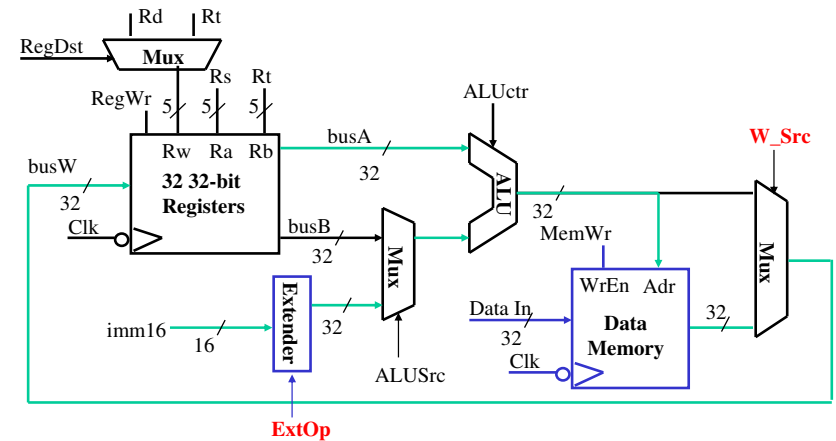


Buses

Lecturer: Dr. Hui Annie Guo
 huig@cse.unsw.edu.au
 K17-501F (ext. 57136)

Recall: Single cycle datapath

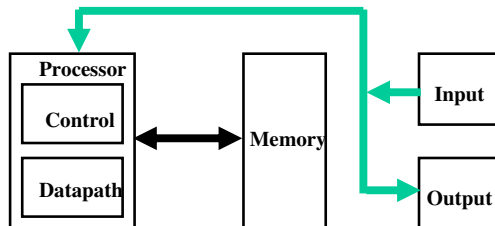


COMP3211/9211

2011S1 wk10_2 P2

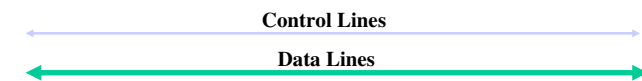
Bus is:

- A **shared communication link** for conveying addresses, data, and control signals
- A single set of wires used to connect multiple subsystems



2011S1 wk10_2 P3

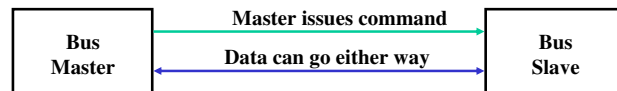
General Organization of a Bus



- **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

2011S1 wk10_2 P4

Master vs Slave



- Components communicate over a bus using so-called **bus transactions**
- A bus transaction includes two parts:
 - Issuing a command (and address) – request
 - Transferring the data – action
- **Master is the device that starts the bus transaction by:**
 - issuing a command (and address)
- **Slave is the device that responds to the request by:**
 - Sending data to the master if the master asks for data
 - Receiving data from the master if the master wants to send data

2011S1 wk10_2 P5

BUS Master Examples

- **Processors**
- **DMA controllers**
 - **DMA: Direct Memory Access**
 - A mechanism to enable direct data transfer between an I/O device and memory without involving the processor
 - **Three steps in a DMA transfer:**
 - **Processor sets up the DMA, by providing**
 - The I/O device
 - The I/O operation
 - The memory address and the size of data
 - **DMA controller transfers the data between the device and memory as a bus master**
 - **DMA informs the processor when the transfer is complete**

COMP3211/9211

2011S1 wk10_2 P6

Types of Buses

- There are likely to be a number of different buses in the typical machine, supporting various devices. They can be divided into
 - **Local buses**
 - Such as PCI
 - Used with internal devices such as graphical cards
 - **External buses**
 - Such as ISA
 - Used with external devices such as scanners.

COMP3211/9211

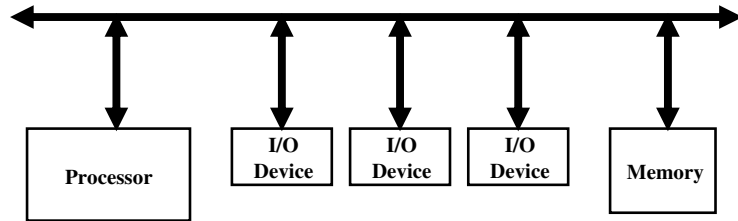
2011S1 wk10_2 P7

Types of Buses (cont.)

- **Processor-Memory Bus (design specific)**
 - Short and high speed
 - Mainly to match the memory system
 - To maximize memory-to-processor bandwidth
 - Connect directly to the processor
 - Probably optimized for cache block transfers
- **I/O Bus (industry standard)**
 - Usually is lengthy and slower
 - Need to match a wide range of I/O devices
 - Connect to the processor-memory bus or backplane bus
- **Backplane Bus (standard or proprietary)**
 - **Backplane:** an interconnection structure within the computer's chassis
 - Allow processors, memory, and I/O devices to coexist
 - Cost advantage: one bus for all components

2011S1 wk10_2 P8

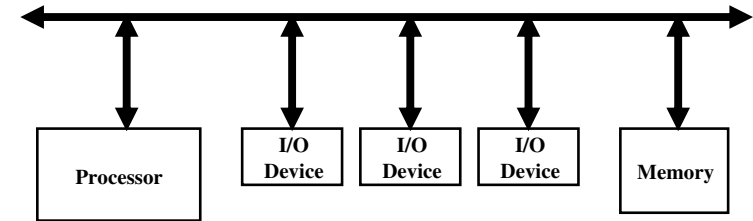
Advantages of Buses



- **Versatility:**
 - New devices can be added easily
 - Peripherals can be moved between computers systems that use the same bus standard
- **Low Cost:**
 - A single set of wires is shared in multiple ways
 - Manage complexity by partitioning the design into multiple, separate modules

2011S1 wk10_2 P9

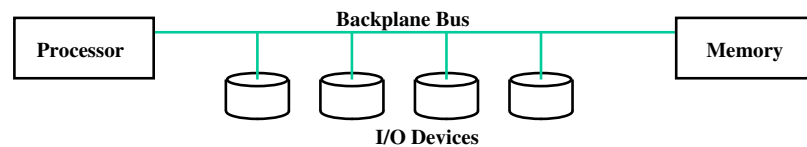
Disadvantage of Buses



- **It can create a communication bottleneck**
 - The bandwidth of the bus 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

2011S1 wk10_2 P10

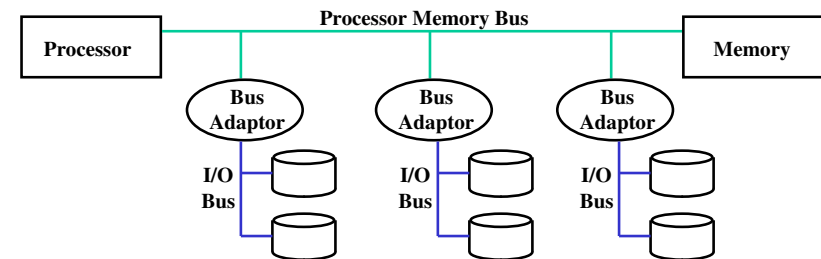
A Computer System with One Bus: Backplane Bus



- **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: IBM PC – AT**
CAN-bus embedded in car

2011S1 wk10_2 P11

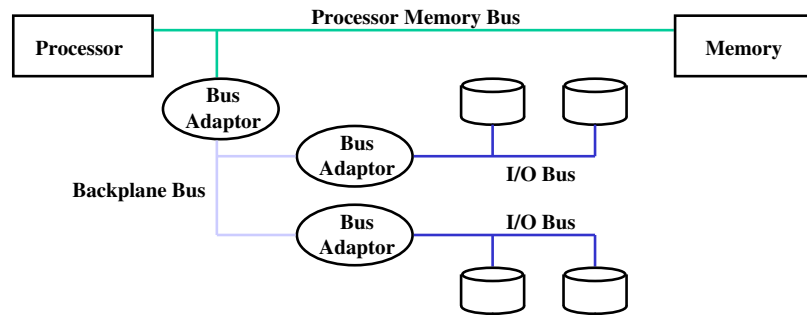
A Two-Bus System



- **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
 - SCSI Bus: the rest of the I/O devices

2011S1 wk10_2 P12

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**
- **Typical for modern, high-performance systems**

2011S1 wk10_2 P13

Principle Bus Design Issues

- **The speed and bandwidth of a bus are influenced by four main factors:**
 - the bus width
 - the bus clocking scheme
 - the bus arbitration method, and
 - the bus operation

2011S1 wk10_2 P14

Increasing Bus Bandwidth

- **Separate (vs multiplexed) address and data lines:**
 - Address and data can be transmitted in one bus cycle if separate address and data lines are available
- **Increase bus width:**
 - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles
- **Increase bus speed – operating frequency**
 - limited by bus skew as signals on different wires travel at slightly different speeds
- **Block transfers:**
 - Allow the bus to transfer multiple words in back-to-back bus cycles for increased throughput
 - Only send one address at the beginning
 - Do not release the bus until the last word is transferred

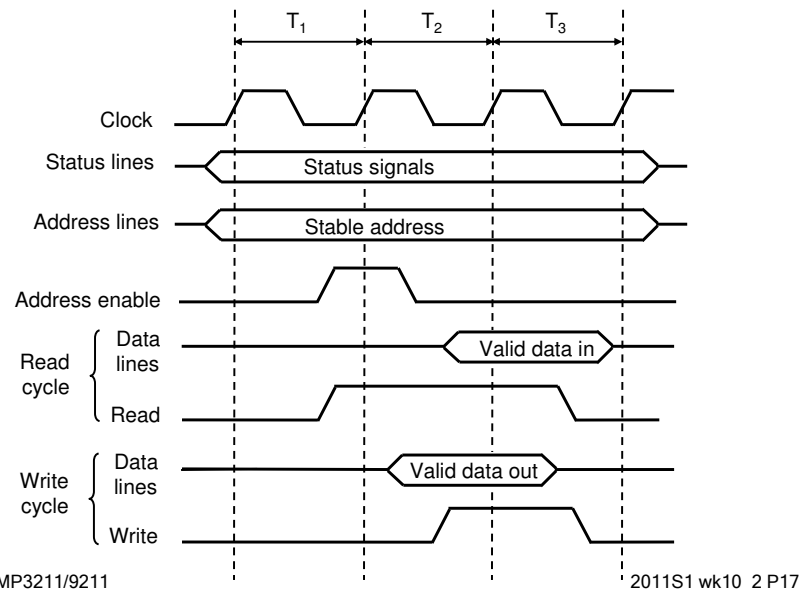
2011S1 wk10_2 P15

Bus Clocking: 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, they cannot be long if they are fast
- **Asynchronous Bus:**
 - Not clocked
 - Accommodates a wide range of devices
 - Can be lengthened without worrying about clock skew
 - Requires a handshaking protocol

2011S1 wk10_2 P16

Representative Synchronous Timing Diagram



Synchronous bus operation

Depending upon the protocol, signals undergo transitions and are latched on either rising or falling clock edges – in this example, values are sampled on rising clock edges

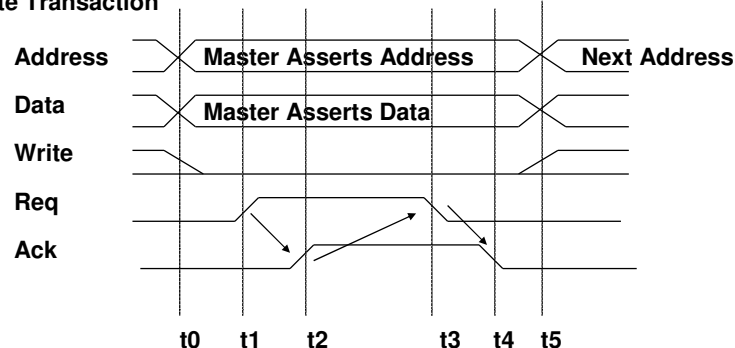
1. Status signals may be asserted to indicate the bus is in use – that a transaction is about to take place
2. The bus master drives an address onto the address lines and asserts an address enable line to indicate to the other devices that a transaction address is available
3. For read transactions the master indicates data is to be read, and in this case, the slave must respond within the third cycle so that the master may latch the result
4. For write transactions, the master places the data onto the bus and asserts a write control signal when the slave is to latch the data

COMP3211/9211

2011S1 wk10_2 P18

Example Asynchronous Handshake

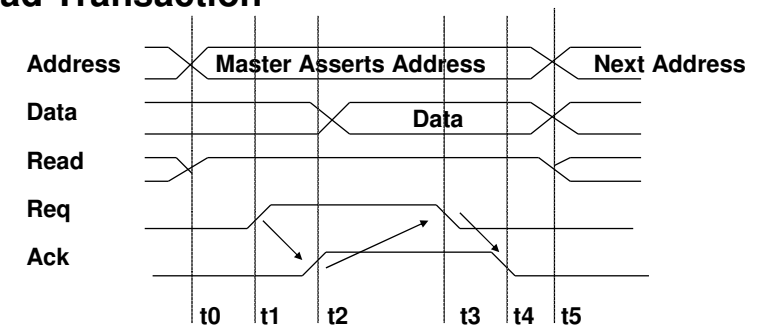
Write Transaction



- t₀ : Master has obtained control and asserts address, direction (r/~w), data; waits a specified amount of time for slaves to decode target address
- t₁: Master asserts req(uest) line
- t₂: Slave asserts ack(nowledge), indicating data received
- t₃: Master releases req indicating ack received
- t₄: Slave releases ack indicating transaction successful

2011S1 wk10_2 P19

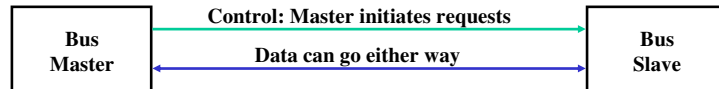
Read Transaction



- t₀ : Master has obtained control and asserts address & direction (r); waits a specified amount of time for slaves to decode target
- t₁: Master asserts req
- t₂: Slave asserts ack, indicating ready to transmit data
- t₃: Master releases req, indicating data received
- t₄: Slave releases ack

2011S1 wk10_2 P20

Arbitration: 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

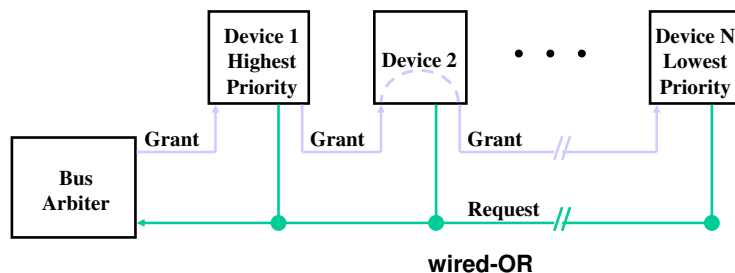
2011S1 wk10_2 P21

Multiple Potential Bus Masters: the need for arbitration

- **Bus arbitration scheme:**
 1. A bus master wanting to use the bus asserts the bus request
 2. A bus master cannot use the bus until its request is granted
 3. A bus master must signal to the arbiter after finished 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

2011S1 wk10_2 P22

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

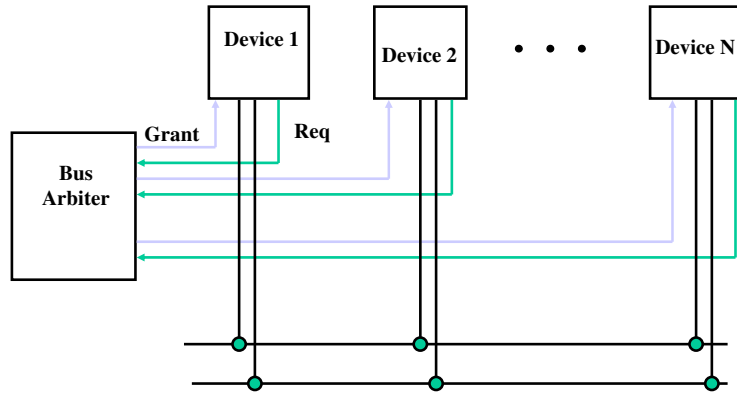
2011S1 wk10_2 P23

Daisy Chain Arbitration Protocol

- **Signal the request line**
- **Wait for a L → H transition on the grant line indicating that the bus is being reassigned**
- **Intercept the grant signal, and do not allow lower-priority devices to see it. Stop asserting the request line**
- **Use the bus**
- **Fairness can be improved by not allowing a device to reassert the request line until it sees the bus request line go low.**
- **Some bus systems use multiple daisy chains with separate request and grant lines for each daisy chain and a priority encoder to select from among the requests on multiple request lines.**

2011S1 wk10_2 P24

Centralized Parallel Arbitration



- Used in essentially all processor-memory busses and in high-speed I/O busses e.g. PCI
- Can use any one of a few arbitration schemes e.g. round robin, priority, etc.