any boot errors because of missing vital cards, such as VGA card, network card, etc.). A few seconds is all that is needed to clear the Plug and Play BIOS resource allocations.

    3) Power off the computer

    4) Power on the computer momentarily, and power off again.

    5) Continue this process of powering on and off the computer, reinstalling each remaining PCI card in between, until all PCI cards have been reinstalled.

    6) Boot the system and check for conflicts. If the EVM is still in conflict, you may need to disable the offending IRQ via the BIOS setup and try reseting the Plug and Play BIOS again, re-enabling the IRQ if needed between the reinstall of two of the PCI cards.

- In some cases, a new version of your system BIOS may be available from the BIOS vendor that supports control over IRQ allocation. Check with your BIOS vendor for further information. Most BIOS vendors provide free BIOS upgrades available from their support web page.

- If you are running under Windows 95, it is possible to manually set the EVM's IRQ assignment. In the BIOS setup, enable the *PnP OS* option if it is available. This will allow Windows 95 to be able to control the plug-and-play information and manually assign interrupts in the Device Manager.

TMS320C62x EVM Interrupt (IRQ) Conflict Resolution

If you machine will not boot Windows at all, then you can force a boot in Safe mode. This should allow you to boot and get to the Device Manager.

The following steps should be peformed:

1) From the Windows 95 desktop, right click on the *My Computer* icon.

2) Select the *Properties* menu item.

3) Select the *Device Manager* tab.

4) Double-click on the *Other devices* item.

5) Double-click on the *TI TMS320C6x EVM* item.

6) The *General* tab should be active. Click on *the Disable in this hardware profile* check box so that it is checked.

7) Click on the *OK* button and reboot.

Once the system is rebooted, the following steps should be performed:

1) From the Windows 95 desktop, right click on the *My Computer* icon.

2) Select the *Properties* menu item.

3) Select the *Device Manager* tab.

4) Double-click on the *Other devices* item.

5) Double-click on the *TI TMS320C6x EVM* item.

6) Click on *the Disable in this hardware profile* check box so that it is **not** checked.

7) Click on the *Resources* tab.

8) Click on the *Use automatic settings* checkbox so that it is not checked.

9) Double click on the *Interrupt Request* entry in the *Resource settings* list.

10) At the *Value:* prompt, select the desired IRQ that does not conflict with other devices, and click on the *OK* button.

11) Reboot the system.

- Since Windows NT 4.0 does not provide plug-and-play support, you are at the mercy of the BIOS interrupt allocation. If your BIOS setup does not provide control over interrupt allocation, then you must resort to vendor or BIOS utilities.

  - Check with your computer manufacturer. Some of them, such as Compaq, include or can provide a utility that allows users to reallocate IRQs as needed.

  - Some BIOS vendors also offer utilities that allow users to manually control IRQs. One of them is available from Intel called ICU configuration utility. It can be used to assign unique IRQs to PCI cards. This action should be attempted as a last resort. This utility is available from the Intel web page below:

    http://developer.intel.com/design/motherbd/gen_indx.htm