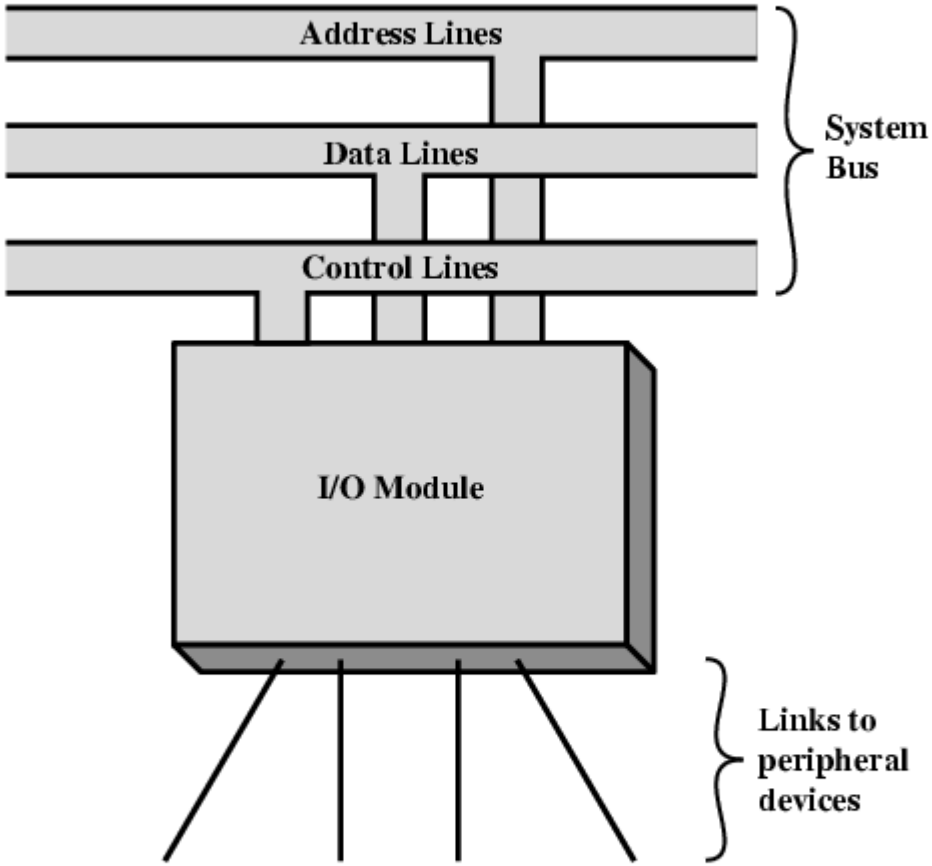
CHAPTER 4
INPUT/OUTPUT

Computer Architecture and Organization

CONTENT

- Why I/O Unit
- External Devices
- Input Output Techniques
- I/O Module Function
  - Programmed
  - Interrupt driven
  - Direct Memory Access (DMA)

GENERIC MODEL OF I/O MODULE



INPUT/OUTPUT PROBLEMS

- Wide variety of peripherals
  - Delivering different amounts of data
  - At different speeds
  - In different formats
- All slower than CPU and RAM
- Need I/O modules

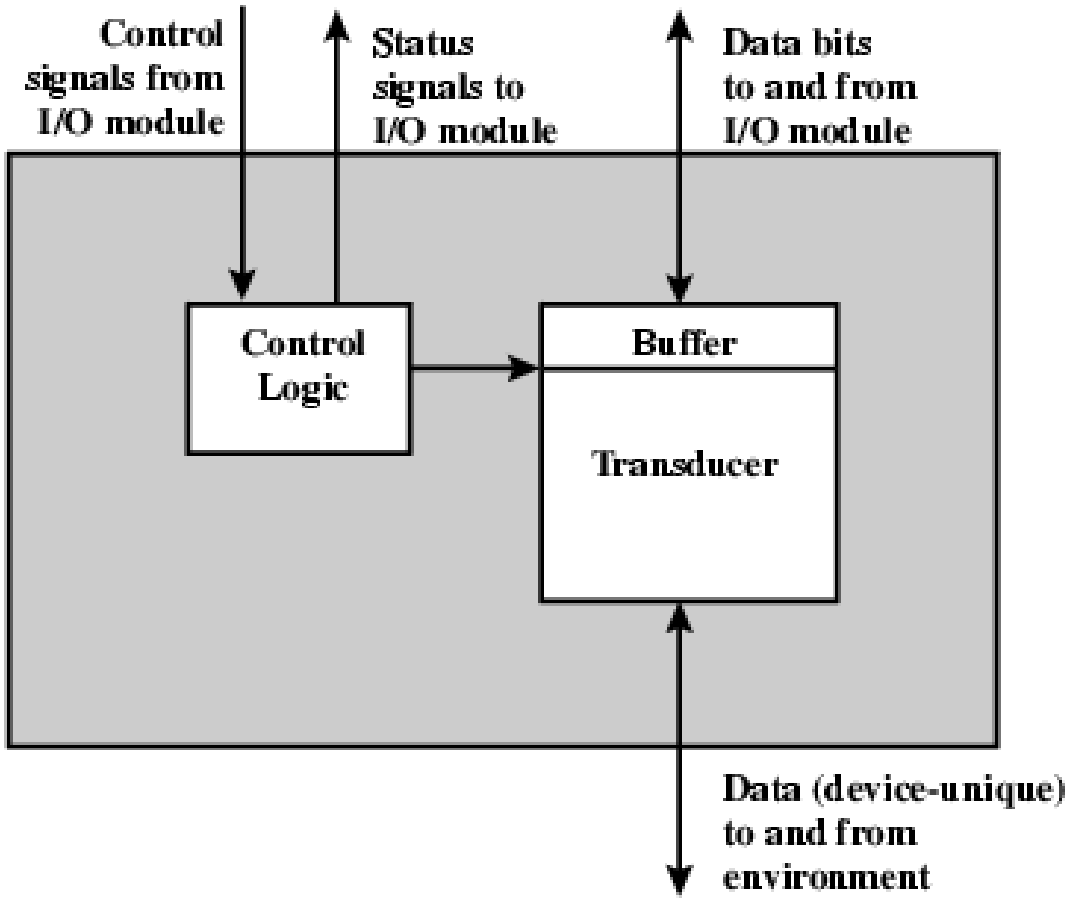
EXTERNAL DEVICES

- Provide a means of exchanging data between the external environment and the computer
- Attach to the computer by a link to an I/O module
  - The link is used to exchange **control**, **status**, and **data** between the I/O module and the external device
- *Peripheral device*
  - An external device connected to an I/O module

Three categories:

- Human readable
  - Suitable for communicating with the computer user
  - Video display terminals (VDTs), printers
- Machine readable
  - Suitable for communicating with equipment
  - Magnetic disk and tape systems, sensors and actuators
- Communication
  - Suitable for communicating with remote devices such as a terminal, a machine readable device, or another computer

EXTERNAL DEVICE BLOCK DIAGRAM



KEYBOARD/MONITOR

Most common means of computer/user interaction

User provides input through the keyboard

The monitor displays data provided by the computer

International Reference Alphabet (IRA)

- Basic unit of exchange is the character
  - Associated with each character is a code
  - Each character in this code is represented by a unique 7-bit binary code
    - 128 different characters can be represented
- Characters are of two types:
  - Printable
    - Alphabetic, numeric, and special characters that can be printed on paper or displayed on a screen
  - Control
    - Have to do with controlling the printing or displaying of characters
    - Example is carriage return
    - Other control characters are concerned with communications procedures

Keyboard Codes

- When the user depresses a key it generates an electronic signal that is interpreted by the transducer in the keyboard and translated into the bit pattern of the corresponding IRA code
- This bit pattern is transmitted to the I/O module in the computer
- On output, IRA code characters are transmitted to an external device from the I/O module
- The transducer interprets the code and sends the required electronic signals to the output device either to display the indicated character or perform the requested control function

I/O MODULE FUNCTION

- Control & Timing
  - Coordinates the flow of traffic between internal resources and external devices
- CPU Communication
  - Involves command decoding, data, status reporting, address recognition
- Device Communication
  - Involves commands, status information, and data
- Data Buffering
  - Performs the needed buffering operation to balance device and memory speeds
- Error Detection
  - Detects and reports transmission errors

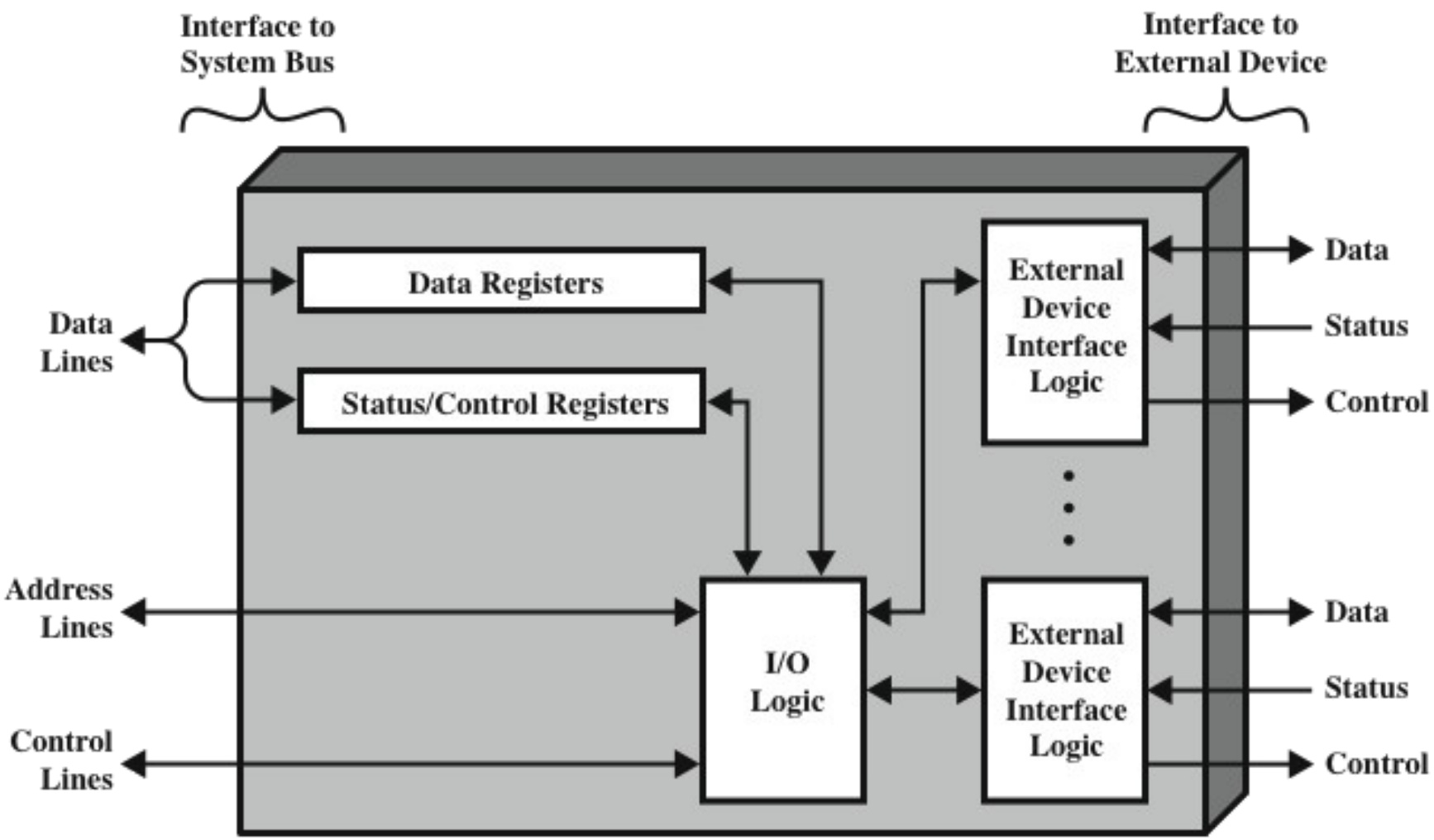
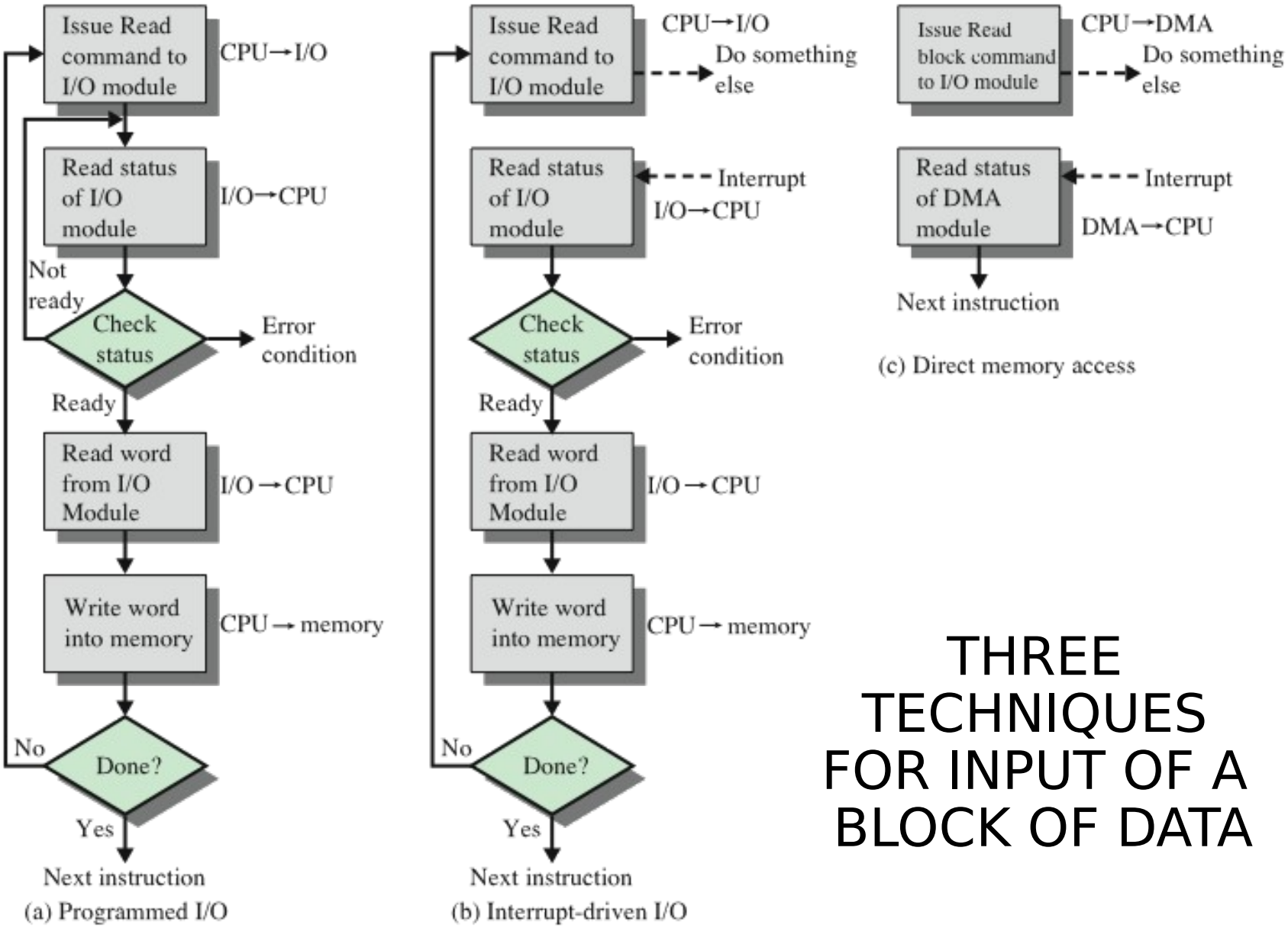


Figure 7.3 Block Diagram of an I/O Module

INPUT OUTPUT TECHNIQUES

- Programmed I/O
  - Data are exchanged between the processor and the I/O module
  - Processor executes a program that gives it direct control of the I/O operation
  - When the processor issues a command **it must wait** until the I/O operation is complete
  - If the processor is faster than the I/O module this is **wasteful of processor time**
- Interrupt-driven I/O
  - Processor issues an I/O command, **continues to execute other** instructions, and is **interrupted** by the I/O module when the latter has completed its work
- Direct memory access (DMA)
  - The I/O module and main memory exchange data directly **without processor involvement**



THREE TECHNIQUES FOR INPUT OF A BLOCK OF DATA

PROGRAMMED I/O

- CPU has direct control over I/O
  - Sensing status
  - Read/write commands
  - Transferring data
- CPU waits for I/O module to complete operation
- Wastes CPU time

PROGRAMMED I/O - DETAIL

- CPU requests I/O operation
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

I/O COMMANDS

- There are four types of I/O commands that an I/O module may receive when it is addressed by a processor:

  1) Control
     - used to activate a peripheral and tell it what to do

  2) Test
     - used to test various status conditions associated with an I/O module and its peripherals

  3) Read
     - causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer

  4) Write
     - causes the I/O module to take an item of data from the data bus and subsequently transmit that data item to the peripheral

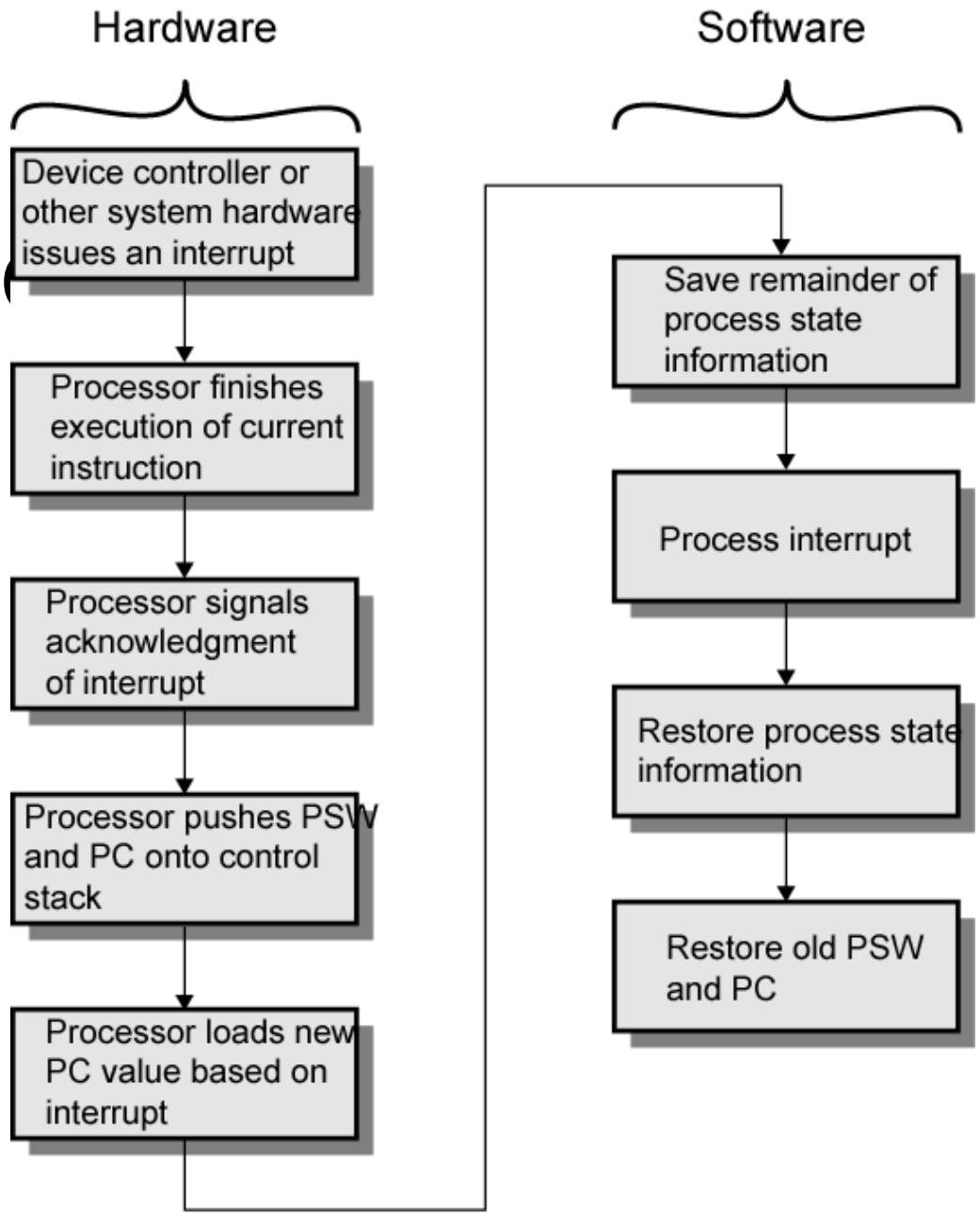
INTERRUPT DRIVEN I/O

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

INTERRUPT DRIVEN I/O BASIC OPERATION

- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data

SIMPLE INTERRUPT PROCESSING



CPU VIEWPOINT

- Issue read command
- Do other work
- Check for interrupt at end of each instruction cycle
- If interrupted:-
  - Save context (registers)
  - Process interrupt
    - Fetch data & store
- See Operating Systems notes

DESIGN ISSUES

- How do you identify the module issuing the interrupt?
- How do you deal with multiple interrupts?
  - i.e. an interrupt handler being interrupted

IDENTIFYING INTERRUPTING MODULE (1)

- Different line for each module
  - PC
  - Limits number of devices
- Software poll
  - CPU asks each module in turn
  - Slow

IDENTIFYING INTERRUPTING MODULE (2)

- Daisy Chain or Hardware poll
  - Interrupt Acknowledge sent down a chain
  - Module responsible places vector on bus
  - CPU uses vector to identify handler routine
- Bus Master
  - Module must claim the bus before it can raise interrupt
  - e.g. PCI & SCSI

MULTIPLE INTERRUPTS

- Each interrupt line has a priority
- Higher priority lines can interrupt lower priority lines
- If bus mastering only current master can interrupt

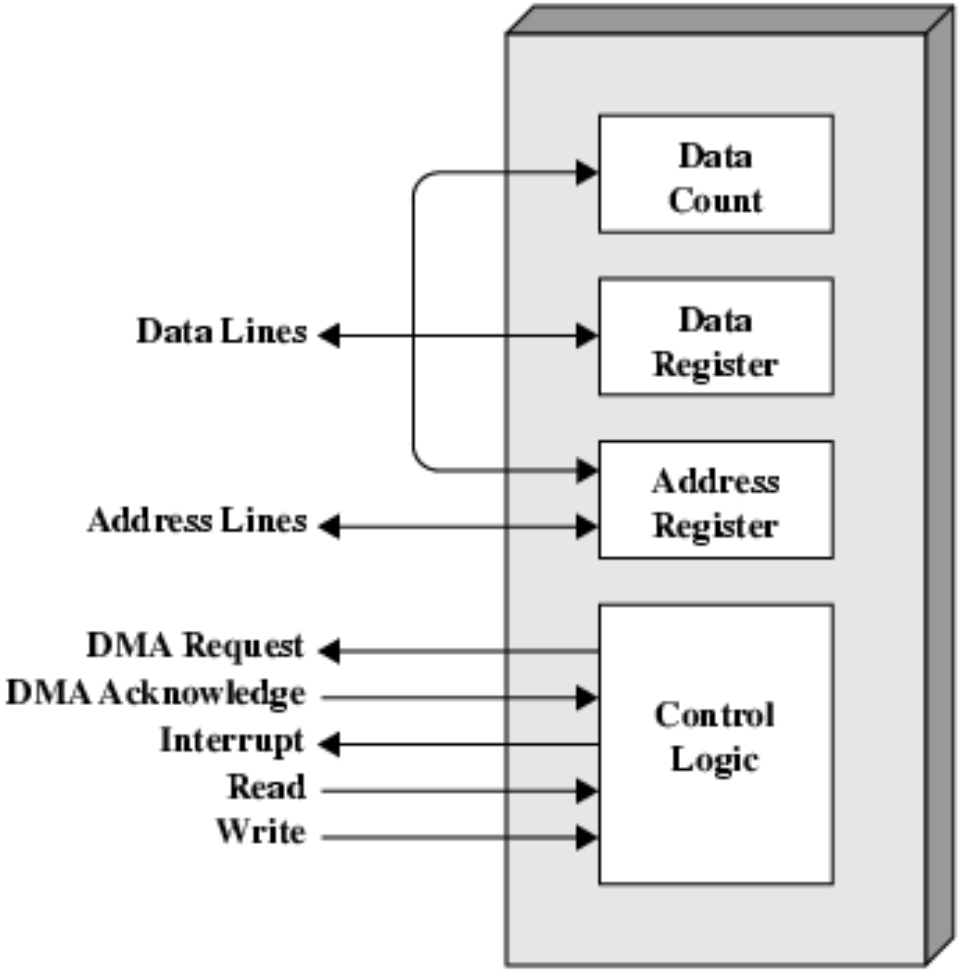
DIRECT MEMORY ACCESS

- Interrupt driven and programmed I/O require active CPU intervention
  - Transfer rate is limited
  - CPU is tied up
- DMA is the answer

DMA FUNCTION

- Additional Module (hardware) on bus
- DMA controller takes over from CPU for I/O

TYPICAL DMA MODULE DIAGRAM



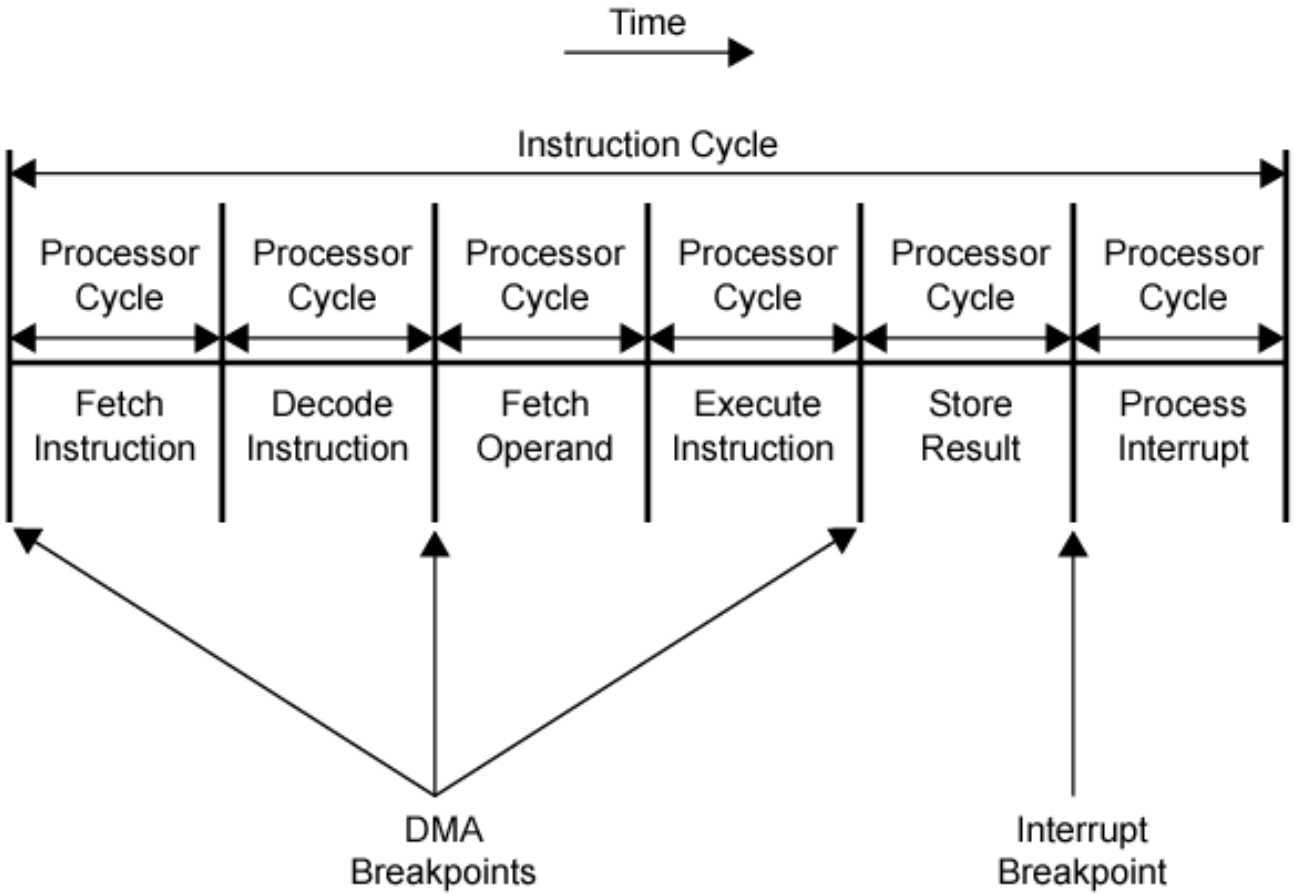
DMA OPERATION

- 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

DMA TRANSFER CYCLE STEALING

- DMA controller takes over bus for a cycle
- Transfer of one word of data
- Not an interrupt
  - CPU does not switch context
- CPU suspended just before it accesses bus
  - i.e. before an operand or data fetch or a data write
- Slows down CPU but not as much as CPU doing transfer

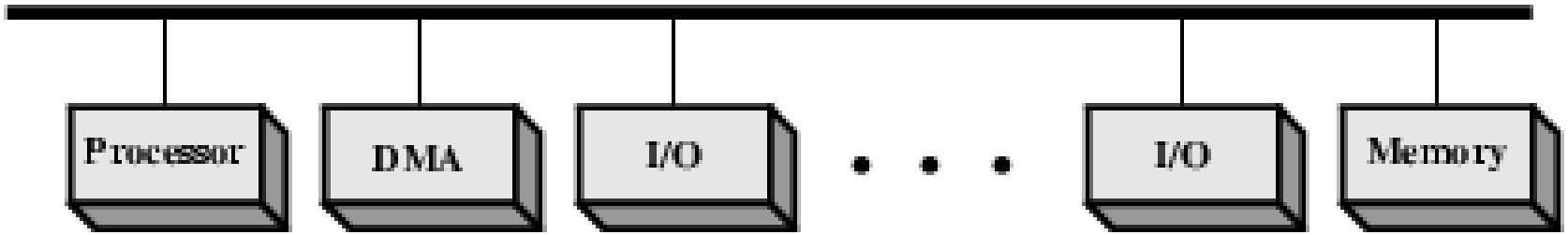
DMA AND INTERRUPT BREAKPOINTS DURING AN INSTRUCTION CYCLE



ASIDE

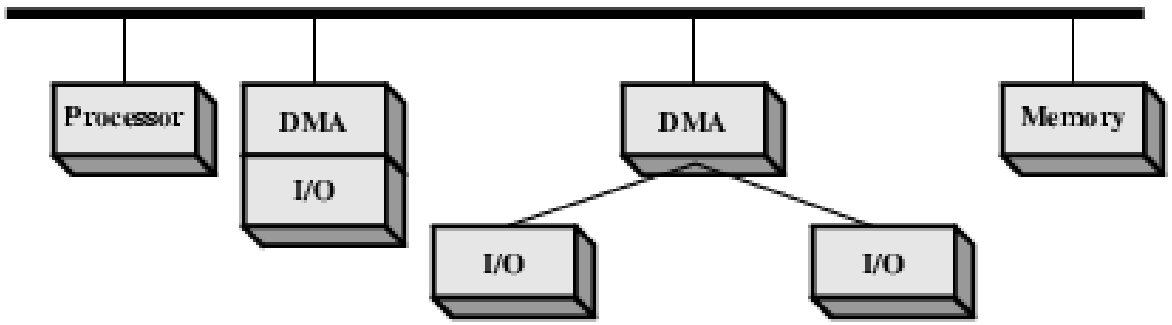
- What effect does caching memory have on DMA?
- What about on board cache?
- Hint: how much are the system buses available?

DMA CONFIGURATIONS (1)



- Single Bus, Detached DMA controller
- Each transfer uses bus twice
  - I/O to DMA then DMA to memory
- CPU is suspended twice

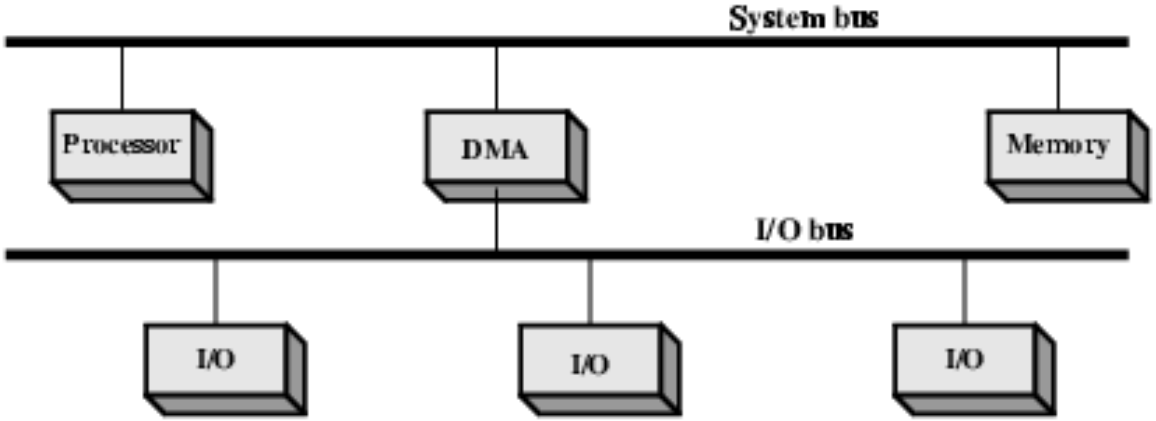
DMA CONFIGURATIONS (2)



(b) Single-bus, Integrated DMA-I/O

- Single Bus, Integrated DMA controller
- Controller may support >1 device
- Each transfer uses bus once
  - DMA to memory
- CPU is suspended once

DMA CONFIGURATIONS (3)



(c) I/O bus

- Separate I/O Bus
- Bus supports all DMA enabled devices
- Each transfer uses bus once
  - DMA to memory
- CPU is suspended once