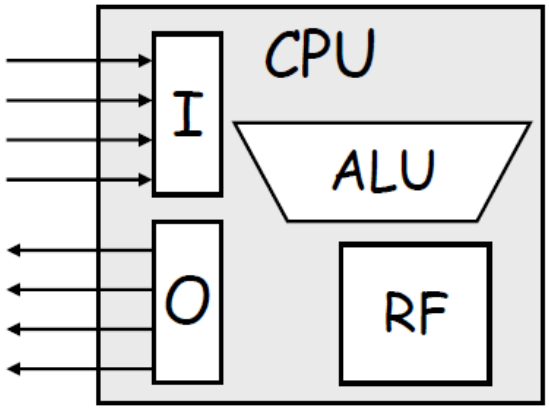
ساختار و زبان کامپیوتر

فصل هفتم

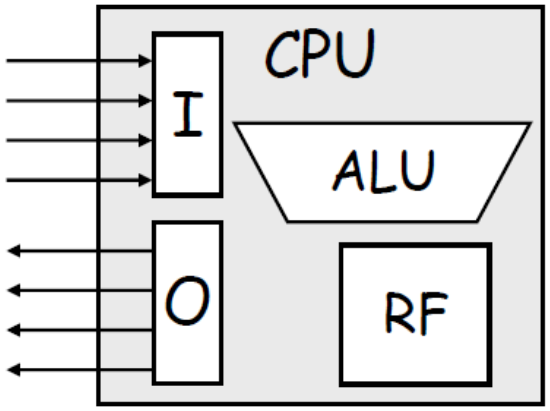
ورودی/خروجی و وقفه



Computer Structure and Machine Language

Chapter Seven

Input/Output & Interrupt



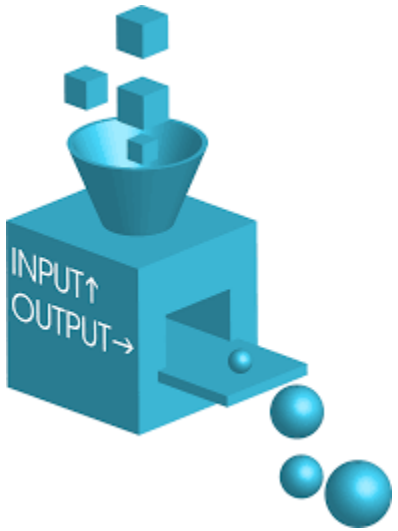
Copyright Notice

Parts (text & figures) of this lecture are adopted from :

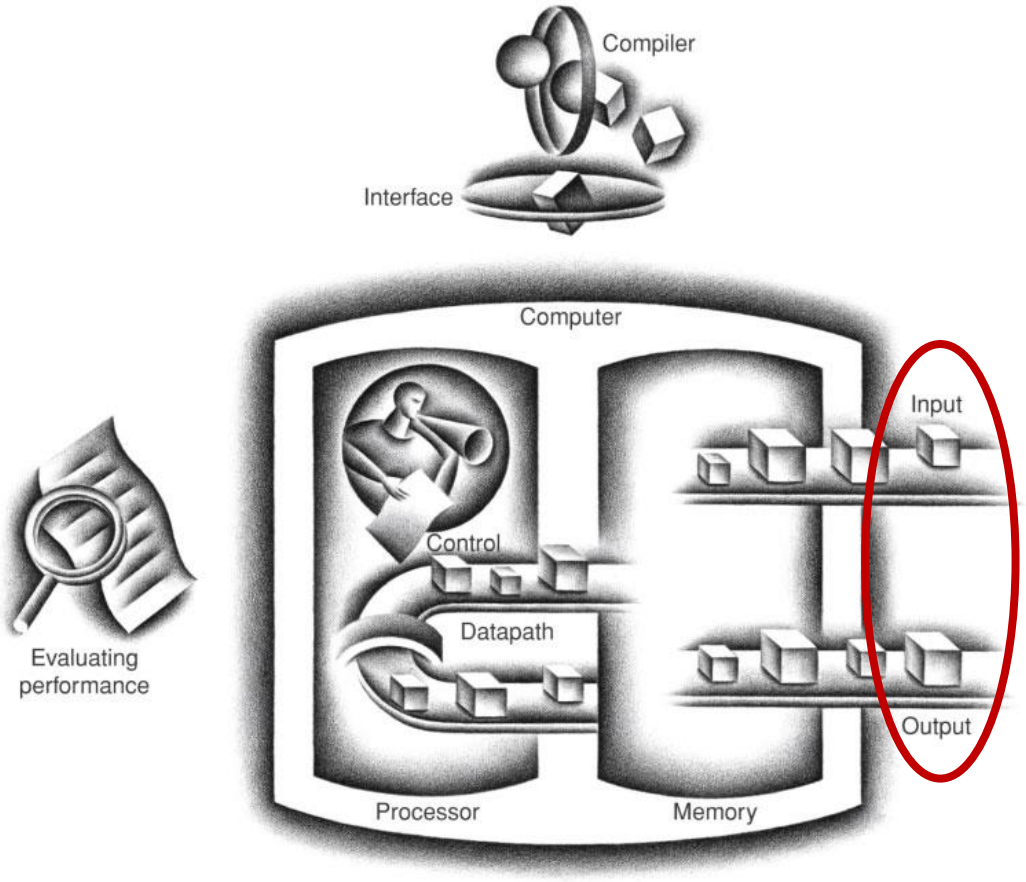
- M. Mano, C. Kime, T. Martin, "Logic and Computer Design Fundamentals", 5th Ed., Pearson, 2015
- D. Patterson & J. Hennessey, "Computer Organization & Design, The Hardware/Software Interface", 4th Ed., MK publishing, 2012
- W. Stallings, "Computer Organization and Architecture, Designing for Performance", 10th Ed., Pearson, 2016

Contents

- I/O Devices (Instances & Characteristics)
- I/O Interfaces
- Modes of Transfer
  - Interrupt vs. Programmed I/O
  - Direct Memory Access (DMA)
- Operating System's Role

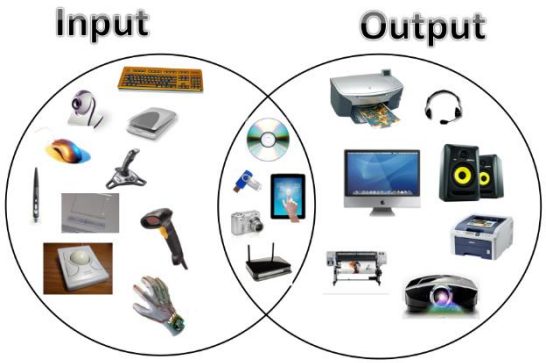


Input/Output Subsystem



I/O Devices

- Same components for all kinds of computers
  - Desktop, server, embedded
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers



I/O Devices Characteristics - 1

- Behavior:
  - Input (read only)
  - output (write only, cannot be read)
  - storage (can be reread & usually rewritten)

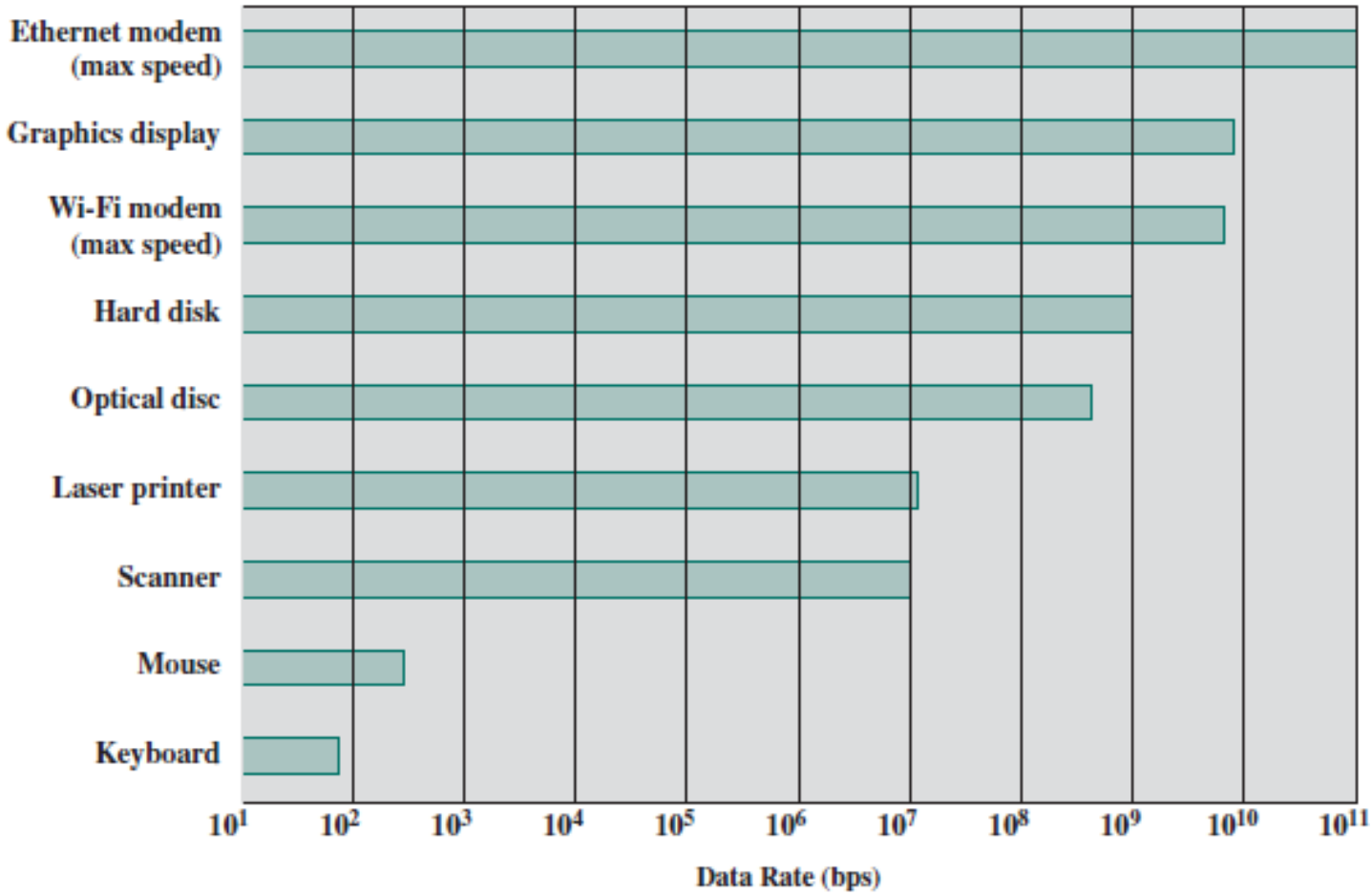
I/O Devices Characteristics - 2

- Partner:
  - A human or a machine is at the other end of the I/O device
    - feeding data on input or
    - reading data on output

I/O Devices Characteristics - 3

- Data rate:
  - The peak rate at which data can be transferred between the I/O device & memory or processor

Typical I/O Device Data Rates



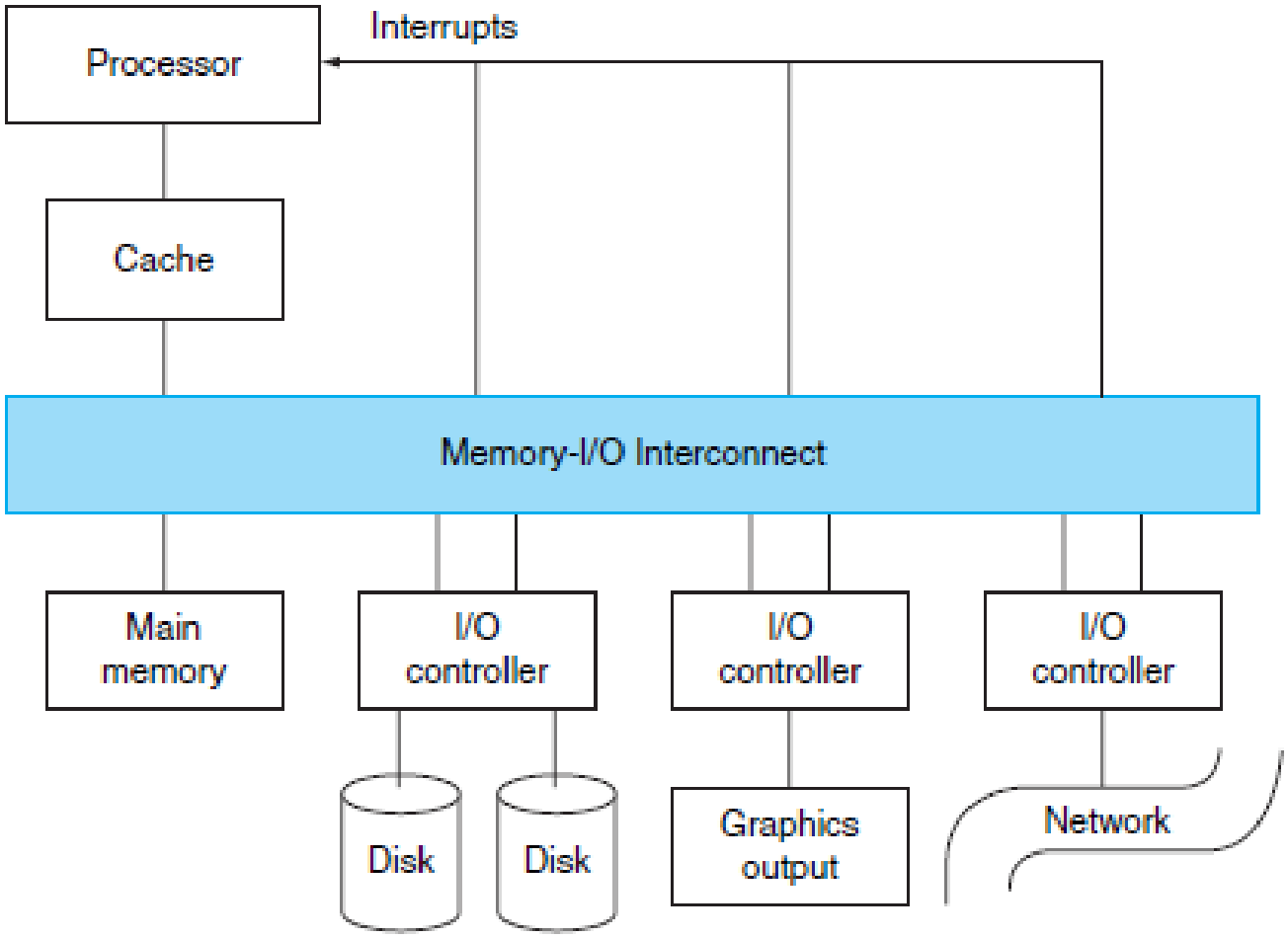
I/O Devices Characteristics

- Behavior (input, output, storage)
- Partner (a human or a machine)
- Data rate
- e.g. keyboard:
  - an input device
  - used by a human
  - with a peak data rate of about 10 bytes per second

The Diversity of I/O Devices

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	Input	Human	0.0001
Mouse	Input	Human	0.0038
Voice input	Input	Human	0.2640
Sound input	Input	Machine	3.0000
Scanner	Input	Human	3.2000
Voice output	Output	Human	0.2640
Sound output	Output	Human	8.0000
Laser printer	Output	Human	3.2000
Graphics display	Output	Human	800.0000–8000.0000
Cable modem	Input or output	Machine	0.1280–6.0000
Network/LAN	Input or output	Machine	100.0000–10000.0000
Network/wireless LAN	Input or output	Machine	11.0000–54.0000
Optical disk	Storage	Machine	80.0000–220.0000
Magnetic tape	Storage	Machine	5.0000–120.0000
Flash memory	Storage	Machine	32.0000–200.0000
Magnetic disk	Storage	Machine	800.0000–3000.0000

Typical Collection of I/O Devices

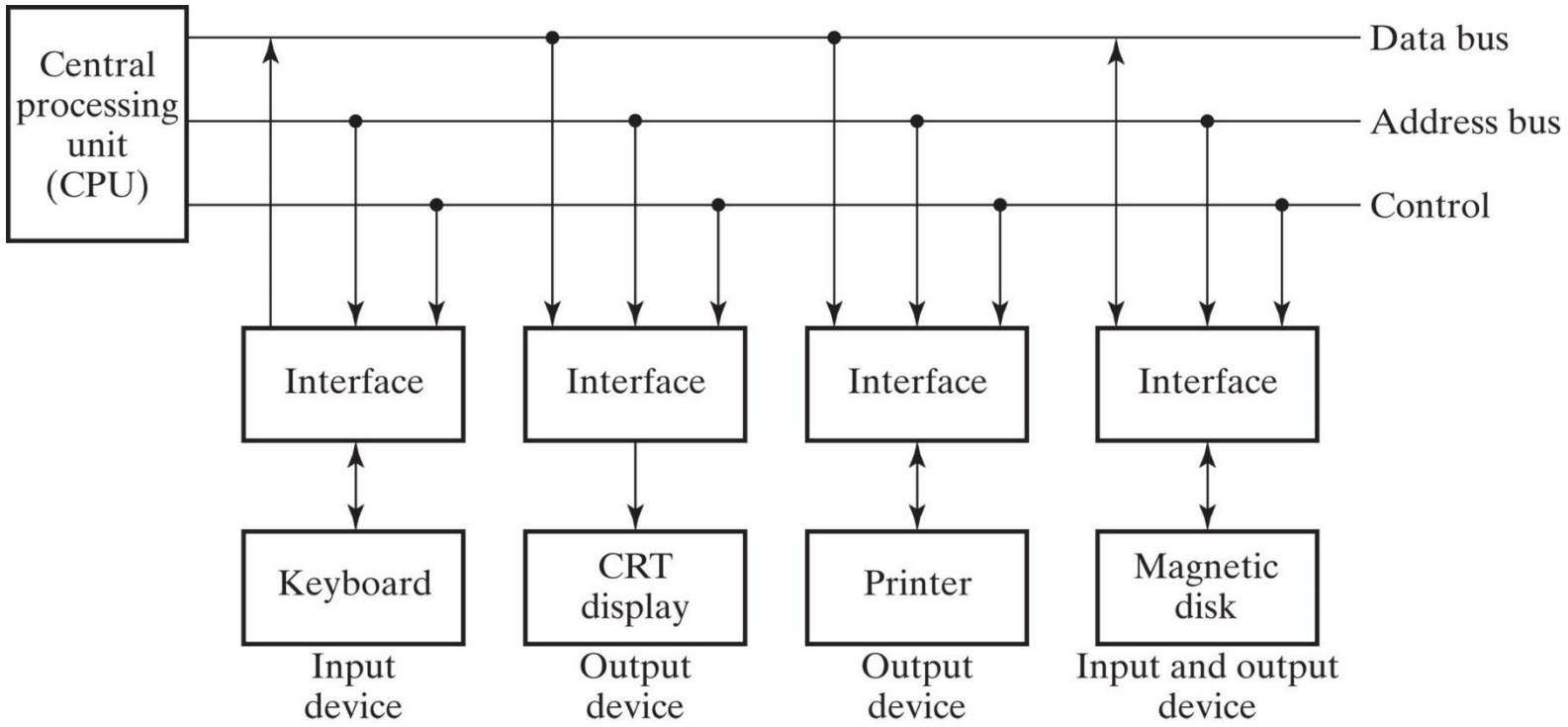


Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

I/O Interfaces

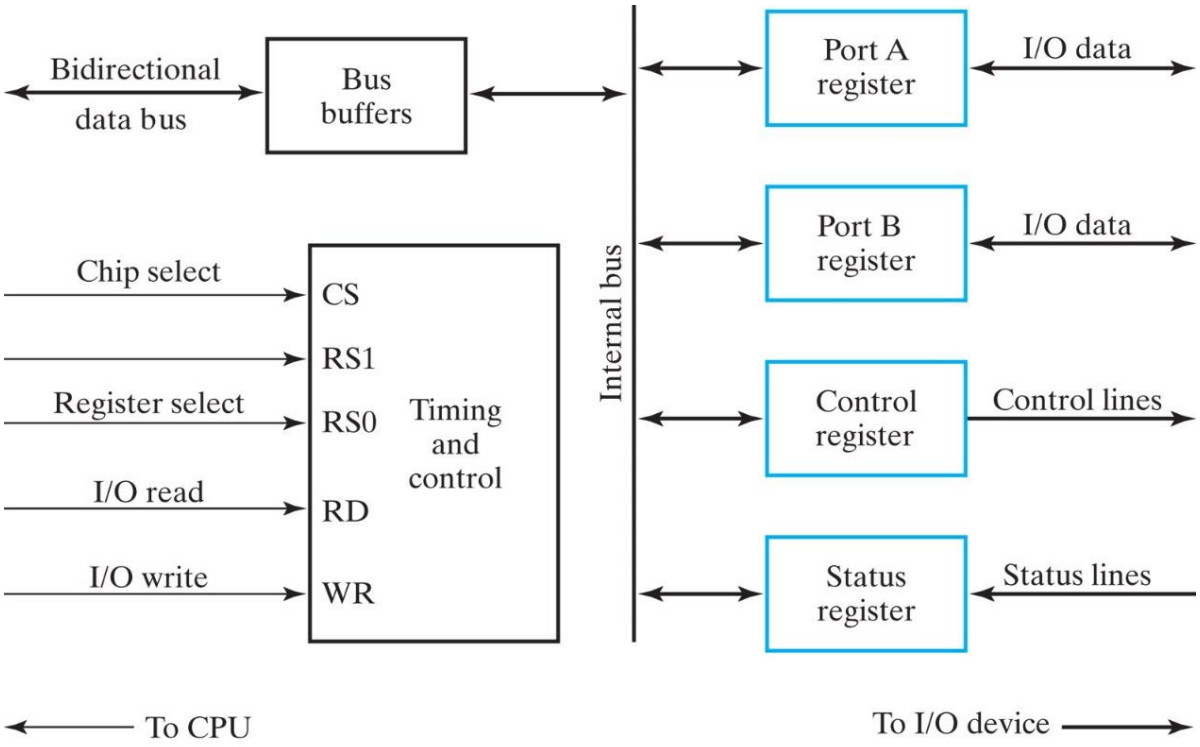
- Special hw components between CPU & peripherals
  - Peripherals are often electromechanical
    - a conversion of signals may be required
  - Data transfer rate of peripherals is usually different from CPU clock rate
    - a synchronization mechanism may be needed
  - Data codes and formats in peripherals differ from CPU and memory word format
  - Operating modes of peripherals differ from each other and each must be controlled independently

Figure 11-5: Connection of I/O Devices to CPU



Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

Figure 11-6: Example of I/O Interface Unit



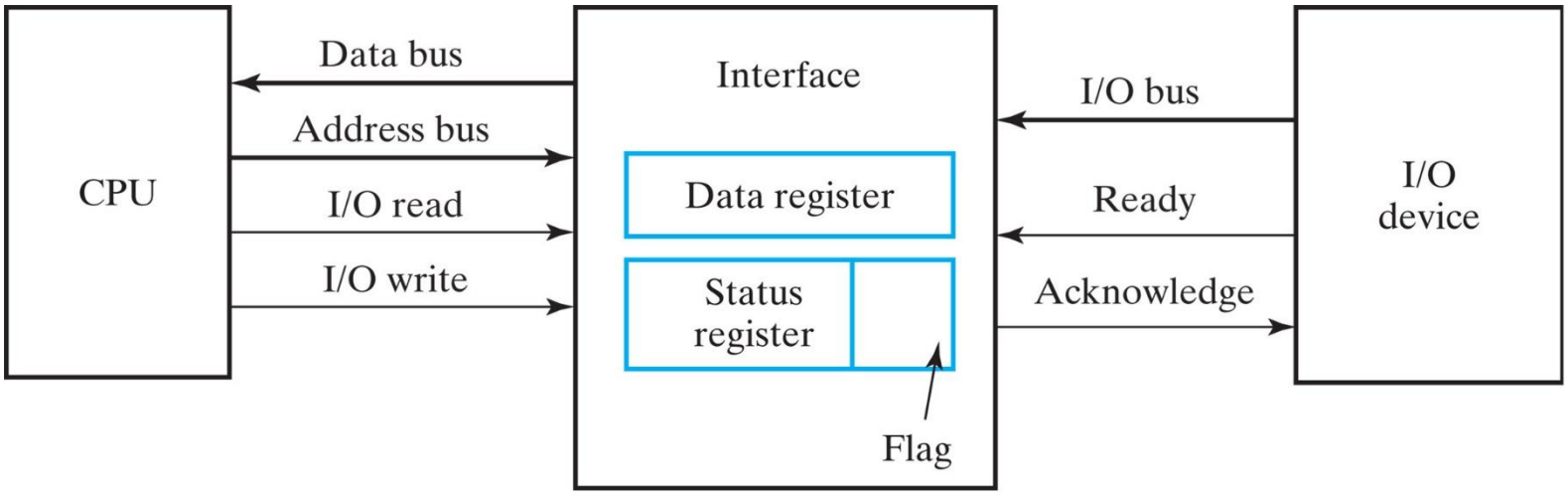
CS	RS1	RS0	Register selected
0	X	X	None: data bus in high-impedance state
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

Modes of Transfer

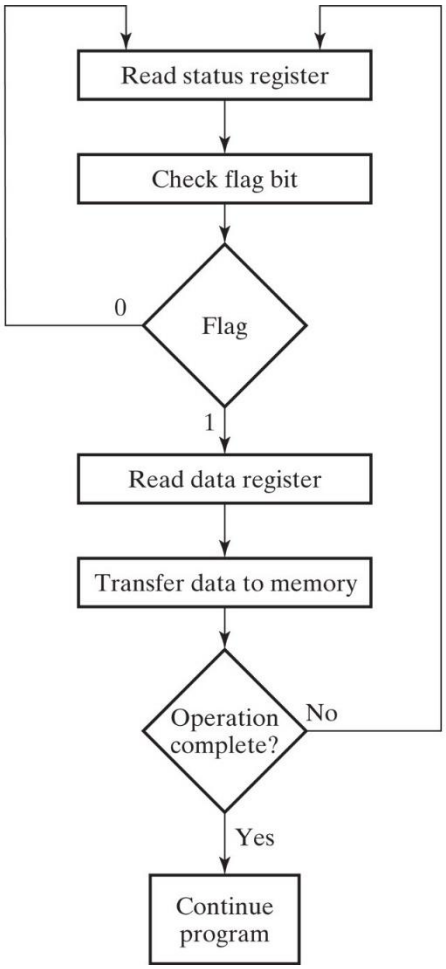
- Data transfer to and from peripherals may be handled in three modes:
  - data transfer under program control
  - interrupt-initiated data transfer
  - direct memory access transfer

Figure 11-13: Data Transfer from I/O Device to CPU



Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

Figure 11-14: Flowchart for CPU Program to Input Data



Pearson

Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

Program-Controlled Transfer

- Checks the status of I/O device periodically to determine the need to service
  - ☺ Simple
  - ☺ More predictable I/O overhead
  - ☹ can waste a lot of processor time
- Used in real-time applications because of predictability

Interrupt-Initiated Transfer

- A scheme that interrupts the processor to indicate that an I/O device needs attention
  - Overcomes CPU waiting
  - No repeated CPU checking of device
  - I/O device interrupts when ready
- I/O device
  - Raises the interrupt
  - Identifies itself

Program Interrupt

- Used to handle situations that require a departure from the normal program sequence
- Transfers control from a program that is currently running to another service program as a result of an externally or internally generated request
- Control returns to the original program after the service program is executed

Interrupt vs. Procedure Call

- Interrupt is usually initiated at an unpredictable point in the program by an external or internal signal, rather than the execution of an instruction
- Address of the service program that processes the interrupt request is determined by a hardware procedure, rather than the address field of an instruction
- In response to an interrupt, it is necessary to store information that defines all or part of the contents of the register set, rather than storing only the program counter

Processing Interrupts

- Very similar to the execution of a procedure call instruction
  - register set of the processor are temporarily stored in memory (pushed onto a memory stack)
  - address of the first instruction of the interrupt service program is loaded into the PC
- The address of the service program is chosen by the hardware

Service Program Address

- Some computers assign one memory location for the beginning address of the service program the service program must then determine the source of the interrupt and proceed to service it
- Other computers assign a separate memory location for each possible interrupt source
- Sometimes, the interrupt source hardware itself supplies the address of the service routine
- In any case, the computer must possess some form of hardware procedure for selecting a branch address for servicing the interrupt

Interrupt Cycle

- Just before going to fetch the next instruction, the control checks for any interrupt signal
- If an interrupt has occurred, control goes to a hardware interrupt cycle
  - Program Counter is pushed onto the stack
  - Branch address for the particular interrupt is transferred to PC
  - Control goes to fetch next instruction (beginning of the ISR)
- At the beginning of the service routine registers are pushed to the stack
- The last instruction in the service routine is a return from the interrupt instruction
  - the stack is popped to retrieve the return address and registers

Types of Interrupts

- External Interrupts
  - come from external sources
- Internal Interrupts
  - arise from the invalid or erroneous use of an instruction or data
- Software Interrupts
  - initiated by executing an instruction

Software Interrupts

- a special call instruction that behaves like an interrupt rather than a procedure call
- can be used by the programmer to initiate an interrupt procedure at any desired point in the program
- Typical use: System call

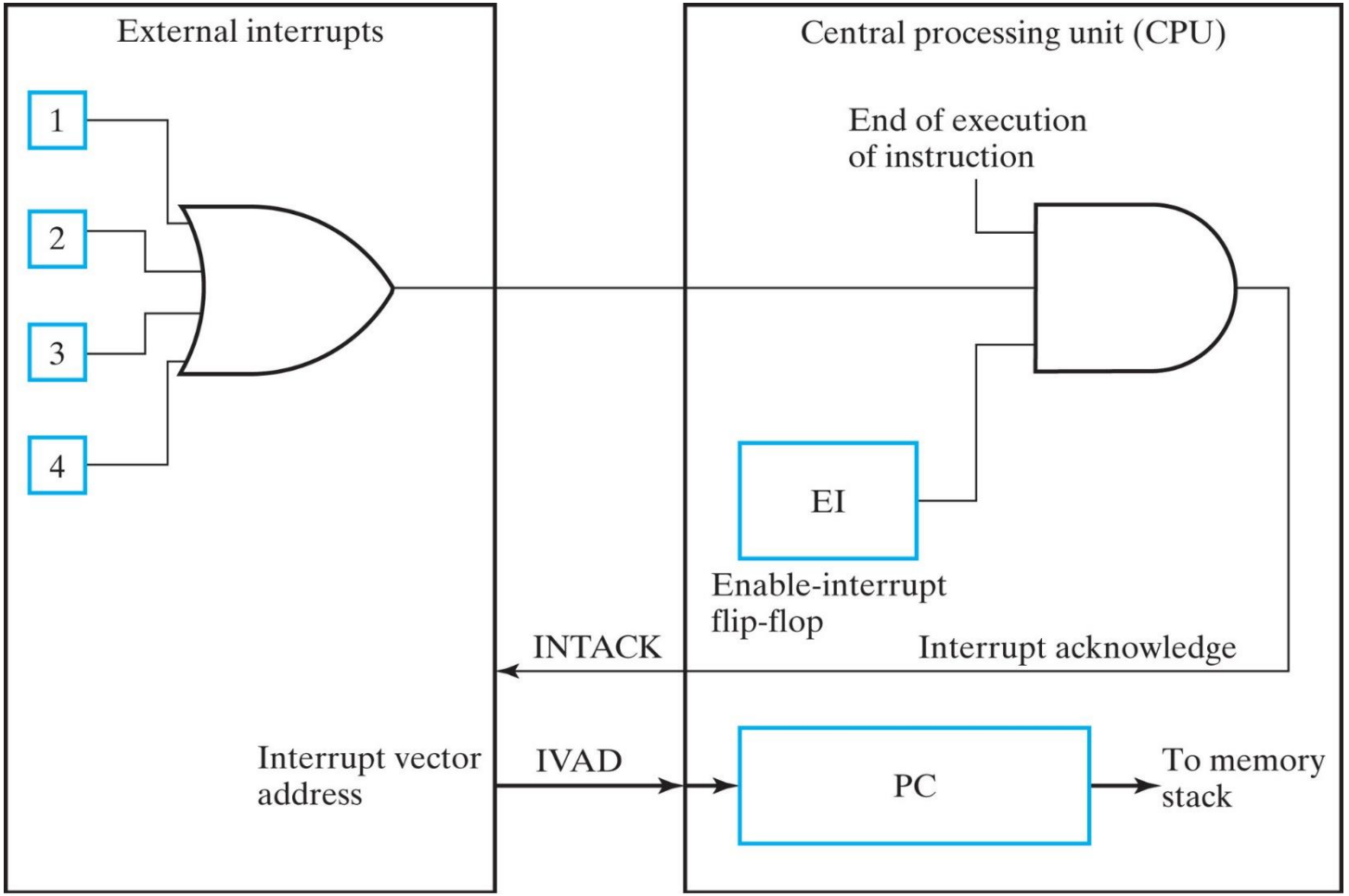
Internal Interrupts

- Also known as exceptions or traps
- Sources:
  - arithmetic overflow
  - attempt to divide by zero
  - invalid opcode
  - memory stack overflow
  - protection violation
- Corresponding service programs determine the corrective measure to be taken in each case

External Interrupts

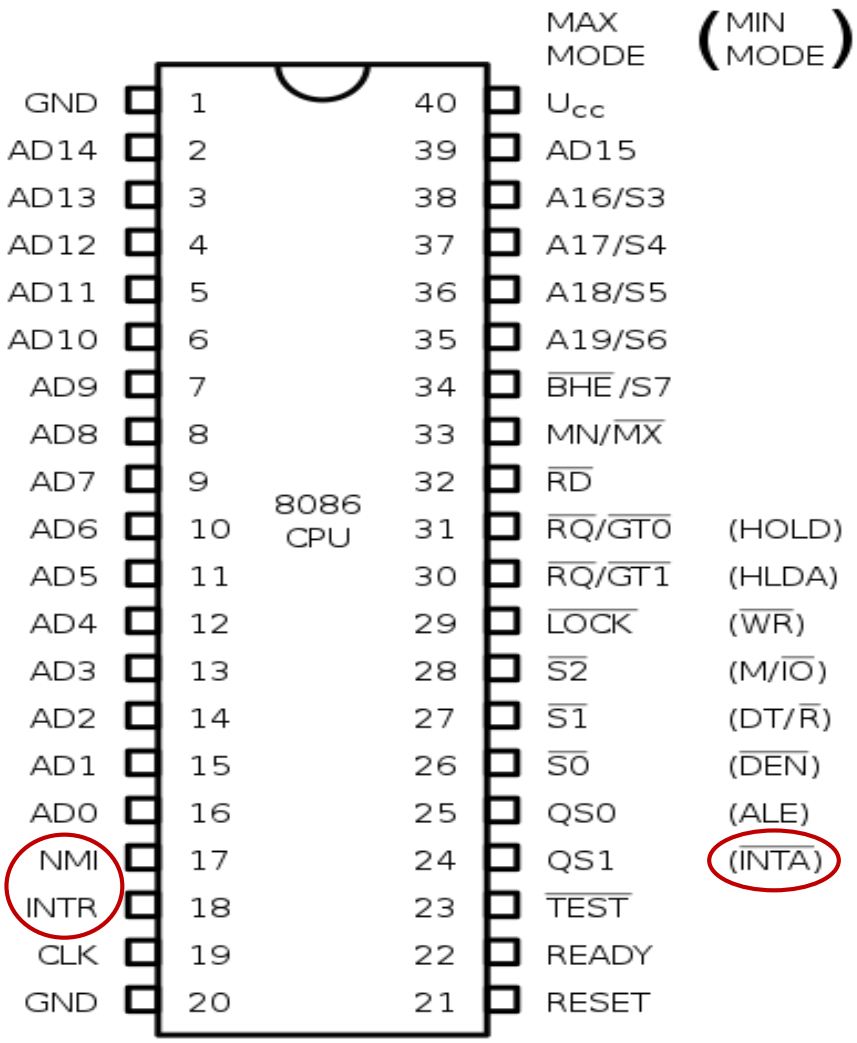
- Sources
  - I/O device
    - requesting transfer of data
    - completing transfer of data
  - Time-out of an event
  - Impending power failure
- may have single or multiple interrupt input lines

Figure 9-9: Example of External Interrupt Configuration



Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

8086 Chip


MAX MODE
(MIN MODE)
GND 1
AD14 2
AD13 3
AD12 4
AD11 5
AD10 6
AD9 7
AD8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
8086 CPU
40 Vcc
39 AD15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 BHE/S7
33 MN/MX
32 RD
31 RQ/GT0 (HOLD)
30 RQ/GT1 (HLDA)
29 LOCK (WR)
28 S2 (M/IO)
27 S1 (DT/R)
26 S0 (DEN)
25 QS0 (ALE)
24 QS1 (INTA)
23 TEST
22 READY
21 RESET

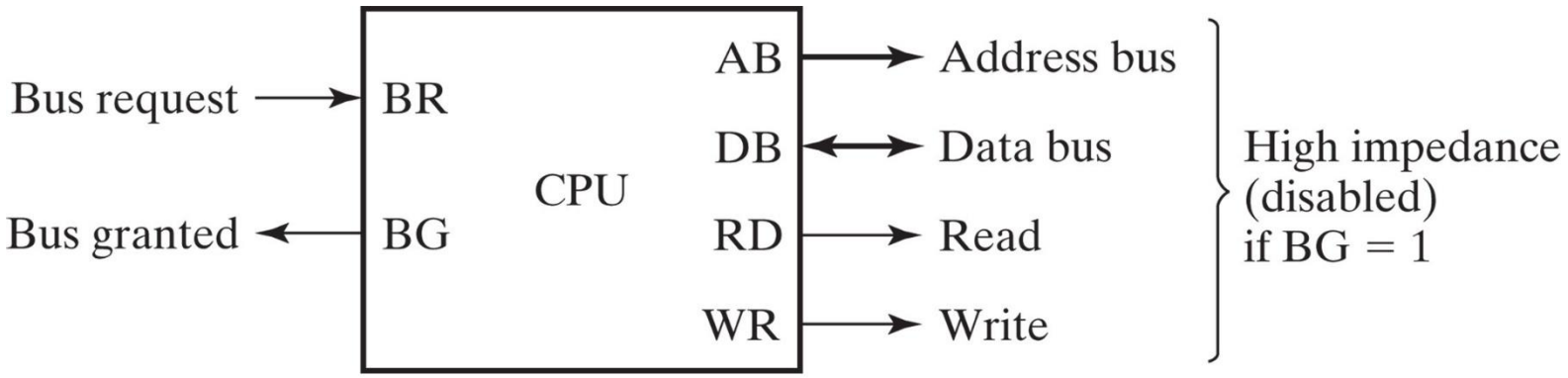
High-bandwidth I/O

- Programmed I/O & interrupt are suitable for low bandwidth devices
  - Data transfer is done in small no of bytes
- For higher bandwidth (e.g. hard disks)
  - Programmed I/O is used in real-time applications
  - Interrupt driven approach
    - Helps OS to work on other tasks while actual I/O transfer is done
    - But the overhead of each transfer is still intolerable
- Solution: Direct Memory Access (DMA)

Direct Memory Access (DMA)

- Provides a device controller with the ability to transfer data directly to or from the memory without involving the processor
- Implemented with a specialized controller that transfers data between an I/O device and memory independent of the processor

Figure 11-18: CPU Bus Control Signals

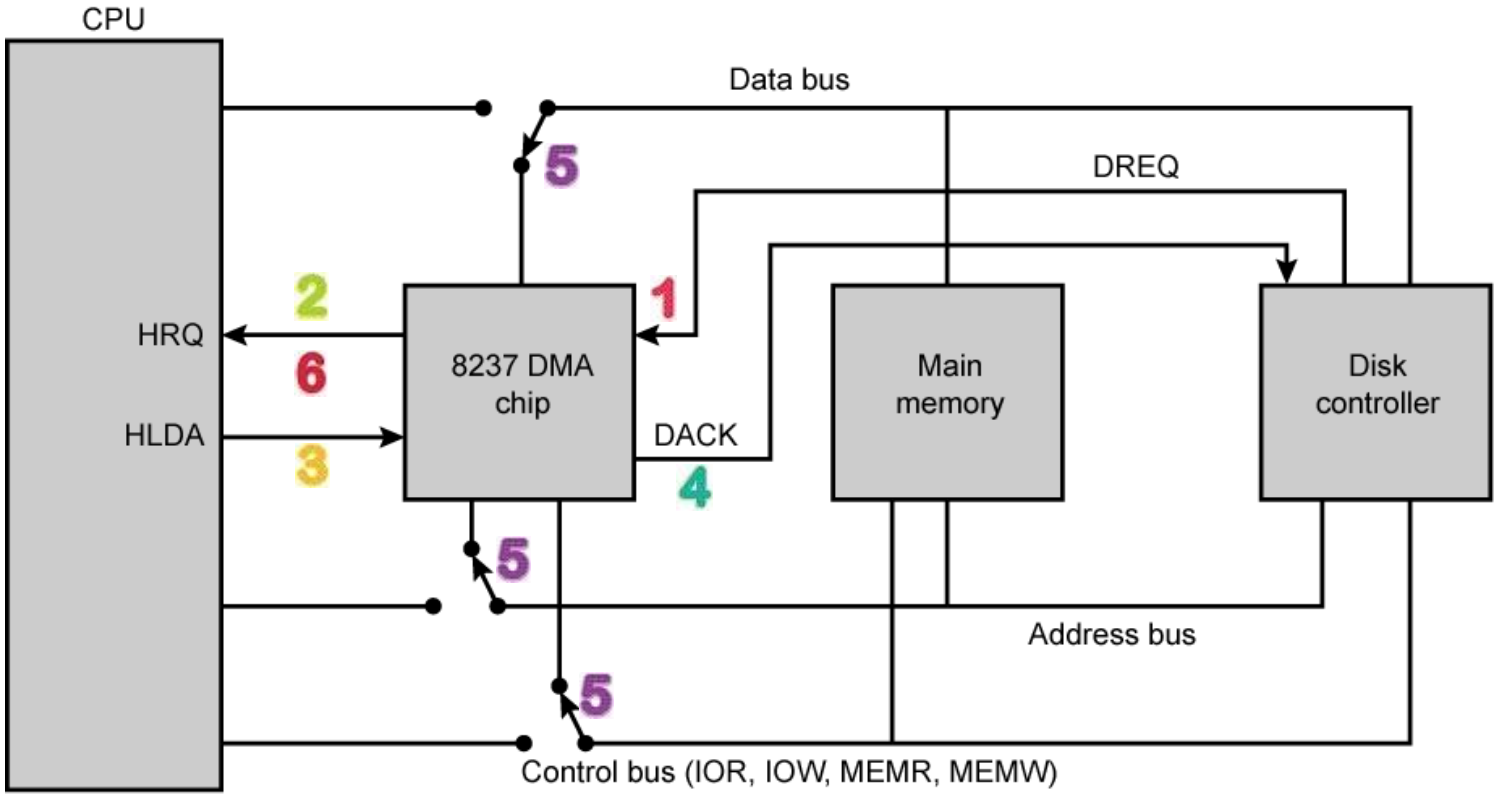


Pearson
Copyright © 2016, 2008, 2004 Pearson Education, Inc. All Rights Reserved

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 Usage of Systems Bus



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

Operating System

- Interface between a user's program and the hw
- Provides variety of services and supervisory functions
  - Handling basic input and output operations
  - Allocating storage and memory
  - Providing for protected sharing of the computer among multiple applications using it simultaneously
- e.g., Linux, MacOS, and Windows

Why Operating System?

- OS responsibilities arise from I/O characteristics:
  - Multiple programs using the processor share the I/O system
  - I/O systems often use interrupts to communicate information about I/O operations. Interrupts cause a transfer to kernel or supervisor mode, so they must be handled by the OS
  - The low-level control of an I/O device is complex:
    - requires managing a set of concurrent events
    - requirements for correct device control are detailed

Operating System's Role

- Guarantees that a user's program accesses only the portions of an I/O device to which the user has rights
  - e.g., must not allow a program to read or write a file on disk if the owner of the file has not granted access to this program
  - In a system with shared I/O devices, protection could not be provided if user programs could perform I/O directly
- ...

Operating System's Role (cont.)

- Guarantees that a user's program accesses only the portions of an I/O device to which the user has rights
- Provides abstractions for accessing devices by supplying routines that handle low-level device operations
- Handles the interrupts generated by I/O devices, just as it handles the exceptions generated by a program
- Tries to provide equitable access to the shared I/O resources and schedule accesses to enhance system throughput

OS Communication with I/O

- Give commands to the I/O devices:
  - read, write, disk seek, ...
- Be notified when the I/O device has completed an operation or has encountered an error
  - e.g., when a disk completes a seek, it will notify OS
- Transfer data between memory and I/O device
  - e.g., the block being read from a disk must be moved from disk to memory

Outlines

- I/O devices
- I/O interfaces
- Program Controlled Transfer
- Interrupt Initiated Transfer
- Direct Memory Access (DMA)
- Operating System's Role