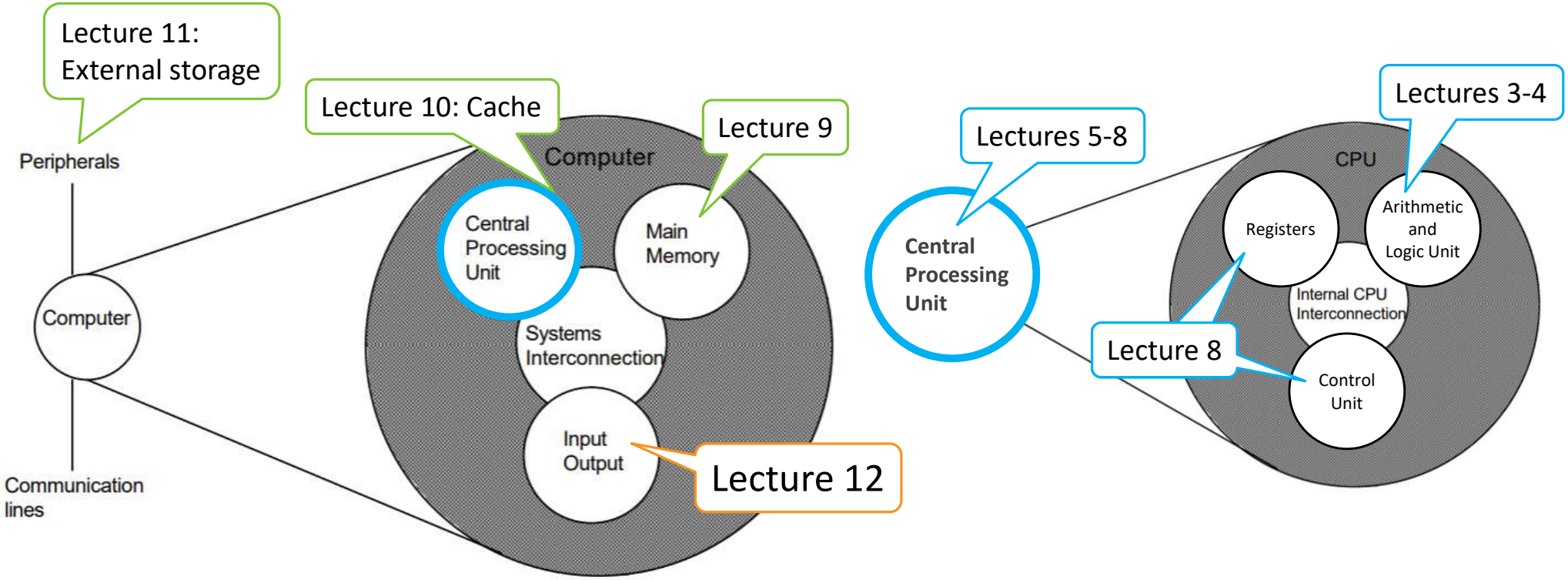
COMP2120A Computer Organization

LECTURE 12 INPUT/OUTPUT

Dr. Carmen Lam
Department of Computer Science, HKU

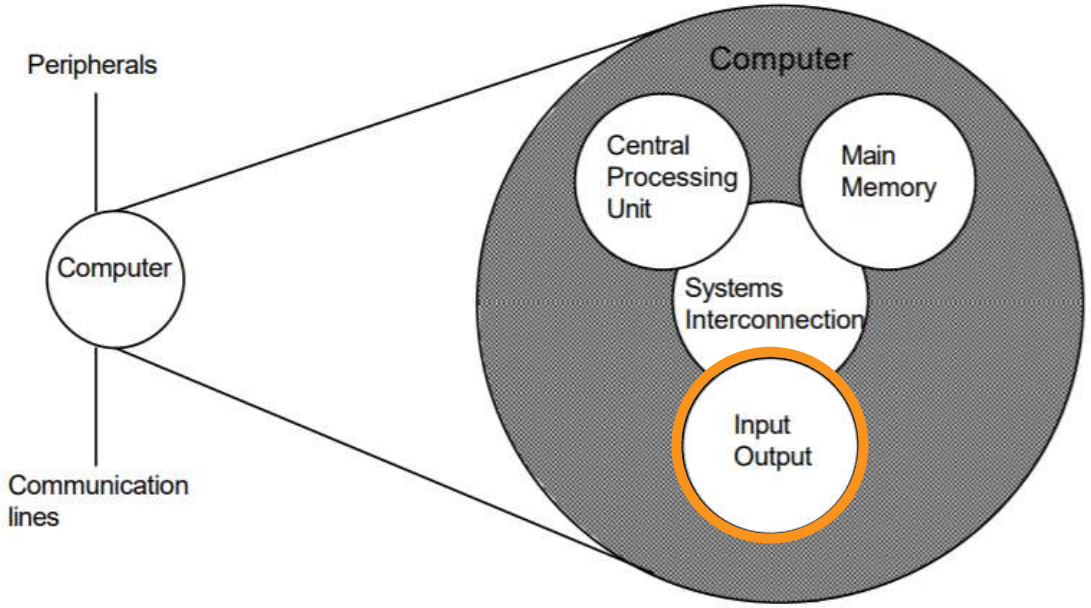
Recalling from Lecture 1: Structure of a computer (single-processor)



2

Outline

- Function and structure of an I/O module
- Internal I/O interface
  - Programmed I/O
  - Interrupt-Driven I/O
  - Direct Memory Access
  - Direct Cache Access
- External I/O interface



3

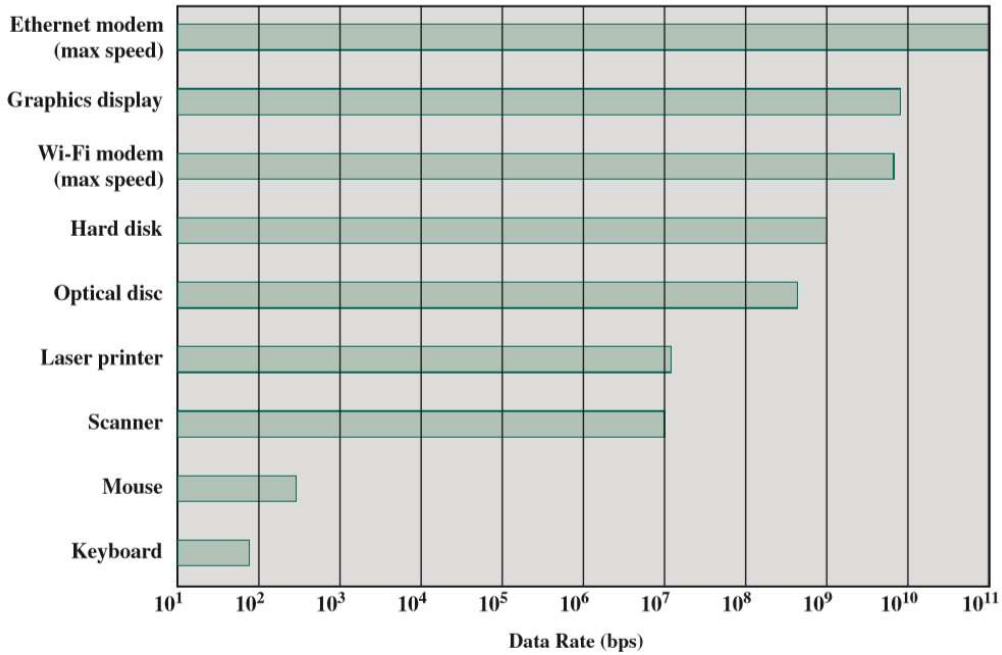
Course Learning Outcomes

1. **[Basic Computer Organization] Lectures** 3-5, 8-11, **12**, 13
   Understand the basic components of a computer, including CPU, memories, and input/output, and their organization.
2. **[Design Issues] Lectures** 8-**12**
   Understand the cost performance tradeoff in designing memory hierarchy and instruction sets.
3. **[Assembly Language Programming]** Lectures 6, 7, 13
   Understand and be able to use assembly languages for solving simple problems.
   Understand the relationship between high level language and assembly language.
4. **[Modern Trends]** Lectures 2, 8, 11
   Be able to follow the trends in computer design and appreciate the design philosophy behind.

4

The need for an I/O module

- Wide variety of peripherals, with various methods of operation, to control.
- Data transfer rate of peripherals is much slower than that of the memory or processor.
  - Cache: 10^{12} bps
  - RAM: 10^{11} bps
- Peripherals often use different data formats and word lengths than the computer.

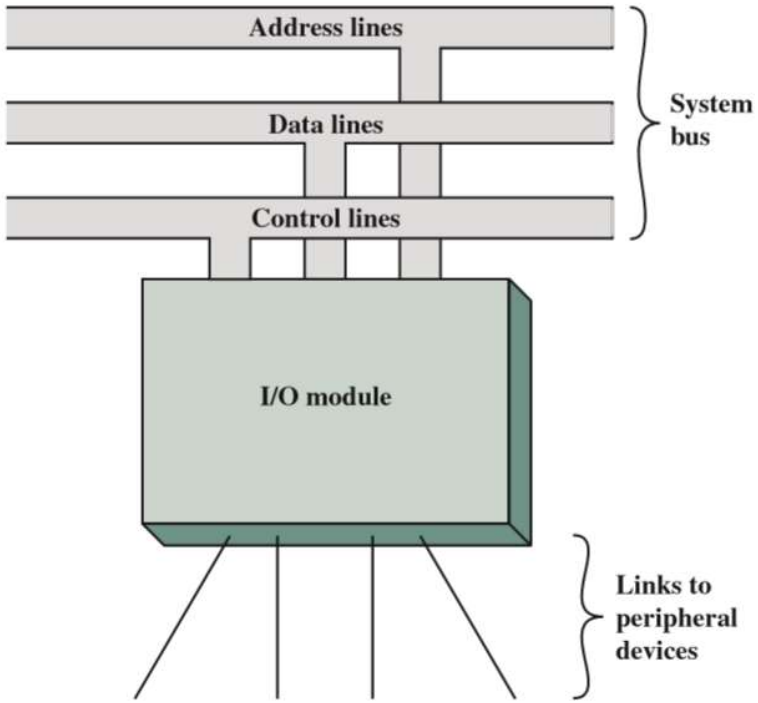


Typical I/O Device Data Rates

5

The need for an I/O module

- An I/O module is required for two major functions:
  - Interface to the processor and memory via the system bus or central switch.
  - Interface to one or more peripheral devices by tailored data links.
- An I/O module functions to allow the processor to view a wide range of devices in a simple-minded way.



Generic Model of an I/O Module

6

I/O Module Function

- Categories of the major functions or requirements for an I/O module:
  - Control and timing
  - Processor communication
  - Device communication
  - Data buffering
  - Error detection

7

I/O Module Function
Control and timing

- Control and timing is to coordinate the flow of traffic between internal resources and external devices.
- For example, the control of data transfer from an external device to the processor might involve the following steps:
  1. Processor sends a request to the I/O module.
  2. I/O module returns the device status.
  3. If device ready, processor requests the transfer of data, by means of a command to the I/O module.
  4. I/O module obtains data from device.
  5. Data transferred back to the processor.

8

I/O Module Function
Processor communication

I/O module must communicate with the processor:

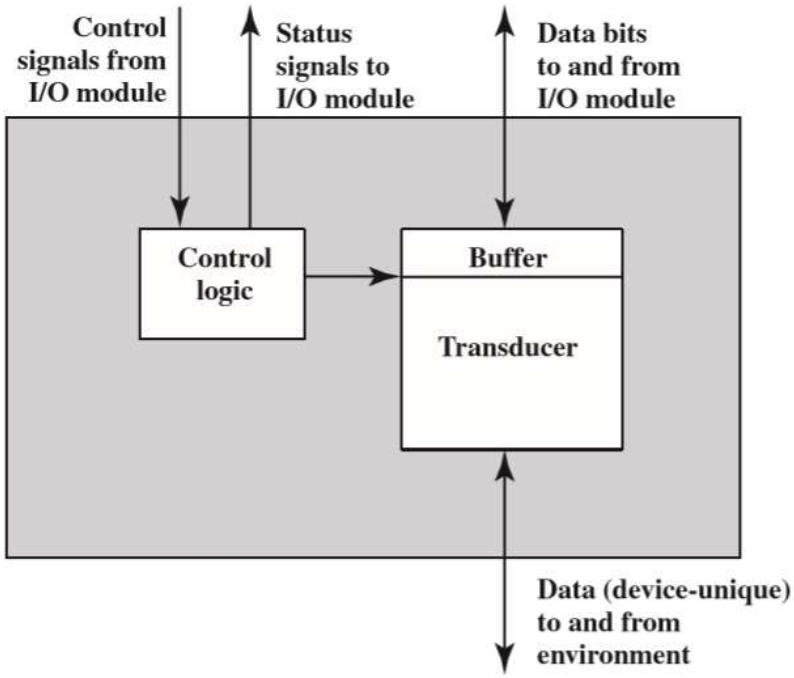
- **Command decoding** — decode commands from the control bus, e.g. READ, WRITE etc.
- **Data** — exchange data via data bus.
- **Status reporting** — e.g. BUSY, READY, paper out (for printer), etc.
- **Address recognition** — identify address of the peripheral.

9

I/O Module Function
Device communication

Device communication involves:

- **Control signals/commands** – determine the function that the device will perform, such as:
  - send data to the I/O module (INPUT/READ),
  - accept data from the I/O module (OUTPUT/WRITE),
  - report status, etc.
- **Status signals** – indicate the state of the device, e.g. READY, NOT-READY.
- **Data** – a set of bits to be sent to or received from the I/O module.

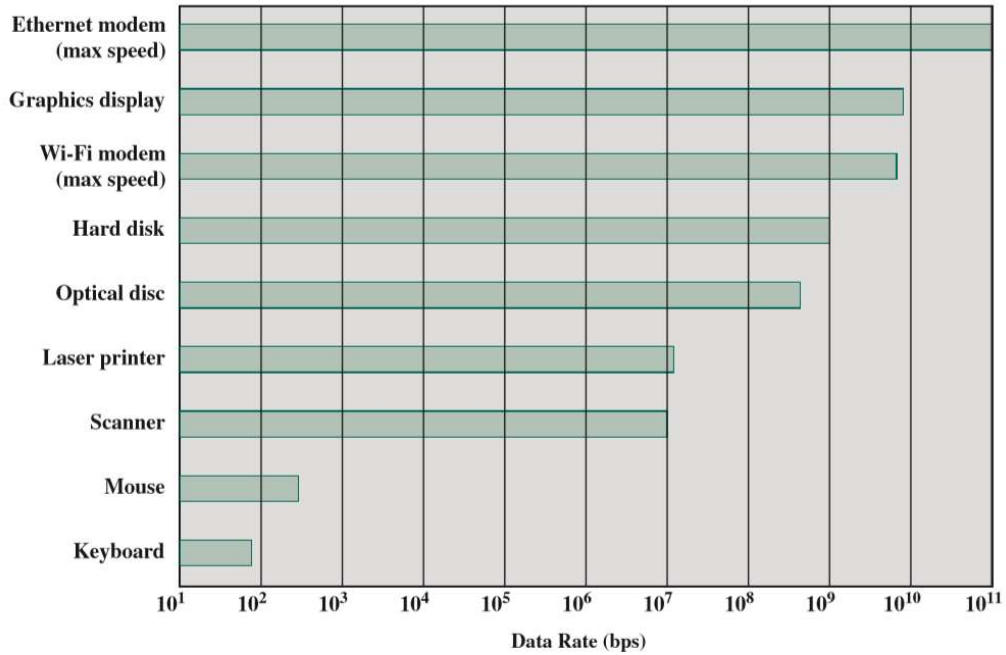


Block Diagram of an External Device

10

I/O Module Function
Data buffering

- Data transfer rate of peripherals is much slower than that of the memory or processor.
  - Cache: 10^{12} bps
  - RAM: 10^{11} bps
- Fast data from main memory
  → data are buffered in the I/O module and then sent to the peripheral device at its data rate.
- Slow data from peripherals
  → data are buffered so as not to tie up the memory in a slow transfer operation.



Typical I/O Device Data Rates

11

I/O Module Function
Error detection

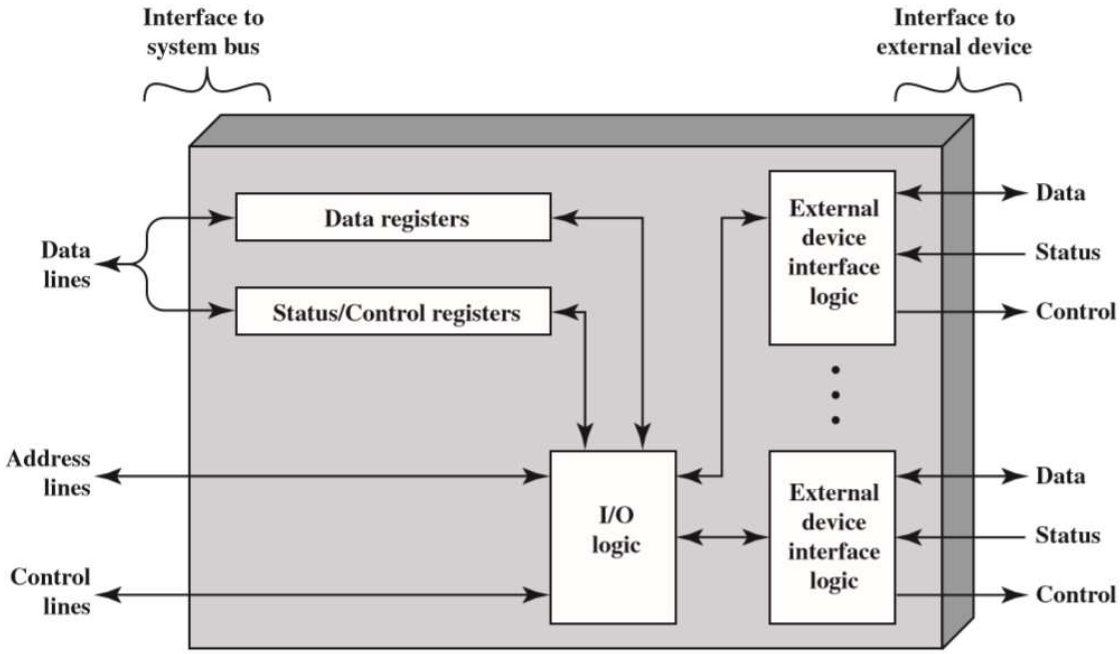
I/O module is responsible for error detection and reporting errors to the processor, for example:

- Mechanical and electrical malfunctions reported by the device (e.g., paper jam, bad disk track).
- Error-detecting code, e.g. parity bit, to detect transmission errors.

12

I/O Module Structure

- Data transferred to and from the module are buffered in one or more **data registers**.
- **Status registers** reflects the status of the devices.
- **Control registers** accept control information from the processor.
- **I/O logic** receives commands from control lines and recognizes addresses of the devices it controls.
- **External device interface logic** – logic specific to the interface with each device that it controls.



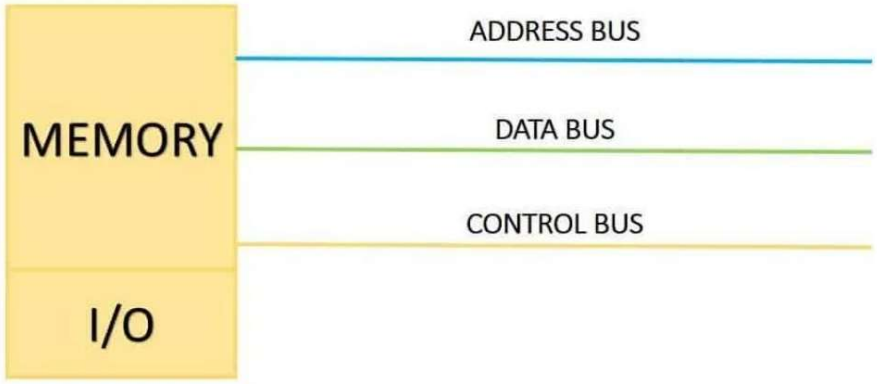
Block Diagram of an I/O Module

13

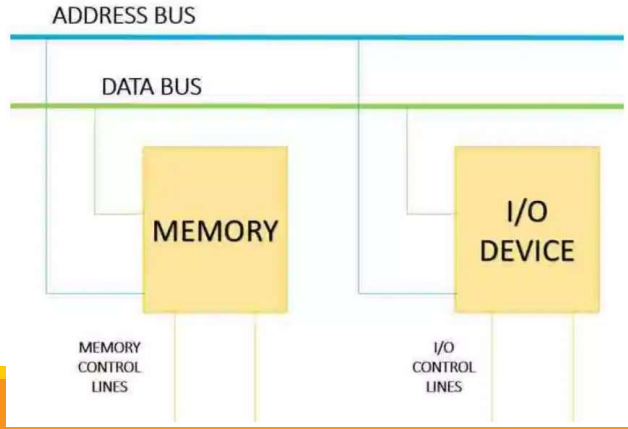
I/O Interfacing
Memory-mapped I/O vs Isolated I/O

- Many I/O devices connected through I/O modules to the system.
- Each device is given a unique identifier or address.
- Two modes of addressing:

Memory-mapped I/O: a single address space for memory locations and I/O devices.



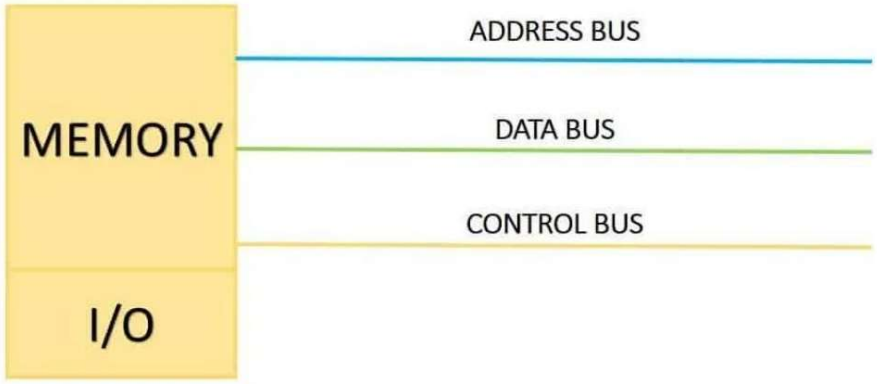
Isolated I/O: separate address space for memory locations and I/O devices.

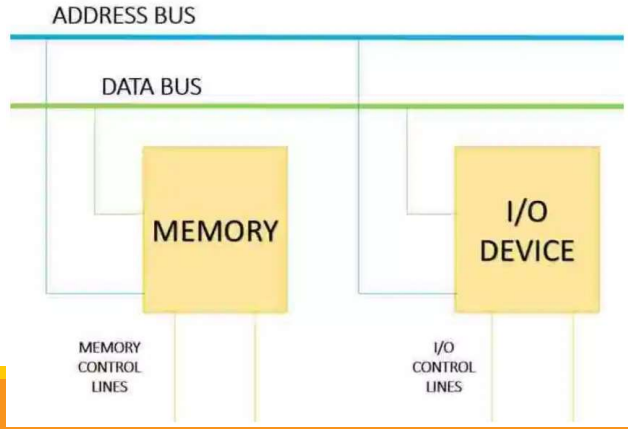


I/O Interfacing
Memory-mapped I/O vs Isolated I/O

- Same address space for memory and I/O.
- Only part of address space used for memory.
- Same instructions for both I/O and Memory.
- Simpler logic as I/O is treated as memory.
- Shared buses: less efficient but smaller in size.

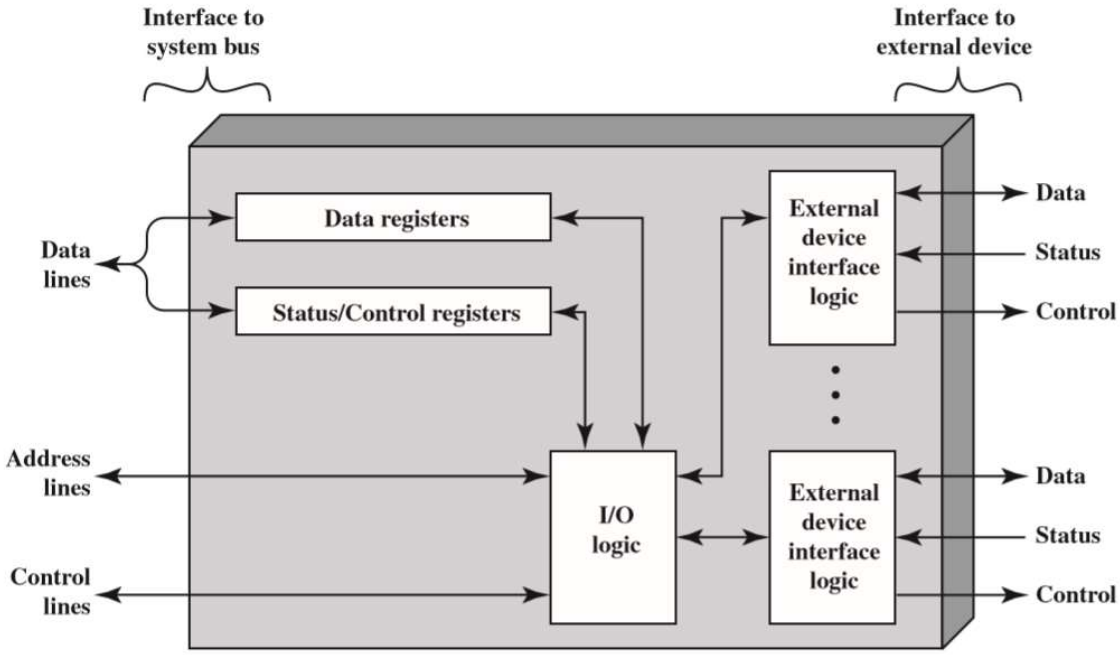
- Separate address space for memory and I/O.
- All addresses can be used by the memory.
- Different set of opcodes for I/O instructions.
- Separate logic, more complex.
- Separate buses: more efficient but larger in size.





I/O Interfacing
Memory-mapped I/O vs Isolated I/O

- CPU controls the operation of the I/O devices by writing/reading the data registers and status/control registers.
- Memory-mapped I/O:
  - CPU treats these registers as memory locations.
  - I/O operations are performed using memory read/write machine instructions.
- Isolated I/O:
  - These registers are in dedicated I/O ports.
  - I/O ports are accessible by special I/O instructions.
  - The control lines specify whether the address refers to a memory location or an I/O device.



Block Diagram of an I/O Module

16

Check your understanding

1. Input or output devices that are connected to computer are called

A. Input/Output Subsystem
B. Peripheral Devices
C. Interfaces
D. Interrupt

17

Check your understanding (Answer)

1. Input or output devices that are connected to computer are called

A. Input/Output Subsystem

B. Peripheral Devices

C. Interfaces

D. Interrupt

18

Check your understanding

2. Each peripheral device has associated with

A. CPU
B. Cache memory
C. Interface
D. Other device

19

Check your understanding (Answer)

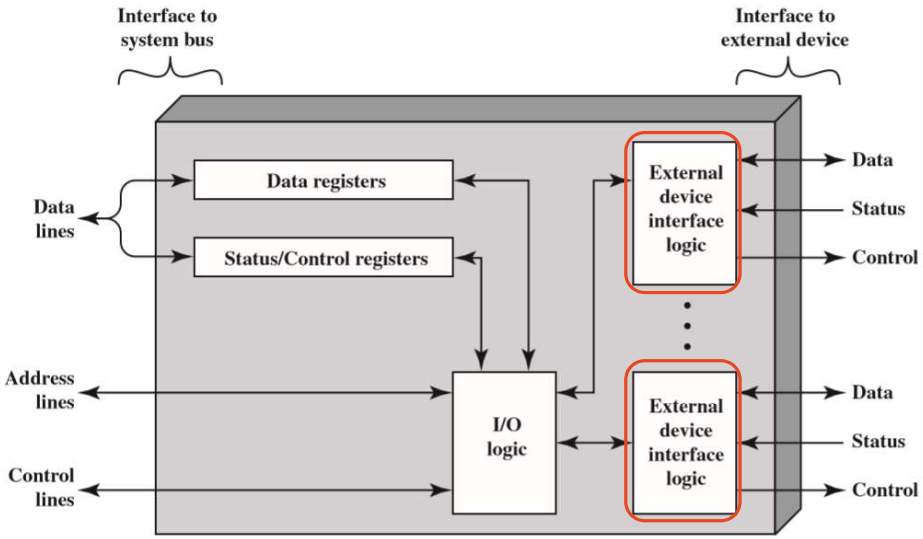
2. Each peripheral device has associated with

A. CPU

B. Cache memory

C. Interface

D. Other device



Block Diagram of an I/O Module

20

Check your understanding

3. The I/O module communicates with the CPU through the

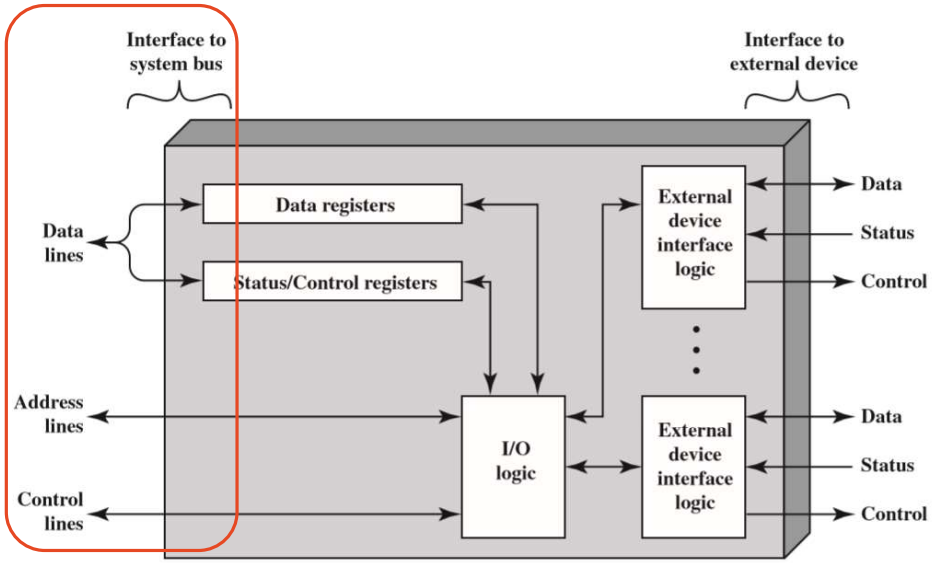
A. Bus
B. Cache
C. Other interface
D. None of the above

21

Check your understanding (Answer)

3. The I/O module communicates with the CPU through the

A. Bus
B. Cache
C. Other interface
D. None of the above



Block Diagram of an I/O Module

22

Check your understanding

4. The system is notified of a read or write operation by

A. appending an extra bit of the address.
B. enabling the read or write bits of the devices.
C. raising an appropriate interrupt signal.
D. sending a special signal along the BUS.

23

Check your understanding (Answer)

4. The system is notified of a read or write operation by

A. appending an extra bit of the address.
B. enabling the read or write bits of the devices.
C. raising an appropriate interrupt signal.
D. sending a special signal along the BUS.

24

Check your understanding

5. To overcome the lag in the operating speeds of the I/O device and the processor we use

A. control signals
B. buffer spaces
C. status flags
D. exceptions

25

Check your understanding (Answer)

5. To overcome the lag in the operating speeds of the I/O device and the processor we use

A. control signals
B. buffer spaces
C. status flags
D. exceptions

26

Check your understanding

6. A scheme in which portions of the address space are given to I/O devices is called

A. Address mapped I/O
B. Data mapped I/O
C. Isolated I/O
D. Memory mapped I/O

27

Check your understanding (Answer)

6. A scheme in which portions of the address space are given to I/O devices is called

A. Address mapped I/O
B. Data mapped I/O
C. Isolated I/O
D. Memory mapped I/O

28

Check your understanding

7. Which of the following is/are advantage(s) of memory-mapped I/O to isolated I/O? (more than one answer)

A. All address can be used by the memory.

B. More efficient programming.

C. More efficient use of bus.

D. Simpler logic.

29

Check your understanding (Answer)

7. Which of the following is/are advantage(s) of memory-mapped I/O to isolated I/O? (more than one answer)

A. All address can be used by the memory.

B. More efficient programming.

C. More efficient use of bus.

D. Simpler logic.

30

I/O Techniques

- There are three techniques for I/O operations:
  - Programmed I/O
  - Interrupt-driven I/O
  - Direct Memory Access (DMA)

I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

31

Programmed I/O

- Processor executes a program and encounters an I/O-related instructions.

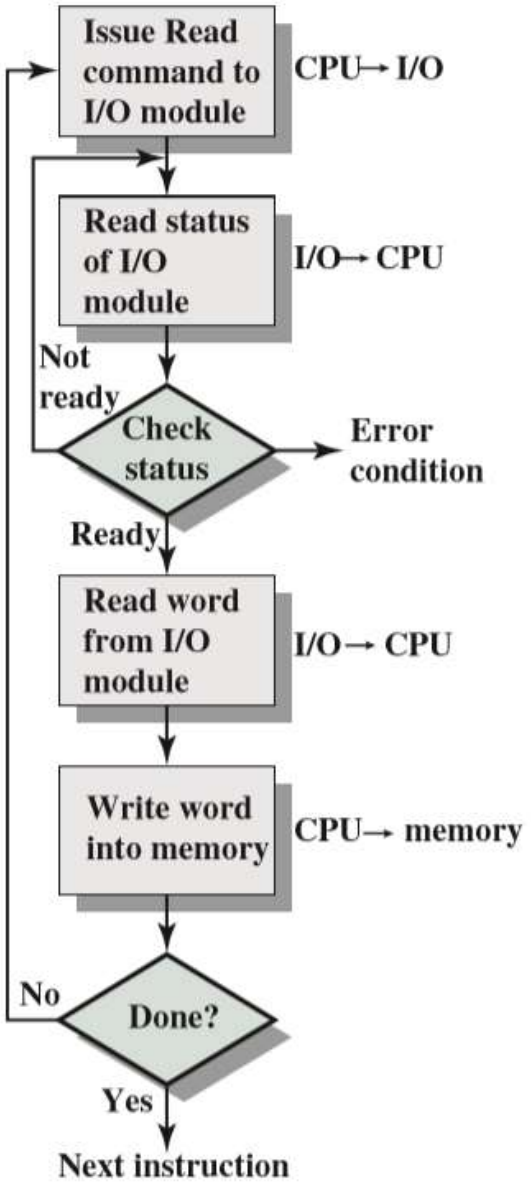
→ Processor issues a command to the I/O module.

→ The I/O module performs the requested action while processor waits until the operation is finished.

→ The I/O module finished the operation and set the appropriate bits in the I/O status register.

→ Processor needs to repeatedly check the status register to see if the I/O operation is finished and then continue the program execution.

- Will waste CPU cycle waiting for the device, especially for slow device like printer.



I/O Commands

- To execute an I/O-related instruction, the processor issues an address, specifying the particular I/O module and external device, and an I/O command.

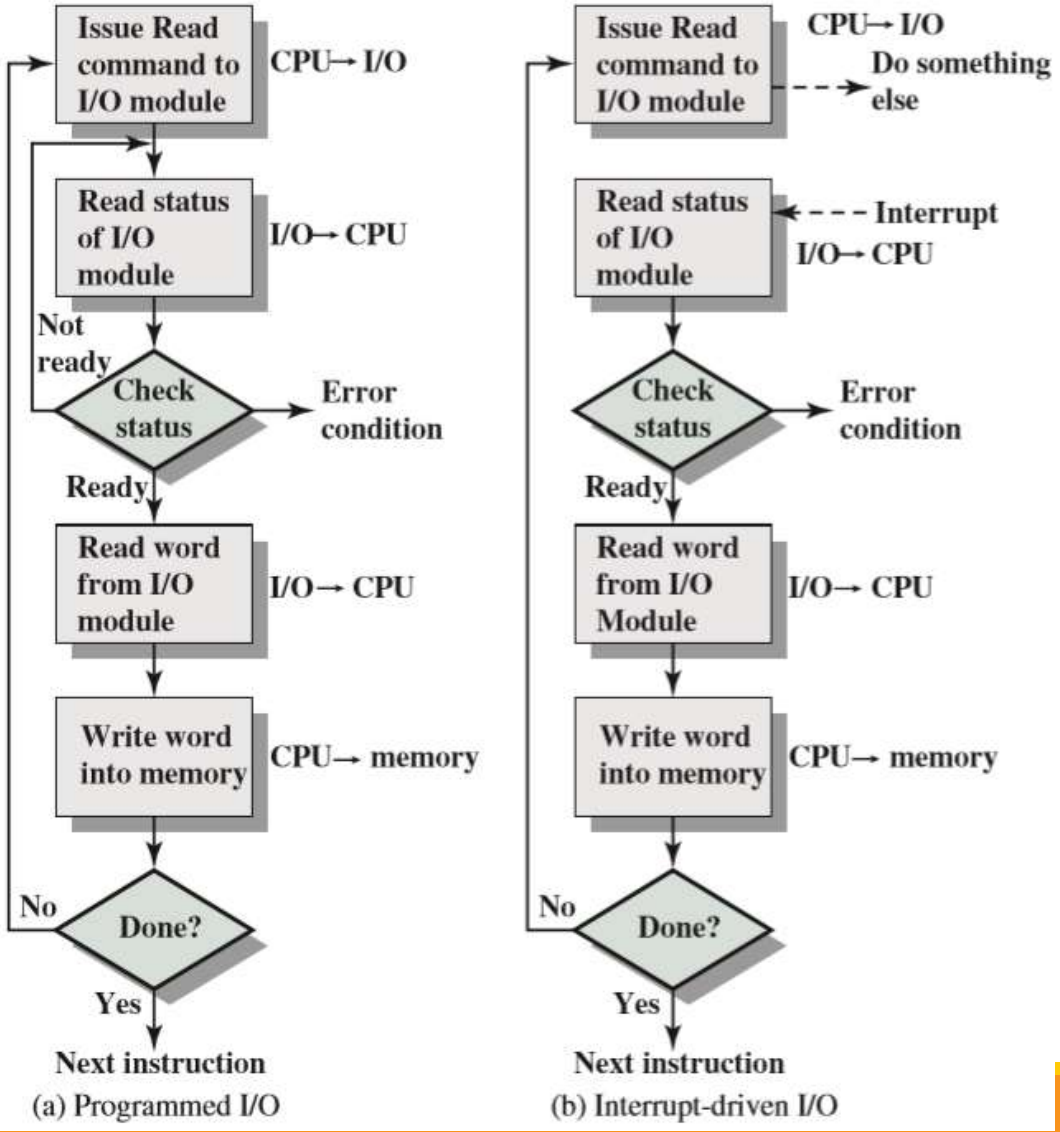
Four types of I/O commands:

- **Control** – Used to activate a peripheral and tell it what to do.
- **Test** – Used to test I/O module/device status.
- **Read**: data from the peripheral → I/O module internal buffer → data bus →CPU.
- **Write**: take data from data bus → peripheral.

33

Interrupt-driven I/O

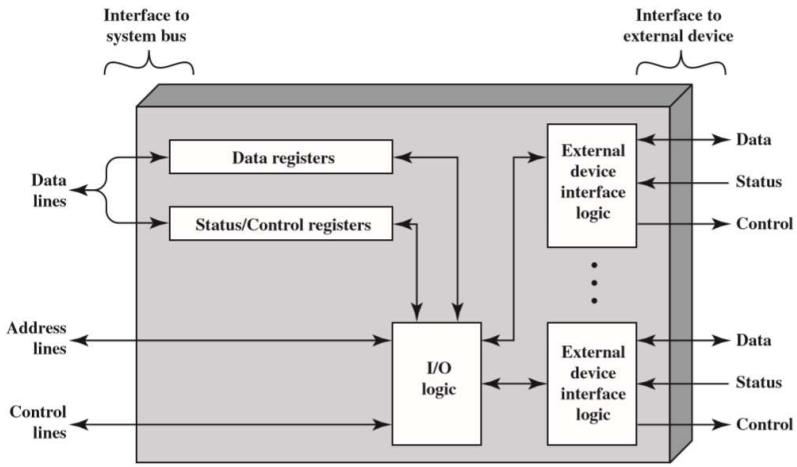
- Programmed I/O: CPU waits for the device to finish its I/O operation → waste CPU time.
- **Interrupt-driven I/O**:
  - CPU issues an I/O command, then continues to execute instructions from other process.
  - When I/O finishes, I/O module interrupts the CPU.
  - CPU will then suspend the current program, executes the remaining I/O operations, then return to the suspended program.



Interrupt-driven I/O Example

From I/O module's point of view, the action for input:

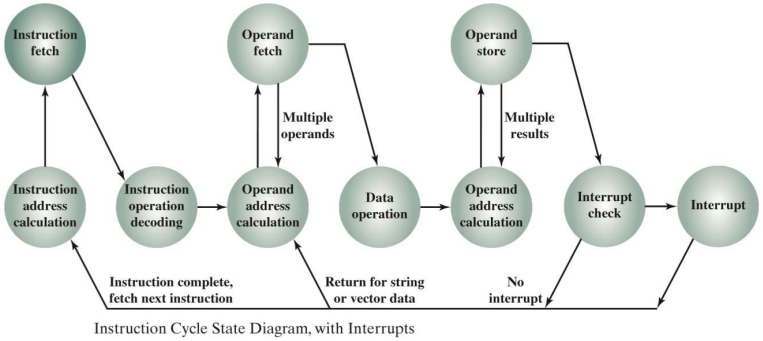
- The I/O module receives a READ command from the processor.
- The I/O module then proceeds to read data in from an associated peripheral.
- Once the data are in the module's data register, the module signals an interrupt to the processor over a control line.
- The module then waits until its data are requested by the processor.
- When the request is made, the module places its data on the data bus and is then ready for another I/O operation.



Block Diagram of an I/O Module

35

Interrupt-driven I/O Example



Instruction Cycle State Diagram, with Interrupts

From processor's point of view, the action for input:

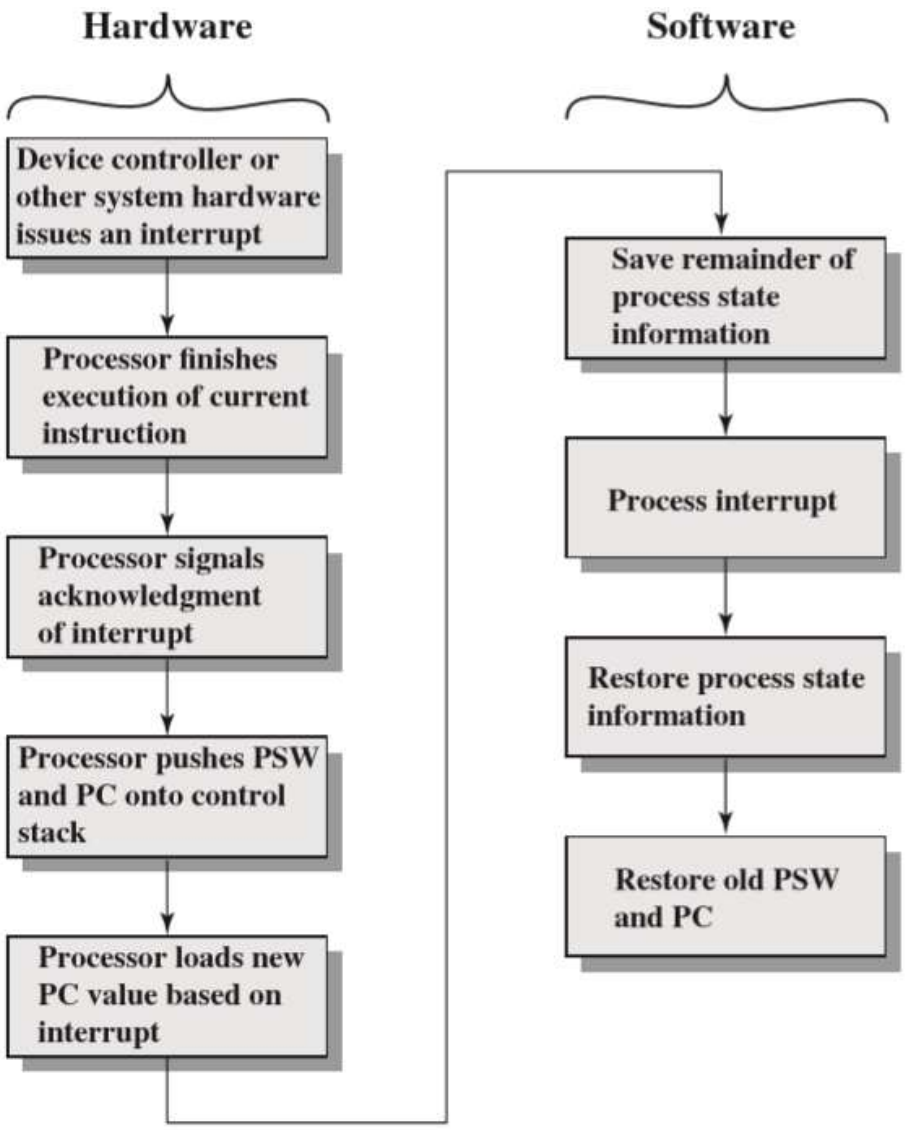
- The processor issues a READ command, then goes off and does something else.
- At the end of each instruction cycle, the processor checks for interrupts.
- When the interrupt from the I/O module occurs, the processor saves the context (e.g. program counter and processor registers) of the current program and processes the interrupt.
- In this example, the processor reads the word of data from the I/O module and stores it in memory.
- The processor then restores the context of the program it was working on and resumes execution.

36

Interrupt Processing

When an I/O device completes an I/O operation, the following sequence of hardware events occurs:

1. The I/O module issues an interrupt signal to CPU.
2. CPU finishes execution of the current instruction before responding to the interrupt.
3. CPU tests for an interrupt, and signals acknowledgement of interrupt if there is one.
4. Save information (PSW and PC) needed to resume the current program at the point of interrupt.
5. CPU loads the program counter with the entry location of the corresponding interrupt-handling program.

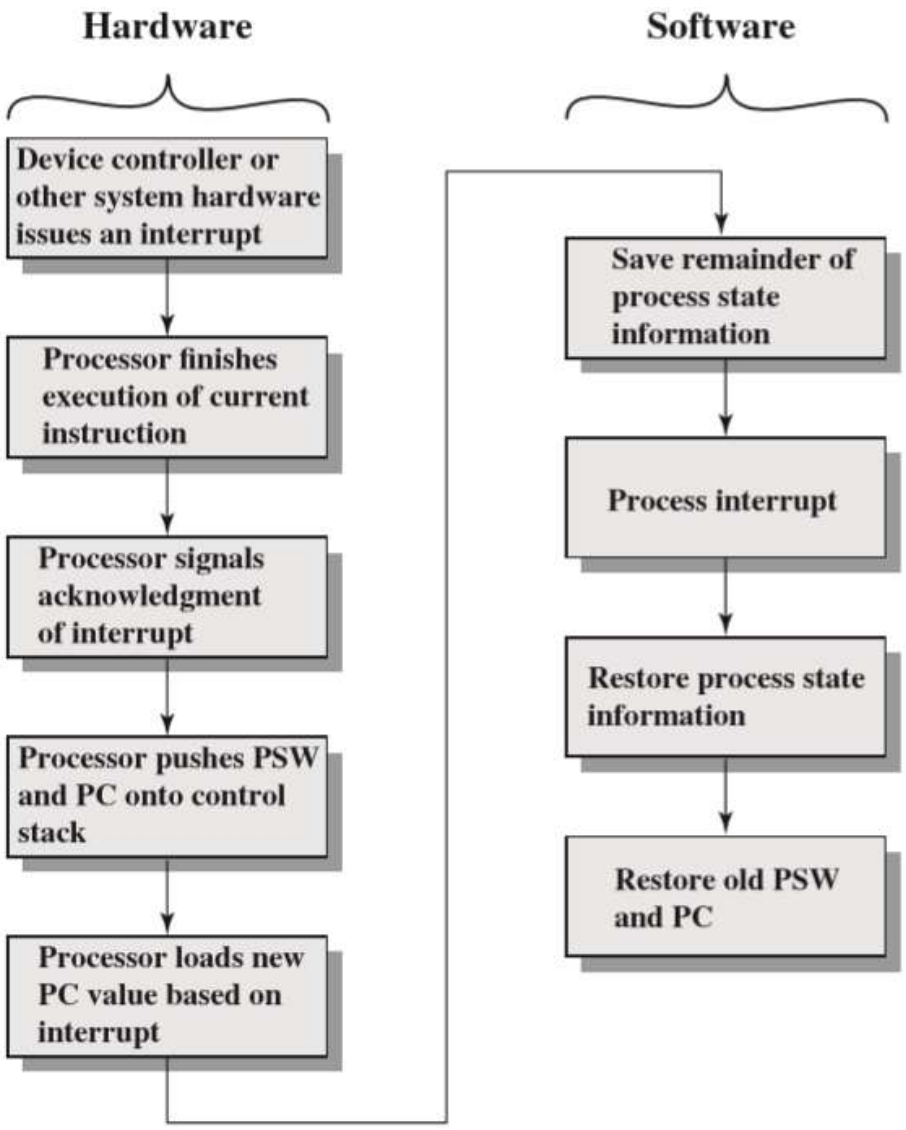


Simple Interrupt Processing

Interrupt Processing

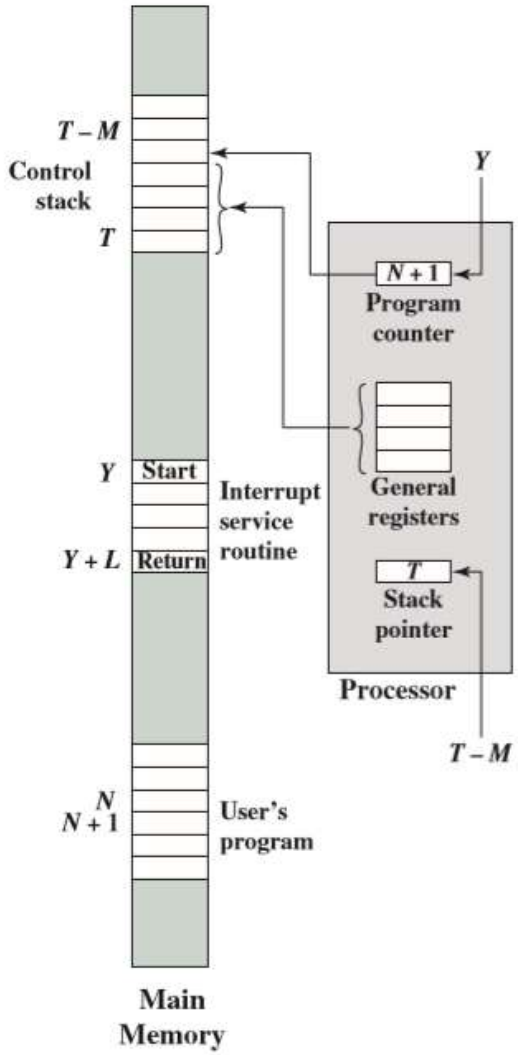
The next instruction cycle, control is transferred to the interrupt-handler program, with the following operations:

6. The interrupt handler saves the contents of all registers.
7. Processes the interrupt.
8. When interrupt processing is complete, the saved register values are retrieved from the stack and restored to the registers.
9. Restore the PSW and program counter values from the stack.

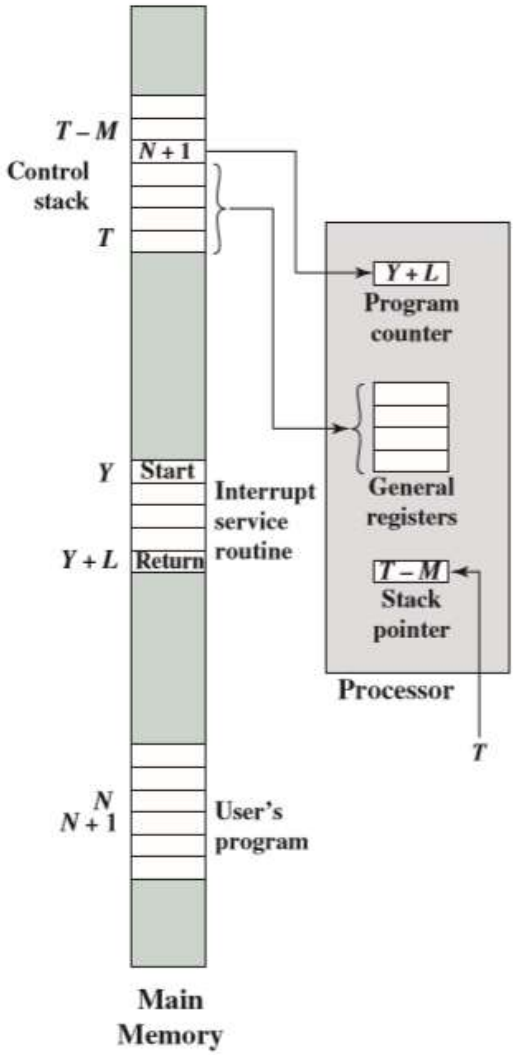


Simple Interrupt Processing

Example



(a) Interrupt occurs after instruction at location N



(b) Return from interrupt

39

Check your understanding

8. The method of synchronizing the processor with the I/O device in which the device sends a signal when it is ready is?

A. Signal handling
B. Exceptions
C. Interrupts
D. DMA

40

Check your understanding (Answer)

8. The method of synchronizing the processor with the I/O device in which the device sends a signal when it is ready is?

A. Signal handling
B. Exceptions
C. Interrupts
D. DMA

41

Check your understanding

9. On receiving an interrupt from an I/O device, the CPU

A. halts for predetermined time.

B. branches off to the interrupt service routine immediately.

C. branches off to the interrupt service routine after completion of the current instruction.

D. hands over control of address bus and data bus to the interrupting device.

42

Check your understanding (Answer)

9. On receiving an interrupt from an I/O device, the CPU

A. halts for predetermined time.

B. branches off to the interrupt service routine immediately.

C. branches off to the interrupt service routine after completion of the current instruction.

D. hands over control of address bus and data bus to the interrupting device.

43

Check your understanding

10. The interrupt-request line is a part of the

A. Data line
B. Control line
C. Address line
D. None of the mentioned

44

Check your understanding (Answer)

10. The interrupt-request line is a part of the

A. Data line

B. Control line

C. Address line

D. None of the mentioned

45

Check your understanding

11. The return address from the interrupt-service routine is stored on the

A. System heap
B. Processor register
C. Processor stack
D. Memory

46

Check your understanding (Answer)

11. The return address from the interrupt-service routine is stored on the

A. System heap
B. Processor register
C. Processor stack
D. Memory

47

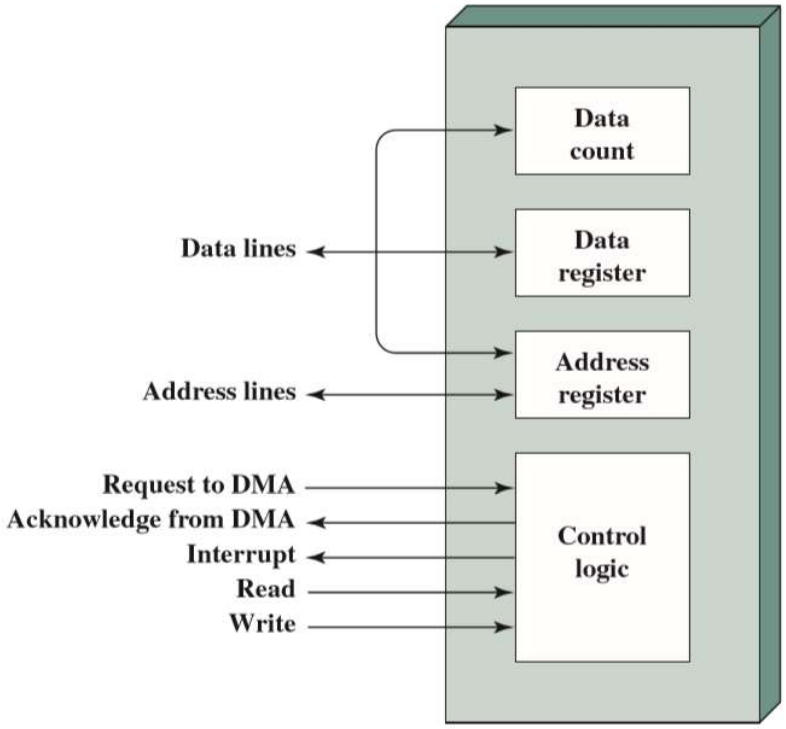
Drawbacks of Programmed and Interrupt-Driven I/O

- Requires the active intervention of the processor to transfer data between memory and an I/O module, drawbacks:
  - I/O transfer rate is limited by the speed with which CPU can test and service a device.
  - CPU is tied up in managing an I/O transfer.
- Programmed I/O: CPU is dedicated to the task of I/O and can move data at a rather high rate, at the cost of doing nothing else.
- Interrupt I/O: frees up CPU for other tasks at the cost of the I/O transfer rate.
- **Direct Memory Access (DMA)**: more efficient, esp. for large volumes of data.

48

Direct Memory Access (DMA)

- Use **DMA module**, an intelligent device controller that contains an I/O processor, to minimize CPU intervention.
- CPU will tell the I/O processor to perform, for example: Read the device and place the data in memory location x to y.
- The I/O processor will directly write to the memory.



Typical DMA Block Diagram

49

Direct Memory Access (DMA)

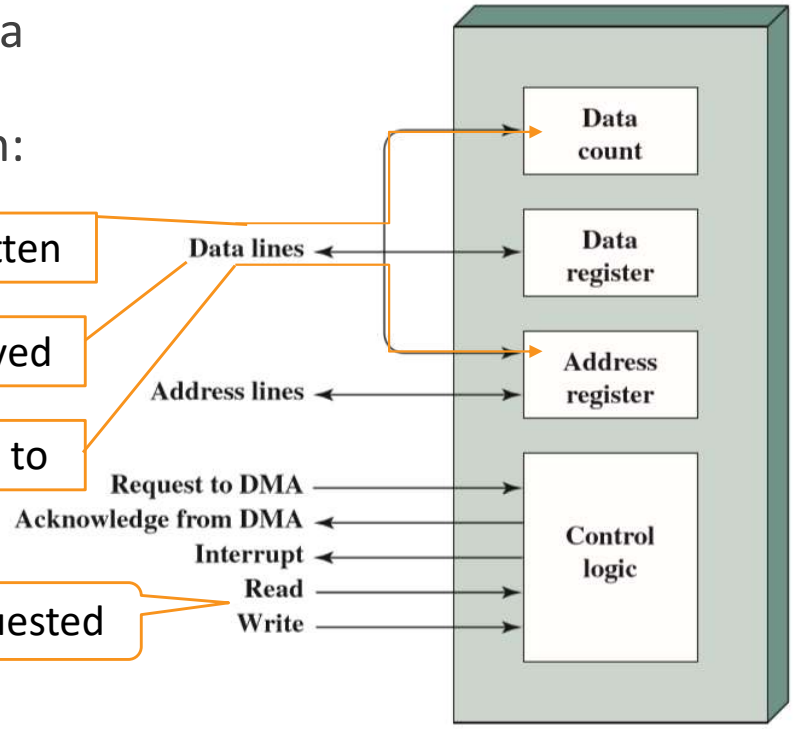
- When the processor wishes to read or write a block of data, it issues a command to the DMA module, with the following information:

The number of words to be read or written

The address of the I/O device involved

The starting location in memory to read from or write to

Whether a read or write is requested

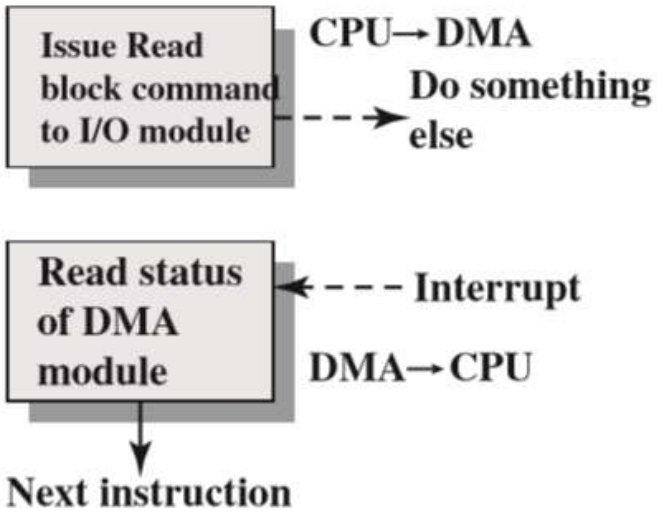


Typical DMA Block Diagram

50

Direct Memory Access (DMA)

- I/O operation delegated to the DMA module, CPU continues with other work.
- The DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through the processor.
- When the transfer is complete, the DMA module sends an interrupt signal to the processor.
- Thus, the processor is involved only at the beginning and end of the transfer

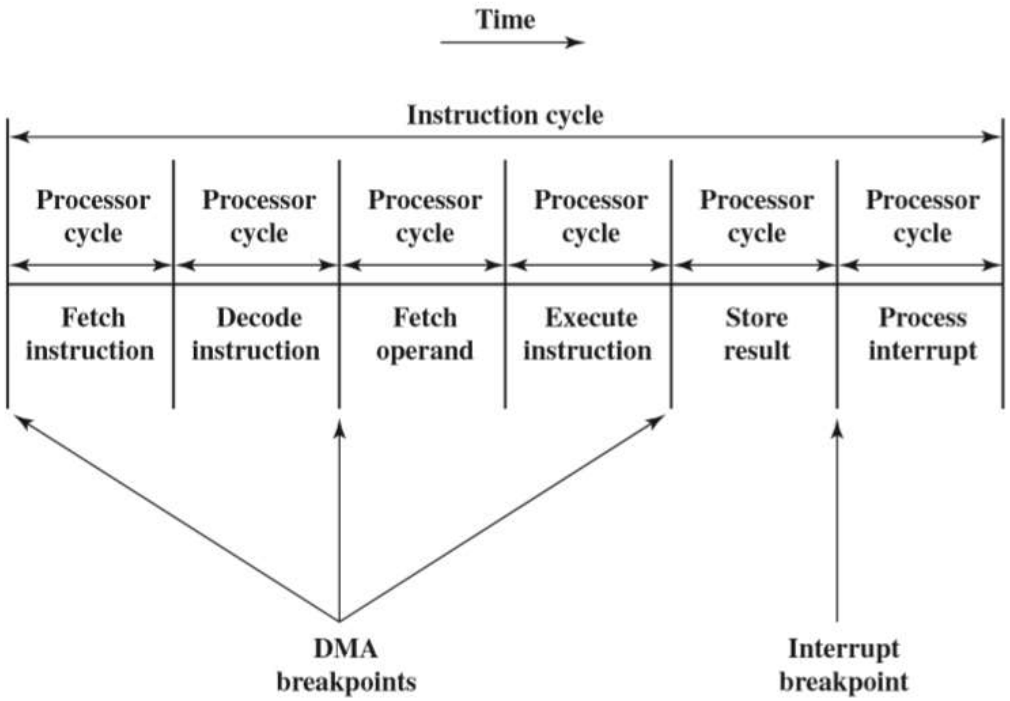


(c) Direct memory access

51

Direct Memory Access (DMA)

- DMA module must use the bus only when the processor does not need it, or it must force the processor to suspend operation temporarily (**Cycle Stealing**).
- Processor pauses for one bus cycle → execute more slowly, but still far more efficient than interrupt-driven or programmed I/O for a multiple-word I/O transfer.



DMA and Interrupt Breakpoints during an Instruction Cycle

52

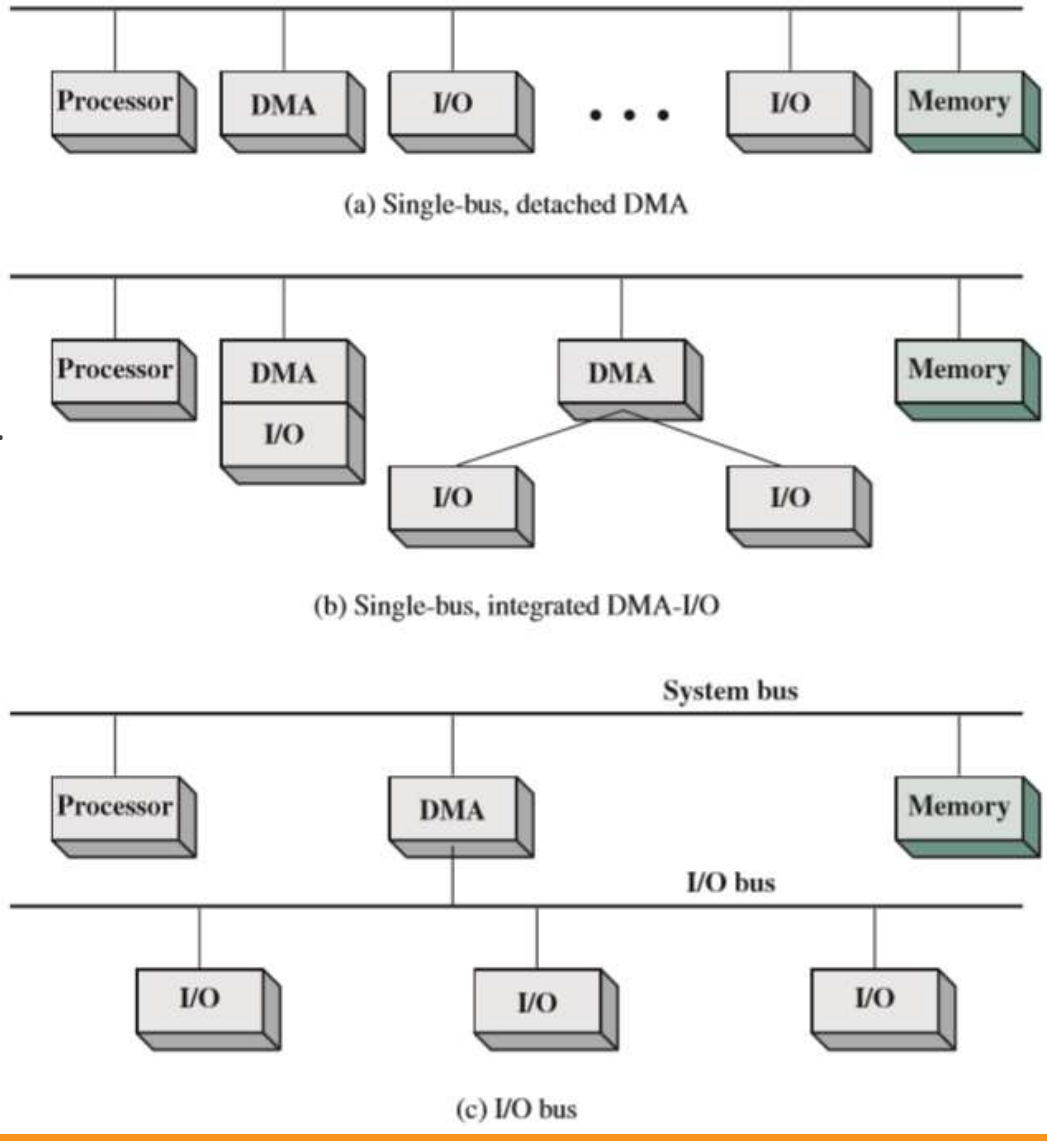
Alternative DMA Configurations

a) **Single-bus, detached DMA**:
- Inexpensive,
- Inefficient: transferring a word needs 2 bus cycles.

b) **Single-bus, integrated DMA-I/O**:
- Expensive with multiple DMA modules.

c) **I/O bus:**
- Easily expandable

- In (b) and (c), system bus is used by the DMA module only to exchange data with memory.



Direct Cache Access

- DMA is not able to scale to meet the increased demand for network I/O (e.g. 10-Gbps. 100-Gbps Ethernet switches, Gbps-WiFi, etc.)
- **Direct Cache Access (DCA)**: performance enhanced by enabling the I/O function to have direct access to the **last-level cache**.

54

External I/O interface

- Some widely used external interface standards:

External interface standards	Used on	Data rate
Universal Serial Bus (USB)	Keyboard, printers, disk drives, network adapters, etc.	USB 1.0: 1.5 Mbps - 12 Mbps USB 2.0: 480 Mbps USB 3.0: 4 Gbps USB 3.1: 9.7 Gbps
FireWire Serial Bus	Data storage devices, digital audio/video equipment	25 Mbps - 3.2 Gbps
Small Computer System Interface (SCSI)	Disks, modems, printers, etc. SCSI is a shared bus, can support up to 16 or 32 devices.	SCSI-1: 5 Mbps SCSI-3 U3: 160 Mbps
Thunderbolt	Combines data, video, audio, and power into a single high-speed connection for peripherals like hard drives, RAID, video-capture boxes, and network interfaces	Up to 10 Gbps

55

External I/O interface

- Some widely used external interface standards:

External interface standards	Used on	Data rate
InfiniBand	Can connect up to 64,000 servers, storage systems, and networking devices.	2 - 3000 Gbps
PCI Express	Motherboard interface for personal computers' graphics cards, hard drives, SSDs, Wi-Fi and Ethernet hardware connections.	Ver. 5: 32 - 500 Gbps
SATA	Disk storage	Up to 6 Gbps
Ethernet	Wired networking	Up to 100 Gbps
Wi-Fi	Wireless Internet connection	Wi-Fi 6, or 802.11ax (2020): up to 10 Gbps

56

Check your understanding

12. The method which offers higher speeds of I/O transfers is

A. Memory mapping
B. Programmed I/O
C. Interrupts
D. DMA

57

Check your understanding (Answer)

12. The method which offers higher speeds of I/O transfers is

A. Memory mapping
B. Programmed I/O
C. Interrupts
D. DMA

58

Check your understanding

13. When a process requests for a DMA transfer,

A. the process is temporarily suspended.
B. the process continues execution.
C. another process gets executed.
D. the process is temporarily suspended and another process gets executed.

59

Check your understanding (Answer)

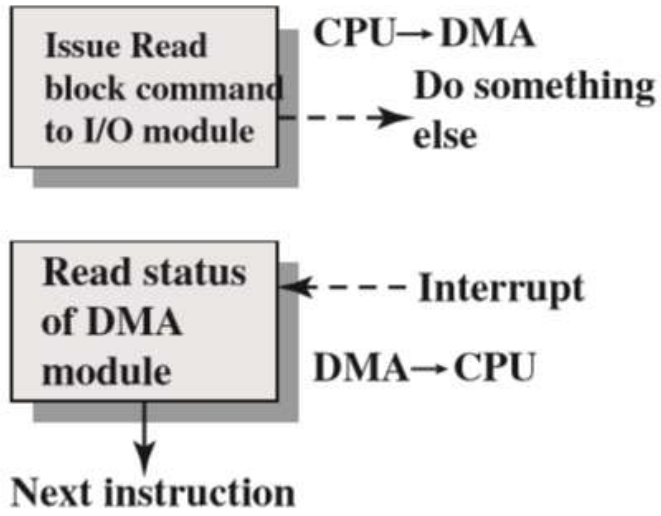
13. When a process requests for a DMA transfer,

A. the process is temporarily suspended.

B. the process continues execution.

C. another process gets executed.

D. the process is temporarily suspended and another process gets executed.



(c) Direct memory access

60

Check your understanding

14. After the completion of the DMA transfer, the processor is notified by

A. Acknowledge signal
B. Interrupt signal
C. DMA signal
D. None of the mentioned

61

Check your understanding (Answer)

14. After the completion of the DMA transfer, the processor is notified by

A. Acknowledge signal

B. Interrupt signal

C. DMA signal

D. None of the mentioned

62

After-class exercises (1)

- Consider a system employing interrupt-driven I/O for a particular device that transfers data at an average of 8 KB/s on a continuous basis.

a) Assume that interrupt processing takes about 100 µs (i.e., the time to jump to the interrupt service routine (ISR), execute it, and return to the main program). Determine what fraction of processor time is consumed by this I/O device if it interrupts for every byte.

b) Now assume that the device has two 16-byte buffers and interrupts the processor when one of the buffers is full. Naturally, interrupt processing takes longer, because the ISR must transfer 16 bytes. While executing the ISR, the processor takes about 8 µs for the transfer of each byte. Determine what fraction of processor time is consumed by this I/O device in this case.

c) Now assume that the processor is equipped with a block transfer I/O instruction such as that found on the Z8000. This permits the associated ISR to transfer each byte of a block in only 2 µs. Determine what fraction of processor time is consumed by this I/O device in this case.

63

After-class exercises (2)

- For the Intel 8237A DMA controller, once a block transfer begins, it takes three bus clock cycles per DMA cycle. During the DMA cycle, the 8237A transfers one byte of information between memory and I/O device.

a) Suppose we clock the 8237A at a rate of 5 MHz. How long does it take to transfer one byte?

b) What would be the maximum attainable data transfer rate?

c) Assume that the memory is not fast enough and we have to insert two wait states per DMA cycle. What will be the actual data transfer rate?

64

After-class exercises (3)

- A computer consists of a processor and an I/O device D connected to main memory M via a shared bus with a data bus width of one word. The processor can execute a maximum of 10^6 instructions per second. An average instruction requires five machine cycles, three of which use the memory bus. A memory read or write operation uses one machine cycle. Suppose that the processor is continuously executing "background" programs that require 95% of its instruction execution rate but not any I/O instructions. Assume that one processor cycle equals one bus cycle. Now suppose the I/O device is to be used to transfer very large blocks of data between M and D.

a) If programmed I/O is used and each one-word I/O transfer requires the processor to execute two instructions, estimate the maximum I/O data-transfer rate, in words per second, possible through D.

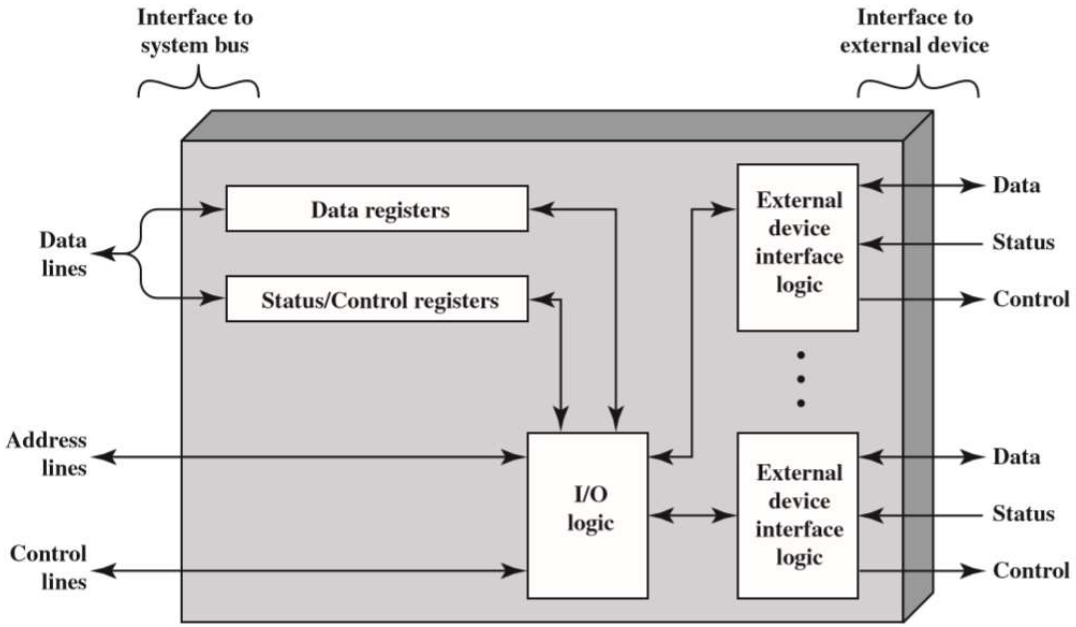
b) Estimate the same rate if DMA is used.

65

Summary

- I/O Module Function
  - Control and timing
  - Processor communication
  - Device communication
  - Data buffering
  - Error detection

- I/O Module Structure



Block Diagram of an I/O Module

66

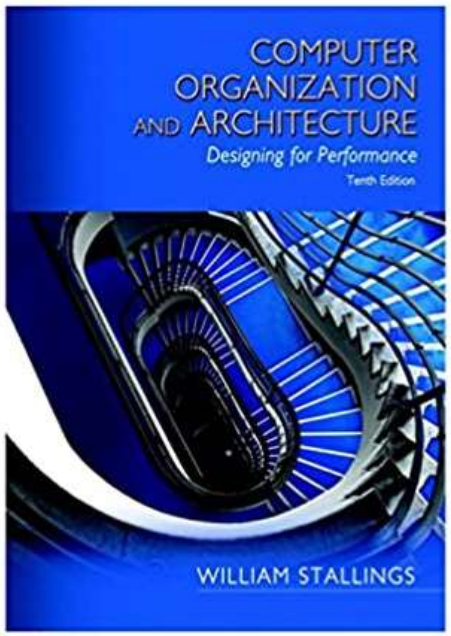
Summary

- Internal I/O Interfacing:
  Memory-mapped I/O vs Isolated I/O
- I/O Techniques
  - Programmed I/O
  - Interrupt-driven I/O
  - Direct Memory Access (DMA)
  - Direct Cache Access
- External I/O Interface:
  - USB
  - FireWire Serial Bus
  - SCSI
  - Thunderbolt
  - InfiniBand
  - PCI Express
  - SATA
  - Ethernet
  - Wi-Fi

67

References

- Computer Organization and Architecture - Designing for Performance, Tenth Edition, William Stallings, Pearson 2010.
  Chapter 7 Input/Output.



68