CPE 186 Computer Hardware Design

Configuration Transactions (Part 1)

Dr. Pang

Who Performs Configuration?

- Initially, the BIOS code performs device configuration. Once a Plug-and-Play OS has been booted and control is passed to it, the OS takes over device management.
- it is a configuration program executing on the processor that performs system configuration.
- The host processor must have some way to instruct the host/PCI bridge to perform configuration read and write transactions on the PCI bus.

Introduction

The programmer must supply the following information to the host/PCI bridge when performing a configuration read or write:

- target PCI bus.
- target PCI device (i.e., package) on the bus.
- target PCI function within the device.
- target dword within the function's configuration space.
- target byte(s) within the dword.

Introduction

The bridge must then determine If the target PCI bus specified **is:**

- the bus immediately on the other side of the host/PCI bridge (in other
- words, Bus 0)
- a bus further out in the bus hierarchy
- none of the buses behind the bridge.

The bridge compares the target bus number specified by the programmer to the range of buses that exists beyond the bridge. There are three possible cases:

- **CASE 1.** The target bus is bus 0.
- **CASE 2.** The target bus isn't bus 0. but is less than or equal to the value in the Subordinate Bus Number Register. In other words, the transaction targets a device on a subordinate bus.
- **CASE 3.** The target bus doesn't fall within the range of buses that exists beyond this bridge.

Case 1: Target Bus Is PCI Bus 0

Type **0** configuration transaction to indicate to the devices on Bus **0** that one of them is the target of this configuration transaction.:

- Bridge that "owns" that bus has already performed the bus number comparison and verified that the request targets a device on its bus.
- Setting AD[1:0] to 00b during the address phase

Case 2: Target Bus Is Subordinate To Bus 0

Type **1** configuration transaction:

- It targets a device on a bus further out in the hierarchy beyond a PCI-to-PCI bridge that is attached to Bus 0**.**
- It instructs all functions other than PCI-to-PCI bridges that the transaction is not for any of them.
- Setting AD[1:0] to 01b during the address phase

- **Must Respond To Configuration Accesses Within 2^25 Clocks After RST#**

Background

- Intel **x86** and PowerPC processors (as two examples processor families) do not possess the ability to perform configuration read and write transactions. They use memory and IO (IO is only in the x86 case) read and write transactions to communicate with external devices.
- This means that the host/PCI bridge must be designed to recognize certain IO or memory accesses initiated by the processor as requests to perform configuration accesses.

Background

The x86 processor family is capable of addressing up to, but no more than, 64KB of IO address space.

- As with any other PCI function, a host/PCI bridge may implement up to 64 dwords of configuration registers.
- Each PCI function on each PCI bus requires 64 dwords of dedicated configuration space.

Configuration Mechanism

This mechanism utilizes two 32-bit IO ports **:**

- The 32-bit Configuration Address Port, occupying IO addresses 0CF8h ~ 0CFBh.
- The 32-bit Configuration Data Port, occupying IO addresses 0CFCh ~ 0CFFh.

Configuration Mechanism

Accessing one of a PCI function's configuration registers is a two step process:

- **STEP 1.** Write the target bus number, device number, function number and dword number to the Configuration Address Port and set the Enable bit in it to one.
- **STEP 2.** Perform a one-byte, two-byte, or four-byte IO read from or an write to the Configuration Data Port.

Configuration Mechanism

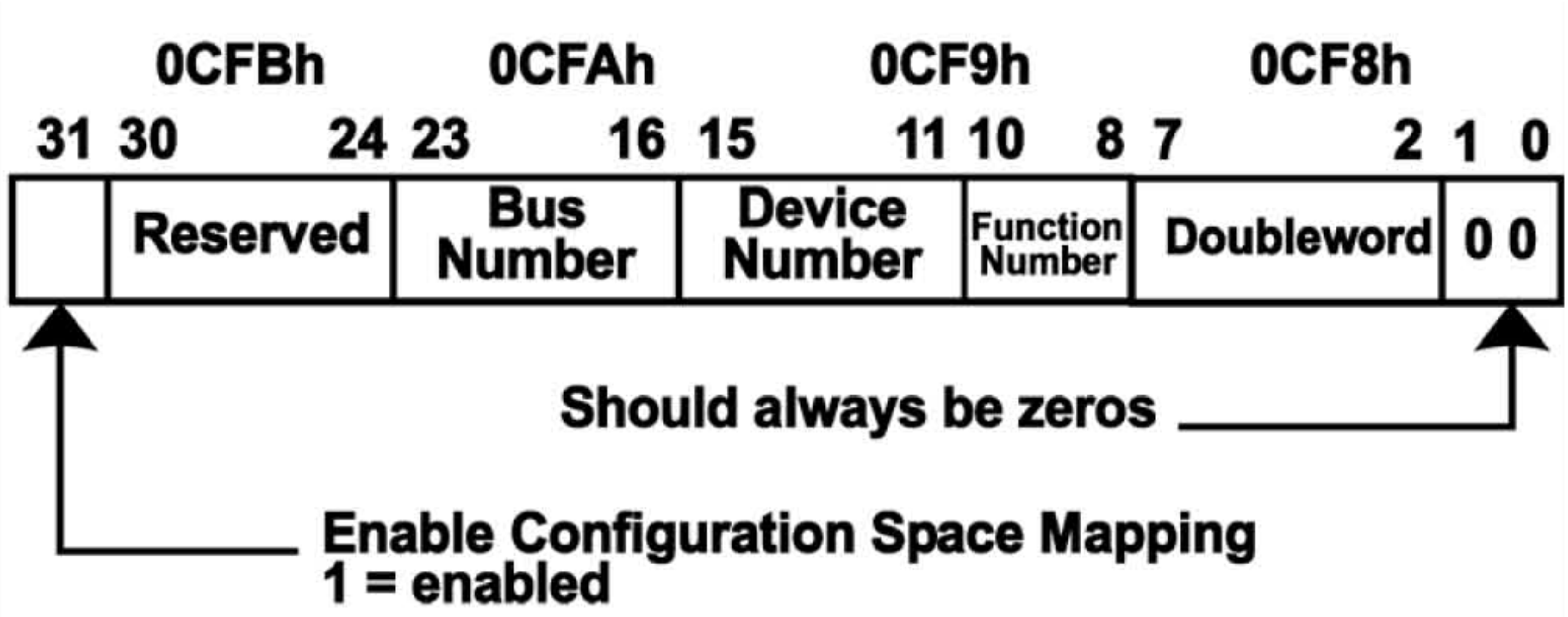
- In response, the host/PCI bridge compares the specified target bus to the range of buses that exist on the other side of the bridge and, if the target bus resides beyond the bridge, it initiates a PCI configuration read or write (based on whether the processor is performing an IO read or write with the Configuration Data Port).

Configuration Address Port

The assertion of reset clears the port to all zeros. Any 8- or 16-bit access within this IO dword is passed directly onto the PCI bus as an 8- or 16-bit PCI IO access. The 32-bits of information written to the Configuration Address Port must conform to the template shown on the next slide:

- bits [1:0] are hard-wired, read-only and must return zeros when read.
- bit 31 must be set to a one, enabling the translation of a subsequent processor IO access to the Configuration Data Port into a configuration access on the PCI bus.
- If bit 31 is zero and the processor initiates an IO read from or IO write to the Configuration Data Port, the transaction is passed through to the PCI bus as a PCI IO transaction.

Configuration Address Port at 0CF8h



0CFBh		0CFAh	0CF9h		0CF8h	
31	30 24	23 16	15 11	10 8	7 2	1 0
	Reserved	Bus Number	Device Number	Function Number	Doubleword	0 0

Should always be zeros

Enable Configuration Space Mapping
1 = enabled

Bus Compare, and Data Port Usage

Each host/PCI bridge implements

- a Bus Number register (in a chipset that only supports one host/PCI bridge, the bridge may have a bus number register that is hardwired to 0, a read/write register that reset forces to 0, or it **Just** implicitly knows that it is the bridge to PCI bus 0)
- and a Subordinate Bus Number register.

Bus Compare, and Data Port Usage

If bit 31 in the Configuration Address Port is enabled (set to one), the bridge compares the target bus number to the range of buses that exists beyond the bridge.

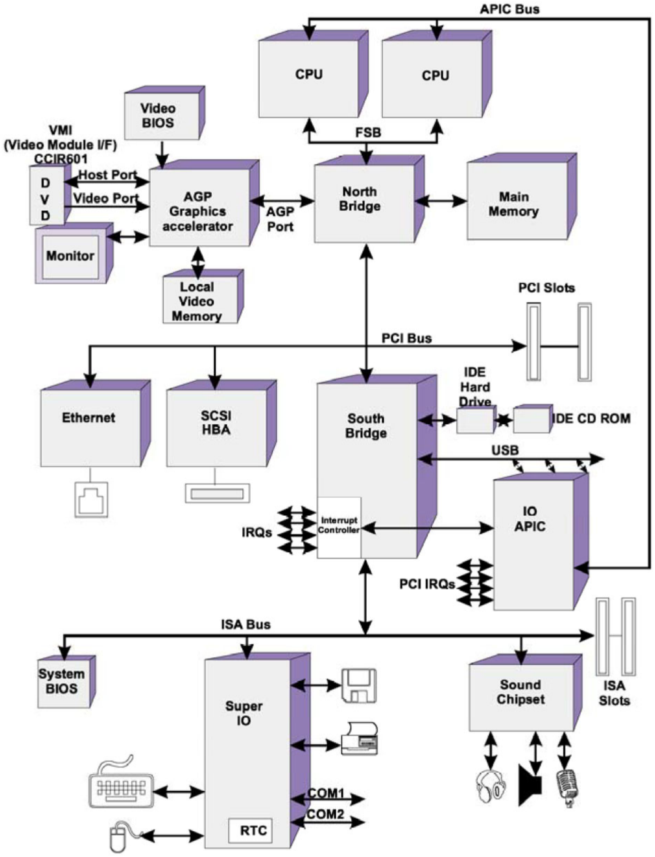
- If the target bus is the **same** as the value in the Bus Number register, this is a request to perform a configuration transaction on PCI bus 0. A subsequent IO read from or write to the bridge's Configuration Data Port at 0CFCh causes the bridge to generate a Type 0 configuration read or write transaction.
- When devices that reside on a PCI bus detect a Type **0** configuration transaction in progress, this informs them that one of them is the target device.
- If the target bus specified In the Configuration Address Port does not match the value in the bridge's Bus Number register, but is equal to or less than the value in the bridge's Subordinate Bus Number register, the bridge converts the subsequent processor IO access to its Configuration Data Port into a Type 1 configuration transaction on its PCI bus.
- When devices that reside on a PCI bus (other than PCI-to-PCI bridges) detect a Type 1 configuration access in progress, they ignore the transaction.

Bus Compare, and Data Port Usage

The only devices on a PCI bus that pay attention to the Type 1 configuration transaction are PCI-to-PCI bridges.

- If the target bus is not within range, then a PCI-to-PCI bridge ignores the Type 1 access.
- If it's in range, the access is passed through the PCI-to-PCI bridge as either a Type **0** configuration transaction (if the target bus compares to the bridge's Secondary Bus Number register), or as a Type 1 transaction (the target bus number is equal to or less than the value in the bridge's Subordinate Bus Number register).

Single Host/PCI Bridge

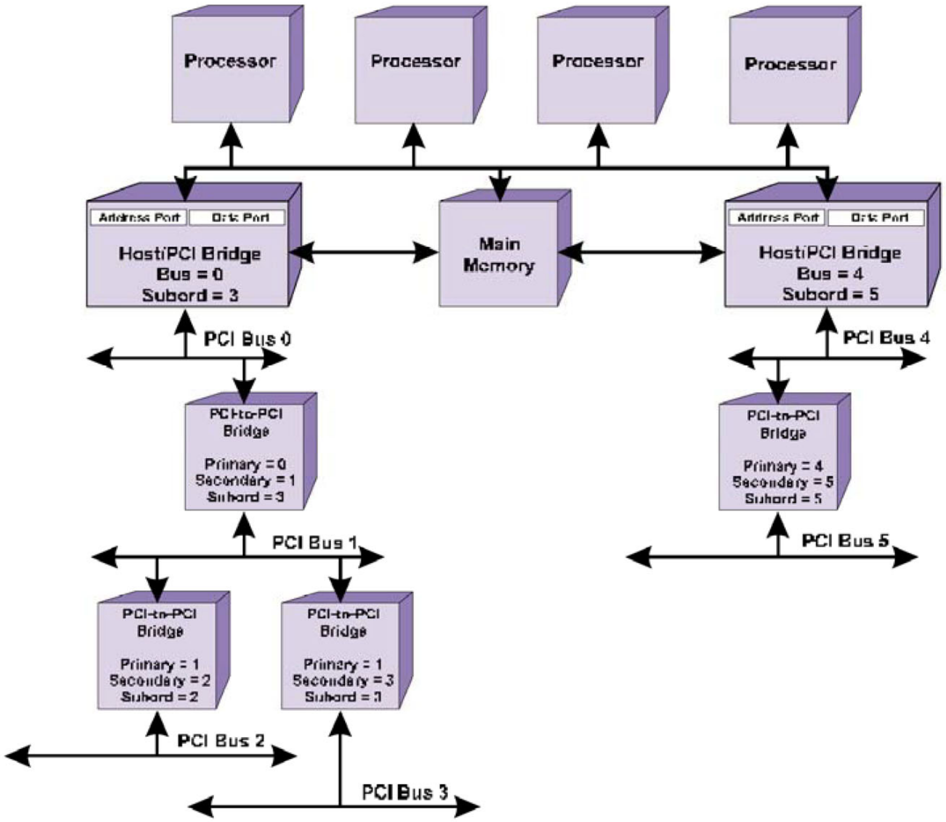


Single Host/PCI Bridge

The information written to the Configuration Address Port is latched by the host/PCI bridge.

- If bit **31** is set to one and the target bus number compares to the bridge's PCI Bus Number register (or is equal to or less than the value in the bridge's Subordinate Bus Number Register), the bridge is enabled to convert a subsequent processor access targeting its Configuration Data Port into a PCI configuration access. The processor then initiates a one-byte, two-byte, or four-byte (indicated by the processor's byte enable signals).

Multiple Host/PCI Bridges



Multiple Host/PCI Bridges

If there are multiple host/PCI bridges present on the processor bus, the Configuration Address and Data Ports are duplicated at the same IO addresses in each of the host/PCI bridges.

In order to prevent contention on the processor's bus signals, only one of the bridges responds to the processor's accesses to the configuration ports.

Multiple Host/PCI Bridges

- **STEP 1.** When the processor initiates the IO write to the Configuration Data Port, only one of the host/PCI bridges (typically, the bridge to PCI bus 0) actively participates in the transaction. The other bridge quietly snarfs the data as it's written to the active participant.
- **STEP 2.** Both bridges then compare the target bus number to their respective Bus Number and Subordinate Bus Number registers. If the target bus doesn't reside behind a particular host/PCI bridge, that bridge doesn't convert the subsequent access to its Configuration Data Port into a PCI configuration access on its PCI bus (in other words, it ignores the transaction).
- **STEP 3.** A subsequent read or write access to the Configuration Data Port is only accepted by the host/PCI bridge that is the gateway to the target bus. This bridge responds to the processor's transaction and the other ignores it.
- **STEP 4.** When the access is made to the Configuration Data Port. the bridge with a bus compare tests the state of the Enable bit in its Configuration Address Port. If enabled, the bridge converts the processor's IO access into a PCI configuration access.
- If the target bus is the PCI bus immediately on the other side of the host/PCI bridge, the bridge converts the access to a Type 0 configuration access on its PCI bus.
- Otherwise, it converts it into a Type 1 configuration access.

Generation of Special Cycles

- Host/PCI bridges are not required to provide a means that allows software to initiate a Special Cycle transaction on a target PCI bus. If this capability is provided by the bridge, however, the following describes how it must be implemented.
- To prime the host/PCI bridge to generate a PCI Special Cycle transaction, the host processor must write a 32-bit value with the following content to the Configuration Address Port at IO address 0CF8h:

Generation of Special Cycles

- Bus Number = the target PCI Bus that the Special Cycle transaction is to be performed on.
- Device Number = all ones (31d. or 1Fh).
- Function Number = all ones (7d).
- Dword Number = all zeros.

Generation of Special Cycles

- After this has been accomplished, the next write to the Configuration Data Port at IO port 0CFCh causes the target bus's bridge to generate a PCI Special Cycle transaction on its secondary PCI bus. The data written to the host/PCI bridge's
- Configuration Data Port is supplied **as** the message on AD[31:0] during the transaction's data phase.
- If the host/PCI bridge's Bus Number does not match the specified target bus number, but the target bus number is within the range of buses subordinate to the bridge's bus, the bridge passes the transaction through as a Type 1 configuration write (so that it can be submitted to PCI-to-PCI bridges further out in the hierarchy).
- If the host/PCI bridge has been primed to generate a PCI Special Cycle transaction (by writing the appropriate data to the Configuration Address Port) and an IO read is performed from the Configuration Data Port, the result is undefined.
- The bridge may pass it through as a Type 0 configuration read (which will result in a Master Abort with all ones returned as data).