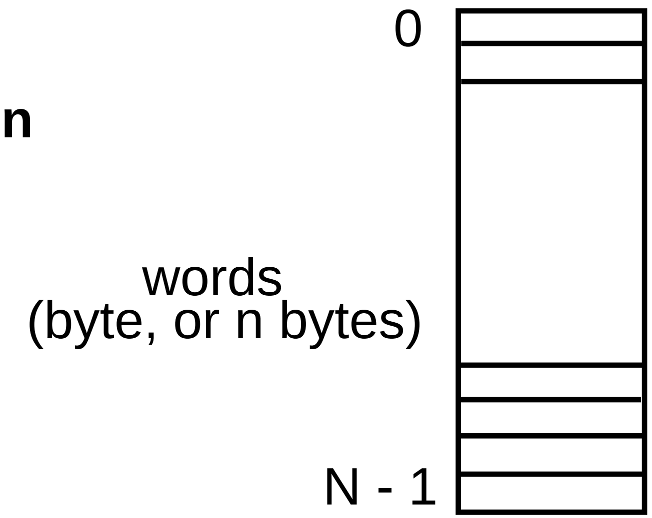
Computer System Architecture

DR. Howida Youssry

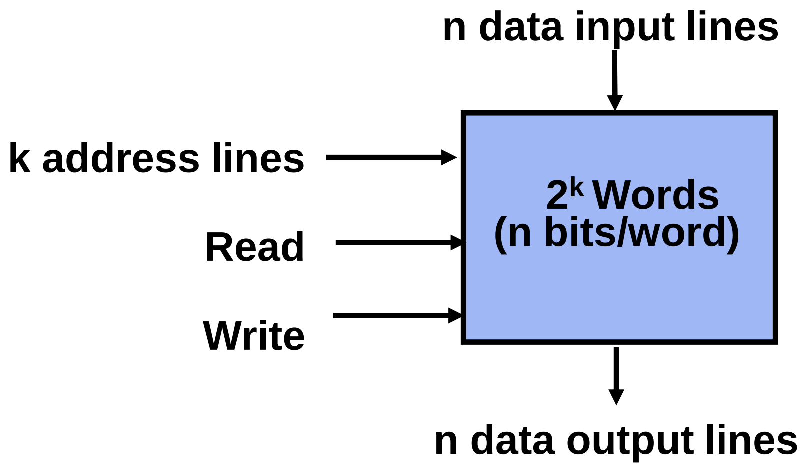
MEMORY COMPONENTS

Logical Organization

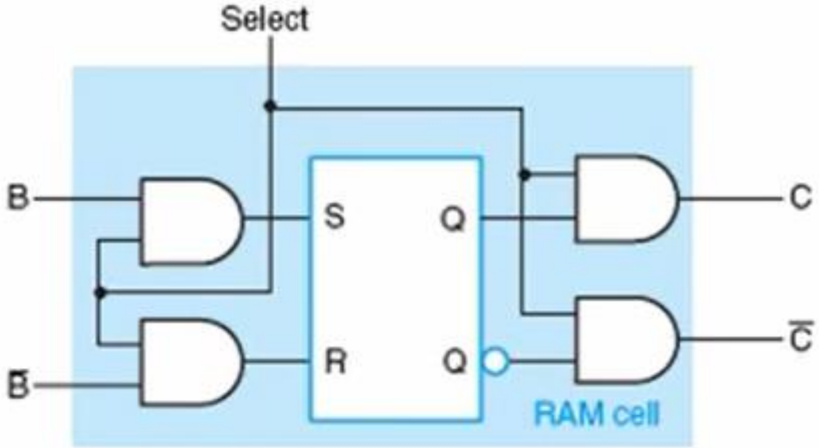


Random Access Memory

- **Each word has a unique address**
- **Access to a word requires the same time independent of the location of the word**
- **Organization**



Memory Components



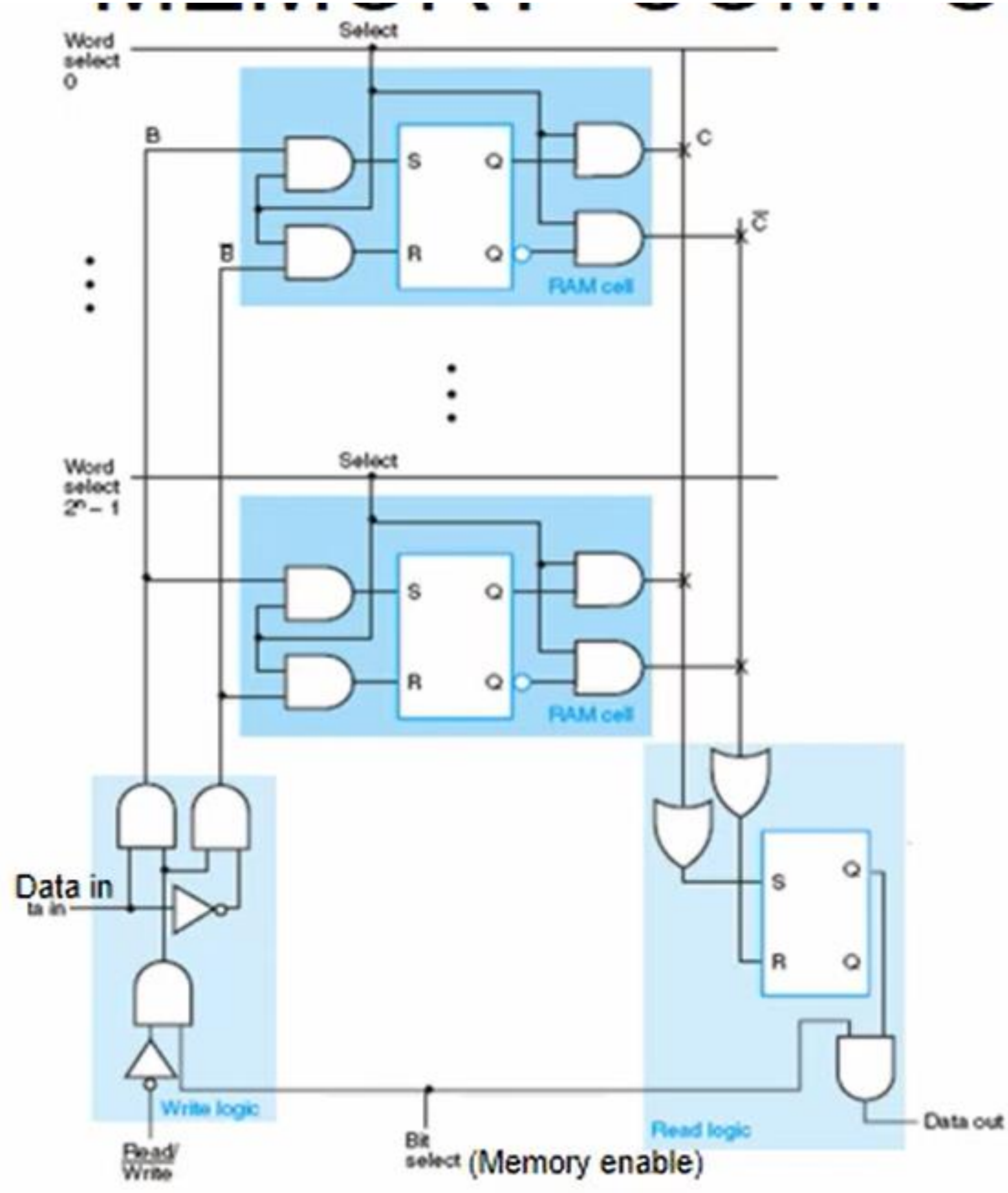
Static RAM Cell

Select = 0 (cell disable)
Select = 1 (cell enable)

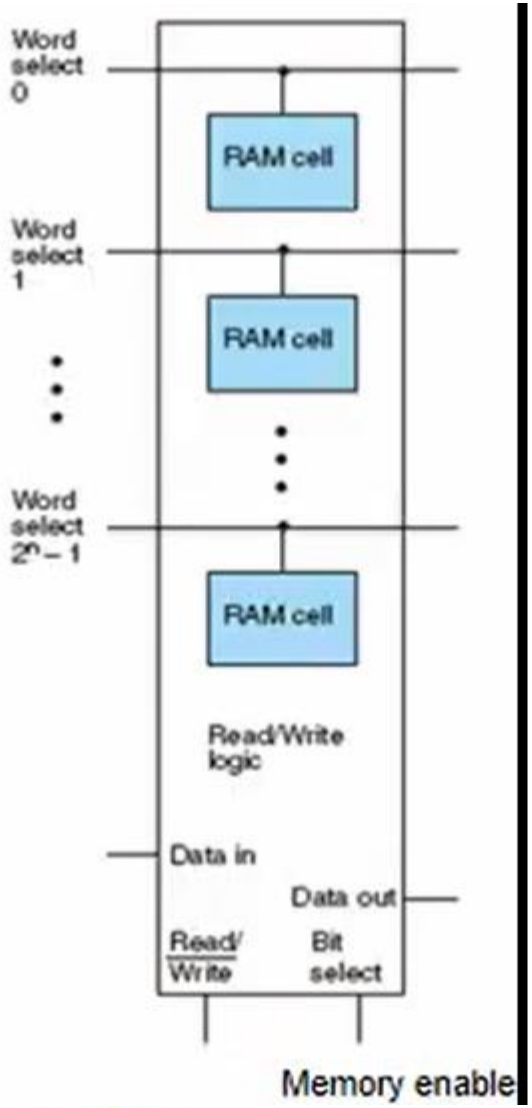
Memory address		
Binary	Decimal	Memory contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	•	•
	•	•
	•	•
	•	•
	•	•
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

Contents of a 1024 $\times$ 16 Memory

Memory Components



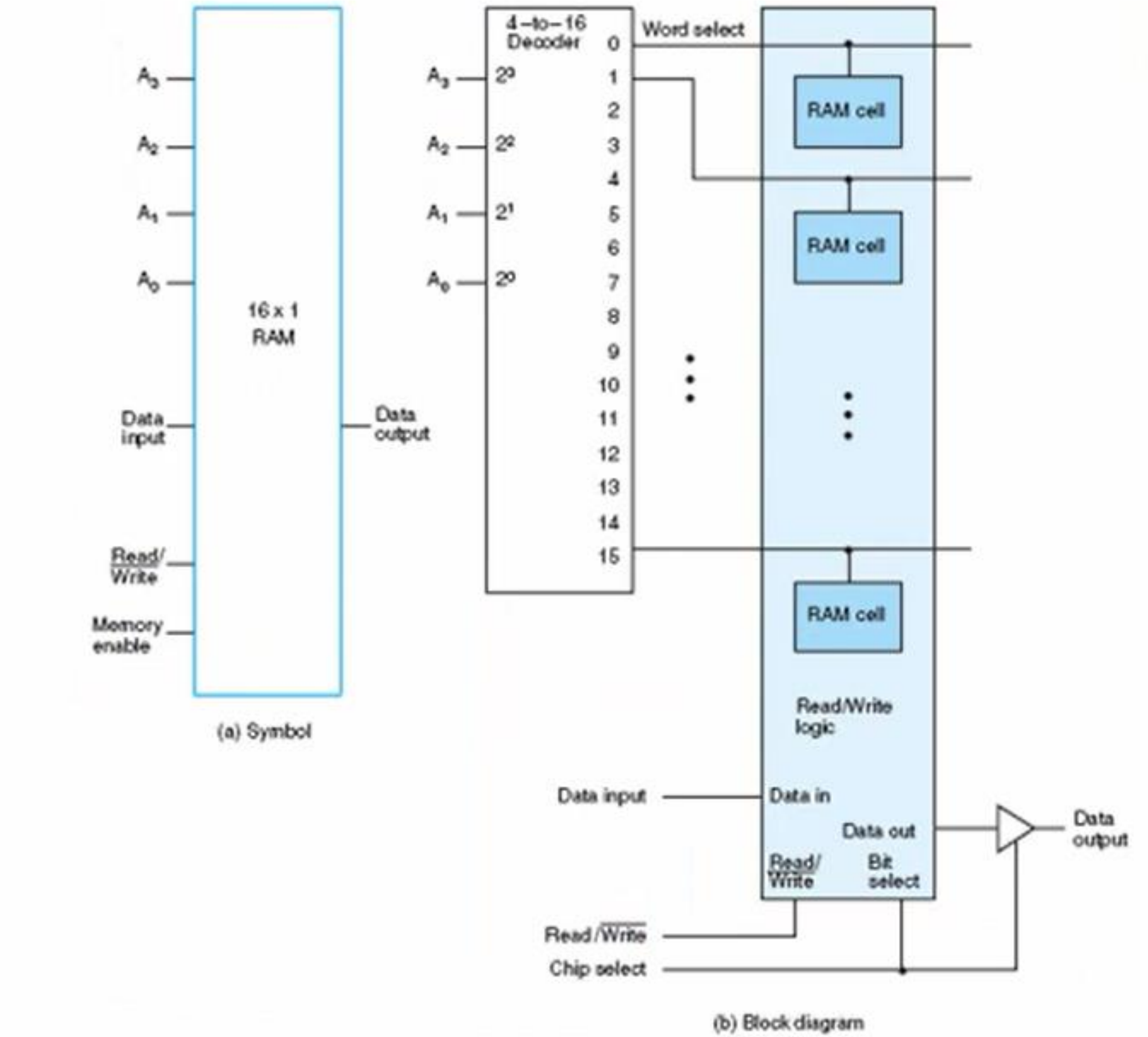
(a) Logic diagram



Control Inputs to Memory Chip

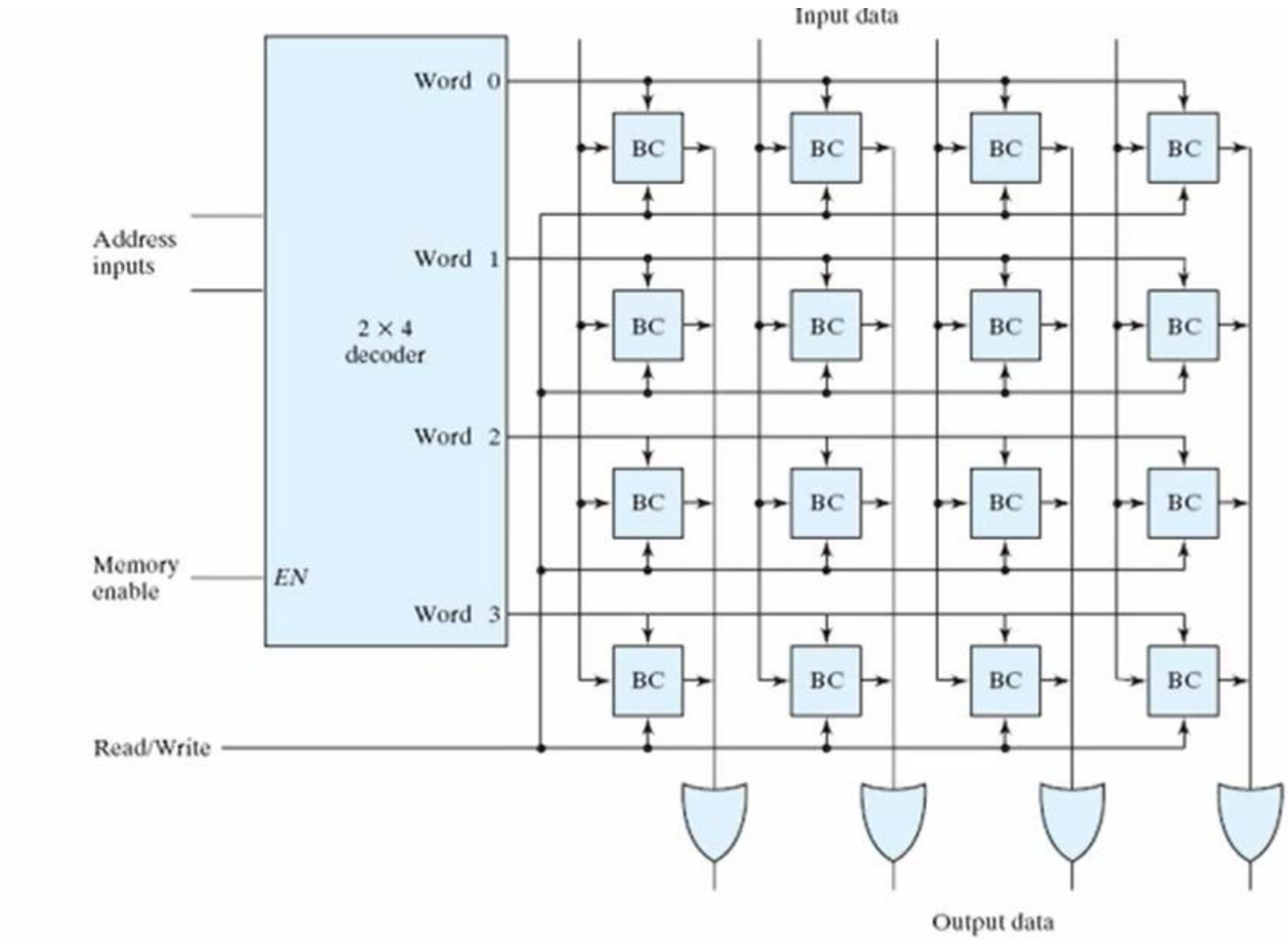
Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Memory Components



16-Word by 1-Bit RAM Chip

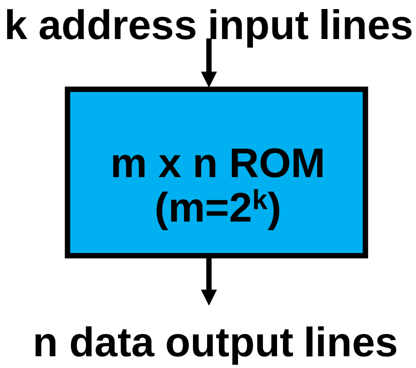
Memory Components



READ ONLY MEMORY(ROM)

Characteristics

- Perform read operation only, write operation is not possible
- Information stored in a ROM is made permanent during production, and cannot be changed
- Organization



Information on the data output line depends only on the information on the address input lines.

--> Combinational Logic Circuit

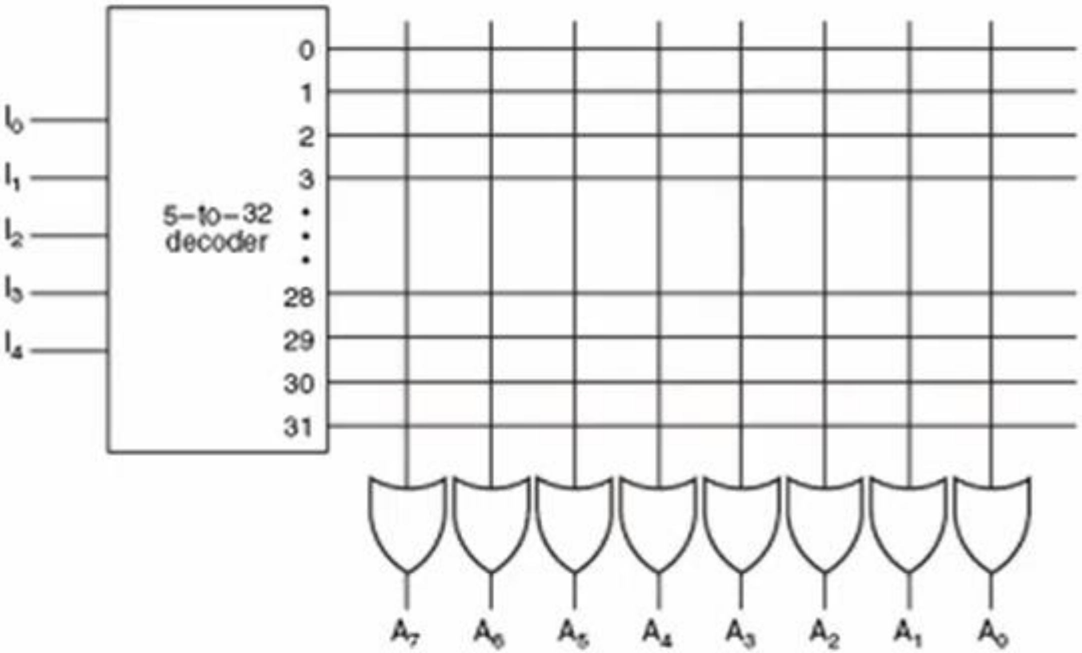
$X_0 = A'B' + B'C$
$X_1 = A'B'C + A'BC'$
$X_2 = BC + AB'C'$
$X_3 = A'BC' + AB'$
$X_4 = AB$

Canonical minterms

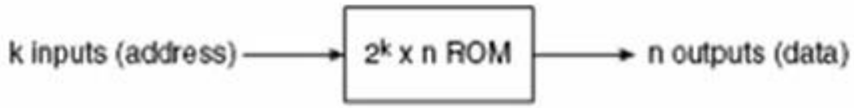
$X_0 = A'B'C' + A'B'C + AB'C$
$X_1 = A'B'C + A'BC'$
$X_2 = A'BC + AB'C' + ABC$
$X_3 = A'BC' + AB'C' + AB'C$
$X_4 = ABC' + ABC$

address ABC	*Output* X_0	X_1	X_2	X_3	X_4
000	1	0	0	0	0
001	1	1	0	0	0
010	0	1	0	1	0
011	0	0	1	0	0
100	0	0	1	1	0
101	1	0	0	1	0
110	0	0	0	0	1
111	0	0	1	0	1

READ ONLY MEMORY(ROM)



Internal Logic of a 32 × 8 ROM



Block Diagram of ROM

TYPES OF ROM

ROM

- Store information (function) during production
- Mask is used in the production process
- Unalterable
- Low cost for large quantity production --> used in the final products

PROM (Programmable ROM)

- Store info electrically using PROM programmer at the user's site
- Unalterable
- Higher cost than ROM -> used in the system development phase
  -> Can be used in small quantity system

EPROM (Erasable PROM)

- Store info electrically using PROM programmer at the user's site
- Stored info is erasable (alterable) using UV light (electrically in some devices) and rewriteable
- Higher cost than PROM but reusable --> used in the system development phase. Not used in the system production due to erasability

INTEGRATED CIRCUITS

Classification by the Circuit Density

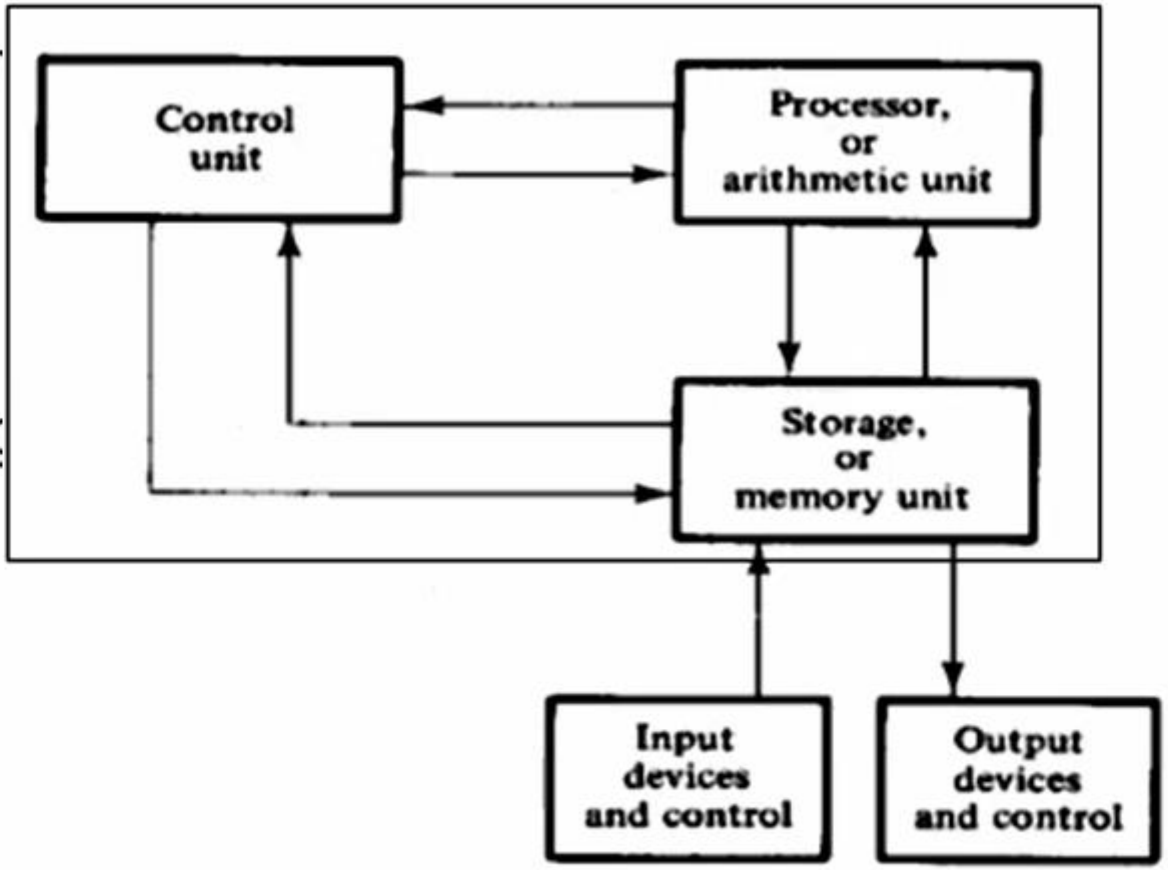
SSI - several (less than 10) independent gates
MSI - 10 to 200 gates; Perform elementary digital functions;
    Decoder, adder, register, parity checker, etc
LSI - 200 to few thousand gates; Digital subsystem
    Processor, memory, etc
VLSI - Thousands of gates; Digital system
    Microprocessor, memory module

Classification by Technology

TTL - Transistor-Transistor Logic
    Bipolar transistors
    NAND
ECL - Emitter-coupled Logic
    Bipolar transistor
    NOR
MOS - Metal-Oxide Semiconductor
    Unipolar transistor
    High density
CMOS - Complementary MOS
    Low power consumption

REGISTER TRANSFER LANGUAGE

- A **digital system** is an interconnection of digital hardware ***modules*** that accomplish a specific information processing task.
- The various modules are *interconnected* with **common data** and **control paths** to form a digital computer system.



General Computer organization

Hardware Organization of Digital Computer

Hardware organization of a digital computer is defined by specifying:

- The **set of register** it contains and their function.
- The **sequence of micro-operations** performed on the binary information stored in the registers.
- The **control** that initiates the sequence of micro-operations

Micro-operation

Micro-operations are the basis for microprocessors.

The operation executed on data stored in register are called micro-operation

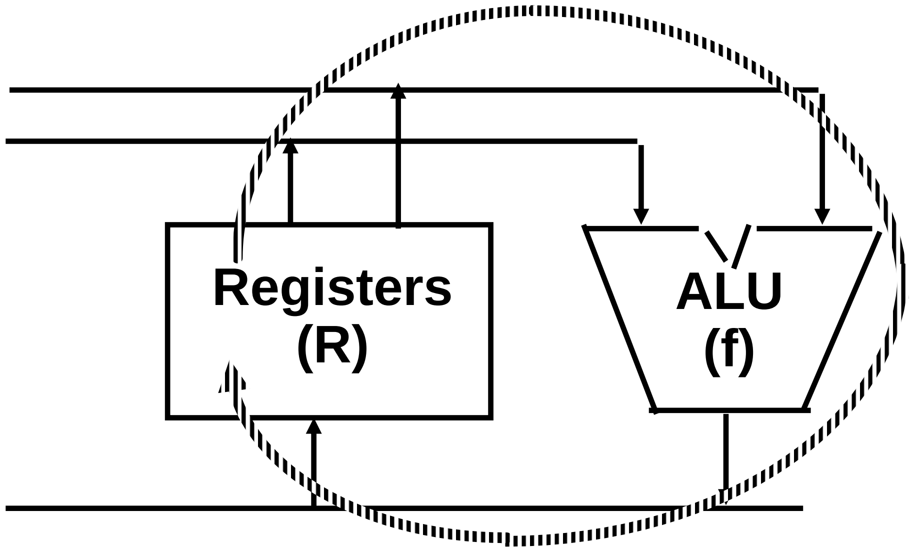
That is, a microprocessor performs the micro-operations in order to realize *the instruction*.

- An instruction is fetched from memory, decoded and executed by performing a sequence of micro-operations.

The micro-operations only specify which data transfers may occur; they don't specify when or how they may occur.

MICROOPERATION

An elementary operation performed during one clock pulse, on the information stored in one or more registers



1 clock cycle

$R \leftarrow f(R, R)$

f: shift, count, clear, load, add,...

MICRO-OPERATIONS

Four types of microoperations

- **Register transfer microoperations**
- **Arithmetic microoperations**
- **Logic microoperations**
- **Shift microoperations**

*** Summary of Arithmetic Micro-Operations**

$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow R2'$	Complement the contents of R2
$R2 \leftarrow R2' + 1$	2's complement the contents of R2 (negate)
$R3 \leftarrow R1 + R2' + 1$	subtraction
$R1 \leftarrow R1 + 1$	Increment
$R1 \leftarrow R1 - 1$	Decrement

REGISTER TRANSFER LANGUAGE

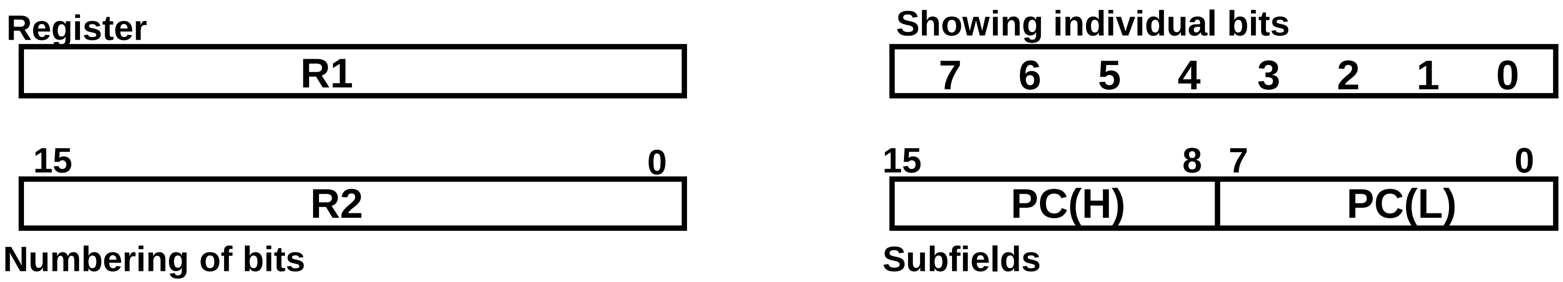
For any function of the computer, a sequence of microoperations is used to describe it
----> *Register transfer language*

- **A symbolic language**
- **A convenient tool for describing the internal organization of digital computers**
- **Can also be used to facilitate the design process of digital systems.**

REGISTER TRANSFER

Designation of a register
- **a register**
- **portion of a register**
- **a bit of a register**

Common ways of drawing the block diagram of a register



Representation of a transfer(parallel)

$R2 \leftarrow R1$

A simultaneous transfer of all bits from the source to the destination register, during one clock pulse

Representation of a controlled(conditional) transfer

P: $R2 \leftarrow R1$

A binary condition(p=1) which determines when the transfer is to occur

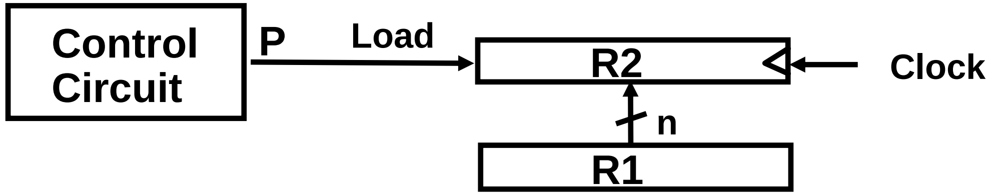
If (p=1) then ($R2 \leftarrow R1$)

HARDWARE IMPLEMENTATION OF CONTROLLED TRANSFERS

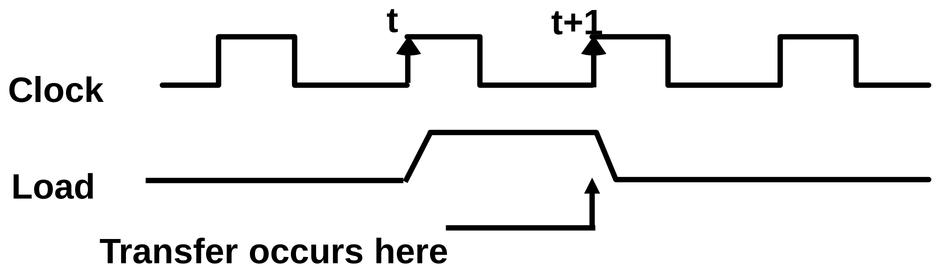
Implementation of controlled transfer

P: R2 ← R1

Block diagram



Timing diagram



Basic Symbols for Register Transfers

Symbols	Description	Meaning
Capital letters and numerals	Denotes a register	MAR, R2
Parentheses ( )	Denotes a part of a register	R2(0-7), R2(L)
Arrow ←	Denotes transfer of information	R2 ← R1
Colon :	Denotes termination of control function	P:
Comma ,	Separates two micro-operations	A ← B, B ← A