Input and Output Fundamentals

Dr Andrew Scott

a.scott@lancaster.ac.uk

1

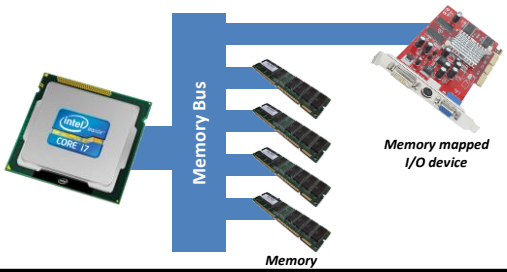
Overview

- Look at I/O mechanisms
  - How we implement and control access to devices
  - Clocks and Timers
    - Used to control and synchronise operations

2

Memory Mapped I/O

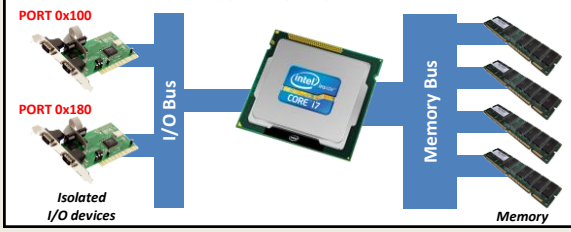
- I/O devices visible as normal memory locations
  - Easily accessed using C pointers



3

Isolated I/O

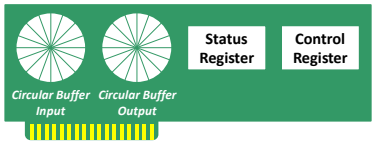
- Separate bus for I/O devices
  - Easier interfacing: distinct from high-speed memory bus
  - Accessed via special instructions
    - Generally not supported by compilers (must use assembler)



4

Optimising Memory Use

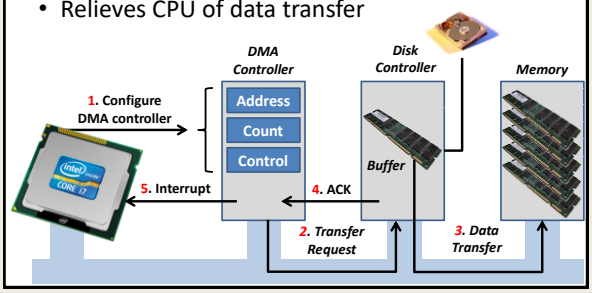
- Circular buffers help improve data throughput
  - Help decouple operation of processor and device
- Status Register holds flags for
  - Device ready
  - Data available, etc.



5

Direct Memory Access

- Alternative to programmed I/O
- Relieves CPU of data transfer



6

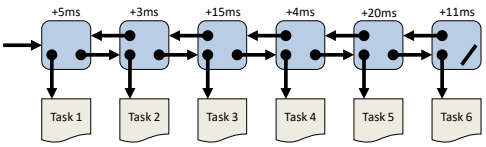
Clocks and Timers

- Real Time Clock (RTC)
  - Returns encoding of current date
- Programmable Interrupt Timer (PIT)
  - Creates periodic event at set/ configurable interval
  - Used to invoke periodic kernel tasks
  - If frequency too
    - Low: Kernel can seem unresponsive
    - High: OS spends too much time servicing timer

7

Avoiding Fine Grained Timers

- Create event queue
  - Timer configured to fire when next task due, rather than at set period
  - Set interval timer to time delta (+*n* ms) to next event
- Technique used widely: OSs, simulators, etc.



8