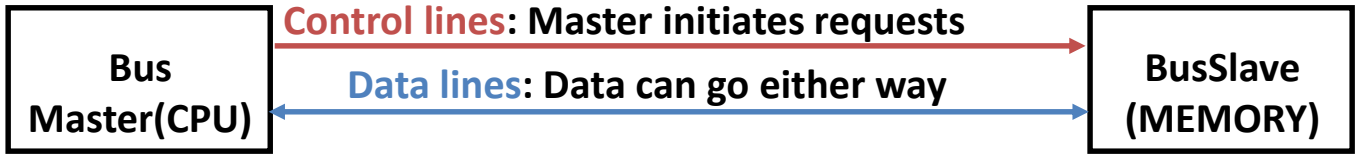
Computer Architecture
Chapter-8

Buses: Connecting I/O Devices to Processor and Memory

I/O System Interconnect Issues

- A bus is a shared communication link (a single set of wires used to connect multiple subsystems) that needs to support a range of devices with widely varying latencies and data transfer rates
  - Advantages
    - Versatile – new devices can be added easily and 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
    - Creates a communication bottleneck – bus bandwidth limits 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

Bus Characteristics



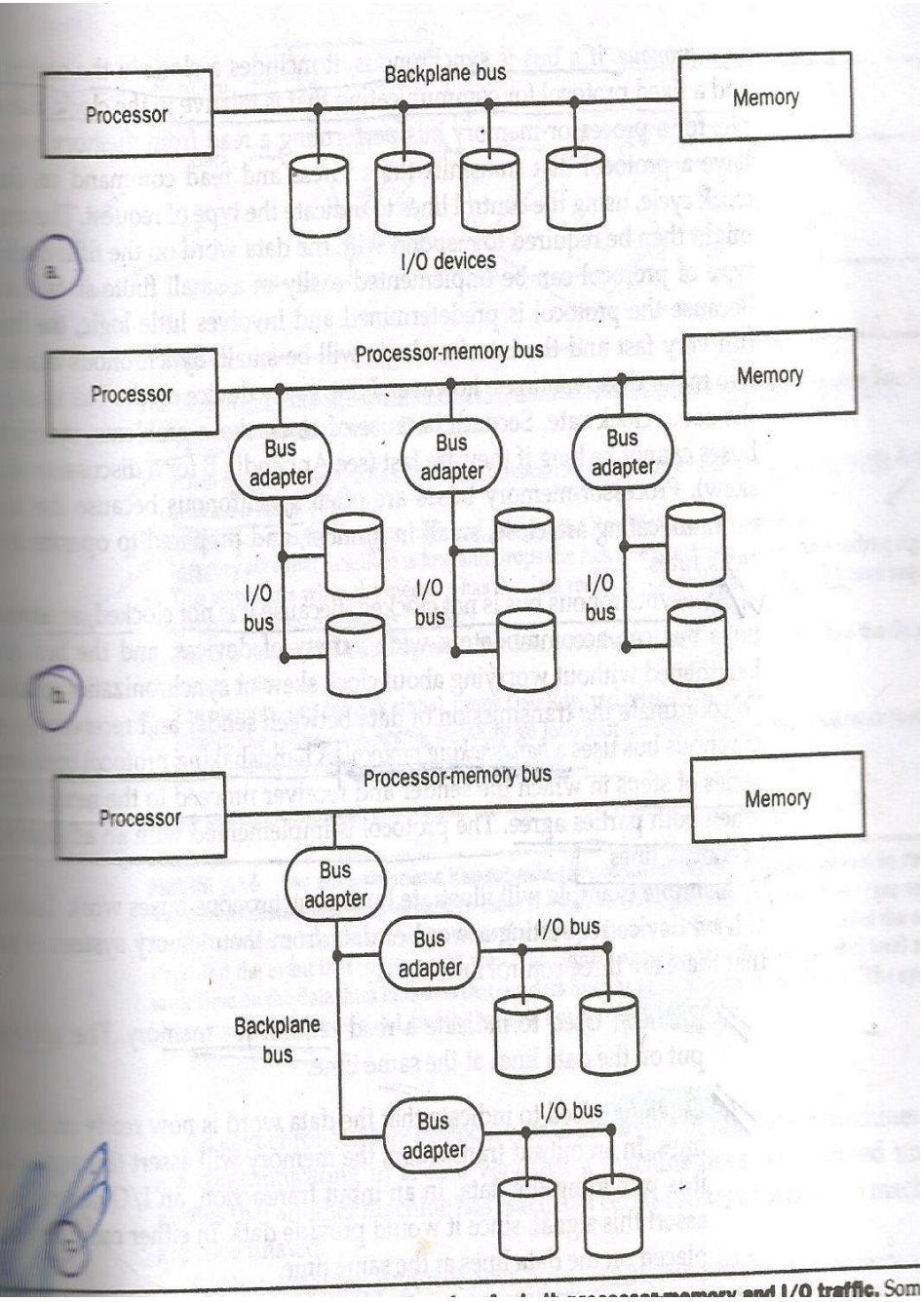
- Control lines
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- Data lines
  - Data, addresses, and complex commands
- Bus transaction consists of
  - Master issuing the command (and address) – request
  - Slave receiving (or sending) the data – action
  - Defined by what the transaction does to memory
    - Input – inputs data from the I/O device to the memory
    - Output – outputs data from the memory to the I/O device

Types of Buses

- **Processor-memory bus (proprietary)**
  - Short and high speed
  - Matched to the memory system to maximize the memory-processor bandwidth
  - Optimized for cache block transfers

- **I/O bus (industry standard, e.g., SCSI, USB, Firewire)**
  - Usually is lengthy and slower
  - Needs to accommodate a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus

- **Backplane bus (industry standard, e.g., ATA, PCIexpress)**
  - The backplane is an interconnection structure within the chassis
  - Used as an intermediary bus connecting I/O busses to the processor-memory bus



- **A)** A single bus used for processor to memory communication between I/O device and memory.**(single bus)**
- B) A separate bus is used for processor- memory traffic. I/O bus interfaces to the processor-memory bus by using bus adapter. Bus adapter provides speed matching between the busses. **(two buses)**
- **C)** A small number of backplane buses tap into the processor-memory bus. The processor-memory bus interface to the backplane bus **(three buses)**

Synchronous and Asynchronous Buses

- **Synchronous bus (e.g., processor-memory buses)**
  - Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
  - **Advantage:** involves very little logic and can run very fast
  - **Disadvantages**:
    - Every device communicating on the bus must use same clock rate
    - To avoid clock skew, they cannot be long if they are fast
- **Asynchronous bus (e.g., I/O buses)**
  - It is not clocked, so requires a **handshaking protocol** and additional control lines (ReadReq, Ack, DataRdy)
  - Advantages:
    - Can accommodate a wide range of devices and device speeds
    - Can be lengthened without worrying about clock skew or synchronization problems
  - Disadvantage: slow

Asynchronous Bus Handshaking Protocol

- A handshaking protocol consists of series of steps in which sender and receiver proceed to the next step only when both parties agree.
- The protocol is implemented within additional set of control lines
- To co-ordinate the transmission of data between the receiver and sender asynchronous bus uses handshaking protocol

- There are three control lines:

1. Read Req:

-used to indicate read request from memory

2. Data Rdy:

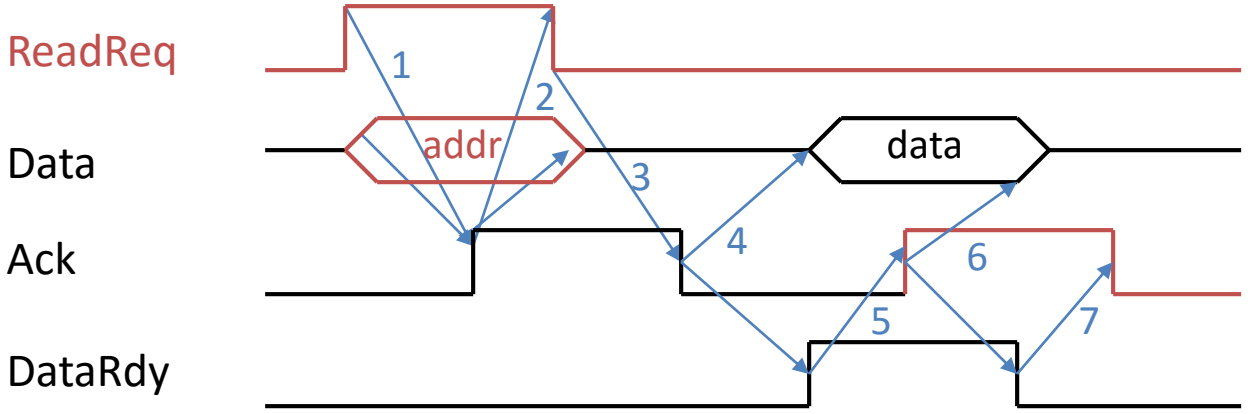
-used to indicate that the data word is now ready on the data lines

3. Ack:

-used to acknowledge the ReadReq or DataRdy signal of the other party

Steps in the Asynchronous Bus Handshaking Protocol

- Output (read) data from **memory to an I/O device**



I/O device signals a request by raising ReadReq and putting the addr on the data lines

1. Memory sees ReadReq, reads addr from data lines, and raises Ack
2. I/O device sees Ack and releases the ReadReq and data lines
3. Memory sees ReadReq go low and drops Ack
4. When memory has data ready, it places it on data lines and raises DataRdy
5. I/O device sees DataRdy, reads the data from data lines, and raises Ack
6. Memory sees Ack, releases the data lines, and drops DataRdy
7. I/O device sees DataRdy go low and drops Ack, which indicates that the transmission is completed