Chapter 2

Computer Organisation

Lesson 3

Memory Unit

Objective —

- **Understand ROM, RAM, Model of memories**
- **Store and load operations**
- **Data alignment and Multibyte loads and stores**

Outline

- **Memory System**
- ROM
- RAM
- Address and Random Access Model
- Data alignment and Multibyte loads and store

The Memory System

- Acts as a storage receptacle for the data and programs used by the computer.

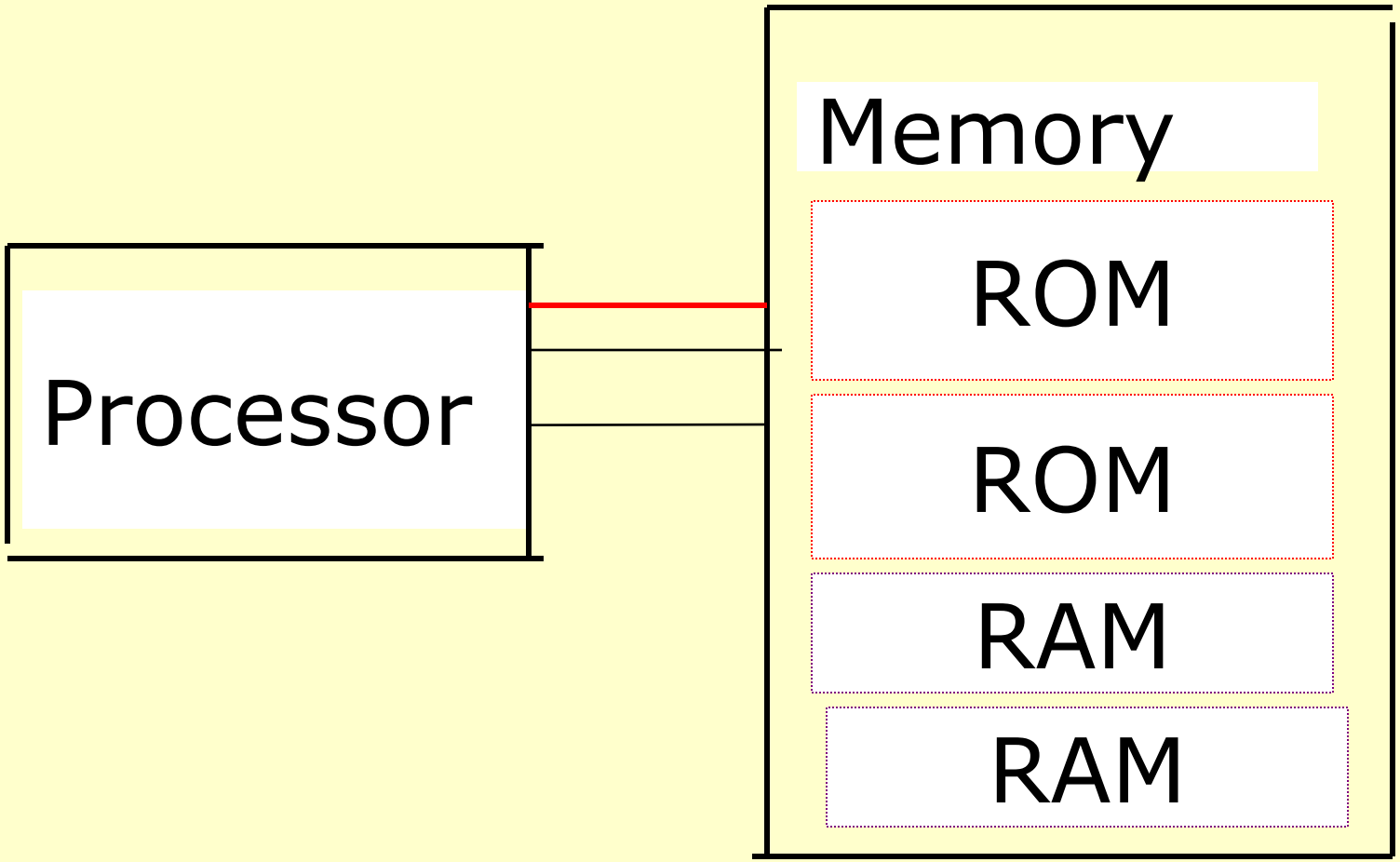
Two types of memory

- Most computers have:

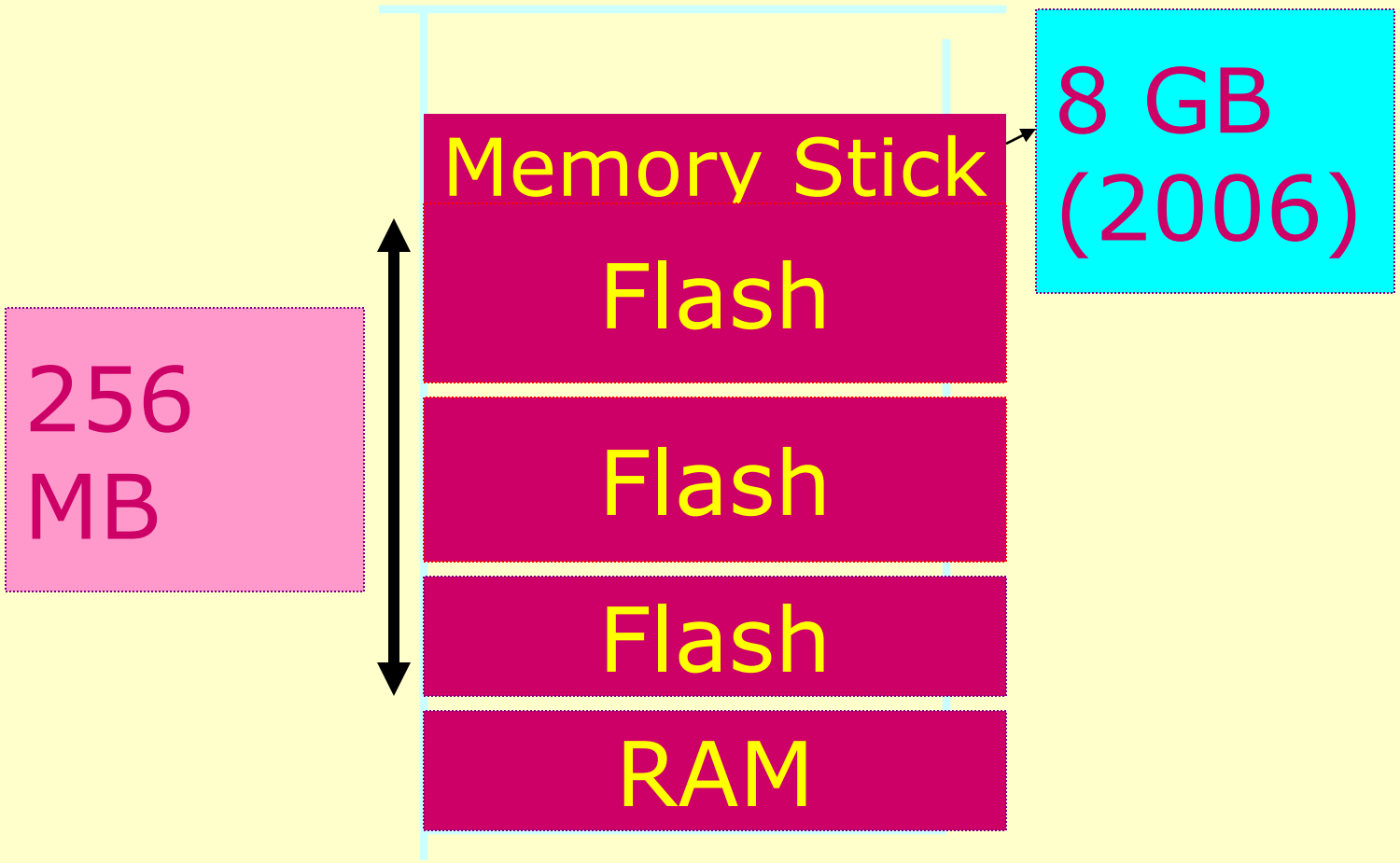
— *read-only memory* (ROM) and

— *random-access memory* (RAM).

Memory System Units



Mobile Computer Memory System Units Flash ROM



Outline

- Memory System
- **ROM**
- RAM
- Address and Random Access Model
- Data alignment and Multibyte loads and store

Contents of the read-only memory

- Cannot be modified by the computer but may be read.
- Nonvolatile (mo effect of power-switch off)

Contents of the read-only memory

- ROM is used to hold a program that is executed automatically by the computer every time it is turned on or reset. This program is called the bootstrap.

Bootstrap, or "boot" loader program

- Instructs the computer to load its operating system off of its hard disk or other I/O device.

Bootstrap Program

- The computer "pulls itself up by its own bootstraps" by executing a program that tells it how to load its own operating system.

An Example

- A Computer has ROM unit(s) for bootstrap program(s), basic input-output system (BIOS) program(s) and for vector addresses for the interrupts

Outline

- Memory System
- ROM
- **RAM**
- Address and Random Access Model
- Data alignment and Multibyte loads and store

Random-access memory

1. Can be both read and, written, and is used to hold the programs, operating system, and data required by the computer.
2. RAM generally volatile, Does not retain the data stored in it when the power switched off.

Random-access memory

3. Any data that needs to be stored while the computer is off must be written to a permanent storage device, such as a hard disk, flash, memory stick,..

Example

A desktop PC nowadays has 512MB of RAM to hold the programs, operating system, and data.

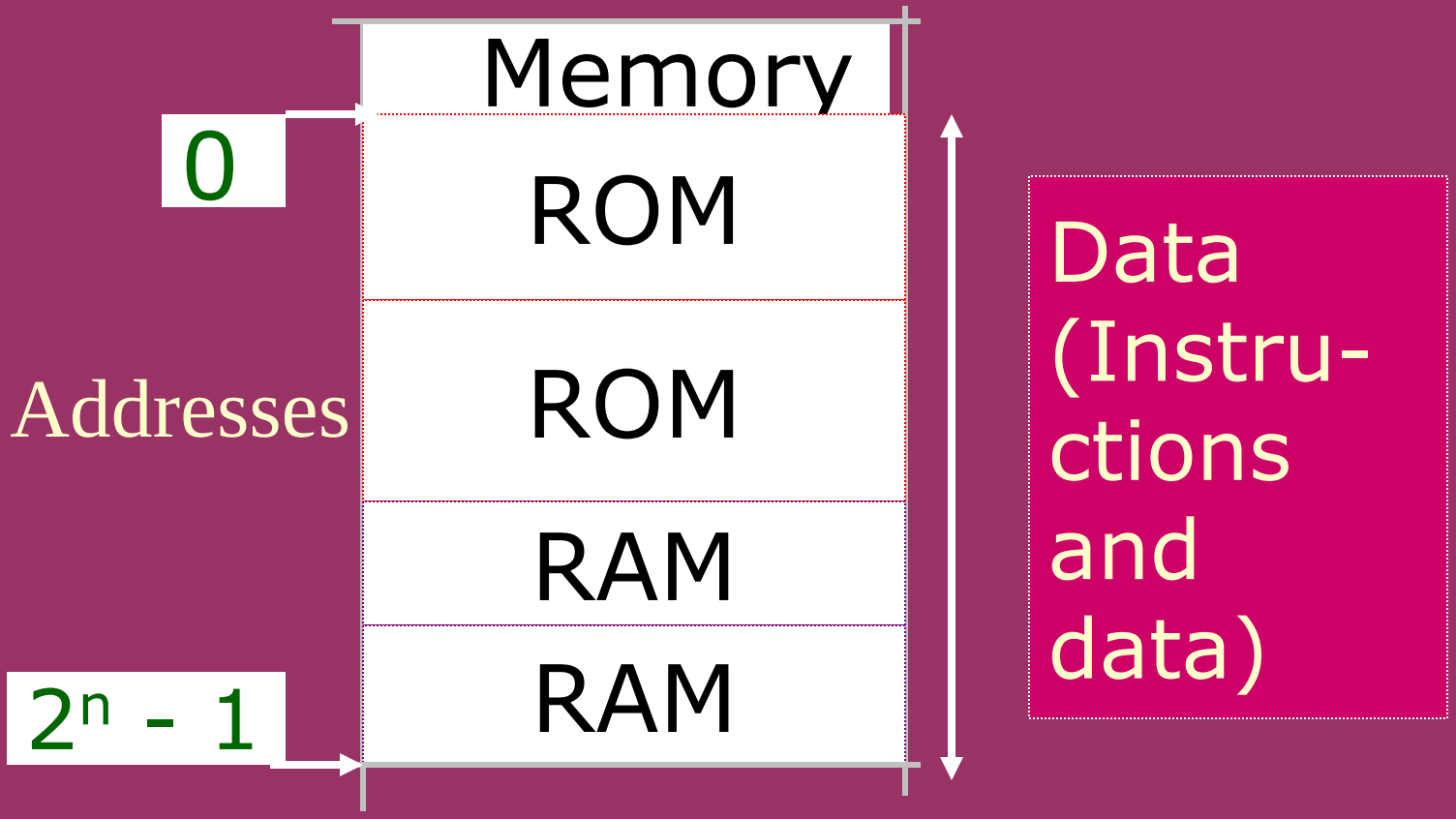
Outline

- Memory System
- ROM
- RAM
- **Address and Random Access Model**
- Data alignment and Multibyte loads and store

ADDRESSES

- Memory (both RAM and ROM) is divided into a set of storage locations, each of which can hold 1 byte (8 bits) of data.
- The storage locations are numbered, and the number of a storage location (called its *address)* is used to tell the memory system which location the processor wants to reference

4096 Addresses when $n = 12$



Computer important Characteristics

- Width of the addresses it uses, which limits the amount of memory that the computer can address.
- Most current computers use either 32-bit or 64-bit addresses, allowing them to access either 2^{32} or 2^{64} bytes of memory.

A 1982 IBM PC EXAMPLE

- 1MB memory (1024 ×1024 bytes).
- It's bootstrap program and BIOS at ROM addresses between 15×2^{16} (≡ 0xF0000) and 2^{20}-1 (≡ 0xFFFFF).
- RAM addresses between 1×2^{16} (≡ 0x10000) and $15 \times 2^{16} - 1$ (≡ 0xEFFFF).

RANDOM ACCESS MODEL OF MEMORY

- A simple random-access model of memory (for both RAM as well as ROM) and in which all memory operations take the same amount of time. Our memory system will support two operations: load and store

Store Operations

- Store operations take two operands, a value to be stored and the address where that the value should be stored in. They place the specified value in the memory location specified by the address

Load Operations

- Load operations take an operand that specifies the address containing the value to be loaded and return (fetch) the contents of that memory location into their destination (register).

The model memory functioning

- is similar to a large sheet of lined paper, where each line on the page represents a 1-byte storage location.

The model memory **write** function

- To write (store) a value into the memory, you count down from the top of the page until you reach the line specified by the address, erase the value written on the line, and write in the new value.

The model memory **read** function

- To read (load) a value, you count down from the top of the page until you reach the line specified by the address, and read the value written on that line.

LOAD OR STORE AT ONE TIME

- Most computers allow more than 1 byte of memory to be loaded or stored at one time.

LOAD OR STORE AT ONE TIME

- Generally, a load or store operation operates on a quantity of data equal to the system's bit width, and the address sent to the memory system specifies the location of the lowest-addressed byte of data to be loaded or stored.

EXAMPLE

- A 32-bit system loads or stores 32 bits (4 bytes) of data with each operation into the 4 bytes that start with the operation's address, so a load from location 424 would return a 32-bit quantity containing the bytes in location 424, 425, 426, and 427.

Outline

- Memory System
- ROM
- RAM
- Address and Random Access Model
- **Data alignment and Multibyte loads and store**

DATA ALIGNMENT

To simplify the design of the memory system, some computers require loads and stores to be "aligned,"

The address of a memory reference must be a multiple of the size of the data being loaded or stored.

DATA ALIGNMENT

A 4-byte load must have an address that is a multiple of 4,

An 8-byte store must have an address that is a multiple of 8, and so on.

Unaligned Load/Store

Other systems allow unaligned loads and stores

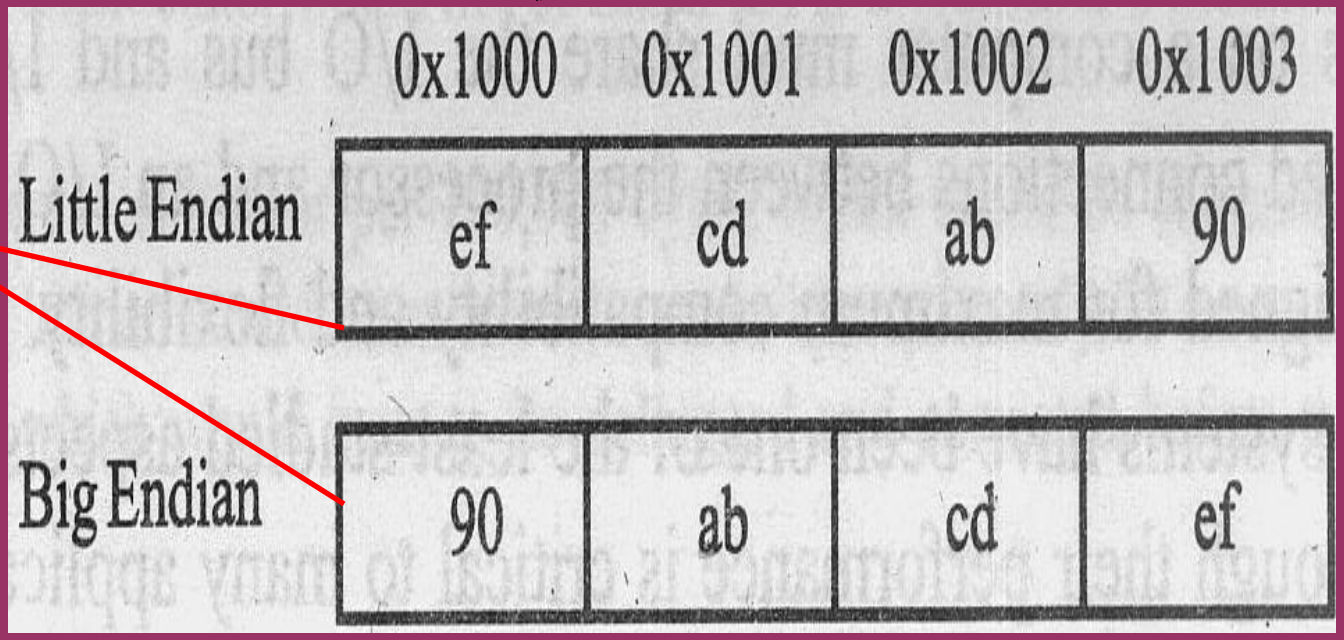
- Take significantly longer to complete such operations than aligned load

EXAMPLE

Little endian versus big endian

Word = 0x90abcdef

Address = 0x1000



4-byte word has been aligned at an address 1000, which is a multiple of 4

Endianness

- **Endianness is often an issue when transmitting data between different computer systems, however, as big-endian and little-endian computer systems will interpret the same aequence of bytes as different words of data.**

Summary

We Learnt

- Memory RAM (volatile) and ROM (non volatile)
- Flash, Memory Stick
- Both ROM and RAM access model is random access

We Learnt

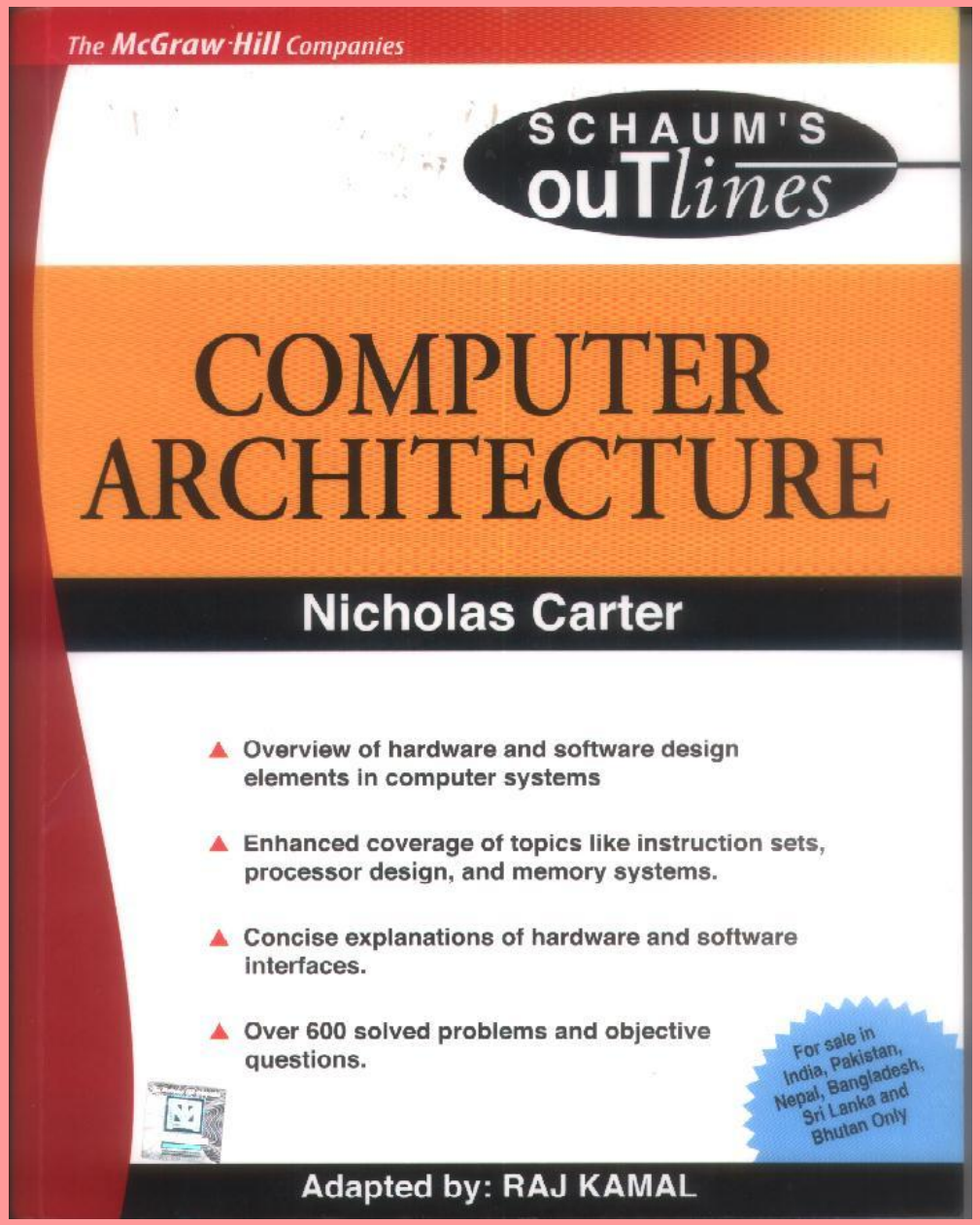
- **Data Align or do not align depend on system**
- **big-endian and little-endian depend on system**

We also learnt

- Every computation can be performed by some Turing machine.
- A universal Turing machine can perform any computation provided given enough memory and time

End of Lesson 3 on

Memory Unit


The McGraw-Hill Companies
SCHAUM'S
ouTlines
COMPUTER ARCHITECTURE
Nicholas Carter
Overview of hardware and software design elements in computer systems
Enhanced coverage of topics like instruction sets, processor design, and memory systems.
Concise explanations of hardware and software interfaces.
Over 600 solved problems and objective questions.
For sale in India, Pakistan, Nepal, Bangladesh, Sri Lanka and Bhutan Only
Adapted by: RAJ KAMAL

THANK YOU