Computer System Architecture

Ms. Yuzana Hlaing
M.E(IT), Ph.D. Candidate(thesis)
Lecturer
Computer Engineering and Information Technology Dept.
Yangon Technological University

Course Information

Course Title: Computer System Architecture

Text Book: ***Computer System Architecture***, International Edition, Third Edition by M. Morris Mano.

Reference: ***Computer Organization and Design,*** Tenth Edition by William Stallings

Course Duration: 12 weeks

Course Objectives

- To understand the basic concepts of the hardware operation of the logic level components in order to have the skill to design the digital circuits.
- To be able to develop the implementation of the computer hardware applications.
- To get the ability of designing and programming the advanced digital circuits to process many functions and operations.
- To support the creative thinking of how to improve the existing computer system in order to increase performance of this system.

Course Preface

This course deals with **computer architecture** as well as **computer organization and design.**

- **Computer organization** is concerned with how the hardware components operate and how they are connected together to form the computer system.

- **Computer design or Computer implementation** is concerned with what hardware should be used and how the parts should be connected.

- **Computer architecture** is concerned with the structure and behavior of the computer as seen by the users.

Course Schedule

Periods	Lectures
Week-1	Introduction to Computer System Architecture
Week-2	Data Presentations
Week-3	Register Transfer and Microoperations
Week-4	Basic Computer Organization and Design
Week-5	Programming the Basic Computer
Week-6	Microprogrammed Control
Week-7	Central Processing Unit
Week-8	Pipeline and Vector Processing
Week-9	Computer Arithmetic
Week-10	Input-Output Organization
Week-11	Memory Organization
Week-12	Multiprocessors

Exam System with Grading

Task	Mark
Final Exam	80%
Lab and Practical Assignments	15%
Tutorials	5%
Total	**100%**

Mark	Grade
85~100	A+
75~85	A
60~75	B+
50~60	B
<50	F

Lecture-1

Introduction to Computer System Architecture

Lecture-1
Introduction to Computer System Architecture

- What is digital computer?
- Boolean Algebra
- Combinational Circuits
- Flip Flops
- Sequential Circuits
- Integrated Circuits
- Memory Unit

What is a Digital Computer?

What is digital computer?

- A **digital computer** is a digital system that performs various computational tasks.
- It is subdivided into
  - Hardware
    - Main Unit
    - Peripherals
  - Software
    - Operating system (also called System Programs)
    - Application programs

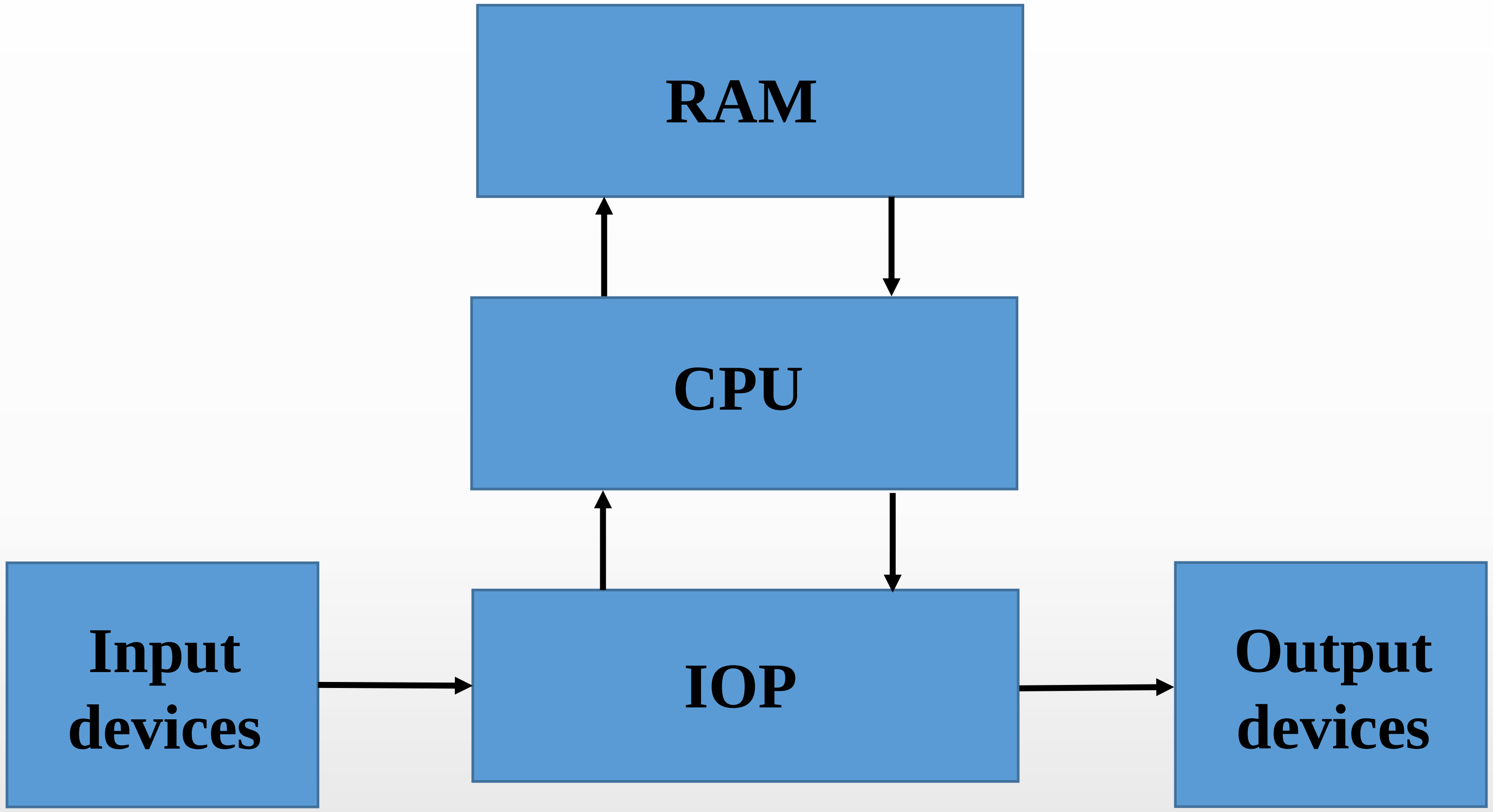


Fig: Block Diagram of a Digital Computer

Boolean Algebra

Boolean Algebra

- A **Boolean algebra** is an algebra that deals with binary variables and logic operations.
- A **Boolean function** can be expressed algebraically with binary variables, the logic operation symbols, parentheses, and equal sign.
- A **truth table** represents the relationship between a function and its binary variables.
- A Boolean function can be transformed from an algebraic expression into a **logic diagram** composed of AND, OR and inverter gates.
- The purpose of Boolean algebra is to facilitate the analysis and design of digital circuits.
- It provides a convenient tool to:
  1. Express in algebraic from a truth table relationship between binary variables.
  2. Express in algebraic form the input-output relationship of logic diagrams.
  3. Find simpler circuits for the same function.

TABLE: Basic Identities of Boolean Algebra	
(1) $x + 0 = x$	(2) $x . 0 = 0$
(3) $x + 1 = 1$	(4) $x . 1 = x$
(5) $x + x = x$	(6) $x . x = x$
(7) $x + x' = 1$	(8) $x . x' = 0$
(9) $x + y = y + x$	(10) $xy = yx$
(11) $x + (y + z) = (x + y) + z$	(12) $x(yz) = (xy)z$
(13) $x (y + z) = xy + xz$	(14) $x + yz = (x + y) (x + z)$
(15) $(x + y)' = x'y'$	(16) $(xy)' = x' + y'$
(17) $(x')' = x$	

- To see how the Boolean algebra manipulation is used to simplify digital circuits in order to reduce the logic gates requirements, the following equation can be studied as example.

$$
\begin{aligned}
F &= ABC + AB\acute{C} + \acute{A}C \\
&= AB(C + \acute{C}) + \acute{A}C \\
&= AB + \acute{A}C
\end{aligned}
$$

For $F = ABC + AB\acute{C} + \acute{A}C$ — For $F = AB + \acute{A}C$

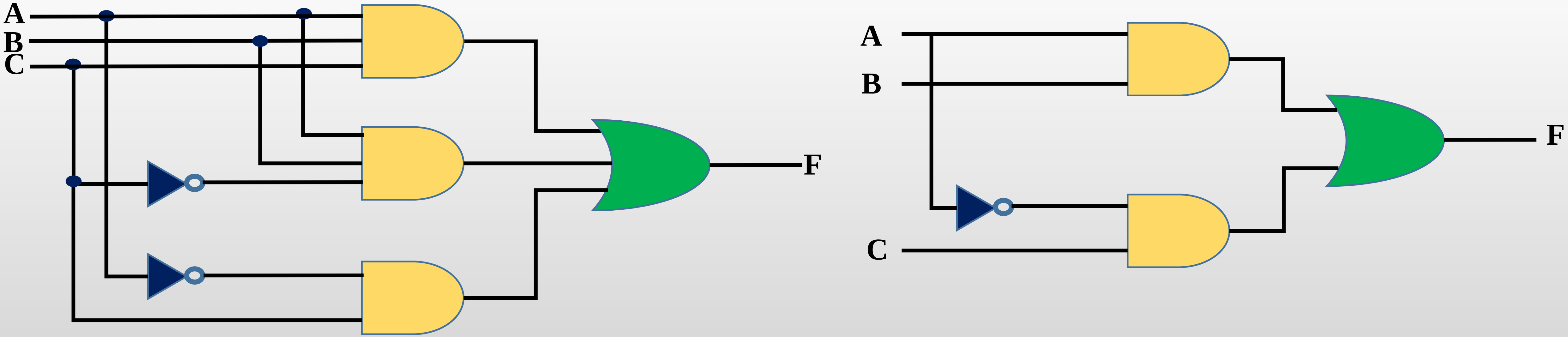


Fig: Two Logic Diagrams for Same Boolean Function

Combinational Circuits

Combinational Circuits

- A **combinational circuits** is a connected arrangement of logic gates with a set of inputs and outputs.
- The procedure to design the combinational circuits involves the following steps.
  - The problem is stated.
  - The input and output variables are assigned letter symbols.
  - The truth table that defines the relationship between inputs and outputs is derived.
  - The simplified Boolean functions for each output are obtained.
  - The logic diagram is drawn.

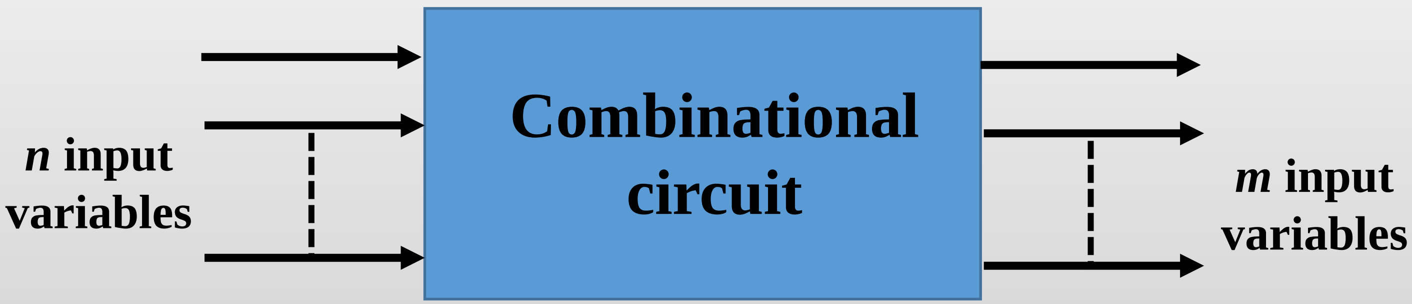


Fig: Block Diagram of a Combinational Circuit

Adders

- Adders are one kind of combinational circuits.
- An adder is a digital circuit that performs addition of binary numbers.
- Two kinds of adders
  - Half adder
  - Full adder.

For Half adder:

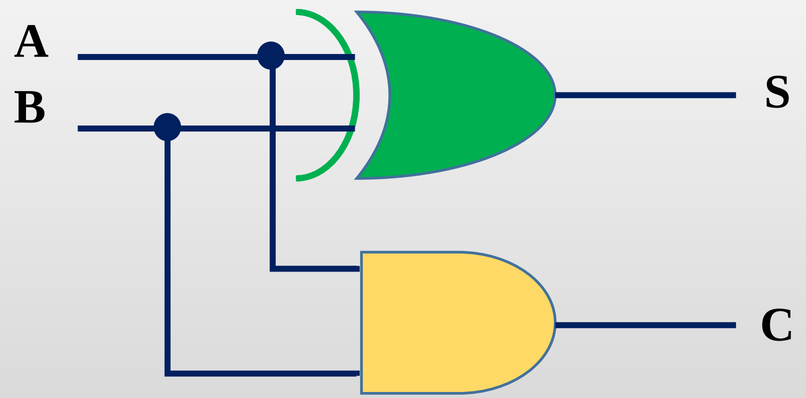
Truth table

A	B	S	C_{in}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Equation

$$S = \overline{A}B + A\overline{B} = A \oplus B$$
$$C = AB$$

Logic diagram



For Full adder:

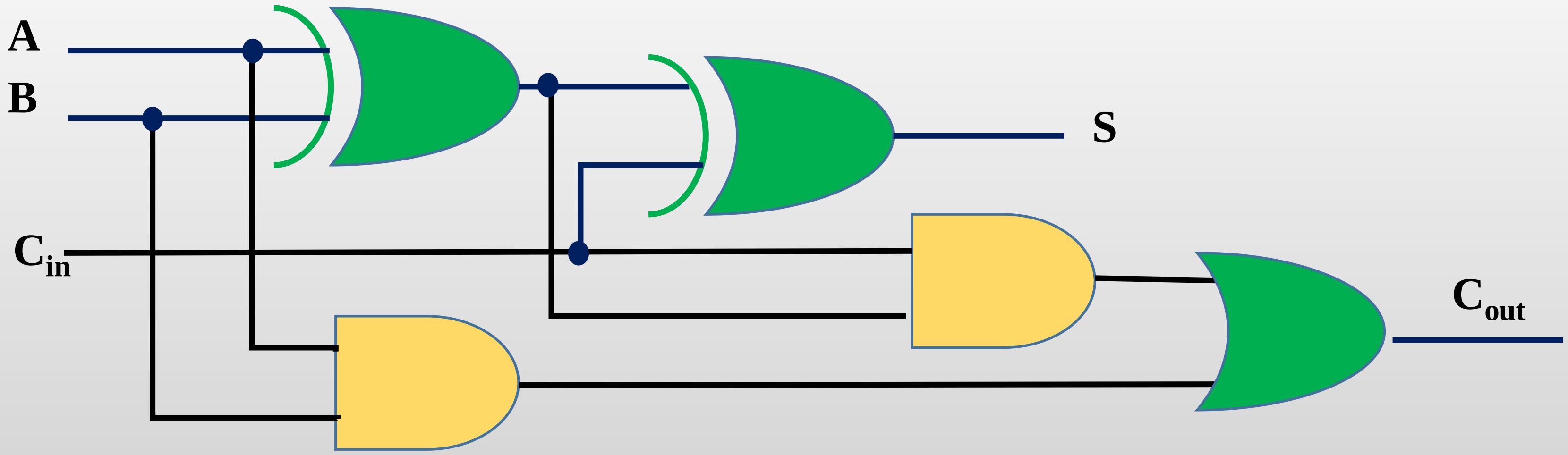
Truth table

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Equation

$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + (A \oplus B)C_{in}$$

Logic diagram



Decoders

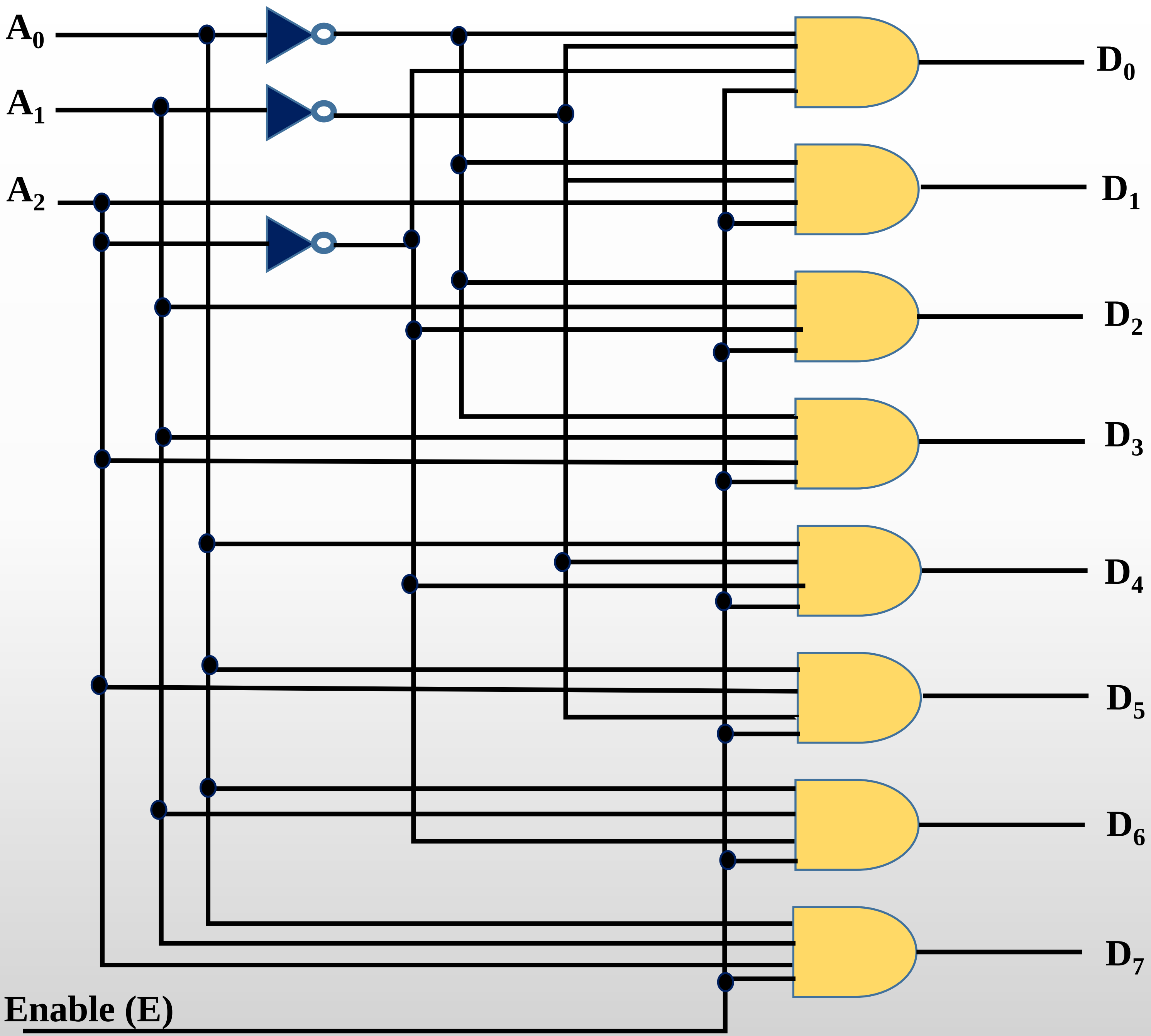
- A **decoder** is a combinational circuit that converts binary information from the n coded inputs to a maximum of 2^n unique outputs.

For 3-to-8-line Decoder:

Truth table

Enable	Inputs			Outputs							
E	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Gate level illustration



NAND Gate Decoders

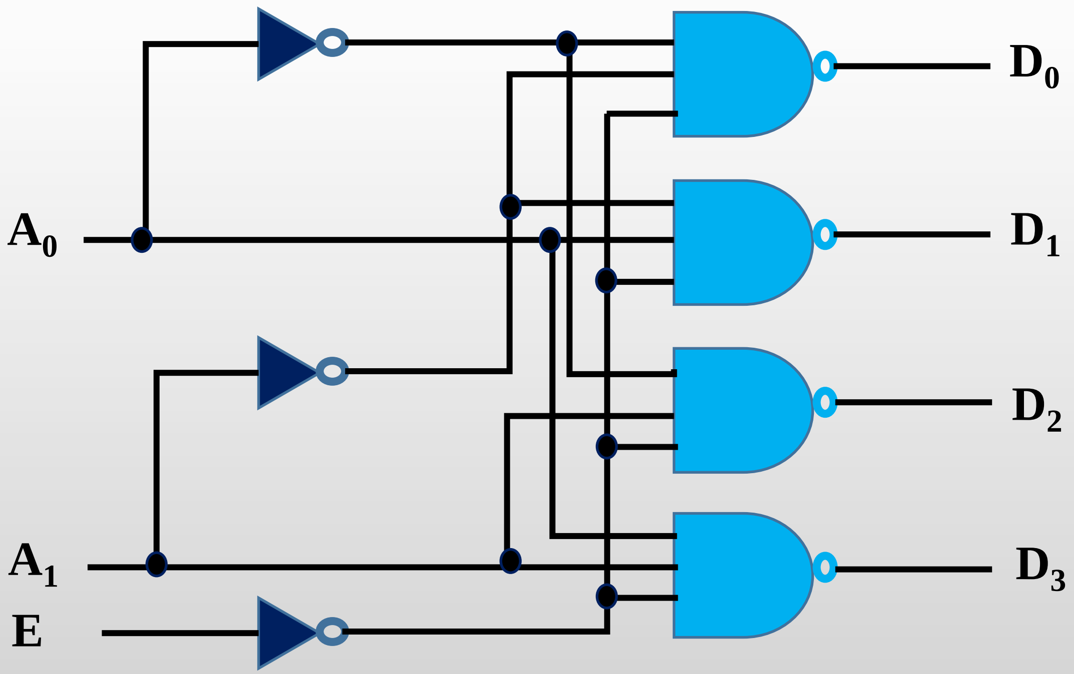
- A decoder constructed with NAND instead of AND is called NAND gate decoder which becomes more economical to generate the decoder outputs in their complement form.

For 2-to-4-line Decoder with nand gate:

Truth table

E	A1	A0	D0	D1	D2	D3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

Gate level illustration



Multiplexer

- A **multiplexer** is a combinational circuit that receives binary information from one of two input data lines and directs it to a single output line.
- The selection of a particular input data line for the output is determined by a set of selection inputs.

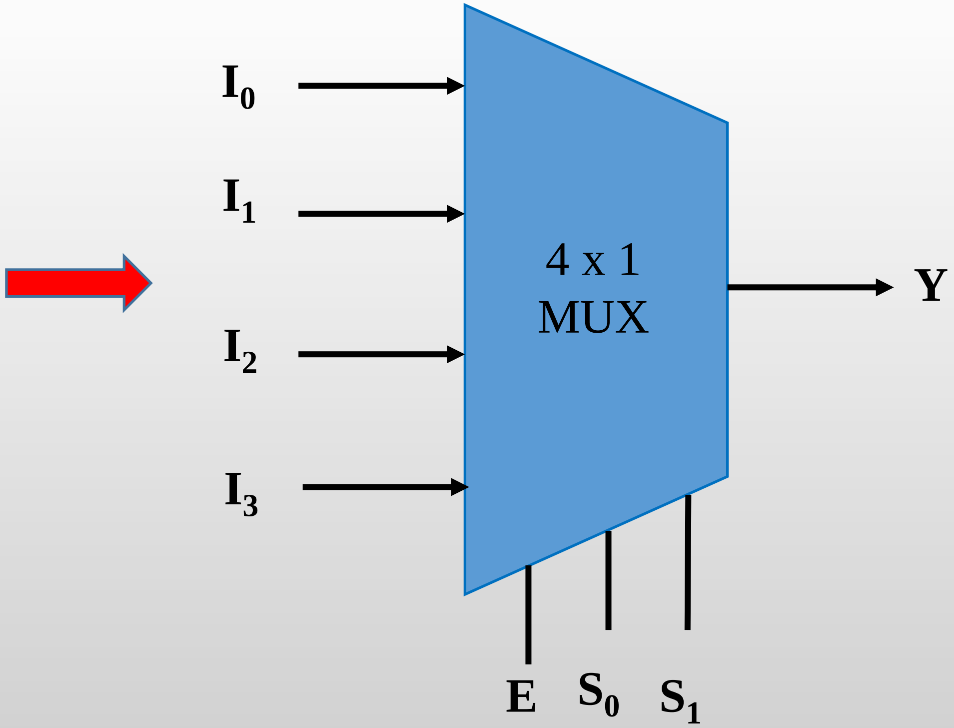
For 4 x 1 multiplexer,

$$Y = \overline{S_1 S_0} I_0 + \overline{S_1} S_0 I_1 + S_1 \overline{S_0} I_2 + S_1 S_0 I_3$$

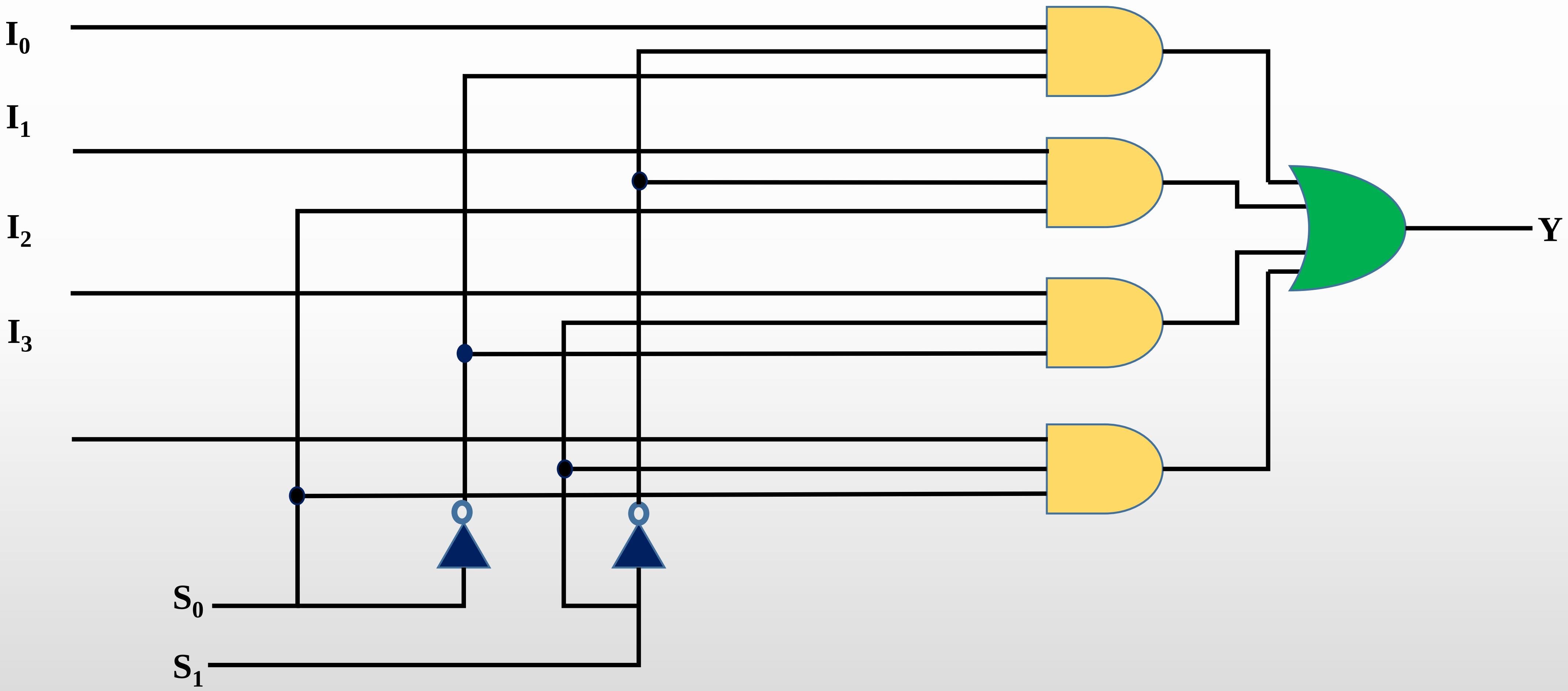
Truth table

Select line		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Block diagram



Gate level illustration



Flip Flops

Flip-flops

- A **flip-flop** is a device that is used to store one single bit of information.
- A flip-flop maintains a binary state until directed by a clock pulse to switch states.
- Different types of flip-flops are
  - SR flip-flop
  - D flip-flop
  - JK flip-flop
  - T flip-flop
- The **difference** among various types of flip-flops is in the number of inputs they possess and in the manner in which the inputs affect the binary state.

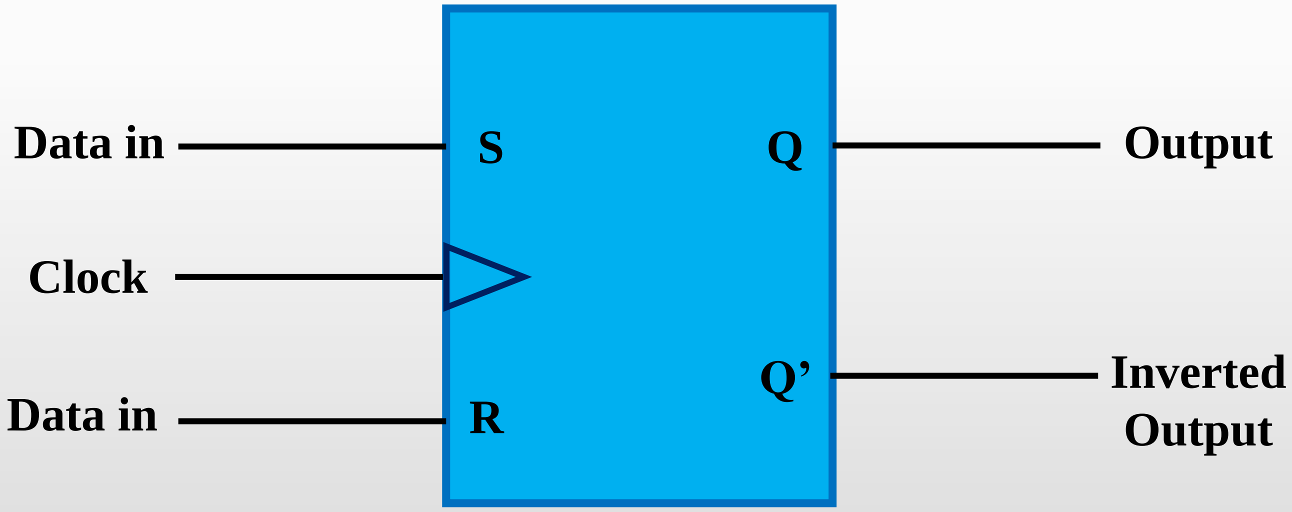
SR Flip-flop

- In **SR flip-flop**, two inputs called S (Set) and R (Reset) and one clock called C control the output of this flip-flop according to the values of these input.

Truth table

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	?

Block Diagram



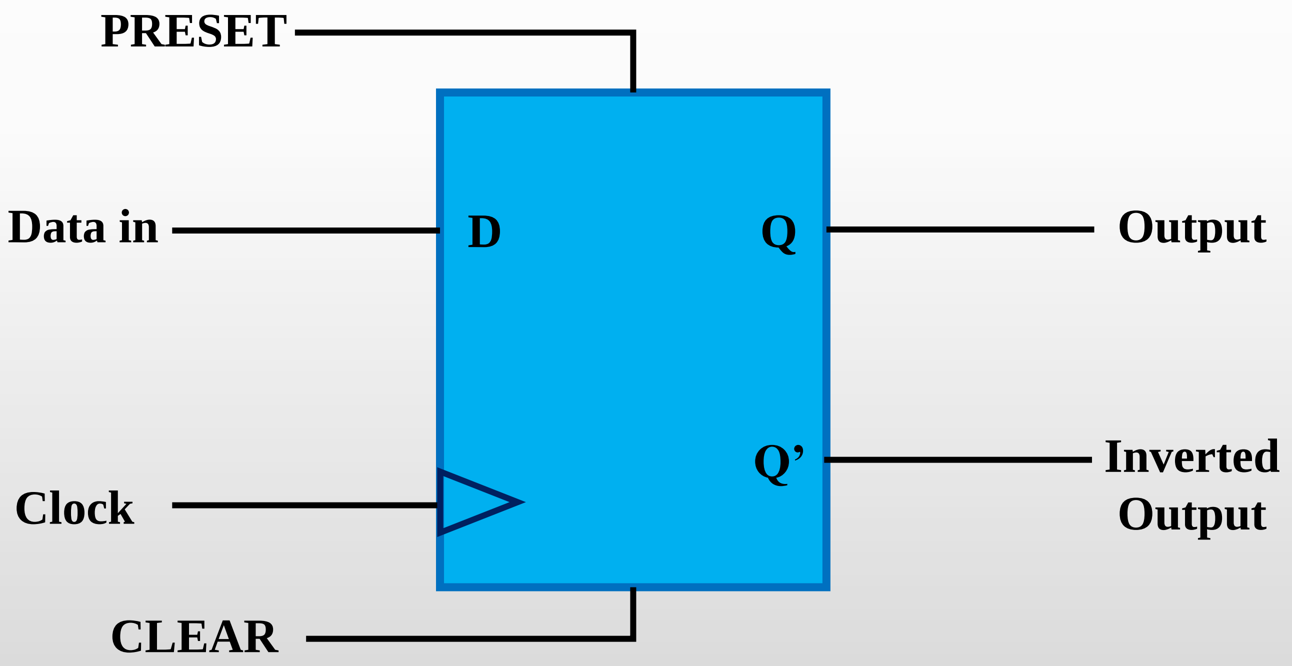
D Flip-flop

- The **D(data) flip-flop** is a slight modification of SR flip-flop by inserting an inverter between S and R and assigning the symbol D to the single output.

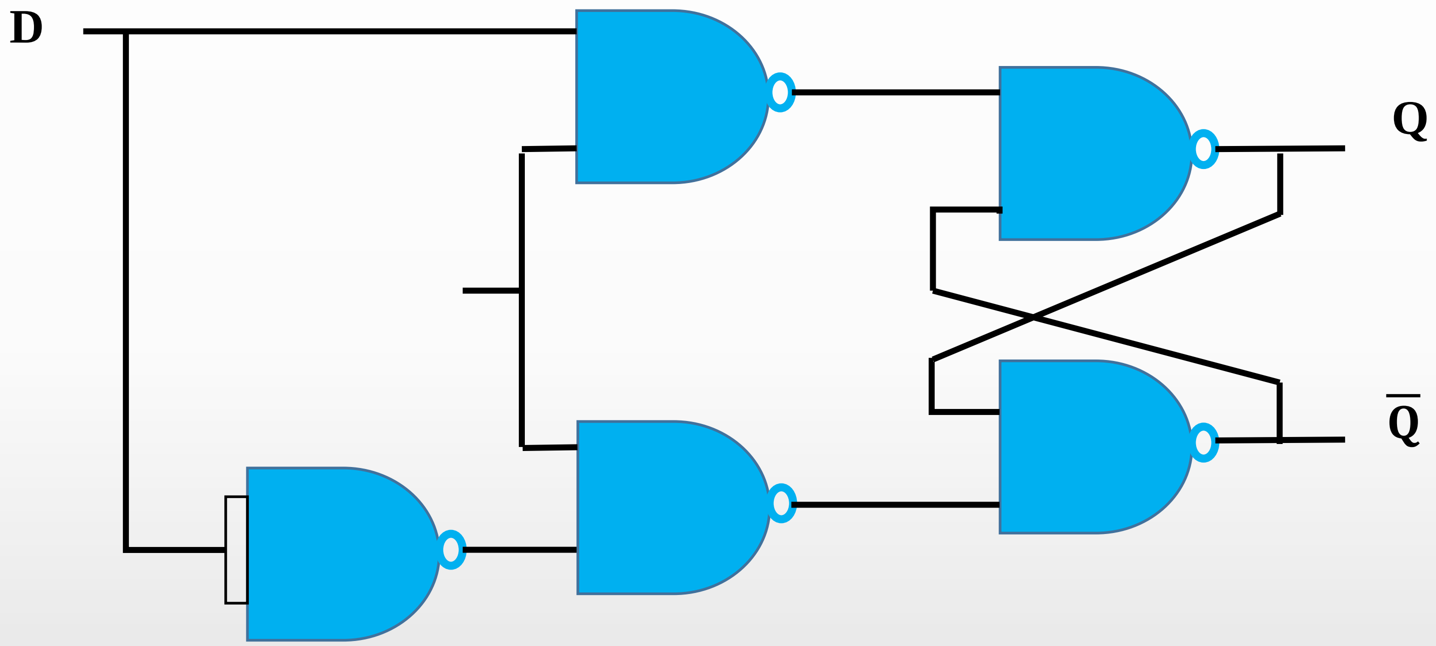
Truth table

Clk	Q(t)	D	Q(t+1)
0	x	x	Q(t)
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Block Diagram



Gate level Illustration of D Flip-flop



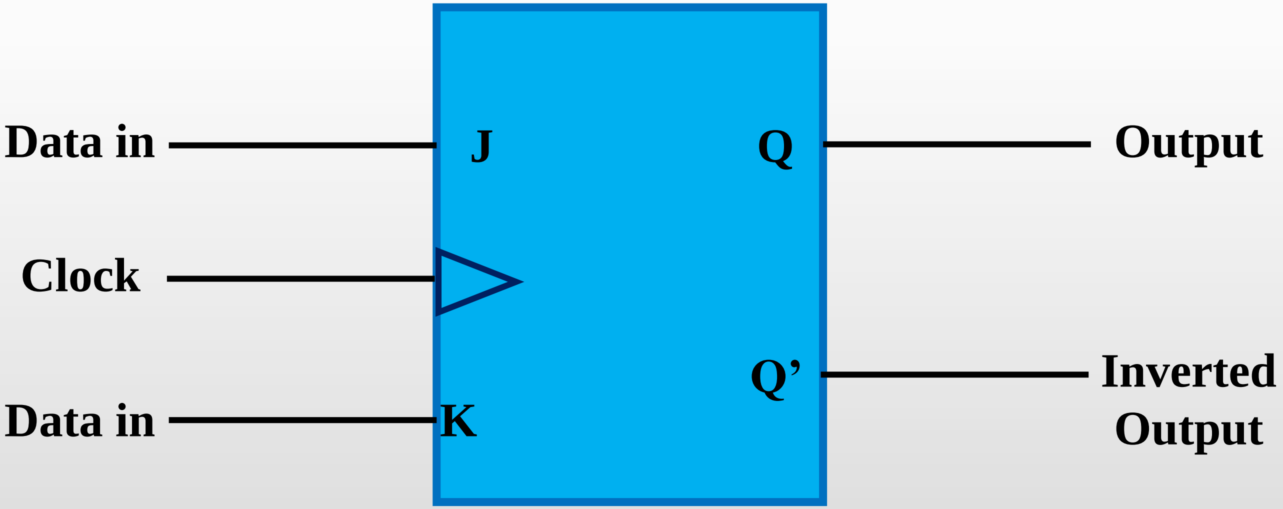
JK Flip-flop

- The **JK flip-flop** is a refinement of SR flip-flop in that the indeterminate condition of SR flip-flop is defined in the JK type.

Truth table

J	K	Q(t+1)	
0	0	Q(t)	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	Q'(t)	Complement

Block Diagram



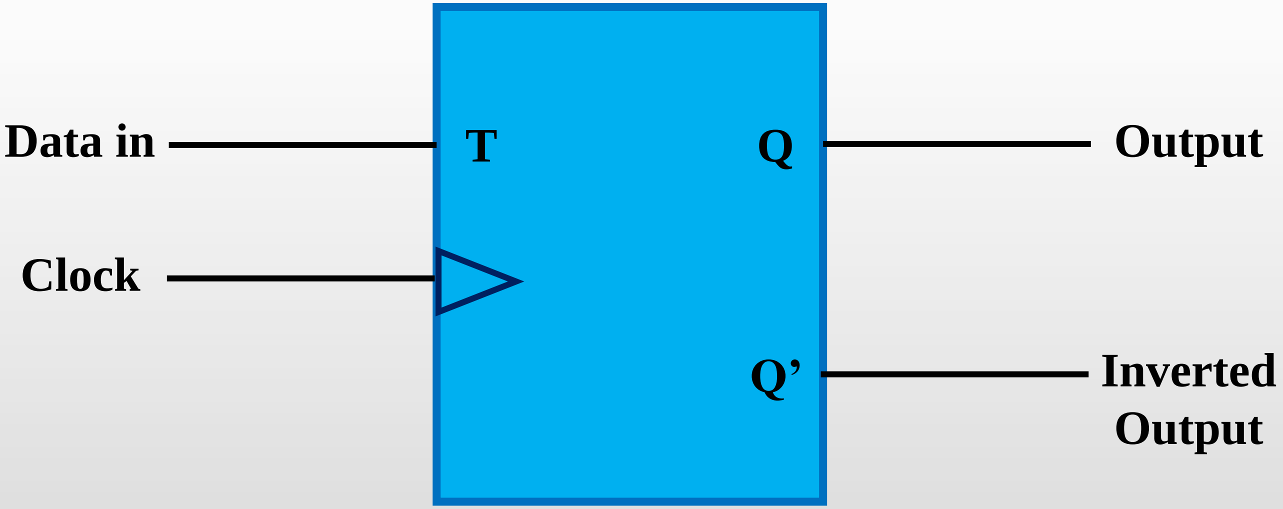
T Flip-flop

- The **T(toggle) flip-flop** is obtained from JK flip-flop in which inputs J and K are connected to provide a single input designated by T.

Truth table

T	Q(t+1)	
0	Q(t)	No change
1	Q'(t)	Complement

Block Diagram



Sequential Circuits

Sequential Circuits

- A **sequential circuit** is an interconnection of flip flops and gates.

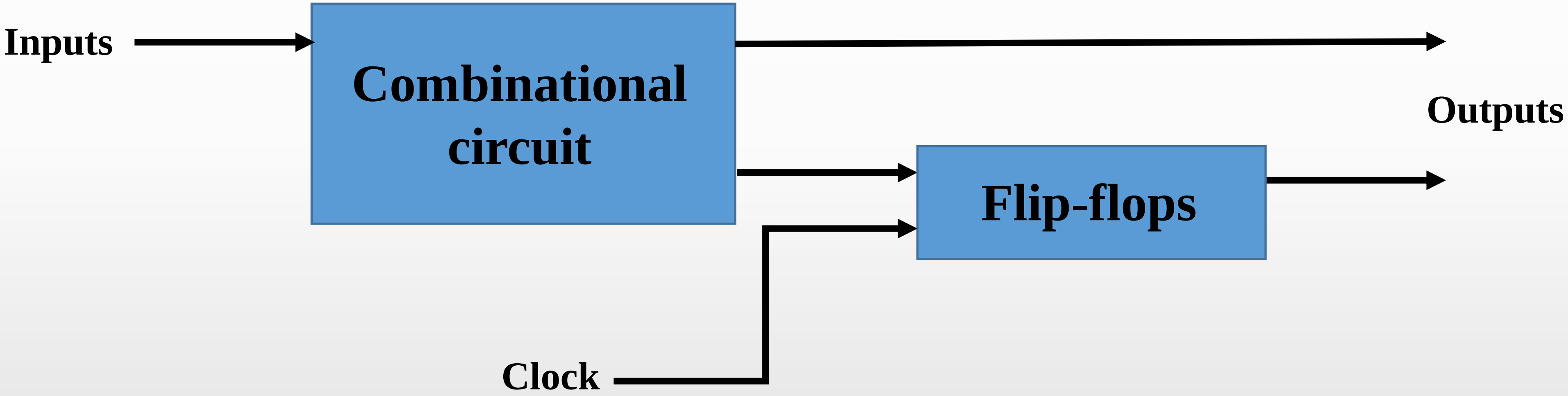


Fig: Block Diagram of a Sequential Circuit

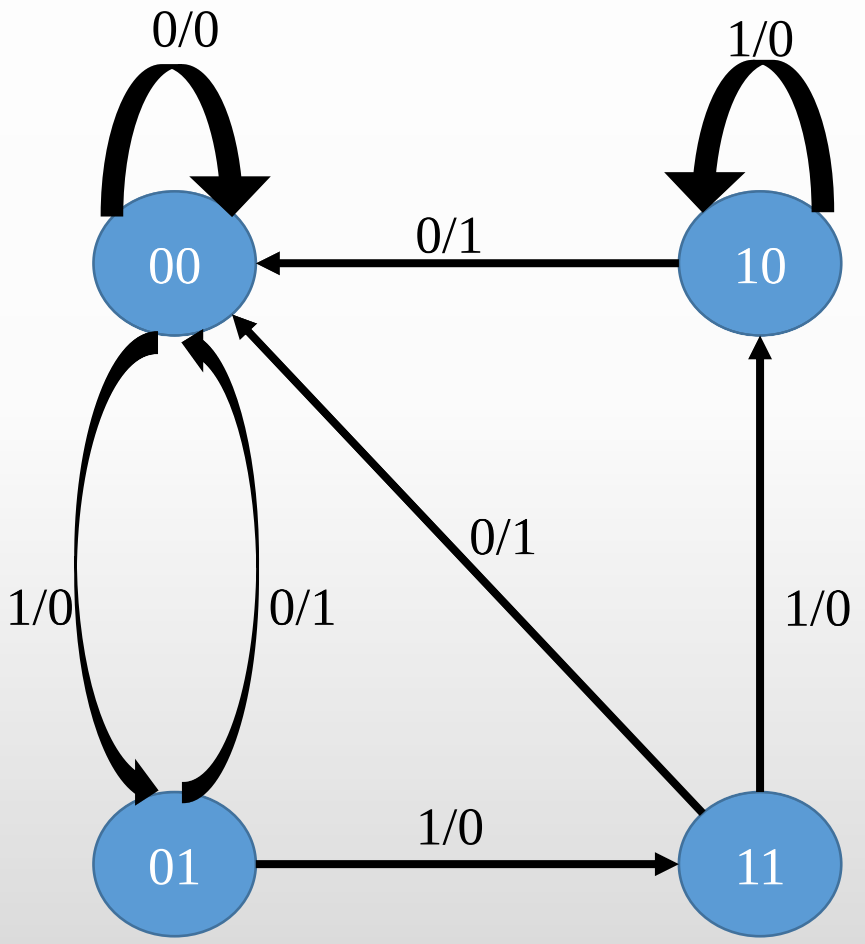
State Table and Diagram of Sequential Circuits

- The behavior of a sequential circuit is determined from the inputs, outputs, and the state of its flip flops.
- The state table of this circuit consists of four sections, present state, input, next state, and output.
- The present-state section shows the states of flip flops A and B at any given time ***t***.
- The input section gives the values of ***x*** for each possible present state.
- The next-state section shows the states of the flip flops one clock period later at time ***t+1***.
- The output section gives the value of ***y*** for each present state and input condition.
- The information available in the state table can be presented graphically in a state diagram.
- In this type of diagram, a state is represented by a circle, and the transition between states is indicated by directed lines connecting the circles.

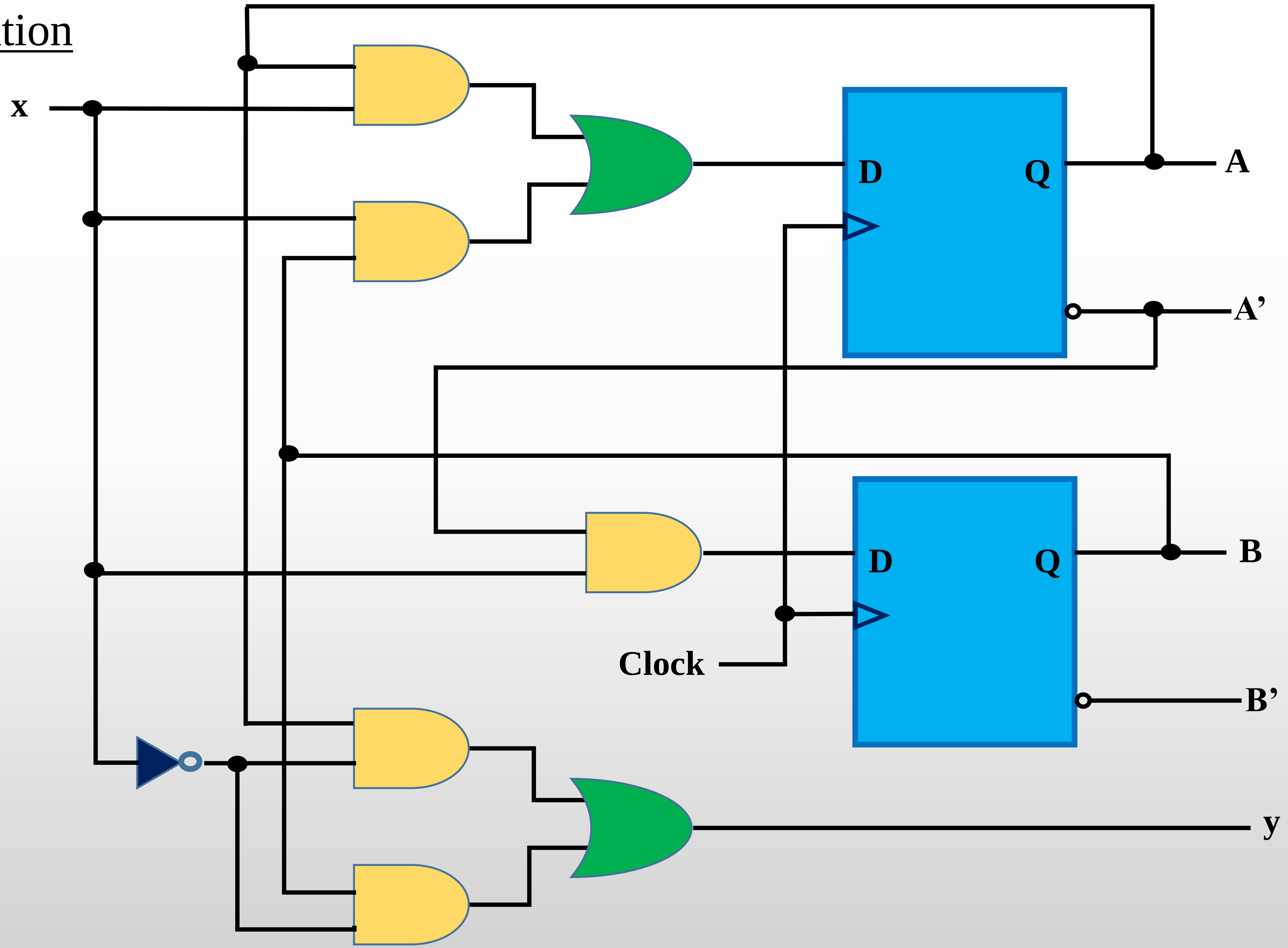
State Table

Present state		Input	Next state		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	`1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

State Diagram



Gate level illustration



Integrated Circuits

Integrated Circuit

An **integrated circuit** is a small silicon semiconductor crystal, called a chip, containing the electronic components for the digital gates.

- The followings are types of integrated circuits.
  - SSI (Small Scale Integration)
    - ❖ less than 10 gates *(eg. Logic gates)*
  - MSI (Medium Scale Integration)
    - ❖ between 10 and 200 gates *(eg. Flip Flops/ Adders/ Counters/…)*
  - LSI (Large Scale Integration)
    - ❖ between 200 to a few thousands of gates *(eg. Programmable Logic Device)*
  - VLSI (Very Large Scale Integration)
    - ❖ between 10,000 to 100,000 gates *(eg. Complex Programmable Logic Device)*
  - ULSI (Ultra Large-Scale Integration)
    - ❖ more than 100,000 gates *(eg. Microprocessors)*

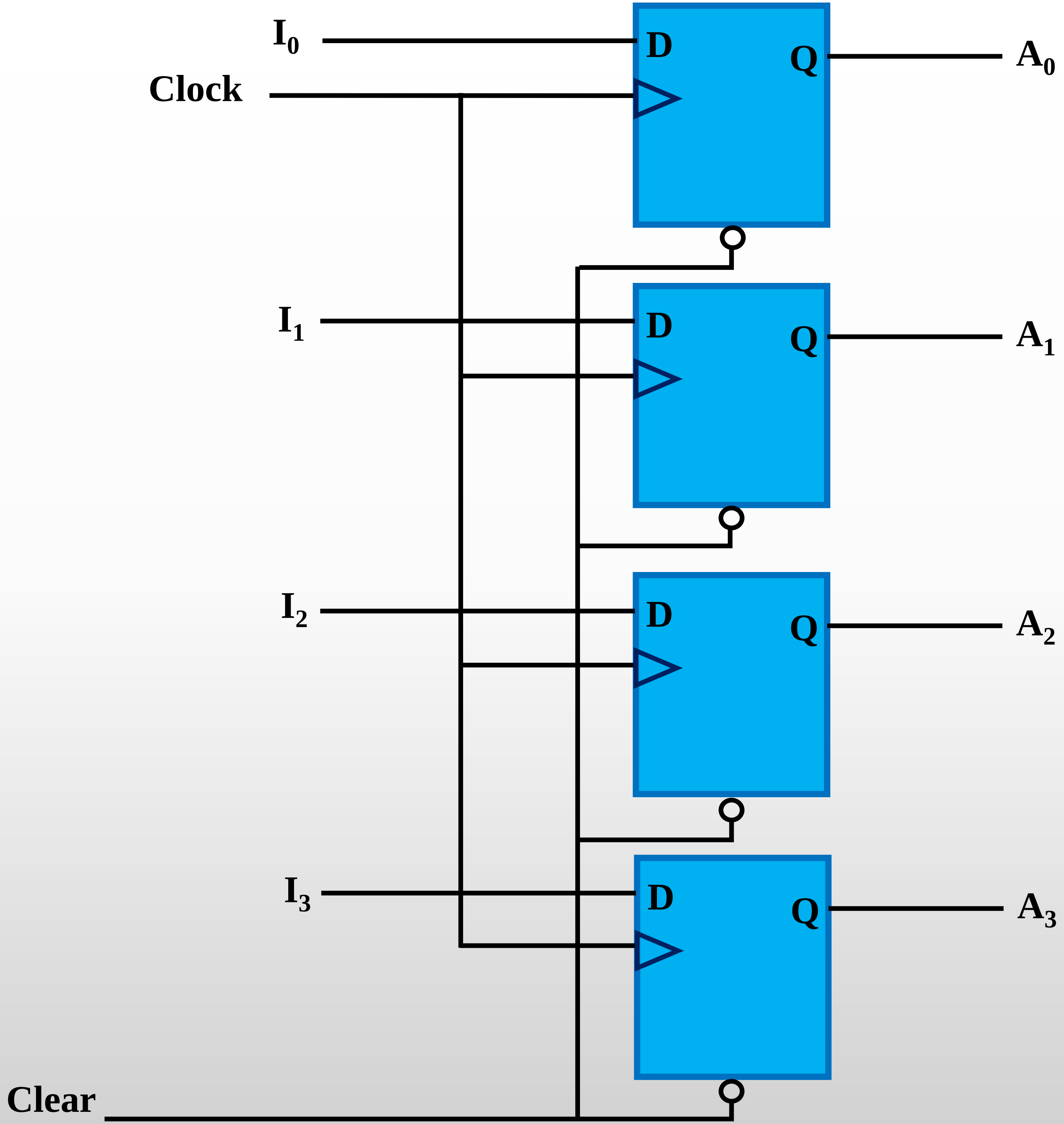
Registers

Registers

- A **register** is a group of flip flops with each flip flop capable of storing one bit of information.
- An ***n***-bit register has a group of ***n*** flip flops and capable of storing any binary information of n-bits.
- In addition to flip flops, a register has combinational gates that perform certain data-processing task.
- The transfer of new information into the register is referred to as **loading** the register.

For 4-bit register:

Block Diagram

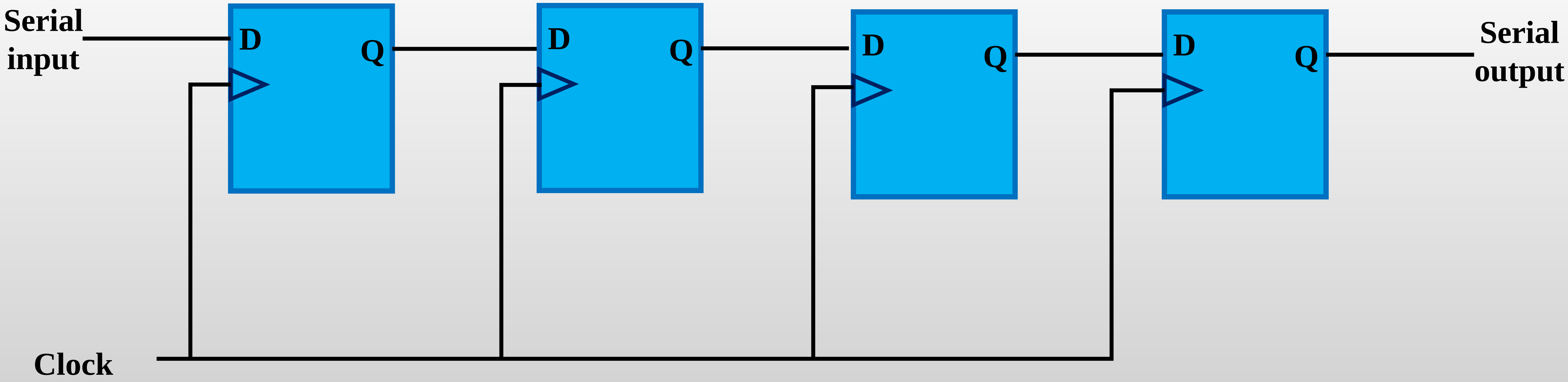


Shift Registers

- A register capable of shifting its binary information in one or both directions is called a **shift register**.
- The **serial input** determines what goes into the leftmost position during the shift.
- The **serial output** is taken from the output of the rightmost flip flop.

For 4-bit register:

<u>Block Diagram</u>



Capabilities of General Shift Register

- The most general shift register has all the capabilities listed below.

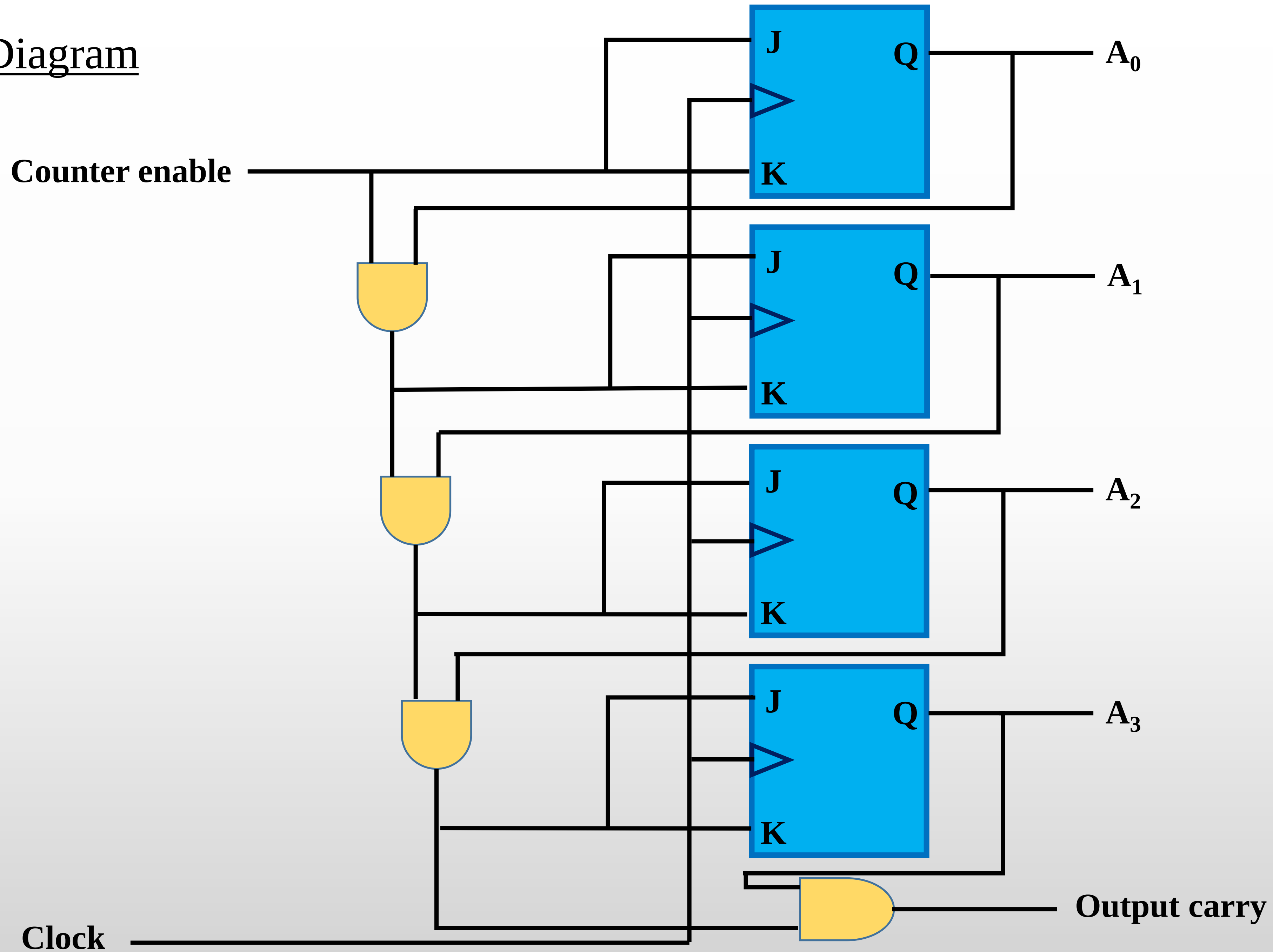
  1. An input for clock pulses to synchronize all operations.
  2. A shift-right operation and serial input line associated with the shift-right.
  3. A shift-left operation and serial input line associated with the shift-left.
  4. A parallel load operation and n input lines associated with the parallel transfer.
  5. ***n*** parallel output lines.
  6. A control state that leaves the information in the register unchanged even though clock pulses are applied continuously.

Binary Counter

Binary Counter

- A **counter** is a register that goes through the predetermined sequence of the stages upon the application of input pulses.
- A **binary counter** is a counter that follows the binary number sequence.
- In addition to flip flops, a register has combinational gates that perform certain data-processing task.
- The transfer of new information into the register is referred to as **loading** the register.

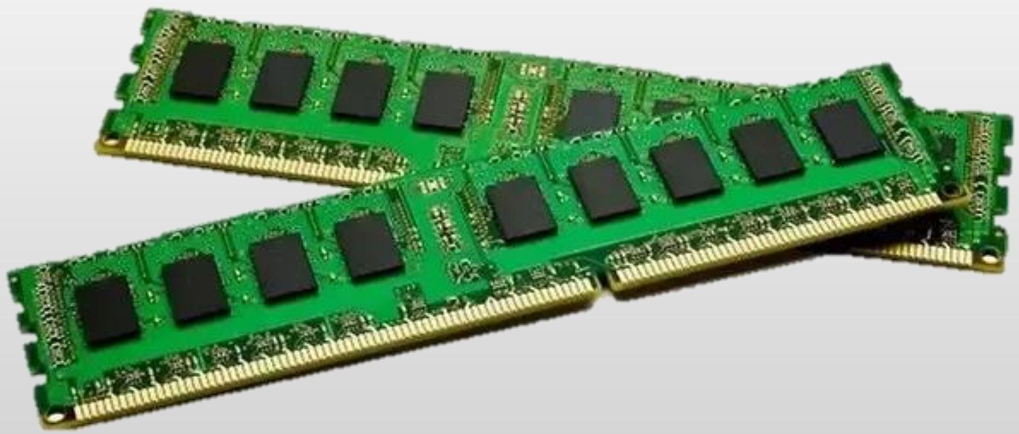
Block Diagram



Memory Unit

Memory Unit

- A **memory unit** is a collection of storage cells together with associated circuits needed to transfer information in and out of storage.
- The memory stores binary information in group of bits called **words**.
- A word in memory is an entity of bits that in and out of storage as a unit.
- The internal structure of memory unit is specified by the number of words it contains and the number of bits in each word.
- Two major types of memories are used in computer system called random access memory (RAM) and read only memory (ROM).



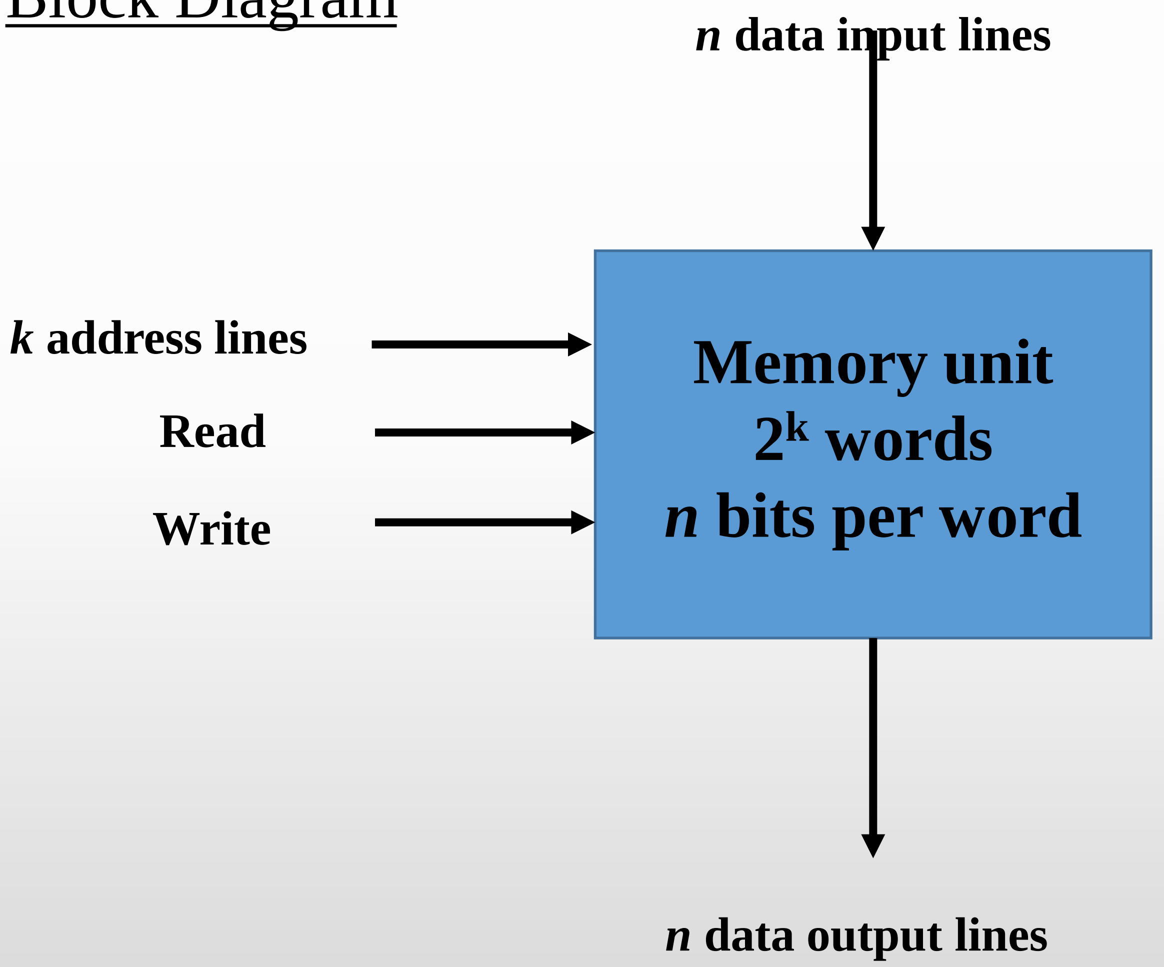
Random Access Memory (RAM)

- In **Random Access Memory (RAM)**, the memory cells can be accessed for information transfer from any desired random location.
- The two operations that a random access memory can perform are read and write operations.
- The write signal specifies a transfer-in operation and the read signal specifies a transfer-out operation.
- The steps of transferring a new word to be stored into memory are as follows.
  - Apply the binary address of the desired word into the address lines.
  - Apply the data bits that must be stored in memory into the data input lines.
  - Activate the write input.
- The steps of transferring a stored word out of memory are as follows.
  - Apply the binary address of the desired word into the address lines.
  - Activate the read input.

Some Features of RAM

- It is used in computer's regular operations, after loading the OS.
- It is a fast process for writing data.
- It is a type of volatile memory, i.e. stored data is lost when powering off.
- It stores quite a lot of data.
- Two types of RAM: Static RAM (SRAM) and Dynamic RAM (DRAM)

Block Diagram



Static RAM (SRAM)

- Static RAM is a type of semiconductor random access memory in which each bit of data is stored in the flip-flop.
- It is a volatile memory that requires power to maintain the stored data and information.
- It provides faster access to data and is more expensive than DRAM.
- It is mainly used for the cache memory in the computer system.

Dynamic RAM (DRAM)

- Dynamic RAM is a type of semiconductor random access memory in which each bit of data is stored in a memory cell.
- It is based on the metal-oxide-semiconductor technology.
- It is more complicated and time-consuming than SRAM, but it is more widely used.
- It is mainly used for the main memory and graphics cards (graphics memory) in the computer system.

Read Only Memory (ROM)

- A **Read Only Memory** is a memory unit that performs the read operation only; it does not have a write capability.
- A ROM is restricted to reading words that are permanently stored within the unit.
- ROM is used for storing fixed programs that are not to be altered and the tables of contents that are not to be changed.
- There are different types of ROM used in today computer system.
  - Programmable Read Only Memory (PROM)
  - Erasable PROM (EPROM)
  - Electrically EPROM (EEPROM)

Thank You

Lecture for Next Week

- **Different types of data presentation**
  - **Data types**
  - **Fixed-point presentation**
  - **Floating-point presentation**
  - **Binary codes**
  - **Error detection codes**