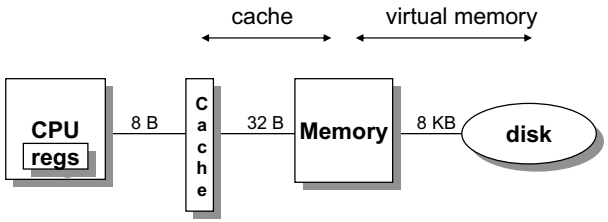
EE108B
Lecture 16
I/O

Christos Kozyrakis
Stanford University
http://eeclass.stanford.edu/ee108b

Announcements

- Remaining deliverables
  - PA2.2. today
  - HW4 on 3/13
  - Lab4 on 3/19
- In class Quiz 2 on Thu 2/15 (11am – 12.30pm)
  - Closed-books, 1 page of notes, green page, calculator
  - All lectures included
- Advice
  - Catch up with lectures and textbook
  - Take advantage of office hours and discussion sessions

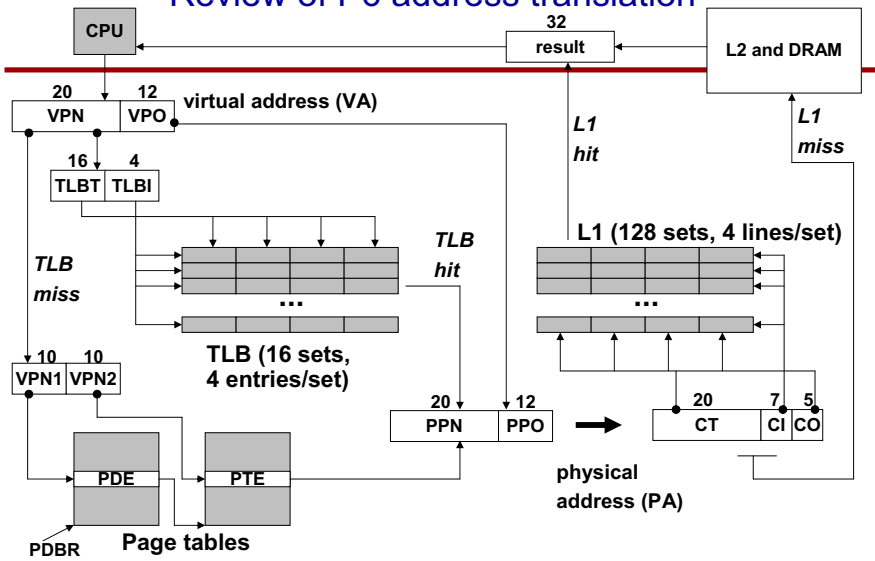
Review: Levels in Memory Hierarchy



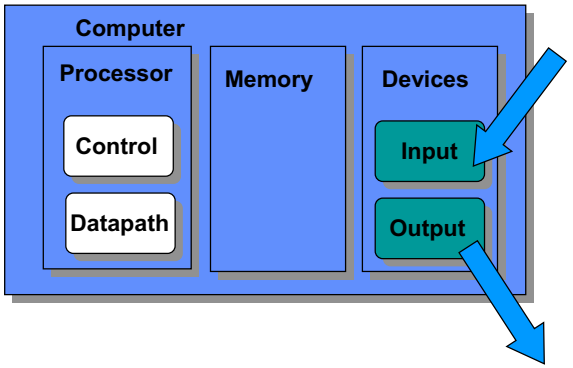
	Register	Cache	Memory	Disk Memory
size:	128 B	<4MB	< 16 GB	> 100 GB
Speed(cycles):	0.5-1	1-20	80-100	5-10 M
$/Mbyte:		$30/MB	$0.128/MB	$0.001/MB
line size:	8 B	32 B	>8 KB	

larger, slower, cheaper

Review of P6 address translation



Five Components



- Datapath
- Control
- Memory
- Input
- Output

Outline

- I/O Systems and Performance
  - Types and characteristics of I/O devices
  - Magnetic disks
- Buses
  - Bus types and bus operation
  - Bus arbitration
- Interfacing the OS and I/O devices
  - Operating System's role in handling I/O devices
  - Delegating I/O responsibility by the CPU
- I/O workloads and performance

Today's Lecture

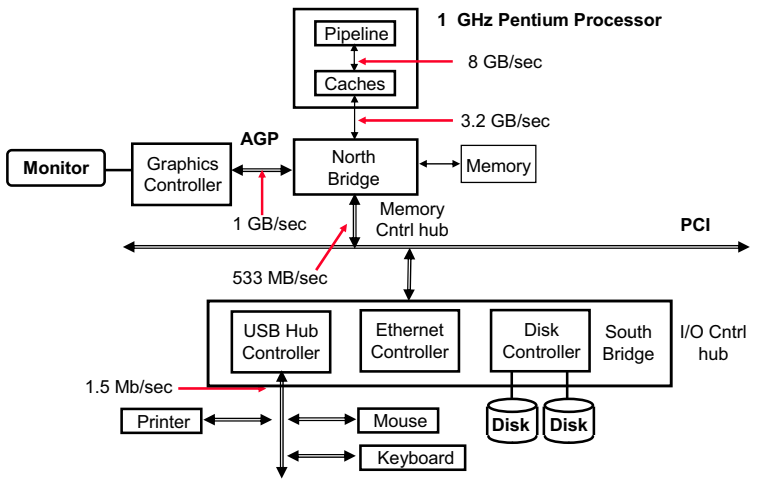
- I/O overview
- I/O performance metrics
- High performance I/O devices
  - Disk

Diversity of Devices

Device	Behavior	Partner	Data Rate (KB/sec)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Line Printer	Output	Human	1.00
Laser Printer	Output	Human	100.00
Graphics	Output	Human	100,000.00
Network-LAN	Communication	Machine	10,000.00
Floppy disk	Storage	Machine	50.00
Optical Disk	Storage	Machine	10, 000.00
Magnetic Disk	Storage	Machine	30,000.00

- Behavior refers to what I/O device does
- Since I/O connects two things, partner refers to the object on the other end of the connection

Speeds and Feeds of a PC System



Throughput vs. Response time

- Throughput
  - Aggregate measure of amount of data moved per unit time, averaged over a window
  - Sometimes referred to as bandwidth
    - Example: Memory bandwidth
    - Example: Disk bandwidth
- Response time
  - Response time to do a single I/O operation
    - Example: Write a block of bytes to disk
    - Example: Send a data packet over the network

I/O System Design Issues

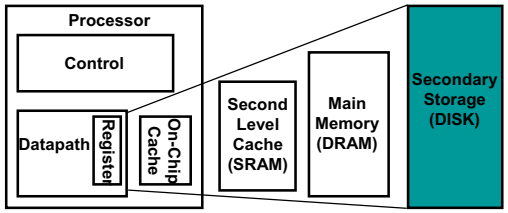
- Performance
  - Is throughput or response time more critical?
  - Huge diversity of devices means wide performance spectrum
  - I/O device performance tends to be technology driven
  - I/O system performance also depends on OS, software, bus performance, etc
- Expandability
- Resilience in the face of failure
- Computer classes
  - Desktop: response time and diversity of devices
  - Server: throughput, expandability, failure resilience
  - Embedded: cost and response time

I/O Devices

- I/O devices leverage various implementation techniques
  - Magnetic disks

Magnetic Hard Disks

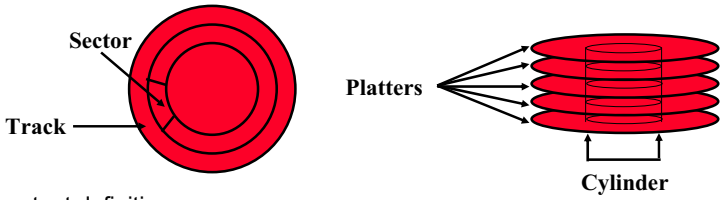
- Characteristics
  - Long term, nonvolatile storage
  - Large, inexpensive, but slow
- Usage
  - Virtual memory (swap area)
  - File system



Hard Disk

- Basic operation
  - Rotating platter coated with magnetic surface
  - Moving read/write head used to access the disk
- Features of hard disks
  - Platters are rigid (ceramic or metal)
  - High density since head can be controlled more precisely
  - High data rate with higher rotational speed
  - Can include multiple platters
- Incredible improvements
  - Example of I/O device performance being technology driven
  - Capacity: 2x every year
  - Transfer rate: 1.4x every year
  - Price approaching 1$/GB = 10^9 bytes

Hard Disk Organization



- Important definitions
  - Each drive uses one or more magnetic platters to store data
  - A head is used to read/write data on each size of each platter
  - Each platter is divided into a series of concentric rings called tracks
  - Each track is in turn divided into a series of sectors which is the basic unit of transfer for disks ("block size")
    - One method is to have a constant number of sectors per track
    - Alternative is constant bit density which places more sectors on outer track
  - A common track across multiple platters is referred to as a cylinder

Measuring Disk Access Time

- Each read or write has three major components
  - *Seek time* is the time to position the arm over the proper track
  - *Rotational latency* is the wait for the desired sector to rotate under the read/write head
  - *Transfer time* is the time required to transfer a block of bits (sector) under the read/write head
- Note that these represent only the "raw performance" of the disk drive
  - Also neglects to account for the I/O bus, controller, other caches, interleaving, etc.

Seek Time

- Seek time is the time to position the arm over the proper track
- Average seek time is the time it takes to move the read/write head from its current position to a track on the disk
- Industry definition is that seek time is the time for all possible seeks divided by the number of possible seeks
- In practice, locality reduces this to 25-33% of this number
- Note that some manufacturers report minimum seek times rather than average seek times

Rotational Latency

- Rotational latency is the time spent waiting for the desired sector to rotate under the read/write head
- Based upon the rotational speed, usually measured in revolutions per minute (RPM)
- Average rotational latency
  - Average rotational latency = 0.5 rotation / RPM
- Example: 7200 RPM

$$Average\ rotational\ latency = \frac{0.5\ rotation}{7200\ RPM} = \frac{0.5\ rotation}{7200\ RPM\ /(60\ \mathrm{sec}/\ \mathrm{min})} = 4.2ms$$

Transfer Time

- Transfer time is the time required to transfer a block of bits
- A factor of the transfer size, rotational speed, and recording density
  - Transfer size is usually a sector
- Most drives today use caches to help buffer the effects of seek time and rotational latency

Typical Hard Drive

- Typical hard disk drive
  - Rotation speed: 3600, 5200, 7200, or 10000 RPM
  - Tracks per surface: 500-2,000 tracks
  - Sectors per track: 32-128 sectors
  - Sectors size: 512 B-1024 KB
  - Minimum seek time is often approximately 0.1 ms
  - Average seek time is often approximately 5-10 ms
  - Access time is often approximately 9-10 ms
  - Transfer rate is often 2-4 MB/s

Average Access Example

- Consider the Seagate 36.4 GB Ultra2 SCSI
  - Rotation speed: 10,000 RPM
  - Sector size: 512 B
  - Average seek time: 5.7 ms
  - Transfer rate: 24.5 MB/s
  - Controller overhead of 1 ms
- What is the average read time?

$$Average\ rotational\ latency = \frac{0.5\ rotation}{10000\ RPM} = \frac{0.5\ rotation}{10000\ RPM / (60\ \text{sec}/\ \text{min})} = 3ms$$

$$Average\ transfer\ time = \frac{0.5\ KB}{24.5\ MB/s} = 0.02\ ms$$

$$Average\ access\ time = seek + rotational + transfer + overhead$$
$$= 5.7\ ms + 3\ ms + 0.02\ ms + 1\ ms = 9.72\ ms$$

Important Footnote

- If the actual seek time is only 25% of the average seek time as a result of locality, we get a very different number

$$Expected\ seek\ time = 0.25 \times 5.7\ ms = 1.43\ ms$$

$$Expected\ access\ time = seek + rotational + transfer + overhead$$
$$= 1.43\ ms + 3\ ms + 0.02\ ms + 1\ ms = 5.45\ ms$$

- Note that the effects of the rotational delay are even more pronounced

Reliability vs. Availability

- *Reliability* refers to the likelihood an individual component has failed whereas *availability* refers to whether the collection of components is available to the user
  - These two terms are frequently misused
- Improving Availability
  - Adding hardware, such as ECC memory (does not improve reliability since the memory is still broken, but corrects for the problem so that the data is still available)
- Improving Reliability
  - Better environmental conditions
  - Building with more reliable components
  - Using fewer components
- Note that improved availability may come at the cost of lower reliability

Disk Arrays

- Disk drives are arranged in an array
  - Combine multiple physical drives into single virtual drive
  - Small independent disks are usually cheaper than a single large drive
- Benefits
  - Increased availability since lost information can be reconstructed from redundant information (note that reliability is actually worse)
    - Mean Time To Failure (MTTF) is often 3-5 years
    - Mean Time To Repair (MTTR) is usually several hours
  - Increased throughput using many disk drives
    - Data is spread over multiple disks
    - Multiple access are made to several disks in parallel
- Known as a redundant array of inexpensive/independent drives (RAID)

I/O System Example : Transaction Processing

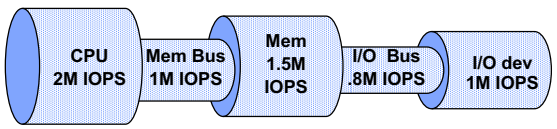
- Examples: Airline reservation, bank ATM, inventory system, e-business
- Workload
  - Many small changes to shared data space
    - Each transaction takes 2-10 disk I/Os
    - Approximately 2M-5M CPU instructions per disk I/O
  - Demands placed on system by many different users
- Important Considerations
  - Terrible locality
  - Requires graceful handling of failures (fault tolerance) by way of built-in redundancy and multiple-phase operations
  - Both throughput and response times are important
    - High throughput needed to keep cost low
    - Measure I/O rate as the number of accesses per second
    - Low response time is also very important for the users

I/O Performance Factors

- Overall performance is dependent upon a great many implementation factors
  - CPU
    - How fast can bits be moved in and out?
    - How fast can the processor operate on the data?
  - Memory system bandwidth and latency
    - Internal and external caches
    - Main memory
  - System interconnection
    - I/O and memory buses
    - I/O controller
    - I/O device
  - Software efficiency
    - I/O device handler instruction path length

Designing an I/O System

- General approach
  - Find the weakest link in the I/O system (consider things such as the CPU, memory system, buses, I/O controllers, and I/O devices)
  - Configure this component to sustain the required throughput
  - Determine the requirements for the rest of the system and configure them to support this throughput



Typical Storage Design Problem

- Analyze a multiprocessor to be used for transaction processing, using a TPC-A like benchmark, with the following characteristics:
  - each transaction = two 128-byte disk accesses + 3.2 M instructions, on a single disk whose account file size must be TPS x 10^9 bytes
  - the base hardware (no CPUs) costs $4,000
  - each processor is a 40 MIPS CPU and costs $3000
  - each processor can have any number of disks
  - disk controller delay = 2 msec
  - can choose between two disk types, but can't mix them

Disk size	Cost	Capacity	Avg seek time	Rotation speed	Transfer rate
3.5 inch	$200	50 GB	8 msec	5400 RPM	4 MB/s
2.4 inch	$120	25 GB	12 msec	7200 RPM	2 MB/s

- What is the highest TPS you can process for $40,000, and with what configuration?

Solution
Part 1: pick a disk

- First calculate how many TPS each disk can sustain
- access time = seek time + rotational delay + transfer + controller
  - 3.5" disk time = 8 + 1/2(1/5400 RPM) + 128B / 4MB/s + 2 = 15.6 msec
  - 2.4" disk time = 12 + 1/2(1/7200 RPM) + 128B / 2MB/s + 2 = 18.2 msec
- Need 2 accesses per transaction, so TPS = 1/(2*time)
  - 3.5" TPS = 1/(2*15.6 msec) = 32.0 TPS
  - 2.4" TPS = 1/(2*18.2 msec) = 27.4 TPS
- But the account file size on each disk = TPS x 10^9 bytes
  - 3.5" size = 32 GB = max 32 TPS (fits!) (I/O limited)
  - 2.4" size = 25 GB = max 25 TPS (doesn't fit!) (capacity limited)
    Must reduce TPS to 25 so that file fits
- Which has better cost/performance?
  - $/TPS for 3.5" = $200/32TPS = 6.25 $/TPS
  - $/TPS for 2.4" = $120/25TPS = 4.8 $/TPS
- Pick the 2.4" disk

Solution
Part 2: pick a CPU configuration

- TPS limit for each CPU =
  400 MIPS / (3.2 M instructions/transaction) = 125 TPS
- To fully utilize the CPU TPS, the number of disks that each can accommodate is
  #disks/CPU = (125 TPS/CPU) / (25 TPS/disk) = 5
- So a system with *n* CPUs and 5*n* disks costing $40,000 means
  $4000 + $3000*n* + $120*5*n* = $40000
  or *n* = 10
- The system has 10 CPUs, 50 2.4" disks, a total account file size of (50 x 250MB) = 12.5 GB, and can process (50 x 25) = 1250 TPS.

I/O Device Summary

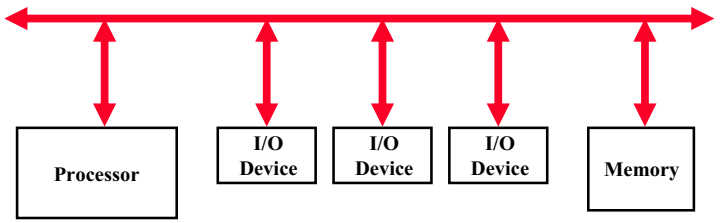
- I/O Performance depends on many factors
  - Device performance (typically technology driven)
  - CPU and OS performance
  - Software efficiency
- I/O System must exploit good aspects and hide bad aspects of the I/O
  - Disk caching and prefetching
  - Color maps for frame buffers
- Some measurements are only meaningful with respect to a particular workload
  - Transactional Processing
- High performance I/O devices
  - Disks
  - Graphics

Today's Lecture

- Buses
- Interfacing I/O with processor and memory
- Read Sections 8.4–8.9

Buses

- A bus is a shared communication link that connects multiple devices
- Single set of wires connects multiple "subsystems" as opposed to a point to point link which only connects two components together
- Wires connect in parallel, so 32 bit bus has 32 wires of data



Advantages/Disadvantages

- Advantages
  - Broadcast capability of shared communication link
  - Versatility
    - New device can be added easily
    - Peripherals can be moved between computer systems that use the same bus standard
  - Low Cost
    - A single set of wires is shared multiple ways
- Disadvantages
  - Communication bottleneck
    - Bandwidth of bus can limit the maximum I/O throughput
  - Limited maximum bus speed
    - Length of the bus
    - Number of devices on the bus
    - Need to support a range of devices with varying latencies and transfer rates

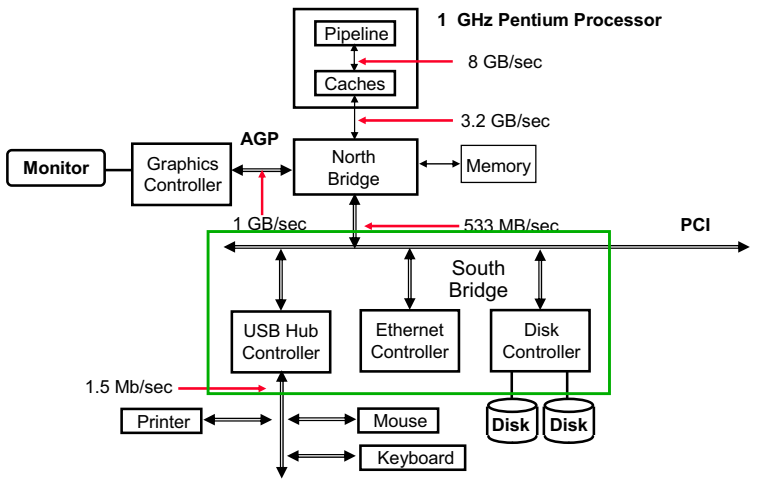
Bus Organization

- Bus Components
  - Control Lines
    - Signal begin and end of transactions
    - Indicate the type of information on the data line
  - Data Lines
    - Carry information between source and destination
    - Can include data, addresses, or complex commands

Types of Buses

- Processor-Memory Bus (or *front-side* bus or *system* bus)
  - Short, high-speed bus
  - Connects memory and processor directly
  - Designed to match the memory system and achieve the maximum memory-to-processor bandwidth (cache transfers)
  - Designed specifically for a given processor/memory system
- I/O Bus (or peripheral bus)
  - Usually long and slow
  - Connect devices to the processor-memory bus
  - Must match a wide range of I/O device performance characteristics
  - Industry standard

Speeds and Feeds of a PC System

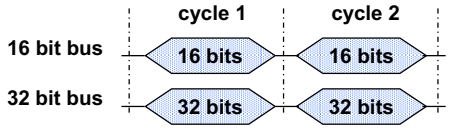


Synchronous versus Asynchronous

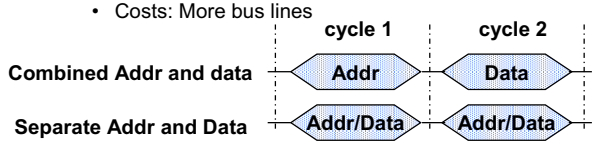
- Synchronous Bus
  - Includes a clock in control lines
  - Fixed protocol for communication relative to the clock
  - Advantages
    - Involves very little logic and can therefore run very fast
  - Disadvantages
    - Every decision on the bus must run at the same clock rate
    - To avoid clock skew, bus cannot be long if it is fast
  - Example: Processor-Memory Bus
- Asynchronous Bus
  - No clock
  - Can easily accommodate a wide range of devices
  - No clock skew problems, so bus can be quite long
  - Requires handshaking protocol

Increasing Bus Bandwidth

- Several factors account for bus bandwidth
  - Wider bus width
    - Increasing data bus width => more data per bus cycle
    - Cost: More bus lines

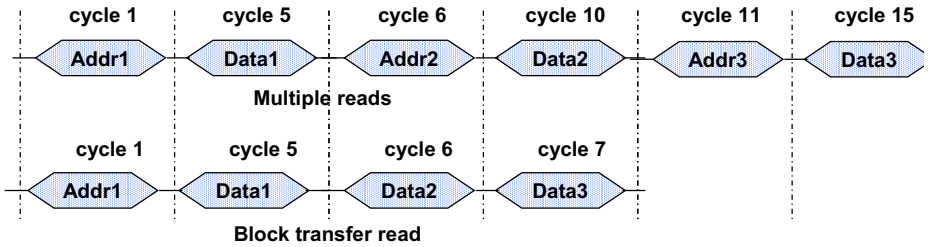


  - Separate address and data lines
    - Address and data can be transmitted in one bus cycle if separate address and data lines are available
    - Costs: More bus lines



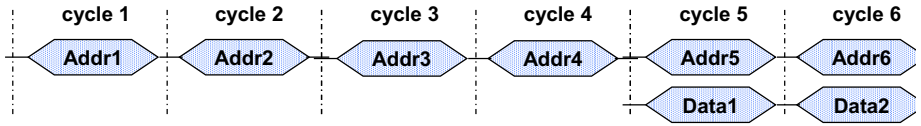
Increasing Bus Bandwidth

- Several factors account for bus bandwidth
  - Block transfers
    - Transfer multiple words in back-to-back bus cycles
    - Only one address needs to be sent at the start
    - Bus is not released until the last word is transferred
    - Costs: Increased complexity and increased response time for pending requests



Increasing Bus Bandwidth

- Split transaction "pipelining the bus"
  - Free the bus during time between request and data transfer
  - Costs: Increased complexity and higher potential latency



Split transaction bus with separate Address and Data wires

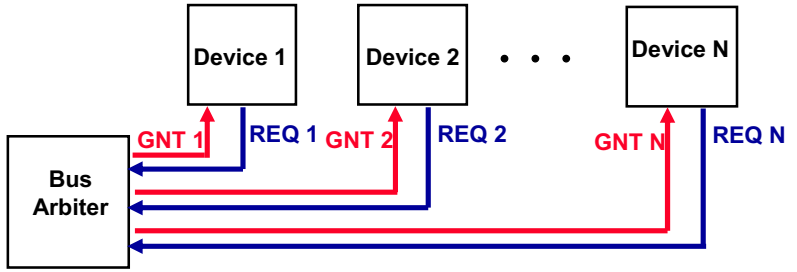
Accessing the Bus

- Up to this point we have not addressed one of the most important questions in bus design: How is the bus reserved by a device that wishes to use it?
- Master-slave arrangement
  - Only the bus master can control access to the bus
  - The bus master initiates and controls all bus requests
  - A slave responds to read and write requests
- A simple system
  - Processor is the only bus master
  - All bus requests must be controlled by the processor
  - Major drawback is the processor must therefore be involved in every transaction!

Multiple Masters

- With multiple masters, arbitration must be used so that only one device is granted access to the bus at a given time
- Arbitration
  - The bus master wanting to use the bus asserts a bus request
  - The bus master cannot use the bus until the request is granted
  - The bus master must signal the arbiter when finished using the bus
- Bus arbitration goals
  - Bus priority – Highest priority device should be serviced first
  - Fairness – Lowest priority devices should not starve

Centralized Parallel Arbitration

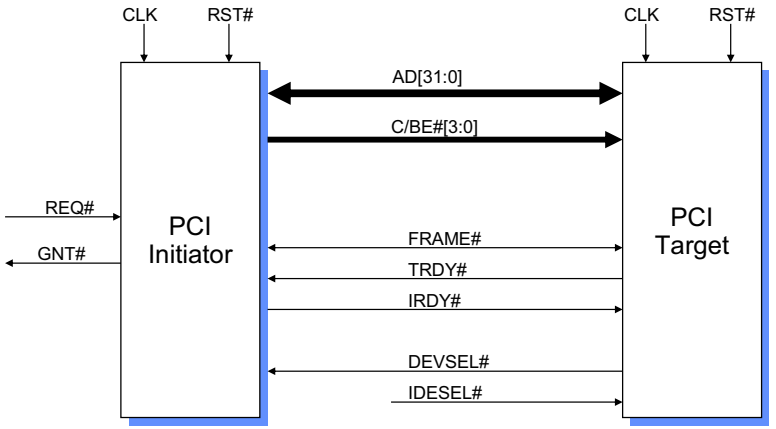


- Advantages
  - Centralized control where all devices submit request
  - Any fair priority scheme can be implemented (FCFS, round-robin)
- Disadvantages
  - Potential bottleneck at central arbiter

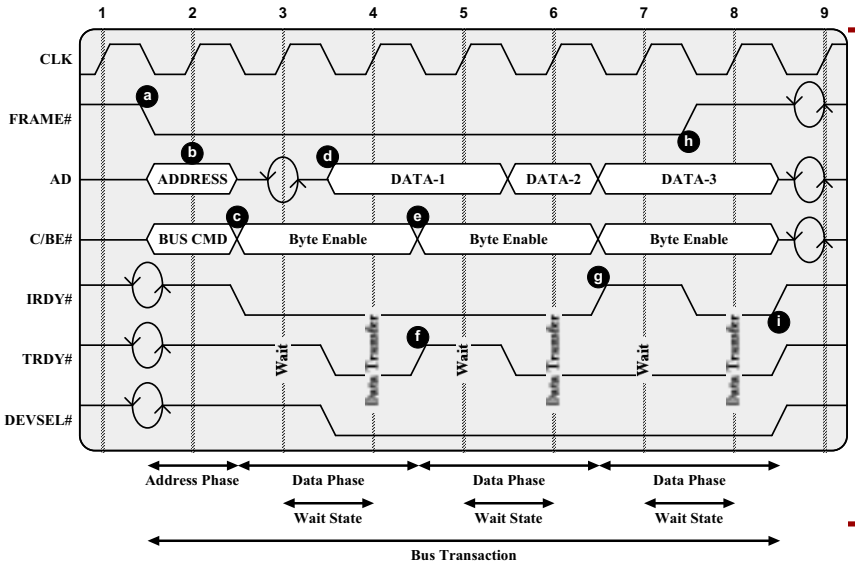
Case Study: PCI

- Peripheral Component Interconnect (PCI) peripheral backplane bus standard
- Clock Rate: 33 MHz (or 66 MHz in PCI Version 2.1) [CLK]
- Central arbitration [REQ#, GNT#]
  - Overlapped with previous transaction
- Multiplexed Address/Data
  - 32 lines (with extension to 64) [AD]
- General Protocol
  - Transaction type (bus command is memory read, memory write, memory read line, etc) [C/BE#]
  - Address handshake and duration [FRAME#, TRDY#]
  - Data width (byte enable) [C/BE#]
  - Variable length data block handshake between Initiatory Ready and Target Ready [IRDY#, TRDY#]
- Maximum bandwidth is 132 MB/s (533 MB/s at 64 bit/ 66 MHz)

32 bit PCI Signals



PCI Read



PCI Read Steps 1

a) Once a bus master has gained control of the bus, it initiates the transaction by asserting FRAME. This line remains asserted until the last data phase. The initiator also puts the start address on the address bus, and the read command on the C/BE lines.

b) The target device recognizes its address on the AD lines.

c) The initiator ceases driving the AD bus. A turnaround cycle (marked with two circular arrows) is required before another device may drive any multiple-source bus. Meanwhile, the initiator changes the C/BE lines to designate which AD lines are to be used for data transfer (from 1-4 bytes wide). The initiator also asserts IRDY to indicate that it is ready for the first data item.

d) The selected target asserts DEVSEL to indicate that it has recognized its address and will respond. It places the requested data on the AD lines and asserts TRDY to indicate that valid data is present on the bus.

PCI Read Steps 2

e) The initiator reads the data at the beginning of clock 4 and changes the byte enable lines as needed in preparation for the next read.

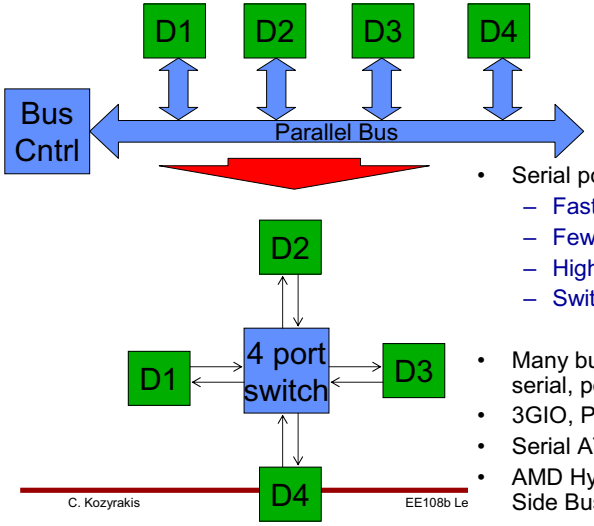
f) In this example, the target needs some time to prepare the second block of data for transmission. Therefore, it deasserts TRDY to signal the initiator that there will not be new data during the coming cycle. Accordingly, the initiator does not read the data lines at the beginning of cycle 5 and does not change the byte enable on that cycle. The block of data is read at the beginning of cycle 6.

g) During clock 6, the target places the third data item on the bus. However, in this example the initiator is not yet ready to read the data item (i.e. temporarily buffers are full). It therefore deasserts IRDY. This will cause the target to hold the data for an extra cycle.

h) The initiator deasserts FRAME to signal the target that the third data transfer is the last, and asserts IRDY to signal that it is ready.

i) Return to the idle state. The initiator deasserts IRDY, and the target deasserts TRDY & DEVSEL.

Trends for Buses
Logical Bus and Physical Switch



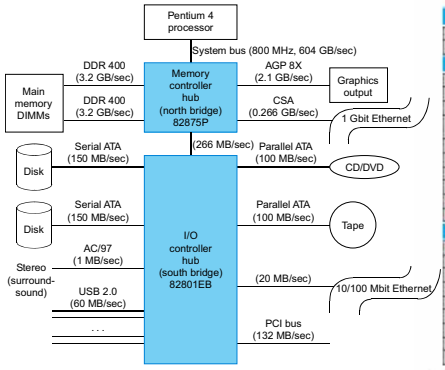
- Serial point-to-point advantages
  - Faster links
  - Fewer chip package pins
  - Higher performance
  - Switch keeps arbitration on chip
- Many bus standards are moving to serial, point to point
- 3GIO, PCI-Express(PCI)
- Serial ATA (IDE hard disk)
- AMD Hypertransport versus Intel Front Side Bus (FSB)

PCI vs. PCI Express

- Same bus protocol
  - Same driver software
- PCI
  - 32–64 shared wires
  - Frequency: 33MHz – 133 MHz
  - Bandwidth: 132 MB/s – 1 GB/s
- PCI Express
  - 4 wires per direction
  - Frequency: 625 MHz
  - Bandwidth: 300 MB/s per direction
- PCI Express Advantage
  - 5-10 x pin bandwidth
  - Multiple links for more bandwidth

Modern Pentium 4 I/O

- I/O Options



Review: Bus Summary

- Bus design issues
  - Bus width
  - Synchronization
  - Arbitration
  - Bus transactions
    - Read/write protocols
    - Block addressing
    - Split transactions
- Three basic buses
  - Processor-memory: Front-side bus
  - Backplane bus: PCI
  - I/O bus: USB
- Bus design trends
  - Point-to-point serial connections with switches