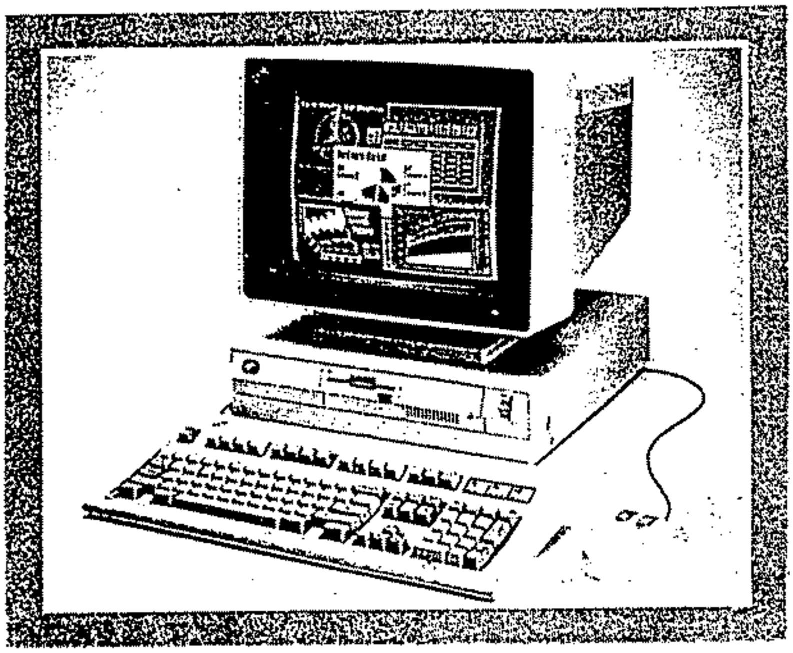
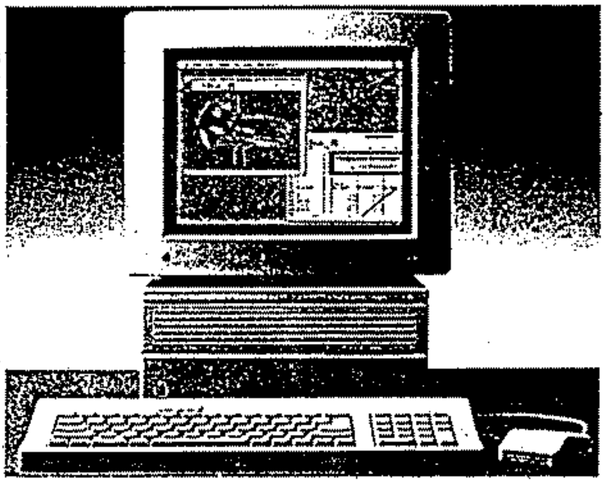
DIGITAL COMPUTER





(Discrete Information Processing System)

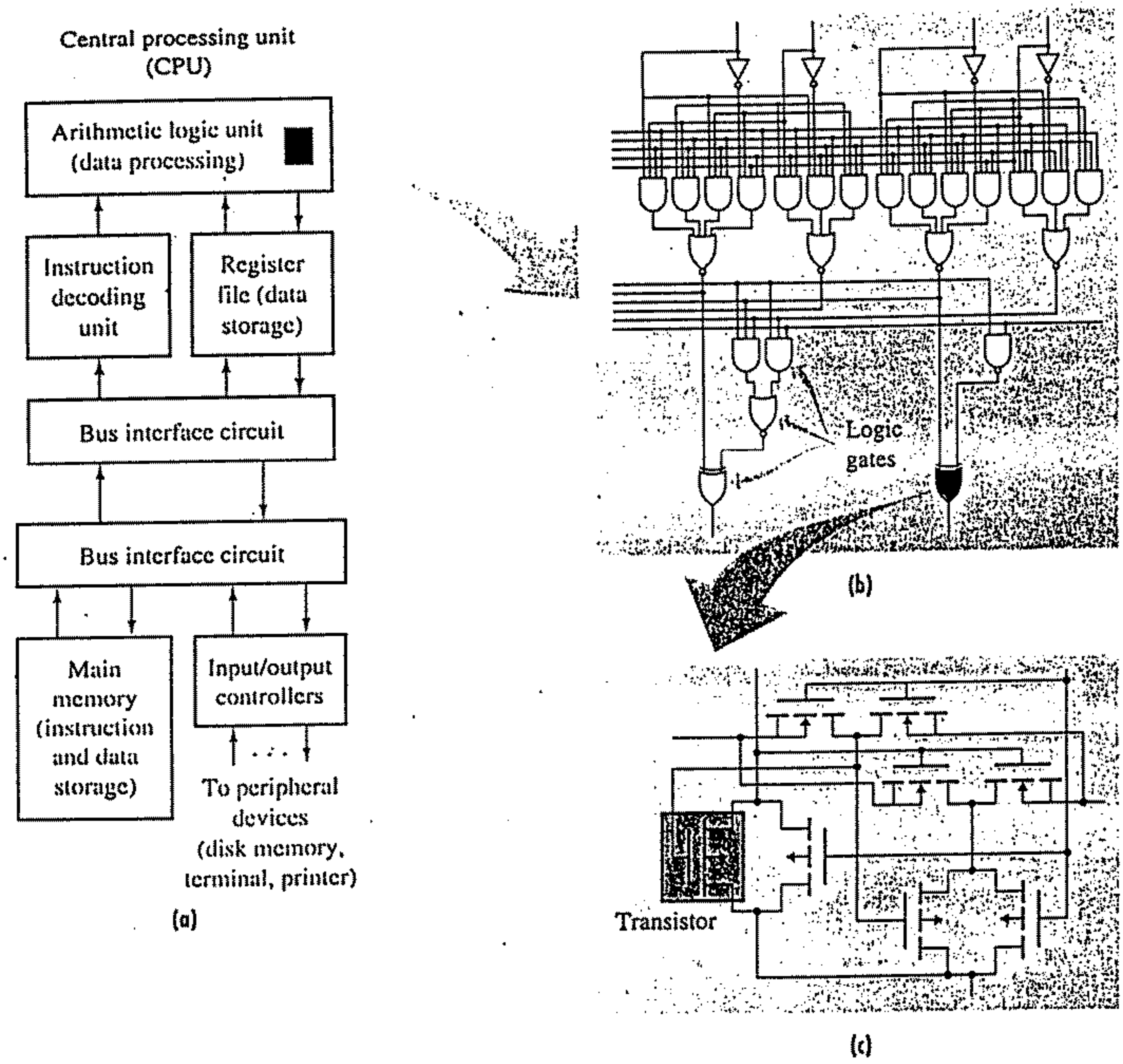
System
is an organized collection of **components** that interact via **links**
among themselves and with their environment
to provide a pre-defined **functionality**

System Structure
defines the composition of the system
in terms of system **components** and **links**

System Functionality
defines the behavior of the system
in terms of system **inputs** and **outputs**

The **structure** of a system is given at different levels of detail e.g.,

a) Compositional
b) Logical
c) Physical


Central processing unit (CPU)
Arithmetic logic unit (data processing)
Instruction decoding unit
Register file (data storage)
Bus interface circuit
Bus interface circuit
Main memory (instruction and data storage)
Input/output controllers
To peripheral devices (disk memory, terminal, printer)
(a)
Logic gates
(b)
Transistor
(c)

The **functionality** of a system with

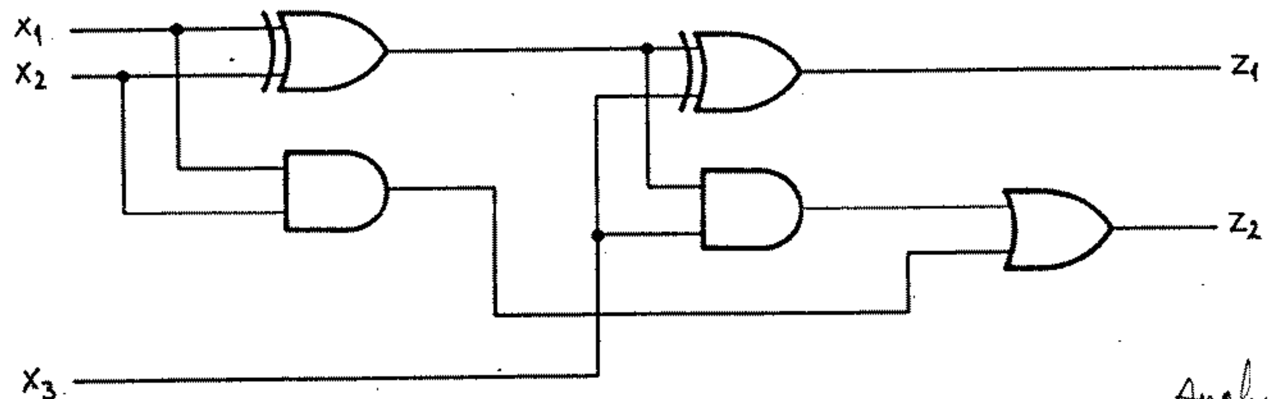
n inputs $x_1, x_2, \ldots, x_n$

m outputs $z_1, z_2, \ldots, z_m$

is defined by

$$z_1 = f(x_1, x_2, \ldots, x_n)$$
$$z_2 = f(x_1, x_2, \ldots, x_n)$$
$$\cdot$$
$$\cdot$$
$$z_m = f(x_1, x_2, \ldots, x_n)$$

e.g.,



Analysis

Design

$$z_1 = f(x_1, x_2, x_3) = (x_1 \oplus x_2) \oplus x_3$$
$$z_2 = f(x_1, x_2, x_3) = (x_1 \oplus x_2) \cdot x_3 + x_1 \cdot x_2$$

Discrete Information

Numeric Information is in Binary Number System

∴ **Number Systems**
(Binary, Octal, Decimal, Hexadecimal)
- **Representation**
- **Conversion**
- **Arithmetic Operations**

Alphanumeric Information is in Binary Coding Systems

∴ **Coding Systems**
- **Encoding**
- **Decoding**
- **Error Detecting Codes**
- **Error Correcting Codes**

Discrete Information Processing

* is expressed in a two valued Boolean Algebra as Boolean functions

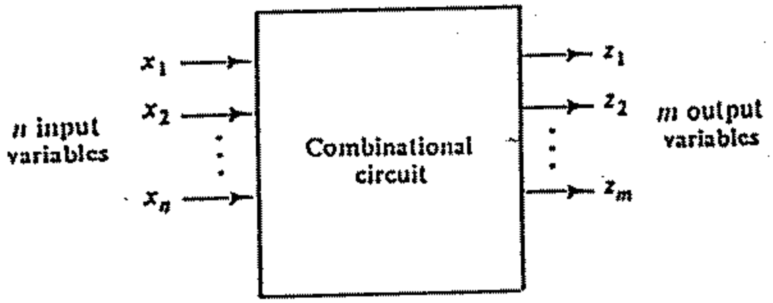
∴ **Switching Algebra**
- **Boolean Expressions (e)**
- **Truth Tables (TT)**
- **Conversion between e and TT**

* is realized by means of Logic Circuits

∴ **Logic Circuits**
- **Logic Gates**
- **Minimization**
- **Combinational Circuits**
- **Sequential Circuits**

Combinational Circuits (CC)

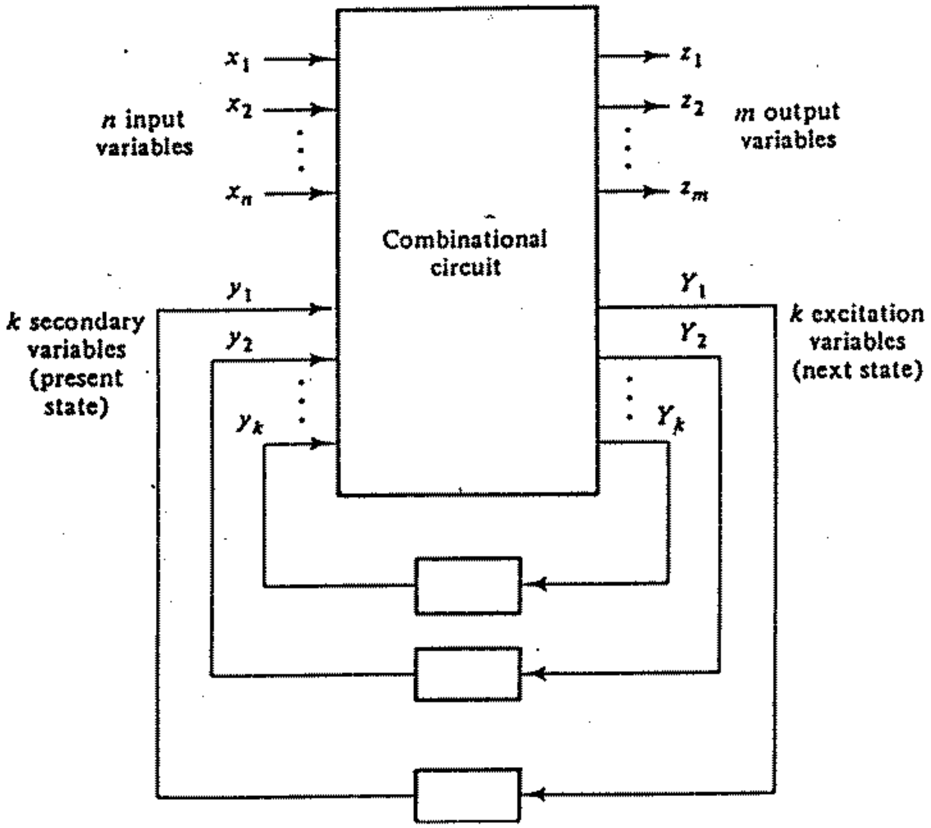
i.e.,



where $z_i = f(x_1, x_2, \ldots, x_n), 1 \leq i \leq m$

- **Half Adder**
- **Full Adder**
- **Binary Ripple Adder/Subtractor**
- **Look-Ahead Carry Adder**
- **Magnitude Comparator**
- **Code Converter**
- **Decoder**
- **Encoder**
- **Demultiplexer**
- **Multiplexer**
- **ROM (Read-Only Memory)**
- **PLA (Programmable Logic Array)**
- **ALU (Arithmetic Logic Unit)**

Sequential Circuits (SC) ... Circuits with Memory



where
$$z_i = f(\overbrace{y_1, y_2, \dots, y_k}^{\text{Present State}}, \overbrace{x_1, x_2, \dots, x_n}^{\text{Present Input}}), \; 1 \le i \le m$$
$$Y_j = f(y_1, y_2, \dots, y_k, x_1, x_2, \dots, x_n), \; 1 \le j \le k$$

- **Asynchronous Sequential Circuits (ASC)** ... without clock
  - **Latches**
  - **Analysis**
  - **Design**

- **Synchronous Sequential Circuits (SSC)** .... with clock
  - **Flip-Flops**
  - **Analysis**
  - **Design**

- **Counters**
- **Registers**

} Examples of Sequential Circuits

Basic Computer Organization and Design

- Data/Address/Control Transfer Units
- Memory Units
- Input/Output Control Units
- Central Processing Units
- Interrupts
- Addressing Modes
- Instruction Types
- Instruction Sets
- Assembly Language Programming

THEN, ...

CSI2121. Computer Architecture II (2 h.p.w.t., 2 h.lab.t.--3 cr.). Data representations. Assembly language concepts. Programming a typical microprocessor, e.g. 386. Basic input/output devices. Programming input/output operations; interrupts, e.g. disk controllers. Prerequisite(s): CSI2111.