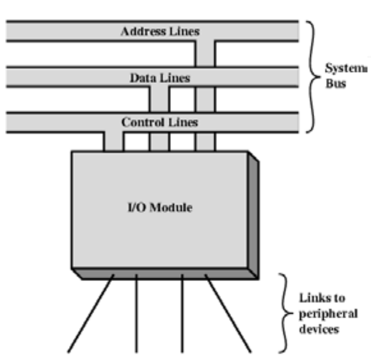
Input/Output

I/O Modules and Peripheral Devices

- I/O modules
  - Wide variety of peripherals
    - Delivering different amounts of data
    - At different speeds
    - In different formats
    - Performance mismatch with processor/memory
  - Interface
    - Between processor/memory and peripheral devices

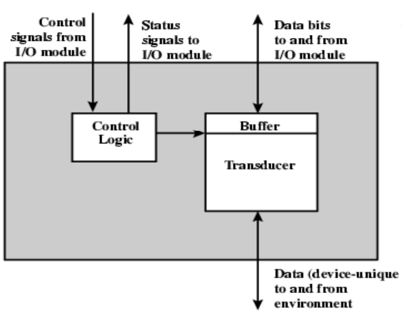
Generic Model of I/O Module



I/O Modules and Peripheral Devices

- I/O modules
- Peripheral devices
  - Provide a means of exchanging data between external environment and computer
  - Different categories
    - Human readable, machine readable, communication
  - Interface
    - Control, data, and status

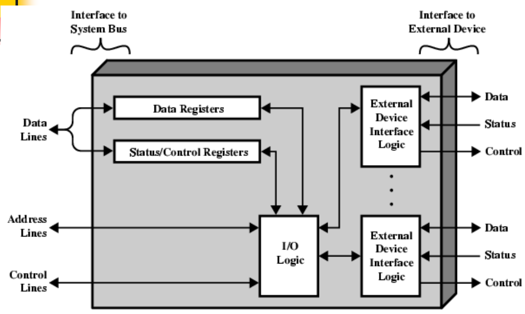
External Device Block Diagram



Typical I/O Steps

- CPU checks I/O module device status
- I/O module returns status
- If ready, CPU requests data transfer
- I/O module gets data from device
- I/O module transfers data to CPU

I/O Module Diagram



Three Types of I/O

- Programmed I/O
- Interrupt-Driven I/O
- Direct Memory Access

Programmed I/O

- CPU has direct control over I/O
  - Sensing status
  - Read/write commands
  - Transferring data
- Memory mapped and Isolated I/O

Memory Mapped I/O vs Isolated I/O

- Memory-mapped I/O
  - Controlling devices just like reading/writing memory
  - Requires no special instructions
- Isolated (Special-purpose) I/O
  - No loss of memory addresses to peripherals
  - Simpler address decoding logic
  - Can be faster
  - Special I/O instructions needed

Interrupt Driven I/O

- Overcomes CPU waiting
- No repeated CPU checking of device
- I/O module interrupts when ready

Interrupt Driven I/O Basic Operation

- Interrupt request signal (device → processor)
- Finishes execution of the current instruction
- Acknowledgement of interrupt
- Save PC and program state information
- Transfer control to interrupt service handler
- Restore PC and program state information
- Execute next instruction

Direct Memory Access

- Why DMA?
  - Interrupt driven and programmed I/O require active CPU intervention
    - Transfer rate is limited
    - CPU is tied up
  - DMA is the answer
- What is DMA (direct memory access)
  - An efficient technique to move large volume of data

How DMA ?

- Additional I/O module on system bus
- DMA controller takes over bus from CPU
  - CPU is suspended if requiring buses

DMA Operation

- CPU tells DMA controller:
  - Read/Write
  - Device address
  - Starting address of memory block for data
  - Amount of data to be transferred
- CPU carries on with other work
- DMA controller deals with transfer
- DMA controller sends interrupt when finished

Data Transfer Using DMA

- Cycle Steal
  - The DMA controller generates read or write signals at cycles when the processor is not using the bus
- Burst Transfer
  - Transfer a block of data within certain consecutive cycles
  - CPU may be halted completely.

Example

- Consider a device that can transfer data at rate of 8KB/s. Assume the device is driven by interrupt and the interrupt takes total 100 us for transferring one byte
  - If the processor has to be interrupted for every byte, what fraction of processor time is consumed by this device?
  - Now assume that the device has two 16-byte buffers and interrupts the processor when one of the buffers is full. The interrupt will take 8us more for each additional byte. How much processor time is consumed by this device?

Summary

- I/O interface and devices
- Three types of I/O
  - Program I/O
  - Interrupt I/O
  - DMA