Hardware White Paper

Designing Hardware for the Microsoft® Windows® Operating Systems

Removing the ISA Architecture in Windows-Based Platforms

Microsoft encourages ISA-free system design based on the hardware incompatibilities and configuration limitations in the ISA legacy architecture under the Microsoft® Windows® 2000 and Windows 98 operating systems.

For the purpose of planning, Microsoft is phasing out ISA support in future versions of Windows 2000 and Windows 98 (after Windows 98 Second Edition). For backward compatibility reasons, Windows 98 will provide the most compatibility for ISA designs; the version after Windows 98 Second Edition will support a completely ISA-less system. Windows 2000 has limited ISA support today and future versions will support a completely ISA-less system.

DRAFT: June 2, 1999

Contents

Introduction

ISA expansion devices are a well-documented cause of costly support problems that continue to burden end users and the PC industry at large because of hardware incompatibilities and configuration limitations inherent in the ISA legacy architecture. Designing ISA-free and legacy-free systems will result in easier and more stable system configuration, lower support cost, the elimination of many driver-related issues, and improved performance.

Microsoft strongly recommends moving away from the continued design and manufacture of ISA devices, as rapidly as possible. Microsoft has limited new work on engineering support for ISA in its operating systems because the ISA expansion bus architecture is fundamentally incapable of accurately representing the topology of devices that reside under it, making it impossible to create true Plug and Play systems.

With the introduction of Plug and Play in the Windows 95 operating system, efforts were made throughout the industry to advance ISA resource allocation capabilities through ISA Plug and Play (see the related Plug and Play specifications at http://www.microsoft.com/hwdev/pnpspecs.htm).

However, industry efforts at solving the ISA resource allocation problems could achieve only limited success because ISA resources can be fixed and therefore cannot be dynamically re-allocated. The inability of the ISA bus to accurately report what resides under it plus its inability to respond dynamically to changing device demands for resources is the root cause of many operating problems. In addition, the ISA bus architecture does not report sufficient information to reliably support advanced platform capabilities such as power management and docking.

In current PCs, the PCI, USB, and IEEE 1394 buses can provide all the expansion capability users need for a true Plug and Play experience. The small cost incurred by phasing out the ISA bus as quickly as possible and putting all functions on the PCI bus is worth paying.

Issues with ISA

The foundation of the true Plug and Play experience is that the operating system always knows how to react when the system configuration changes. The operating system uses the topology of the various devices residing under a given bus to determine what to do with the devices. Each bus can have a number of child devices and might also feature a bridge to another bus, which could also have child devices residing under it.

All this information about devices within the system, and the specific location of each one, represents a logical hierarchy that the operating system uses to assign resources in a true Plug and Play environment. For example, if a USB hub is removed, the operating system quickly becomes aware that all child devices under the USB hub are no longer available. The operating system can then free the resources the child devices were using, so other devices within the system can take advantage of the full range of available resources. To support this capability, the operating system must know what types and amounts of resources these child devices need so it can intelligently assign proper resources and determine how to reallocate resources when they become available again. With PCI, for example, everything is inherently configurable by the PCI controller chip.

A fundamental issue with the ISA bus architecture is that it cannot accurately represent and report the topology of devices that reside under it. The ISA bus is hard-coded; it is not self-configurable. All the addresses are set and unchangeable. With ISA, the operating system cannot know absolutely where the devices are located and what resources these devices need. Because the resources are fixed, the operating system may not know that the device has been removed; nor can it know what resources are now available to reallocate to other parts of the system.

Windows and ISA Plug and Play

Although the ISA Plug and Play effort attempted to address the limitations of the ISA bus architecture, the results are still disappointing, especially in comparison with PCI and technologies. ISA Plug and Play remains an unreliable solution. Although ISA Plug and Play at least answers the question of *what* an ISA device is (identification of the device), the operating system still cannot know where the ISA device is or when it arrived.

A “true” Plug and Play system needs the following in a bus architecture and from the devices that reside on that bus:

Discovery

- A mechanism for detecting when a device is inserted and when a device is removed.
- A mechanism for identifying a device. In PCI, for example, this is accomplished through the device ID, vendor ID, and subsystem vendor ID.
- The ability to determine where in the bus/device hierachy a given bus/device resides.

Configuration

- The ability to determine the resources required by a device
- The ability to programatically assign resources to a device (examples include the I/O address range, Interrupt Request Lines, Memory Window)

Power Management (heavily dependent on topological information established in discovery phase)

- Ability to query for the power management capabilities of a bus/device.
- Ability to manipulate power states of a bus/device.

Legacy ISA systems provide none of the above capabilities. ISA Plug and Play only partially addresses the shortcomings of Legacy ISA. Here is how ISA Plug and Play in Windows 2000 addresses the same needs:

Discovery

- Windows is able discover a new ISA Plug and Play card only at boot. That is, no events exist to tell the operating system when a device is inserted or removed. The operating system has no way of determing where a device resides in the system topology.

Configuration

- Windows can determine an ISA Plug and Play card's resource requirements and can assign it resources. However, ISA Plug and Play devices are not as flexible as PCI devices in regard to the range of resources they will accept.

Power Management

- No support whatsoever is possible. Without topological information, the operating system is incapable of establishing device relations. Without device relations, Windows cannot power manage a device.

ISAPNP is not an acceptable alternative to legacy ISA.

Windows 95/98 includes extensive device-specific coding to ensure that many common ISA components that are not fully Plug and Play-compliant can still be recognized and configured by the operating system. Very modest success for Plug and Play has been achieved on systems that have BIOSes and ISA devices compliant with the *Plug and Play ISA Specification, Version 1.0a*.

Under Windows 2000 and Windows 98, Plug and Play capabilities are fully realized on systems that have ACPI-compliant BIOSes and that contain buses and devices that can report hardware and resource requirements using ACPI mechanisms. However, Windows 2000 also provides support for ISA Plug and Play for devices compliant with the *Plug and Play ISA Specification, Version 1.0a*, but Windows 2000 does not contain extensive coding to support a broad range of individual non-compliant devices.

However, Windows 2000 does not support ISA Plug and Play on docking stations. Eliminating this support made possible many optimizations in the operating system—for example, Windows 2000 does not perform a full re-enumeration before and after a dock or suspend, with the result that Windows 2000 is much faster on docking and suspend/resume than Windows 98. The operating system assumes for many other operations that there will be no ISA add-on cards in the docking station, making it impossible to ensure what might work.

Because Windows 2000 cannot determine what happens with ISA devices on docking stations, Microsoft will not provide product support to help customers debug problems with ISA add-on cards in the docking station, and will tell customers not to put ISA add-on cards in docking stations.

ISA Limitations and Problems

This section lists some of the specific problems and poor performance associated with the ISA bus.

Non-dynamic Resource Allocation. ISA pre-allocates resources for legacy devices within the operating system. As a result, resources are allocated for a device even if it is not present in the system. This wastes resources that could be used by the system or other devices, and often makes it impossible to correctly configure a new device being added into the system, because there are no free resources.

Configuration Space Limitations. Configuration registers allow the operating system to programmatically determine the true resources required by devices at any given moment, helping it to assign the appropriate resources to each device. ISA devices lack built-in configuration registers; they do not have the configuration space that allows the system to soft configure the resources allocated to a given device. Hard-coded ISA resources cannot be dynamically reallocated to other devices. This causes a number of problems, including:

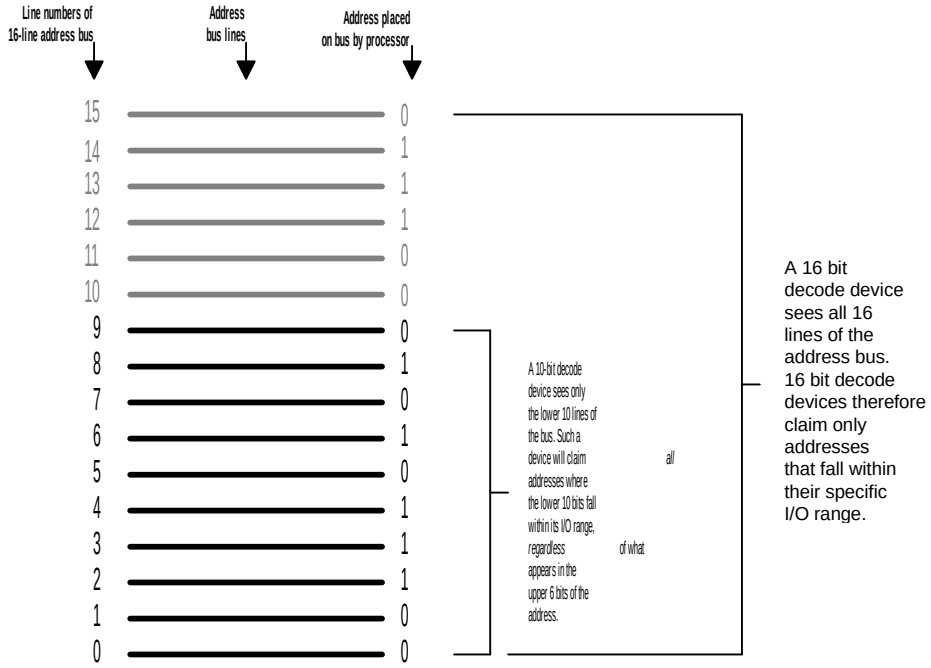
- Resources cannot be programmatically assigned to a card.
- Resources—such as interrupts—cannot be dynamically reallocated by the operating system after a card is removed.
- Devices cannot be reliably configured due to limitations inherent in the hardware, even with ISA Plug and Play.
- Devices cannot be hot-swapped because resources that might be required are already hard-coded to ISA devices.
- Interrupts cannot be shared among devices.
- Systems with VGA devices behind PCI-PCI bridges are prone to BIOS bugs.

PCI-to-ISA Bridge Complications. Because PCI and ISA perform so differently, it is a complex problem to bridge between the two buses. Despite the inherent difficulties, the PCI-to-ISA bridge is a common system scenario because of the volume of computer systems manufactured and sold that feature both systems. A bridge is required between the two buses in order to expand the system, but the requirements for the PCI and ISA buses are significantly different, making bridging extremely

complicated. In a strictly PCI system, these complications disappear and the system performs significantly better; it is also less susceptible to other malfunctions.

Resource Allocation Problems with 10-bit Decoding. Reliance on 10-bit decoding is a significant problem with ISA devices. Because a system design cannot rely on any ISA device appearing uniquely in the 16-bit address space, a design must assume that 10-bit devices will also appear as phantom devices above the 10-bit range. One effect of this dependence is that resources are scattered throughout the configuration space. Because ISA devices interpret resources being assigned to them in a different manner than do PCI devices, the operating system is not able to assign contiguous resources to ISA devices. Resources for these ISA devices end up scattered throughout the configuration space, which is confusing for the operating system that is attempting to identify and configure the device—there is no mechanism for the operating system to determine whether a device is real or its "phantom" counterpart. Therefore, to ensure system stability, the operating system must treat all phantom devices as real resources. In addition to being inefficient, this can affect system performance and reliability.

The horizontal lines in the following figure represent an address bus with 16 lines. A device that is capable of handling full 16-bit decode sees all 16 lines of the address bus (both the black and the grey lines.) However, a device that is only capable of handling 10-bit decode sees only the lower 10 lines of the bus (the black lines only.)



To address this issue, reserve *all* 16 bit addresses claimed by a 10 bit card, for use by the 10 bit card. This guaruntees no conflicts, but is very inefficient use of I/O space. Implement Subtractive Address decoding on ISA cards or bridges to ISA buses. Subtractive decoding ensures an ISA device or PCI/ISA bridge will only have an opportunity to claim an address after PCI devices have had a chance to claim the address. Subtractive decode places limitations on system design (only one subtractive decode device per bridge) and increases complexity of resource arbitration code in the operating system.

No Hot Swapping of Cards. Hot swapping of cards (similar to PCI hot-swapping) is not possible with ISA devices because of the requirements for flexible resources, interrupts, and I/O ranges.

Slow DMA. Direct Memory Access (DMA) is significantly slower with ISA. Systems with ISA audio or modem devices reduce available bandwidth to the host processor and other devices in the system during DMA operations.

Modems and Audio Device Demands. ISA modems and audio devices require more CPU resources to run than their PCI counterparts.

Slow Bus Speed. ISA is extremely slow compared to bus technologies such as PCI and IEEE 1394. All transfers performed over the ISA bus are synchronized to roughly an 8 MHz bus clock signal. Because the data path on the ISA bus is only 16 bits wide, a maximum of only two bytes can be transferred during each transaction. This results in an approximate effective data transfer rate of 1.2 Mb/s.

VGA Devices behind PCI-to-PCI Bridges. The Windows test teams have seen a number of system BIOSes that do not correctly configure PCI-to-PCI bridges in systems that have a VGA device behind a bridge. Specifically, these BIOSes do not correctly set the VGA Enable and ISA Enable bits on their bridges, causing the bridges to be in conflict with each other.

To properly configure a system at boot time, the BIOS must set the VGA Enable register on all bridges that have a VGA device behind them. This includes the Accelerated Graphics Port (AGP) bridge, which is treated like a normal PCI-to-PCI bridge in this situation. In addition, the BIOS must set the ISA Enable bit on all bridges that are peers to bridges that have the VGA Enable bit set.

When a bridge does not have the ISA Enable bit set, it will decode all addresses within the bridge address window. This includes the VGA aliases, because the bridge forwards those addresses to its secondary bus. Because the bridge is decoding those addresses, the operating system allocates those addresses to the bridge. Thus, these addresses are not available for use by any other peer device, including other bridges.

The ISA bit must be set on bridges in order to be able to communicate with ISA devices in the system. This is a specific example of bridging technology becoming more complicated because the operating system design has to deal with the *potential* case of an ISA device present somewhere within the system, even if it is not actually there. The ISA bit must be set on the peer bridges of a particular bridge. For more information, see *Configuring PCI-to-PCI Bridges with VGA Cards* at http://www.microsoft.com/hwdev/pci/vgacard.htm.

Call to Eliminate ISA in PC Systems

The hardware incompatibilities and configuration limitations inherent in the ISA legacy architecture in a true Plug and Play environment make continued support of ISA a questionable strategy for vendors. The goal for IHVs and OEMs should be to switch to legacy-free or legacy-reduced system designs as soon as possible.

ISA-Free Design Guidelines. The following summarizes the guidelines for designing new systems that do not rely on ISA. These guidelines are particularly important for docking stations for mobile systems that will run the Windows 2000 operating system:

- No ISA slots
- No PCI-to-ISA bridges in the docking station
- Include only ACPI-enumerated ISA devices, if any

- Implement solutions that follow the *Low Pin Count (LPC) Interface Specification* (see http://developer.intel.com/design/chipsets/industry/lpc.htm), which defines a new interface between the core logic chipset and motherboard legacy I/O functions

For more information about the migration path for manufacturers to remove ISA and other legacy from PCs, see the white paper *Legacy I/O Removal to Advance the PC Architecture* at http://www.microsoft.com/hwdev/newpc/.

For additional design guidelines for mobile PCs, see the presentations from the *1998 Mobile Platform Design Review* available at http://www.microsoft.com/hwdev/mobilepc/.

See also *ACPI Docking for Windows 2000* at http://www.microsoft.com/hwdev/onnow/acpidock.htm.

ISA Implementation Guidelines. If you are still supporting and implementing ISA in your system and device designs, see *Legacy Plug and Play Guidelines* at http://www.pcdesguide.org for the compatibility requirements for ISA in Windows 98 and Windows 2000 to ensure that cards will perform correctly within Windows.

For Windows 2000 driver implementation guidelines for ISA Plug and Play devices, see "Part 2: Plug and Play" in the *Setup, Plug & Play, Power Management Design Guide* in the Windows 2000 DDK at http://www.microsoft.com/ddk/ddkdocs/win2k/.

Note: The "Designed for Windows" Logo program requires ISA-free systems, and does not accept ISA devices for testing.

Resources:

IDs and Serial Numbers for ISA Plug and Play
http://www.microsoft.com/hwdev/busbios/idpnp.htm

Request an ISA Plug and Play ID
http://www.microsoft.com/hwdev/pnpid.htm.

Plug and Play Technology
http://www.microsoft.com/hwdev/plugnplay/default.htm.

System Hardware Compatibility Tests (HCTs) from Windows Hardware Quality Labs
http://www.microsoft.com/hwtest/.