

Peripheral Component Interconnect (PCI)

Babak Kia
Adjunct Professor
Boston University
College of Engineering



ENG SC757 - Advanced Microprocessor Design

Why PCI?

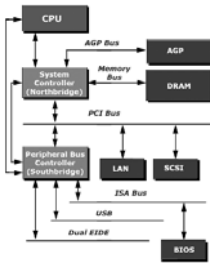
- The original PC bus developed by IBM in 1982 was 16 bits wide and operated at 4.77 MHz
- This was known as the ISA bus, capable of transferring data at a whopping rate of 9 Mbytes per second!
- Gradually peripherals such as video cards and hard drives required a higher bandwidth, and in 1992 Intel introduced the PCI bus standard to allow connected devices direct access to the system memory
- It detached the speed of the peripheral bus from that of the memory, so even though today's Front Side Buses can run up to 800 MHz, the PCI devices can operate at their own independent speed

What is PCI?

- The Peripheral Component Interconnect is an interconnect bus developed by Intel in 1992 which runs at 33 MHz and supports plug-and-play
- It allows high speed connection between peripherals, and from the peripherals to the processor
- Allows for transfer of data amongst peripherals independently of the processor
- Found on many desktops, but not limited to them, the PCI bus is a 32 bit wide bus capable of transferring at data rates up to 132 MBytes per second
- A 66 MHz, 64-bit version is capable of transfer rates of up to 524 Mbytes/second

PC System Architecture

- Northbridge is the chipset which interfaces with memory, PCI bus, level 2 caches, Accelerated Graphics Port (AGP), on the Front Side Bus (FSB)
- Southbridge is the chipset which handles the basic I/O such as USB, serial, audio, IDE devices and ISA bus



PC System Architecture

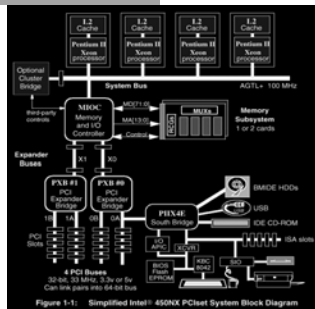
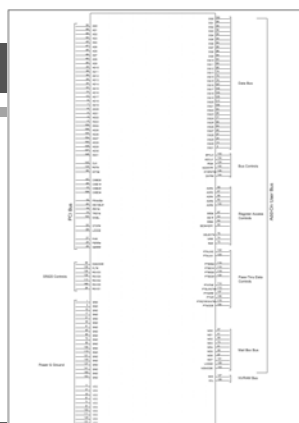
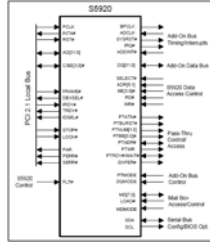


Figure 1-1: Simplified Intel® 450NX PCle System Block Diagram



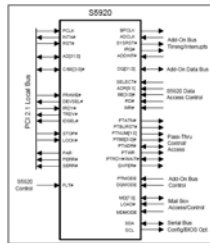
PCI Bus Signals

- AD[31:0] – Multiplexed Address/Data
- C/BE[3:0]# - Command/Byte Enable
 - 0110: Memory Read
 - 0111: Memory Write
- PAR – Parity, always driven as even for all AD[31:0] and C/BE[3:0] signals
- PCLK – PCI Clock, operates from DC (0 Hz) to 33 MHz
- RST# - Reset



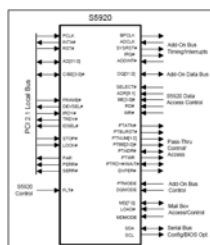
PCI Bus Signals

- FRAME# - Driven by bus master to indicate the beginning and duration of a bus transaction
- IRDY# - Initiator Ready, during which AD[31:0] contains valid data
- TRDY# - Target Ready. During read, it indicates valid AD[31:0] data
- STOP# - Conveys the request to bus master to stop current transaction



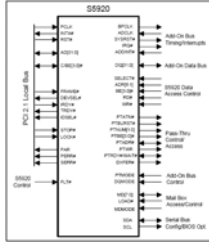
PCI Bus Signals

- LOCK# - Provides for the exclusive use of a resource
- IDSEL – Initialization Device Select, used as a chip select during initialization
- DEVSEL# - Device Select
- INTA# - Interrupt A, driving it low will interrupt the host
- PERR# - Parity Error
- SERR# - System Error, indicates catastrophic system error



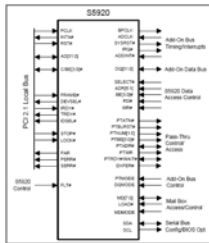
Add-on Bus Signals

- SCL – Serial Clock
- SDA – Serial Data
- MDMODE – Mailbox Data Mode
- LOAD# - Defines the MD[7:0] signals as an input bus
- MD[7:0] – Mailbox Data Bus
- PTMODE – Pass-thru mode
- PTATN# - Pass-thru attention, signals a decoded PT region
- PTBURST# - Indicates current operation is burst



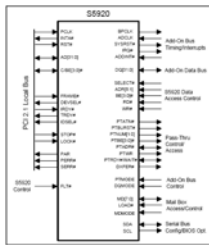
Add-on Bus Signals

- PTRDY#/WAIT# - PTRDY# is asserted low to indicate add-on logic has read/written data in response to PTATN#
- PTNUM[1:0] – Pass-thru number
- PTBE[3:0]# - Pass-thru byte enable
- PTADR# - Pass-thru Address
- PTWR – Indicates whether current operation is a read or a write
- DXFR# - Transfer complete signal



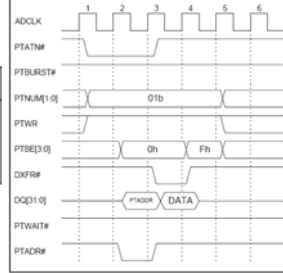
Add-on Bus Signals

- DQ[31:0] – Address/Data
- ADR[6:2] – Internal register address
- BE[2:0]# - Register byte enable signals
- BE3#/ADR1
- SELECT# - Enables internal logic to decode WR# and RD# for internal register access
- WR# - Write Enable
- RD# - Read Enable



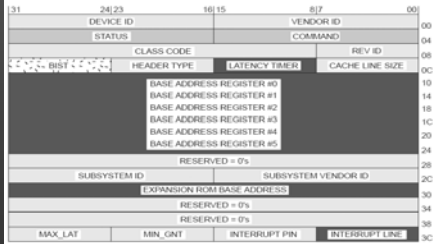
PCI Write Example

Defined PT Bus Width	APTD Register Write Byte Lane Steering			
	BYTE3	BYTE2	BYTE1	BYTE0
32 Bit Data Bus	DQ[31:24]	DQ[23:16]	DQ[15:8]	DQ[7:0]
16 Bit Data Bus	DQ[15:8]	DQ[7:0]	DQ[15:8]	DQ[7:0]
8 Bit Data Bus	DQ[7:0]	DQ[7:0]	DQ[7:0]	DQ[7:0]



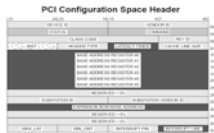
PCI Configuration Register

PCI Configuration Space Header



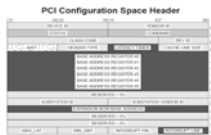
Configuration Register

- Every PCI device in a system, including a PCI-to-PCI bridge has a configuration data space.
- Configuration register is what allows the system to identify and control the device.
- Typically systems are designed so that every PCI device has its configuration register in an offset that is related to the slot it is at



Configuration Register

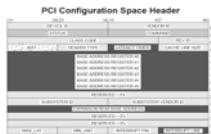
- **Vendor Identification**
 - A unique number describing the chipsets manufacturer. Assigned by the PCI SIG, Intel's is 0x8086
 - This ensures the PCI device's uniqueness amongst all manufacturers
- **Device Identification**
 - A unique number describing the device



The diagram shows a PCI Configuration Space Header with fields: Vendor ID (0x8086), Device ID (0x0001), Revision ID (0x00), Command (0x00000000), Status (0x00000000), Command/Status (0x00000000), and Device/Function Select (0x00000000).

Configuration Register

- **Status**
 - Gives the status of the device as defined in the standard
- **Command**
 - Provides the system with control of the PCI device
 - For example, allowing it to be a bus master, or giving it access to PCI I/O memory



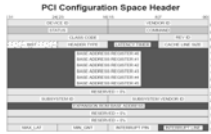
The diagram shows a PCI Configuration Space Header with fields: Vendor ID (0x8086), Device ID (0x0001), Revision ID (0x00), Command (0x00000000), Status (0x00000000), Command/Status (0x00000000), and Device/Function Select (0x00000000).

Command Register

- [9] Fast back-to-back enable
- [8] System Error Enable
 - When set to a 1, the device can drive the SERR# line
- [7] Wait Cycle Enable
- [6] Parity Error Response
 - Required for all devices, when set to a 1, the device can report parity errors (PERR#)
- [5] VGA Palette Snoop Enable
- [4] Memory Write and Invalidate
 - Mandatory for Master devices, if set to 0 it only performs a memory write
- [3] Special Cycle Recognition
- [2] Master Enable
- [1] Memory Access Enable
 - When set, the device responds to memory accesses
- [0] I/O Access Enable

Configuration Register

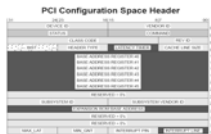
- **Class Code**
 - Identifies the type of device
 - For example, 0x01 is a mass storage controller, 0x02 is a network controller, 0xDA is a docking station
- **Base Address Register**
 - Are used to define and allocate the type, amount, and location of PCI I/O or memory available on the device



The diagram shows a PCI Configuration Space Header with fields: Vendor ID (0000), Device ID (0000), Revision ID (0000), Class Code (0000), and Base Address Registers (BARs) 0 through 5. It also shows the PCI Configuration Space Header structure with fields: Vendor ID, Device ID, Revision ID, Class Code, Base Address Registers, Command Register, Status Register, and Interrupt Pin Register.

Configuration Register

- **Interrupt Pin**
 - Identifies which of the four interrupt pins the PCI device uses
 - The four interrupt pins are INTA# through INTD#
- **Interrupt Line**
 - Used by the operating system to configure the device's interrupt request line



The diagram shows a PCI Configuration Space Header with fields: Vendor ID (0000), Device ID (0000), Revision ID (0000), Class Code (0000), Base Address Registers (BARs) 0 through 5, Command Register, Status Register, and Interrupt Pin Register.

PCI Express

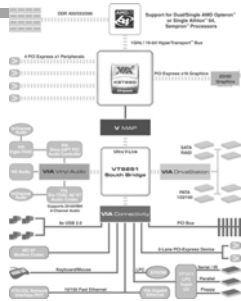
- **PCI Express is the third generation of high performance I/O bus used to interconnect peripheral devices in a system**
- **Nevertheless, because PCI Express uses the same memory and communications model as its predecessors, existing OS and drivers will work without any modifications**
- **Also supports chip-to-chip and board-to-board interconnects**

PCI Express

- Unlike its predecessors however, PCI Express is not a multi-drop parallel interconnect bus
- Instead it implements a serial, point-to-point interconnect scheme for communication between two devices to enable transfer rates of up to 2.5 Gbits/second
- Fewer pins also means smaller package size and less system cost
- Employs a switched-based technology to interconnect devices. Communication occurs using a packet-based communication protocol

PCI Express





Part of this image and its contents may have been taken from other sources and technical manuals. Original content Copyright © 2005 - Babak Kia
