

# Large CPCI Passive Backplanes and Real Time

**This article deals with the problem of PCI-to-PCI bridges interrupt dispatching. To avoid the latency introduced by the cascading of bridges, Digital has developed a hardware component: The DMCC Interrupt Accelerator. It always guarantees an interrupt dispatch latency of two bus-read cycles.**

## INTRODUCTION

Passive backplane technology now plays a major role in the embedded and real time market and, over the last decade, backplanes based on the Peripheral Computer Interconnect (PCI) bus have gained an increasing share of that market. In response to the increasing importance of PCI backplanes, the leading vendors formed the PCI Industrial Computer Manufacturers Group (PICMG). The goal of this consortium is to provide an open standard for PCI backplanes, allowing the customer to develop a solution around components which can be interchanged and combined irrespective of vendor.

The demand for more PCI device configurations beyond the strict limits set by the PCI local bus specification has prompted the development of several PCI-to-PCI bridge solutions. When used in combination with a Real Time Operating System (RTOS) however, these bridges can become the source of serious timing problems, due to the non-deterministic nature of interrupt dispatching.

This article will discuss the reasons behind this non-deterministic behavior, and introduce a solution based upon DIGITAL's Interrupt Accelerator technology, as provided on all DIGITAL Modular Computing Components (DMCC) backplanes.

## THE ISSUE

### PCI Interrupt Handling

Interrupts originating in I/O options and peripheral devices have a direct impact on the predictability, reliability and performance of the whole system.

The crucial bottleneck for these interrupts is the bus used for communication between the CPU and the I/O subsystem. In the case of the PCI bus a mechanism is needed to determine which device was responsible for the interrupt. Understanding the handling of interrupts is key to determining whether or not a system will perform adequately in a particular real time environment. The rules for the interrupt handling on a PICMG compliant backplane are defined in the PCI and PICMG standards.

A PICMG single board computer (SBC) slot has four interrupt lines assigned to INTA#, INTB#, INTC# and INTD#, as does each PCI slot. To avoid time consuming interrupt conflicts between I/O components, such as SCSI Host-adapters or FDDI controllers potentially using the same interrupt line, a routing or binding strategy is required. This is achieved by connecting the INTx# line of the SBC to different lines at each slot.

Figure 1 PICMG Single PCI Bus Interrupt Binding, shows the binding strategy for a system without PCI-TO-PCI bridging.

After the hardware has provided the information about the occurrence of an interrupt, a software procedure is necessary to find its source. This typically means polling each device which uses that interrupt. Under Microsoft Windows NT a part of the kernel itself, called the trap handler, does this polling. The trap handler distinguishes between interrupts and exceptions. In Windows NT terms, an exception is synchronous event which can be reproduced when the system runs

**The PICMG standards describe no specific method to make this time deterministic and thus the interrupt service latency is unpredictable.**

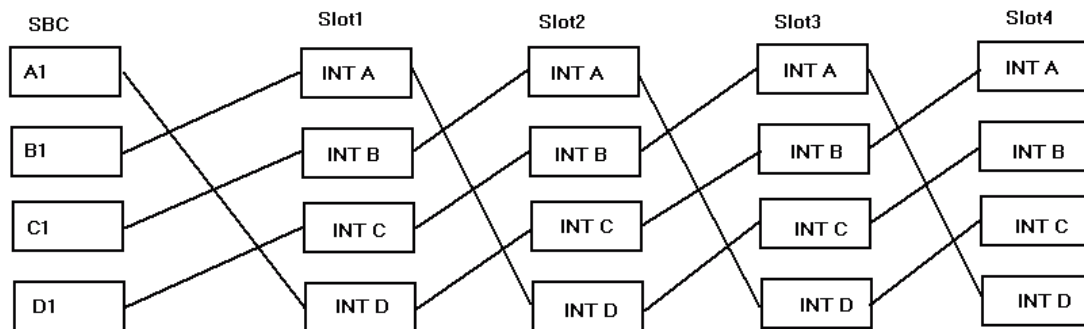


Figure 1 PICMG Single PCI Bus Interrupt Binding

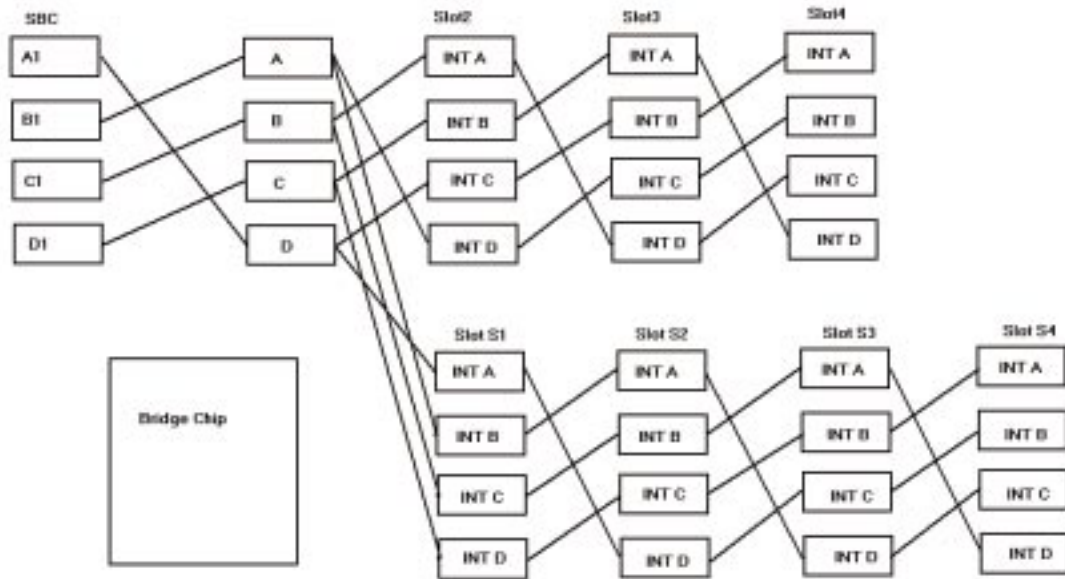


Figure 2. PCI-TO-PCI Bridge Implementation

into the same situation again: for instance a memory access violation. An interrupt is an asynchronous event which can be caused either by hardware, e.g. I/O devices or timer clocks, or software. The routine within the Trap Handler responsible for the Interrupt handling is called the Interrupt Dispatcher. This submodule determines the source of the an interrupt and transfers control either to an external routine, called interrupt service routine (ISR), that handles the interrupt, or to an internal kernel routine. In the case of interrupts from I/O devices, the system always calls an external routine provided by the device driver. Due to the multitude of tasks which this trap handler has to fulfill, a deterministic serving of the interrupt can not be expected.

### Large Passive Backplanes and Interrupt Latency

Nevertheless this process works very well on smaller boards with up to four primary slots and no PCI-TO-

PCI bridging. It provides performance which meets the requirements of a general purpose operating system (GPOS) or network operation system (NOS), and it can even satisfy the needs of a soft real time system.

However there are boards on the market with many more slots, used in a real time environment. All these backplanes use PCI-TO-PCI bridges. Interrupt binding in such configurations is defined in the PCI-to-PCI Bridge Architecture Specification Revision 1.0. This is merely a rudimentary enhancement of the original interrupt binding, and describes only the simple concatenation of two or more PCI-Buses as shown in Figure 2.

**The DMCC Interrupt Accelerator always guarantees an interrupt dispatch latency is of two bus read cycles.**

Interrupt handling becomes of course more complex in this case. It is no longer possible to create a straight assignment between the interrupt line of the SBC and the interrupt line of an particular slot. Two or more devices may now share one interrupt line at the

## REAL TIME ENCYCLOPAEDIA

### Real-Time Magazine

Current issue, Next issue, Index, Subscribe, Contribute, Advertising.

### Real-Time Gazette

Product releases, Commercial news.

### Publications

Books, Magazines, Sites, Directories.



Real-Time Encyclopaedia

### Technology

Research, Terms, Organisations.

### Market

RTOS-list, Players, Careers, User Groups.

### Calendar

Europe, Northern US, Rest of the world.

<http://www.realtime-info.be>

# HW SUPPORT

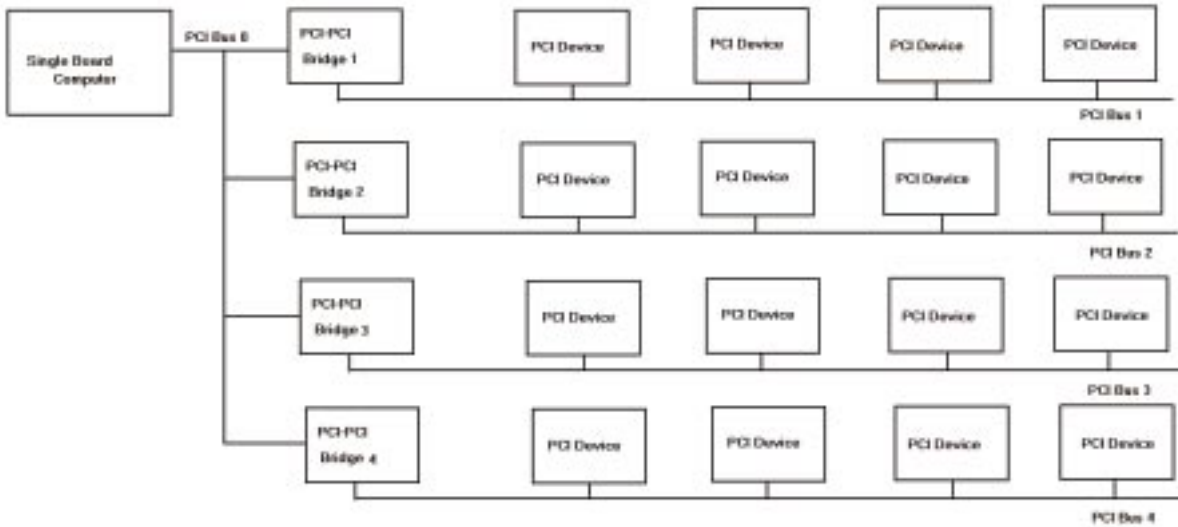


Figure 3. Maximum allowable PCI Configuration with Interrupt Accelerator

SBC level (Figure 3). Hence the time needed to conclude which PCI device sent the interrupt may become much longer. The key problem is the unpredictability of the time needed for interrupt handling. One very critical factor in this context is the interrupt latency. In the case of Windows NT one has to distinguish between interrupt dispatch latency and the interrupt service latency. Interrupt dispatch latency is the elapsed time from receipt of an interrupt request to dispatch to the interrupt service routine. Interrupt service latency is the elapsed time from entry of the interrupt service routine to its completion. The PICMG standards describe no specific method to make this time deterministic and thus, on a typical bridged PICMG compliant backplane, the interrupt service latency is unpredictable. The time determining the originator of the PCI interrupt request could take from

a minimum of 1 up to a maximum of (N-1) bus read cycles where N is the number of the PCI Slots. Thus, assuming a system with only eight PCI Slots, the interrupt service latency can take between one and seven bus read cycles. On a board with the maximum 16 PCI Slots the time could be as high as 15 bus read cycles!

In addition to the problem of unpredictability, the interrupt service latency in very large systems can result in severe degradation of the system performance.

## THE INTERRUPT ACCELERATOR FROM DIGITAL

Digital Equipment Corporation provides a solution for this problem with their DIGITAL Modular Computing

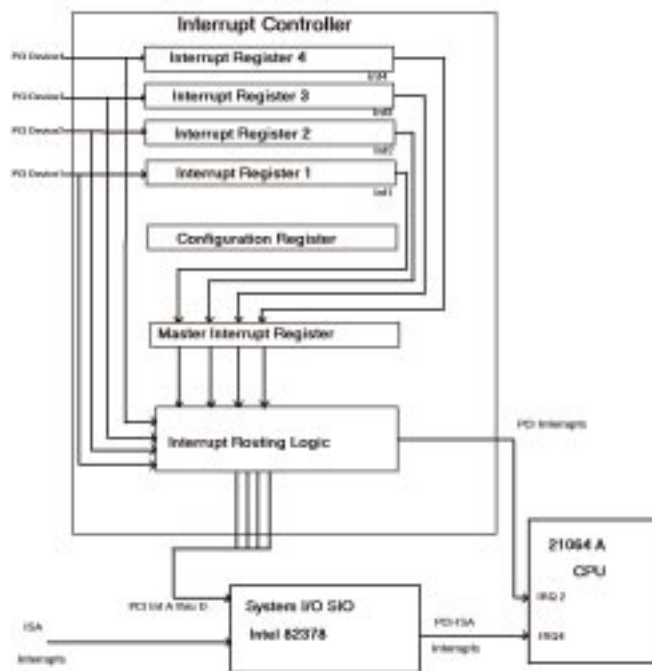


Figure 4. DMCC Interrupt Controller Block

Components (DMCC) backplanes. Known as the Interrupt Accelerator (patents pending) the design goal was to provide improved performance while maintaining a system architecture capable of supporting both conventional (PICMG) and accelerated modes of interrupt source identification. Interrupt Accelerator is detected at boot time and can be activated by the operating system.

In hardware terms the solution is based upon a PCI interrupt controller. This controller must be able to support up to four primary devices. These devices can either be PCI-TO-PCI bridges or PCI slots. Each PCI bridge can support up to four secondary devices behind the bridge. Each bridge can have up to four secondary devices implemented behind the bridge. The maximum configuration would mean a 16 individual PCI slots, as shown in Figure 3.

A controller that can service all of these product scenarios must do the following:

- support up to 4 PCI devices (bridges or slots)
- be able to identify whether a primary device is bridge or a slot
- be able to identify uniquely each of the 16 potential interrupts that can be generated from the PCI bridge devices
- be able to identify uniquely each of the 4 potential interrupts that come from a physical slot

To fulfill these requirements a controller has to have :

- a register to store whether a primary device is a physical slot or a PCI-PCI Bridge
- a register for each primary device to provide the status of each PCI interrupt supported by that primary device
- a register to identify which primary device caused an interrupt.

The basic form of the interrupt controller is shown in Figure 4 DMCC Controller Block Diagram. To maintain 100% compatibility with PICMG standards, the DMCC backplanes support PICMG mode, i.e. where interrupt acceleration is disabled and interrupt handling performed by conventional interrupt routing logic.

The interrupt accelerator needs to

perform two steps to identify an device, in step number one it checks whether the primary device is a primary slot or a PCI-TO-PCI bridge, if the device is an PCI-TO-PCI bridge the Interrupt Accelerator determines in a second step which secondary device caused the interrupt. In other words, the DMCC Interrupt Accelerator always guarantees an interrupt dispatch latency is of two bus read cycles. This is a solid foundation on which developers can implement real time functionality with Windows NT on even huge passive backplanes. This technology has been so successful that DIGITAL can offer DMCC passive backplanes today which scale between 4 and 13 PCI slots. Windows NT on DIGITAL Alpha already supports Interrupt Accelerator Technology, providing the perfect platform for the development of high reliability and high performance real time solutions. Due to this fact, DIGITALs Interrupt Accelerator technology is gaining more and more acceptance on the market not only used along with DMCC boards, DIGITAL already licensed Interrupt Accelerator technology to ICS.

## SUMMARY

The problem of large passive backplanes in a real time environment, especially under Windows NT, has been a major challenge in recent years. DIGITAL has found a reliable answer to this problem in the Interrupt Accelerator for its DMCC platform. DMCC provides the deterministic behavior which developers need to implement real time systems successfully. Together with leading 64-Bit DIGITAL Alpha technology, DMCC can provide a platform powerful enough to meet even the highest demands of today's developers. ■

---

*Werner Reuss is a 30 years old engineer that studied computer science. He joined DIGITAL in 1992 and held several positions as an engineer in the support and service area. Mr. Reuss is now working for round about three years for the Components division. He is focused on all aspects of windows NT within his group, like NT, real-time NT and VME fault tolerant systems.*

## DEC\_2 Ad