I/O—I/O Busses

Professor Alvin R. Lebeck
Computer Science 220
Fall 2001

Admin

• Homework #5 Due Nov 20
• Projects...
  – Good writeup is very important (you’ve seen several papers…)

• Reading:
  – 8.1, 8.3-5 (Multiprocessors)
  – 7.1-3, 7.5, 7.8 (Interconnection Networks)
Review: Storage System Issues

- Why I/O?
- Performance
  - queue + seek + rotational + transfer + overhead
- Processor Interface Issues
  - Memory Mapped
  - DMA (caches)
- I/O & Memory Buses
- Redundant Arrays of Inexpensive Disks (RAID)

Review: Processor Interface Issues

- Interconnections
  - Busses
- Processor interface
  - Instructions
  - Memory mapped I/O
- I/O Control Structures
  - Polling
  - Interrupts
  - Programmed I/O
  - DMA
  - I/O Controllers
  - I/O Processors
**Review: Device Controllers**

- **Bus**
- **Device Controller**
  - Command
  - Status
  - Data 0
  - Data 1
  - Data n-1

Controller deals with mundane control (e.g., position head, error detection/correction)

Processor communicates with Controller

**I/O Data Flow**

**Impediment to high performance: multiple copies, complex hierarchy**

- Memory-to-Memory Copy
- DMA over Peripheral Bus
- Xfer over Disk Channel
- Xfer over Serial Interface

Application Address Space

OS Buffers (>10 MByte)

HBA Buffers (1 M - 4 MBytes)

Track Buffers (32K - 256KBytes)

I/O Device

Head/Disk Assembly
Communication Networks

Performance limiter is memory system, OS overhead, not HW protocols

- Send/receive queues in processor OS memory
- Network controller copies back and forth via DMA
- No host intervention needed
- Interrupt host when message sent or received
- Memory-to-Memory copy to user space

Network Connected Devices

- High speed networks (10Gb Ethernet soon)
- How can we eliminate overheads?
  Page Flipping
  - OS places aligned data into memory and remaps pages
  RDMA
  - Idea is to eliminate kernel-to-user copy (User-level messaging)
  - Requires “translation” on Network Interface (NI)
  - Application registers region with OS
  - OS stores pointer in NI
  - On Receive, pointer says where data should go
A bus is a shared communication link
It uses one set of wires to connect multiple subsystems

Advantages of Buses

Versatility:
- New devices 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 in multiple ways
Disadvantages of Buses

- The bus creates a communication bottleneck
  - Bus bandwidth can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus
  - The need to support a range of devices with:
    - Widely varying latencies
    - Widely varying data transfer rates

The General Organization of a Bus

- **Control lines:**
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- **Data lines** carry information between the source and the destination:
  - Data and Addresses
  - Complex commands
Master versus Slave

- A bus transaction includes two parts:
  - Sending the address
  - Receiving or sending the data

- Master is the device that starts the bus transaction by:
  - Sending the address

- Slave is the device that responds to the address by:
  - Sending data to the master if the master asks for data
  - Receiving data from the master if the master wants to send data

Output Operation

- Output: **Processor** sending data to the I/O device:

  **Step 1: Request Memory**
  
  Processor → Control (Memory Read Request) → Memory

  I/O Device (Disk) → Data (Memory Address)

  **Step 2: Read Memory**
  
  Processor → Control → Memory

  I/O Device (Disk) → Data

  **Step 3: Send Data to I/O Device**
  
  Processor → Control (Device Write Request) → Memory

  I/O Device (Disk) → Data (I/O Device Address and then Data)
Input Operation

- Input is defined as the Processor receiving data from the I/O device:

**Step 1: Request Memory**

- Processor
- Control (Memory Write Request)
- Data (Memory Address)
- I/O Device (Disk)
- Memory

**Step 2: Receive Data**

- Processor
- Control (I/O Read Request)
- Data (I/O Device Address and then Data)
- I/O Device (Disk)
- Memory

Types of Buses

- **Processor-Memory Bus (design specific)**
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor

- **External I/O Bus (industry standard)**
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus

- **Backplane Bus (industry standard)**
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one single bus for all components

- **Bit-Serial Buses (New trend: USB, Firewire, .. )**
  - Use High speed unidirectional point-to-point communication
A Computer System with One Bus: Backplane Bus

- A single bus (the backplane bus) is used for:
  - Processor to memory communication
  - Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: Early IBM PC

A Two-Bus System

- I/O buses tap into the processor-memory bus via bus adaptors:
  - Processor-memory bus: mainly for processor-memory traffic
  - I/O buses: provide expansion slots for I/O devices
- Example: Apple Macintosh-II
  - NuBus: Processor, memory, and a few selected I/O devices
  - SCCI Bus: the rest of the I/O devices
A Three-Bus System

- A small number of backplane buses tap into the processor-memory bus
  - Processor-memory bus is used for processor memory traffic
  - I/O buses are connected to the backplane bus
- Advantage: loading on the processor bus is greatly reduced

The Many Buses of Today's Systems
Synchronous and Asynchronous Bus

- **Synchronous Bus:**
  - Includes a clock in the control lines
  - A fixed protocol for communication that is relative to the clock
  - Advantage: involves very little logic and can run very fast
  - Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew, bus must be short if it is fast.

- **Asynchronous Bus:**
  - It is not clocked
  - It can accommodate a wide range of devices
  - It can be lengthened without worrying about clock skew
  - It requires a handshaking protocol

A Handshaking Protocol

- **Three control lines**
  - **ReadReq**: indicates a read request for memory
    - Address is put on the data lines at the same line
  - **DataRdy**: indicates the data word is now ready on the data lines
    - Data is put on the data lines at the same time
  - **Ack**: acknowledge the ReadReq or the DataRdy of the other party
Increasing the Bus Bandwidth

- **Separate versus multiplexed address and data lines:**
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available.
  - Cost: (a) more bus lines, (b) increased complexity.

- **Data bus width:**
  - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles.
  - Example: USB vs. 32-bit PCI vs. 64-bit PCI.
  - Cost: more bus lines.

- **Block transfers:**
  - Allow the bus to transfer multiple words in back-to-back bus cycles.
  - Only one address needs to be sent at the beginning.
  - The bus is not released until the last word is transferred.
  - Cost: (a) increased complexity, (b) decreased response time for request.

Obtaining Access to the Bus

- **One of the most important issues in bus design:**
  - How is the bus reserved by a device that wishes to use it?

- **Chaos is avoided by a master-slave arrangement:**
  - Only the bus master can control access to the bus:
    - It initiates and controls all bus requests.
  - A slave responds to read and write requests.

- **The simplest system:**
  - Processor is the only bus master.
  - All bus requests must be controlled by the processor.
  - Major drawback: the processor is involved in every transaction.
Multiple Potential Bus Masters: the Need for Arbitration

- **Bus arbitration scheme:**
  - A bus master wanting to use the bus asserts the bus request
  - A bus master cannot use the bus until its request is granted
  - A bus master must signal to the arbiter after finish using the bus

- **Bus arbitration schemes usually try to balance two factors:**
  - **Bus priority:** the highest priority device should be serviced first
  - **Fairness:** Even the lowest priority device should never be completely locked out from the bus

- **Bus arbitration schemes can be divided into four broad classes:**
  - **Distributed arbitration by self-selection:** each device wanting the bus places a code indicating its identity on the bus.
  - **Distributed arbitration by collision detection:** Ethernet uses this.
  - **Daisy chain arbitration:** single device with all request lines.
  - **Centralized, parallel arbitration:** see next-next slide
The Daisy Chain Bus Arbitration Scheme

- **Advantage:** simple
- **Disadvantages:**
  - Cannot assure fairness:
    - A low-priority device may be locked out indefinitely
  - The use of the daisy chain grant signal also limits the bus speed

---

Summary of Bus Options

<table>
<thead>
<tr>
<th>Option</th>
<th>High performance</th>
<th>Low cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus width</td>
<td>Separate address &amp; data lines</td>
<td>Multiplex address &amp; data lines</td>
</tr>
<tr>
<td>Data width</td>
<td>Wider is faster (e.g., 64 bits)</td>
<td>Narrower is cheaper (e.g., 8 bits)</td>
</tr>
<tr>
<td>Transfer size</td>
<td>Multiple words has less bus overhead</td>
<td>Single-word transfer is simpler</td>
</tr>
<tr>
<td>Bus masters</td>
<td>Multiple (requires arbitration)</td>
<td>Single master (no arbitration)</td>
</tr>
<tr>
<td>Clocking</td>
<td>Synchronous</td>
<td>Asynchronous</td>
</tr>
</tbody>
</table>
SCSI: Small Computer System Interface

- Up to 8 devices to communicate on a bus or “string” at sustained speeds of 4-5 MBytes/sec
- SCSI-2 up to 20 MB/sec
- Devices can be slave (“target”) or master (“initiator”)
- SCSI protocol: a series of phases, during which specific actions are taken by the controller and the SCSI disks
  - **Bus Free**: No device is currently accessing the bus
  - **Arbitration**: When the SCSI bus goes free, multiple devices may request (arbitrate for) the bus; fixed priority by address
  - **Selection**: informs the target that it will participate (Reselection if disconnected)
  - **Command**: the initiator reads the SCSI command bytes from host memory and sends them to the target
  - **Data Transfer**: data in or out, initiator: target
  - **Message Phase**: message in or out, initiator: target (identify, save/restore data pointer, disconnect, command complete)
  - **Status Phase**: target, just before command complete

PCI

- Synchronous Bus
- 33 MHz, 66MHz
- 32-bit multiplexed Address/Data Bus
  - Can have 64-bit Data bus (32 are multiplexed with address)
- 132MB/s or 264MB/s
- Frame signal identifies start address phase, followed by one or more data phases
  - Can send 64-bit address over 32-bit bus, takes two cycles
- C/BE[3:0] during address phase indicates type
  - Memory read/write, I/O read/write, ...
- C/BE[3:0] byte valid bits for data phase
  - Wait states supported by separate signals
  - Stop signal to terminate
PCI (continued)

- Centralized arbitration
- Pipelined with ongoing transfers
- Auto configuration to identify type (SCSI, Ethernet, etc.), manufacturer, I/O addresses, memory addresses, interrupt level, etc.

Accelerated Graphics Port (AGP)

- Overview
- Data Transfers
- Split Transaction Bus
  - Separate Bus Transaction for Address and Data
  - Increases controller complexity, but AGP shows advantages
Manufacturing Advantages of Disk Arrays

Disk Product Families

Conventional: 4 disk designs
- 3.5”
- 5.25”
- 10”
- 14”

Low End → High End

Disk Array: 1 disk design
- 3.5”

Redundant Arrays of Disks

- Files are "striped" across multiple spindles
- Redundancy yields high data availability

Disks will fail
- Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store it
  - Bandwidth penalty to update

Techniques:
- Mirroring/Shadowing (high capacity cost)
- Horizontal Hamming Codes (overkill)
- Parity & Reed-Solomon Codes
- Failure Prediction (no capacity overhead!)
Array Reliability

- Reliability of N disks = Reliability of 1 Disk ÷ N
  50,000 Hours ÷ 70 disks = 700 hours
  Disk system MTTF: Drops from 6 years to 1 month!
- Arrays without redundancy too unreliable to be useful!

Hot spares support reconstruction in parallel with access: very high media availability can be achieved

Redundant Arrays of Disks (RAID) Techniques

- Disk Mirroring, Shadowing
  Each disk is fully duplicated onto its "shadow"
  Logical write = two physical writes
  100% capacity overhead

- Parity Data Bandwidth Array
  Parity computed horizontally
  Logically a single high data bw disk

- High I/O Rate Parity Array
  Interleaved parity blocks
  Independent reads and writes
  Logical write = 2 reads + 2 writes
  Parity + Reed-Solomon codes
Problems of Disk Arrays: Small Writes

**RAID-5: Small Write Algorithm**

1 Logical Write = 2 Physical Reads + 2 Physical Writes

**Diagram:**
- D0', D0, D1, D2, D3, P
- New data
- Old data (1. Read)
- Old parity (2. Read)
- XOR
- (3. Write)
- XOR
- (4. Write)

Redundant Arrays of Disks

**RAID 1: Disk Mirroring/Shadowing**

- Each disk is fully duplicated onto its "shadow"
  - Very high availability can be achieved
- Bandwidth sacrifice on write:
  - Logical write = two physical writes
- Reads may be optimized
- Most expensive solution: 100% capacity **overhead**

Targeted for high I/O rate, high availability environments
Redundant Arrays of Disks RAID 3: Parity Disk

Parity computed across recovery group to protect against hard disk failures. 33% capacity cost for parity in this configuration. Wider arrays reduce capacity costs, decrease expected availability, increase reconstruction time.

Arms logically synchronized, spindles rotationally synchronized. Logically a single high capacity, high transfer rate disk.

*Targeted for high bandwidth applications: Scientific, Image Processing*

Redundant Arrays of Disks RAID 5+: High I/O Rate Parity

A logical write becomes four physical I/Os. Independent writes possible because of interleaved parity. Reed-Solomon Codes ("Q") for protection during reconstruction.

*Targeted for mixed applications*
Next Time

- Multiprocessors