I/O Management

All computers have physical devices for acquiring input and producing output.

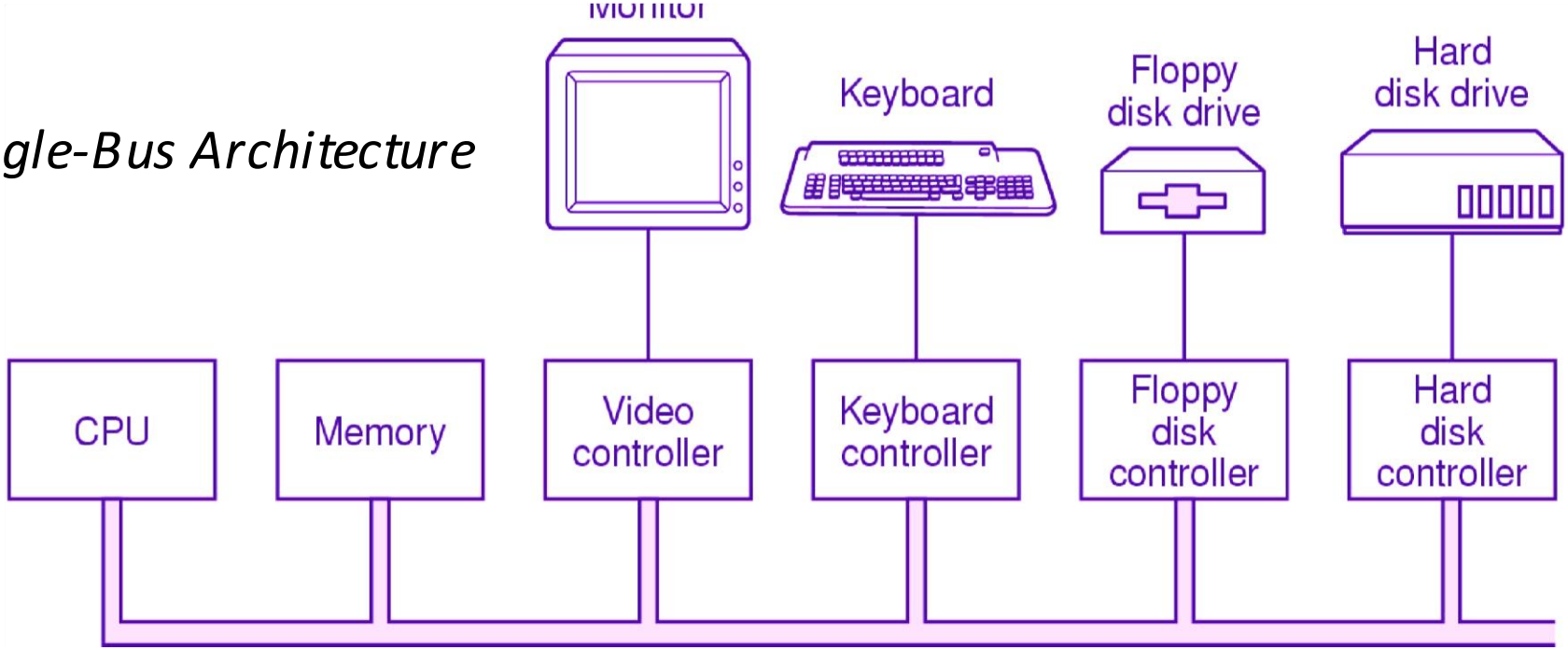
OS is responsible to manage and control all the I/OOperations and I/O devices.

Hardware Organization

Source: www.csitnepal.com

The I/O devices, memory, and the CPU communicate with each other by way of one or more communication buses.

Single-Bus Architecture



Hardware Organization

I/O Devices

The I/O units which consist mechanical components are called I/O devices such as hard-disk drive, printer etc.

There are two types of devices: *block* and *character* devices.

Block devices: Stores information in fixed-sized blocks, each one with its own address. Read or write is possible independent to other blocks-direct access.

Example: *disk*.

Character devices: Delivers or accepts a stream of characters without regard to any block structure. It is not addressable such as *printer*.

Some devices are neither block nor character such as clock.

Hardware Organization

Device Controllers

A controller is a collection of electronics that can operate a bus or a device.

On PC, it often takes the form of printed circuit card that can be inserted into an expansion slot.

A single controller can handle multiple devices; some devices have their own built-in controller.

The controller has one or more registers for data and signals. The processor communicates with the controller by reading and writing bit patterns in these registers.

When transferring a disk block of size 512 bytes, the block first assembled bit by bit in a buffer inside the controller. After its checksum has been verified and the block declared to be error free, it can than be copied to main memory.

Hardware Organization

Memory-Mapped I/O

Device controller have their own register and buffer for commnicating with the CPU, by writing and reading these register OS perform the I/O operation.

The device control registers are mapped into memory space, called memory-mapped I/O.

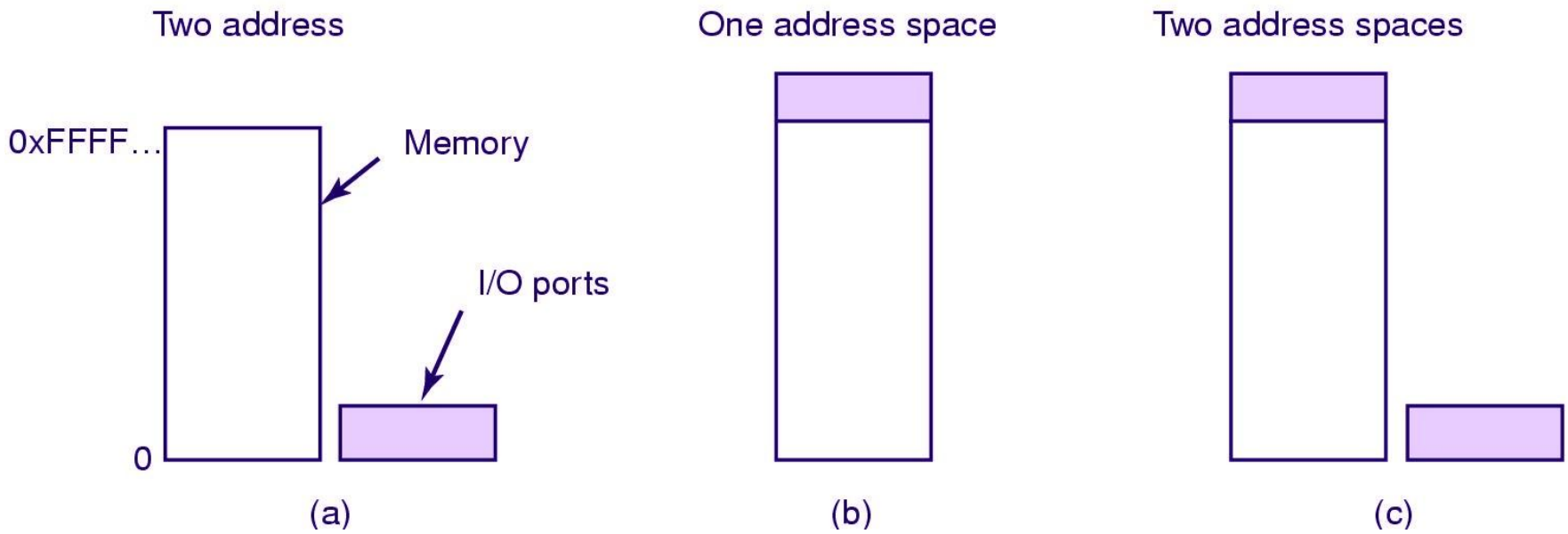
Usually, the assigned addresses are at the top of the address space. Some system (e.g. Pentium) use both techniques(hybrid).

The address 640K to 1M being reserved for device data buffers in addition to I/O ports 0 through 64K.

Hardware Organization

Memory-Mapped I/O

Source: www.csitnepal.com



a) separate I/O port and memory b) Memory-mapped I/O c) Hybrid.

When a CPU wants to read a word, either from memory or from I/O port, it puts the address needs on the bus' address line and then asserts the read signals on a bus' control line. A second signal is used to tell whether I/O or memory space is needed.

Hardware Organization

Memory-Mapped I/O

Advantages:

Can be implemented in high-level languages such as C.

No separate protection mechanism is needed.

Every instruction that can reference the memory can also reference the control registers (in hybrid).

Disadvantages:

Adds extra complexity to both hardware and OS.

All memory modules and all I/O devices must examine all memory references to see which one to respond to.

Hardware Organization Direct-Memory-Access (DMA)

What happens when CPU directly reads data one byte at a time from an I/O controller?

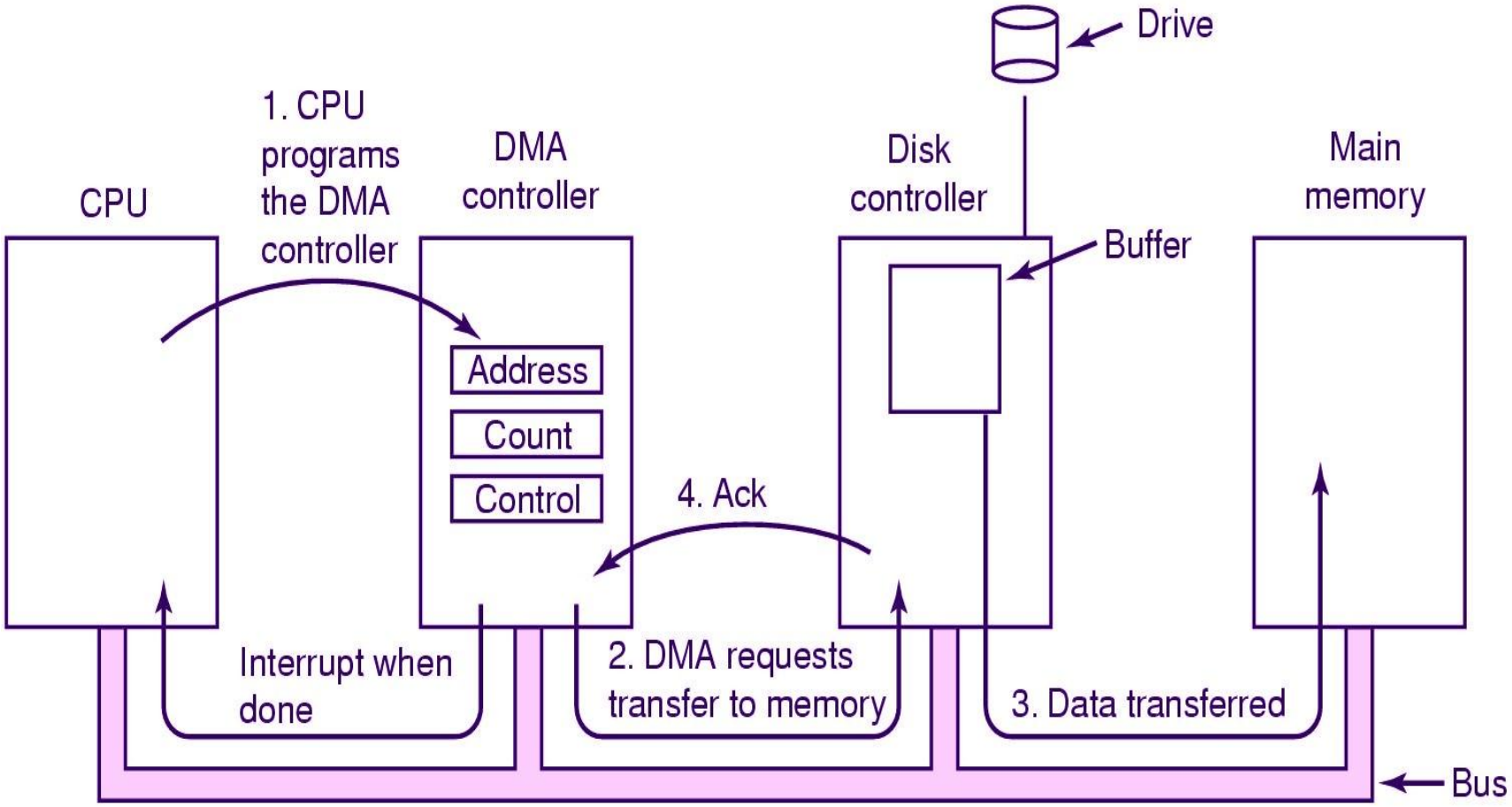
Wastes the CPU's time (busy wait) - The solution is DMA.

Many computers avoid burdening the CPU by offloading some of its work to special propose processor called DMA controller. It consists memory address register, a byte count register, and one or more control registers.
More commonly, a single DMA controller is available (in-built in main-board) for regulating transfer to multiple devices but some systems may have integrated with disk controllers.

Hardware Organization

Direct-Memory-Access (DMA)

Source: www.csitnepal.com



Steps in DMA Transfer

Hardware Organization

Direct-Memory-Access (DMA) How it works?

1. The CPU programs the DMA controller by its registers so it knows what to transfer where.

It also issue a command to disk controller telling it to read data from disk to its internal buffer and verify the checksum. When valid data are in disk's controller buffer, DMA can begin.

2. The DMA controller initiates the transfer by issuing a read request over the bus to disk controller.
3. Data transferred from disk controller to memory.
4. When transferred completed, the disk controller sends an acknowledgment signal to DMA controller. The DMA controller then increments the memory address to use and decrement the byte count. This continues until the byte count greater than 0.
5. When transfer completed the DMA controller interrupt the CPU.

I/O Handling

Polling

Processing is interrupted at brief intervals to allow the CPU to check back to I/O to see if the I/O operation has completed.

Well manner way of getting attention.

Advantages: Simple

Problems: May bear long wait.

Can be a high overhead operation-inefficient.

Used in embedded system, where CPU has nothing else to do.

I/O Handling

Interrupt

The hardware mechanism that enables a device to notify the CPU is called an interrupt.

Interrupt forced to stop CPU what it is doing and start doing something else.

Advantages: Improves efficiency.

Problem: *Because of the selfish nature, the first interrupt may not be served if the similar second happened before the time needed to serve the first.*

Example

Scenario: a cook is cooking something in modern microwave oven equipped kitchen.

CASE A: The cook may regularly peek through the oven's glass door and watch as roast cooks under cook and watch; this kind of regular monitoring is *polling*.

CASE B: The cook may set a timer to expire after as appropriate number of minutes; the buzzer sounding after this interval is an *interrupt*.

I/O Software Issues

Device Independence.

Uniform Naming.

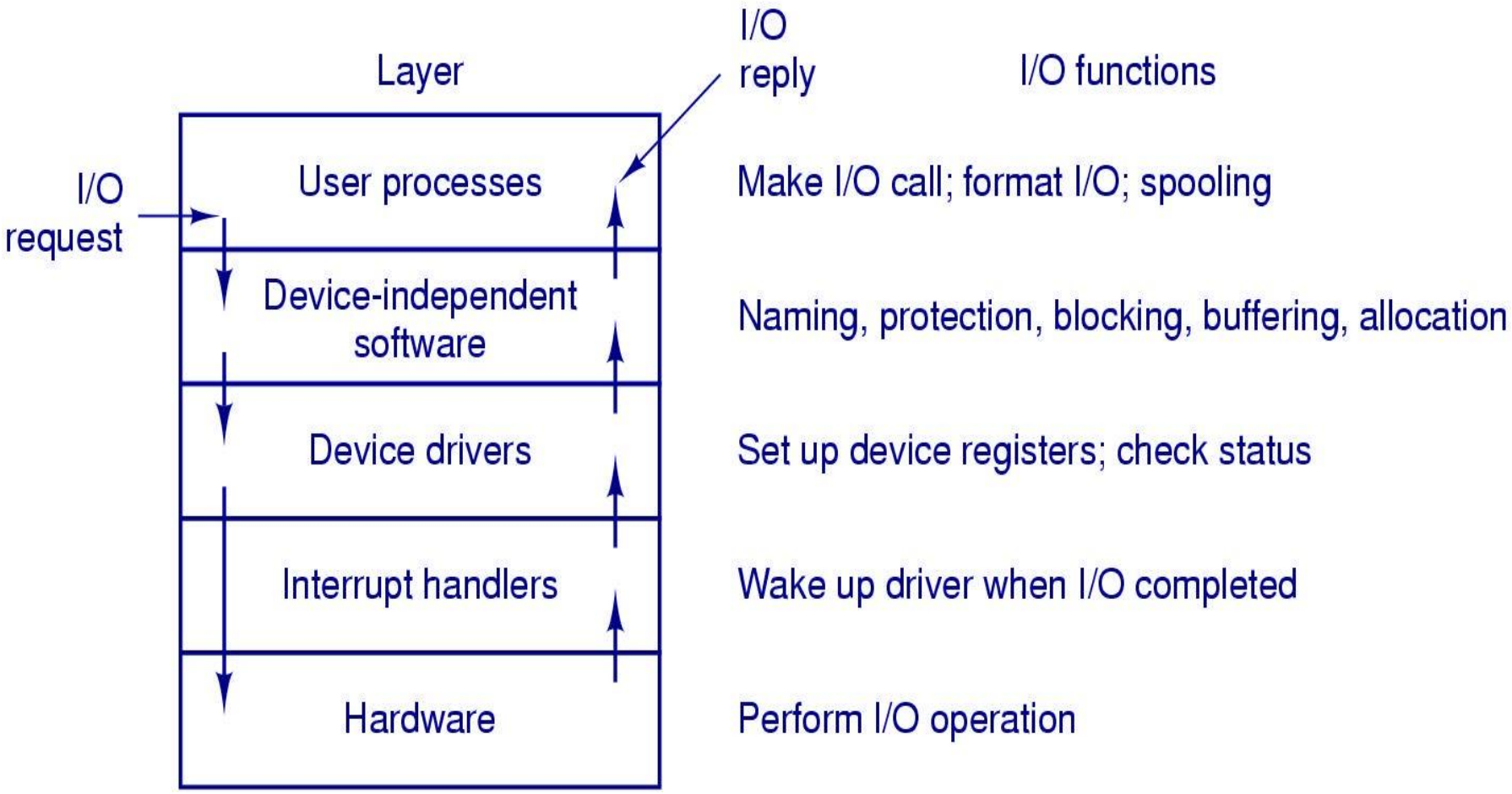
Error Handling.

Synchronous vs. Asynchronous transfer.

Buffering.

Layer Structure

Source: www.csitnepal.com



Interrupt Handlers

Block the driver until the I/O has completed and the interrupt occur.

The interrupt-handler determines the cause of the interrupt and performs the necessary processing.

1. Save any registers (including the PSW) that have not already been saved by the interrupt hardware.
2. Set up a stack for interrupt service procedure.
3. Ack interrupt controller.
4. Copy registers from where they were saved (stack) to the process table.
5. Run the interrupt service procedure.
6. Set up the MMU context for the process to run next.
7. Load new process' registers, including PSW.
8. Start running the new process.

Devices Drivers

Encapsulate detail of devices.

Each I/O device attached to a computer needs some device-specific code for controlling it, called device driver, is generally written by the device's manufacturer and delivered along with the device.

Each device driver normally handles one device type, or at most one class of closely related devices.

In some systems, the OS is a single binary program that contains all of the drivers that it will need compiled into it (e.g., UNIX).

Devices Drivers

Functions:

Accept read and write requests from the deviceindependent software above it.
Initialize the devices if necessary.
Manage power requirement and log events.
It checks the status of devices- in use or free. Decides which command to issue if there is command queue.

Device-Independent OS software

The devices independent-software is to perform the I/O functions that are common to all devices and

to provide a uniform interface to the user-level-software.

Some common functions are:

Uniform Interfacing for devices drivers-naming and protection.
Buffering.
Error reporting.
Allocating and releasing dedicated devices.
Providing a device-independent block size.

User Processes

System calls, including the I/O system calls are normally made by the user processes. User

processes put their parameters in the appropriate place for the system calls, or other procedures that actually do real work.

Home Works

HW #12

1. 4, 5, 8, 9, 10 & 13 from Textbook (Tanenbaum) Ch. 5.
2. How does DMA increase system concurrency? How does it complicate the hardware design?
3. Which one suited, polling/interrupt, for the following types of system? Give reason.
   a) A system dedicated to controlling a single I/O devices.

b) A personal computer running a single-tasking OS.
c) A work station running as heavily used web server.