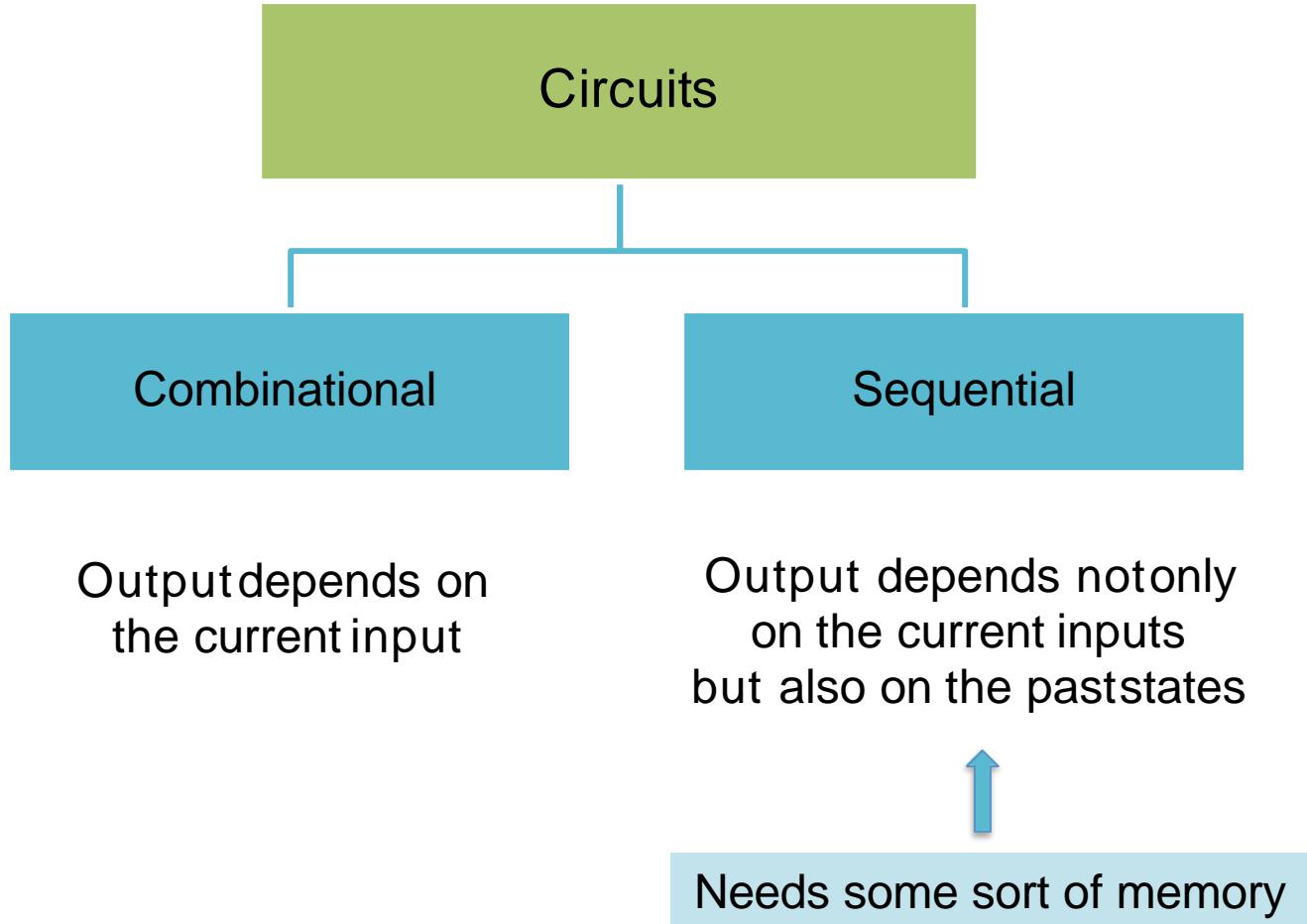


Sequential Circuits

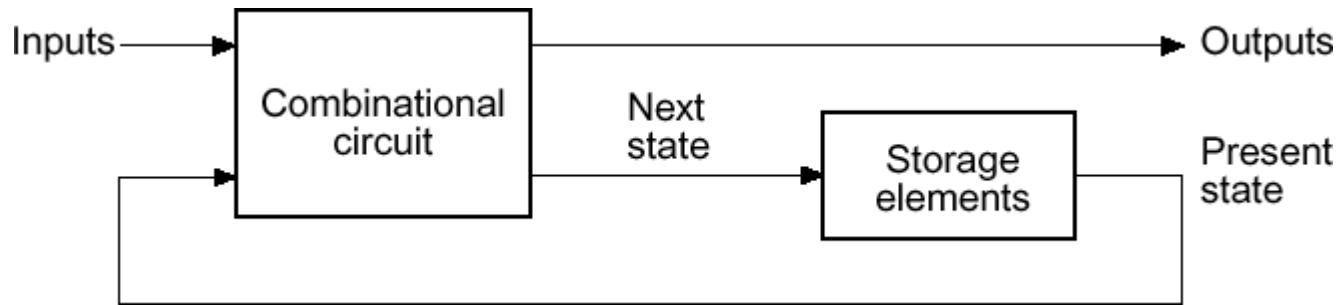
In this lecture, we will study

- i. Flip-flops
- ii. Sequential circuits (decoders and encoders, multiplexers, registers, counters, and memory - RAM, ROM)

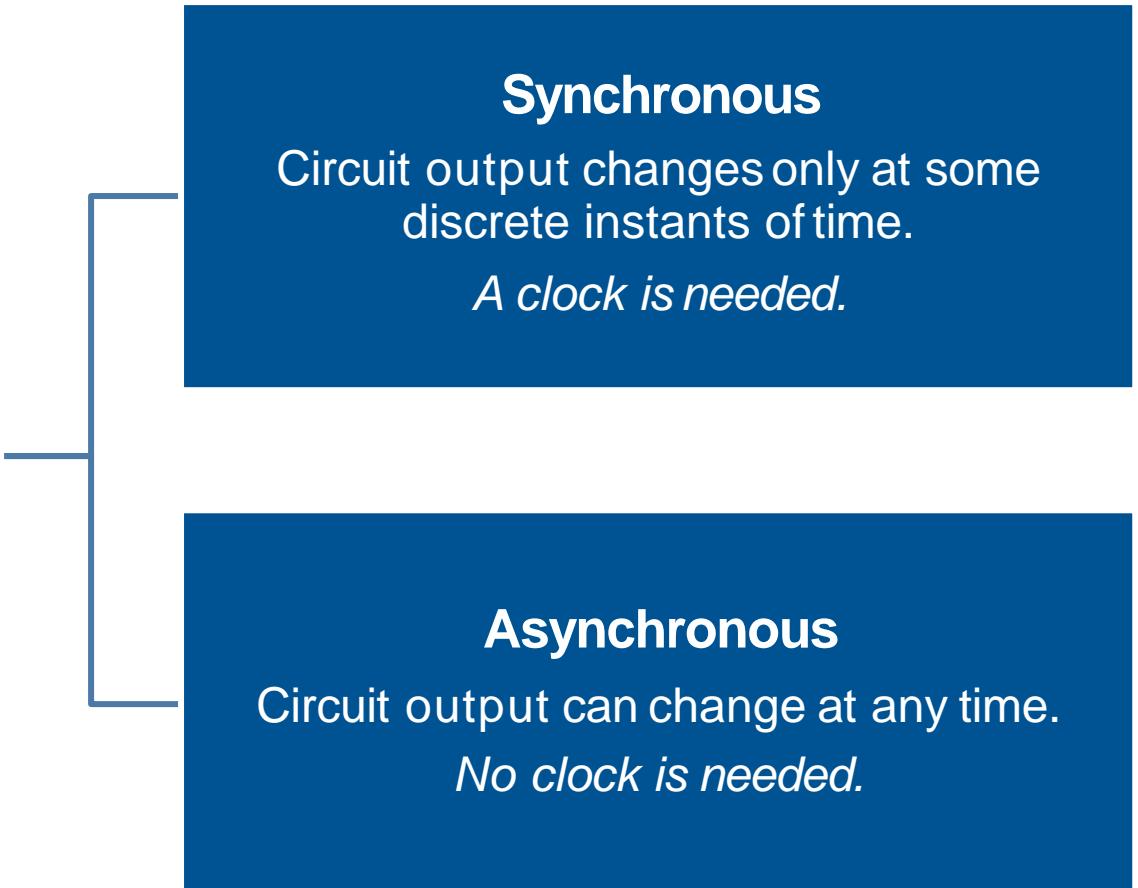
Sequential Circuits



Sequential Circuit



Sequential Circuits

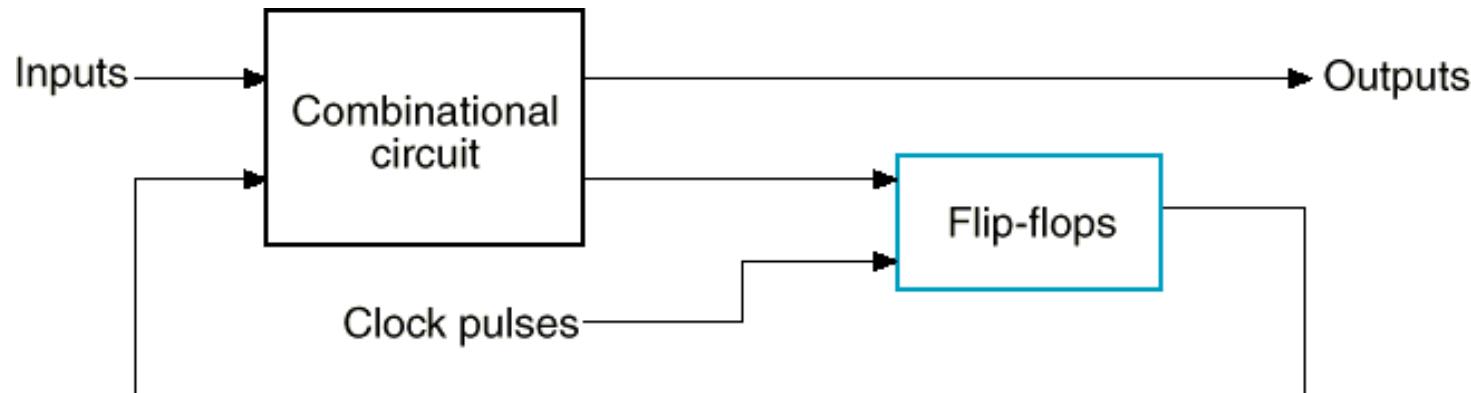


Circuits

Combinational circuits
are made up of GATES

Sequential circuits
are made up of
GATES and FLIP-FLOPS

Synchronous Sequential Circuits with Flip-Flop as State Memory



(a) Block diagram



(b) Timing diagram of clock pulses



Flip-Flops

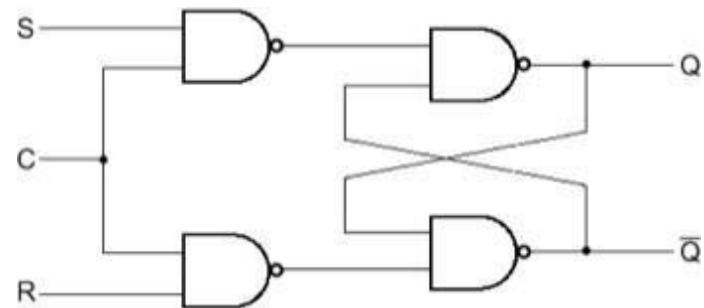
SR(Set-Reset) Flip-Flop aka SR Latch

An SRflip-flop has three inputs:

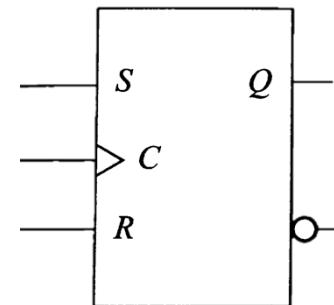
S (set)
R(reset)
C (clock)

It has an output Q, and sometimes has another, complemented output, denoted by a little circle at the other output terminal.

(a) Logic Diagram



(b) Graphical Symbol



Operation of SRFlip-Flop

If there is no signal at the clock input, the output of the circuit cannot change irrespective of the input.

When clock signal changes from 0 to 1, the following states are possible:

S	R	$Q(t+1)$	
0	0	$Q(t)$	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	?	Indeterminate

Characteristic Table

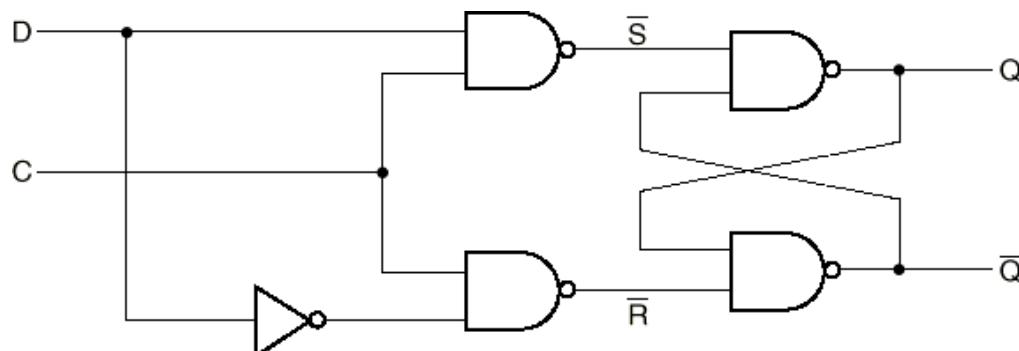
Function Table for SR Flip-Flop

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

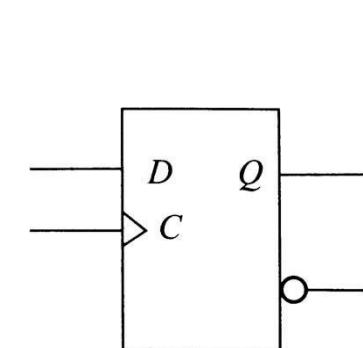
D (Data) Flip-Flop: A slight modification of the SR flip-flop

An SRflip-flop can be converted to a D flip-flop by inserting an inverter between S and R, and assigning the symbol D to the single input.

The inverter makes sure that the inputs are never 1 simultaneously, thereby eliminating the ‘indeterminate state’ that could occur in case of an SRflip-flop.



(a) Logic Diagram



(b) Graphical Symbol

Operation of D Flip-Flop

If there is no signal at the clock input, the output of the circuit cannot change irrespective of the input.

When clock signal changes from 0 to 1, the following states are possible:

D	$Q(t + 1)$	
0	0	Clear to 0
1	1	Set to 1

Characteristic Table

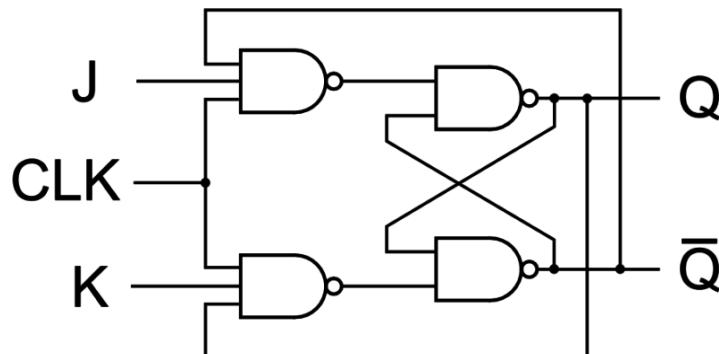
Function Table for D Flip-Flop

C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

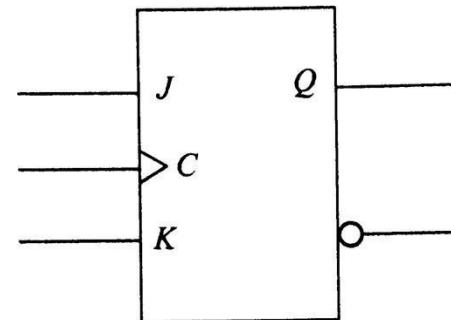
JK Flip-Flop: A refinement of the SR flip-flop

Named after its inventor, Jack Kilby.

The inputs J and K behave like the inputs S and R, except when both J and K are simultaneously 1, in which case, a clock transition switches the output of the flip-flop to its complement state.



(a) Logic Diagram



(b) Graphical Symbol

Operation of JK Flip-Flop

If there is no signal at the clock input, the output of the circuit cannot change irrespective of the input.

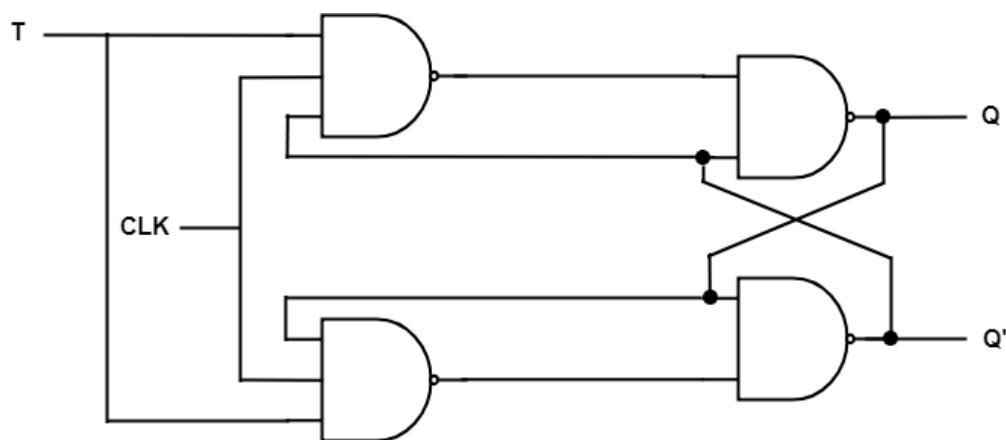
When clock signal changes from 0 to 1, the following states are possible:

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

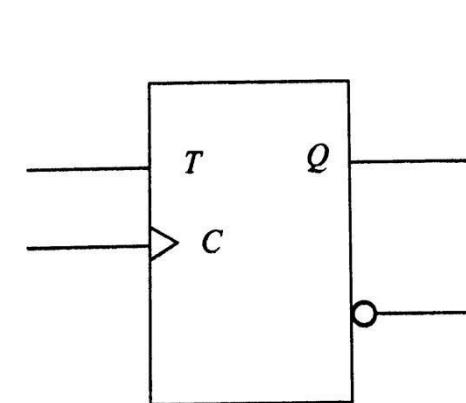
Characteristic Table

T (Toggle) Flip-Flop: A modification of the JKflip-flop

A T flip-flop is obtained from a JK flip-flop when inputs J and K are connected to provide a single input (T).



(a) Logic Diagram



(b) Graphical Symbol

Operation of T Flip-Flop

When $T=0$, a clock transition does not change the state of the flip-flop.

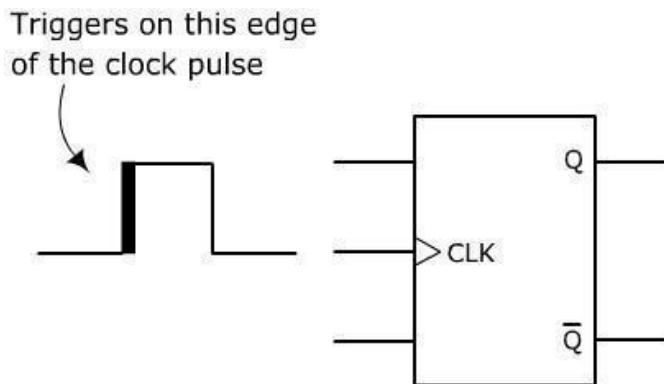
When $T=1$, a clock transition complements the state of the flip-flop.

K	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

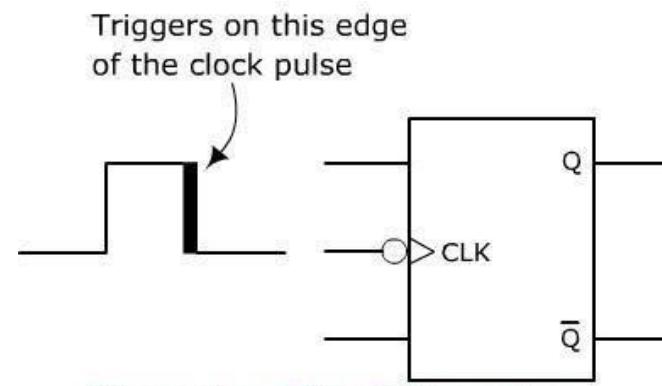
Characteristic Table

Edge-Triggered Flip-Flops

An edge-triggered flip-flop changes states either at the positive/rising edge or at the negative/falling edge of the clock pulse.



Positive Edge-Triggered Flip-Flop



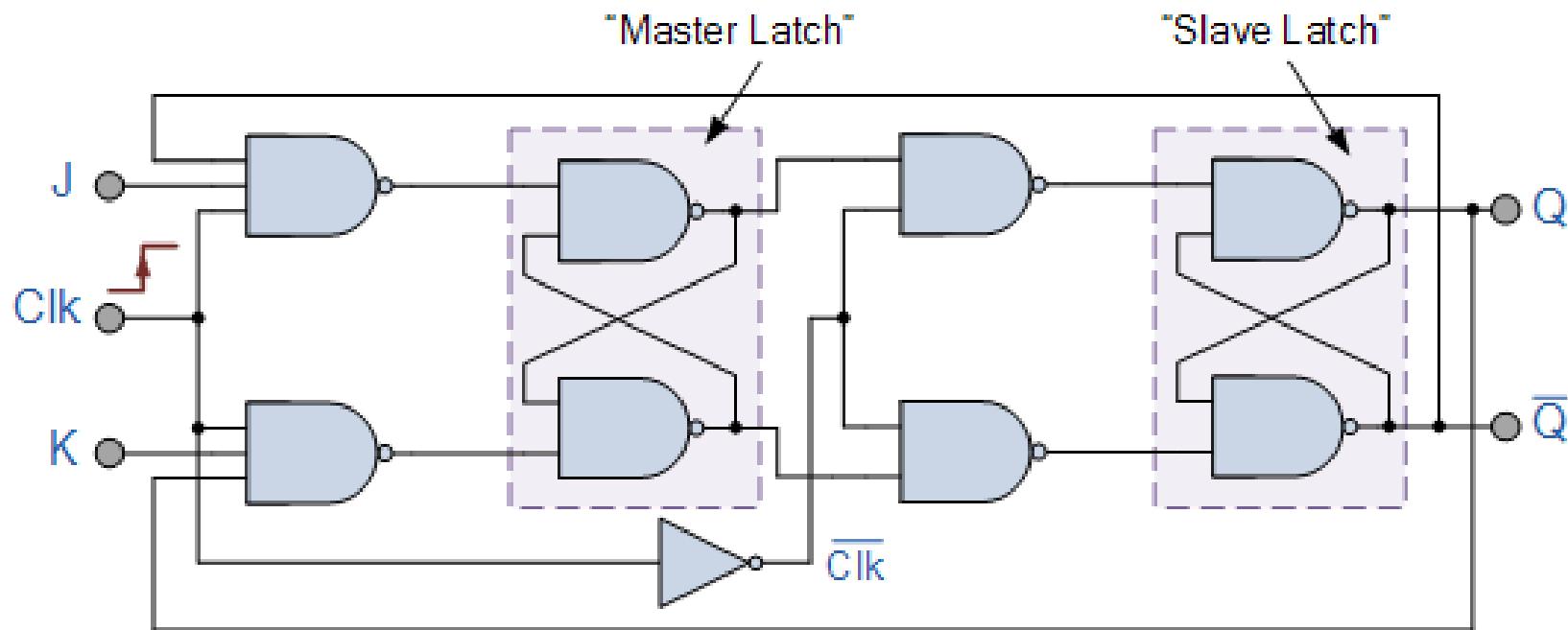
Negative Edge-Triggered Flip-Flop

Master-Slave Flip-Flops

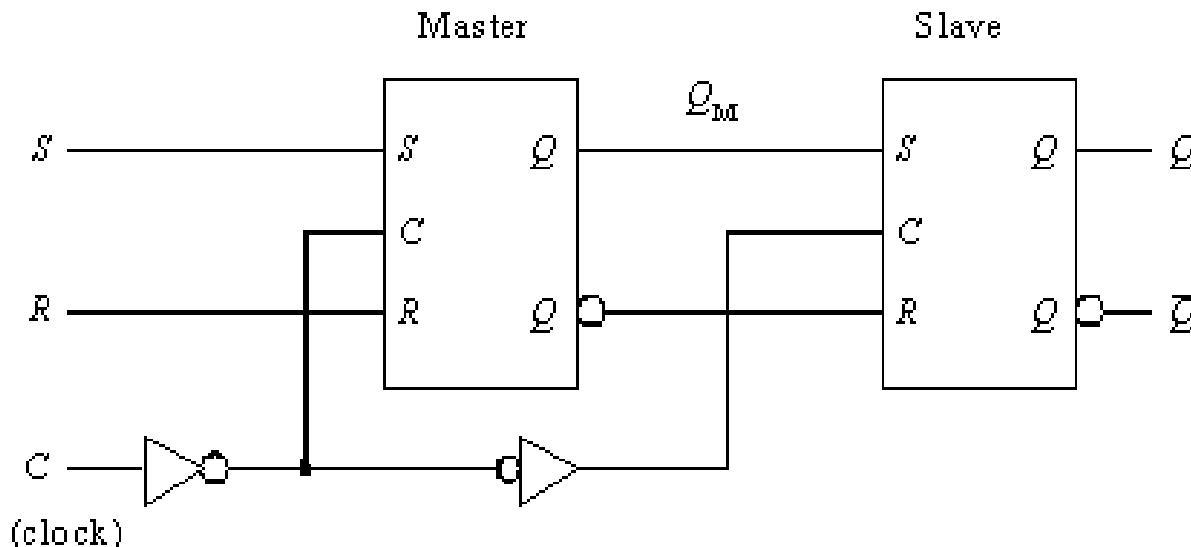
Some circuits use two flip-flops; one is the MASTER and the other is the SLAVE.

The ‘master’ flip-flop responds to the POSITIVE level of the clock, while the ‘slave’ flip-flop responds to the NEGATIVE level of the clock.

Master-Slave JK Flip-Flop



Master-Slave SR Flip-Flop



Excitation Tables

Characteristic tables specify the *next state* of the flip-flop WHEN THE INPUTS AND THE PRESENT STATE ARE KNOWN.

But what about when we need to find out WHICH INPUT CONDITIONS WILL GENERATE THE DESIRED TRANSITION?

For that, we need Excitation Tables.

Returning to Sequential Circuits...

State Diagrams

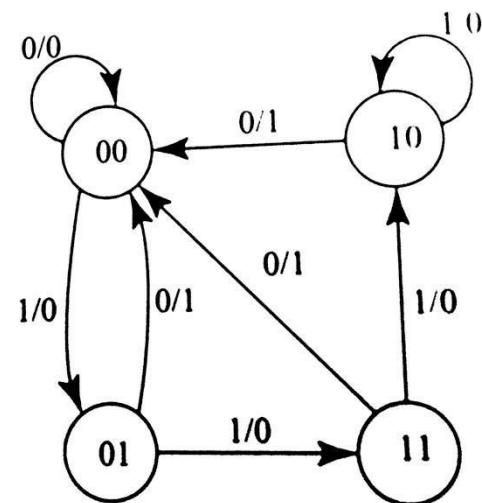
The information available in a state table can be represented graphically using a 'State Diagram'.

In a state diagram:

- each STATE is represented by a CIRCLE
- each TRANSITION between states is represented by DIRECTED LINES connecting the states

State Diagrams

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



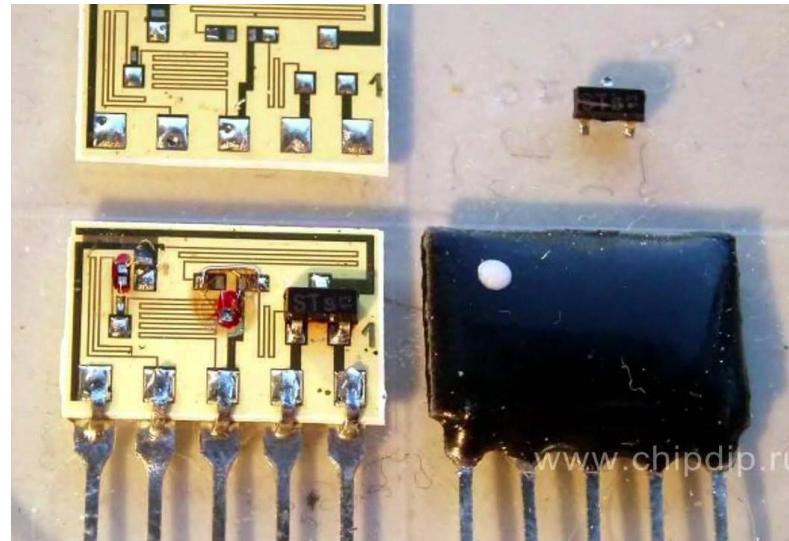
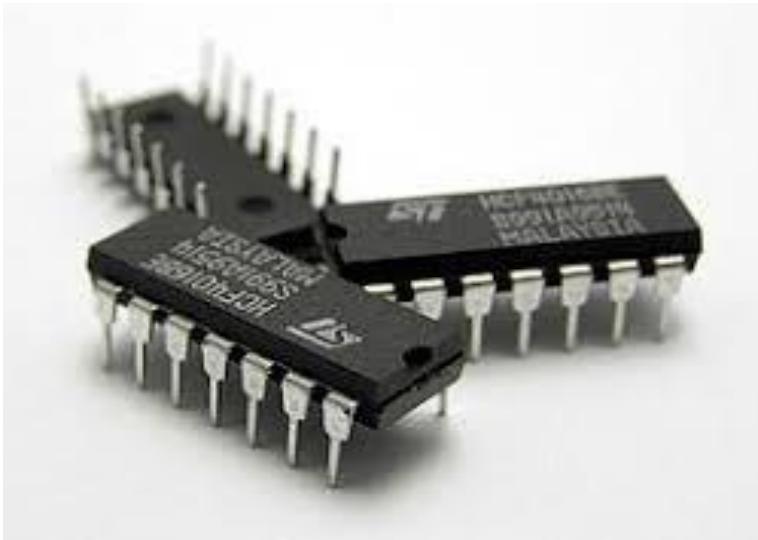
Integrated Circuits

Integrated Circuits

An Integrated Circuit (IC) is a small silicon semiconductor chip that contains electrical components for the digital gates.

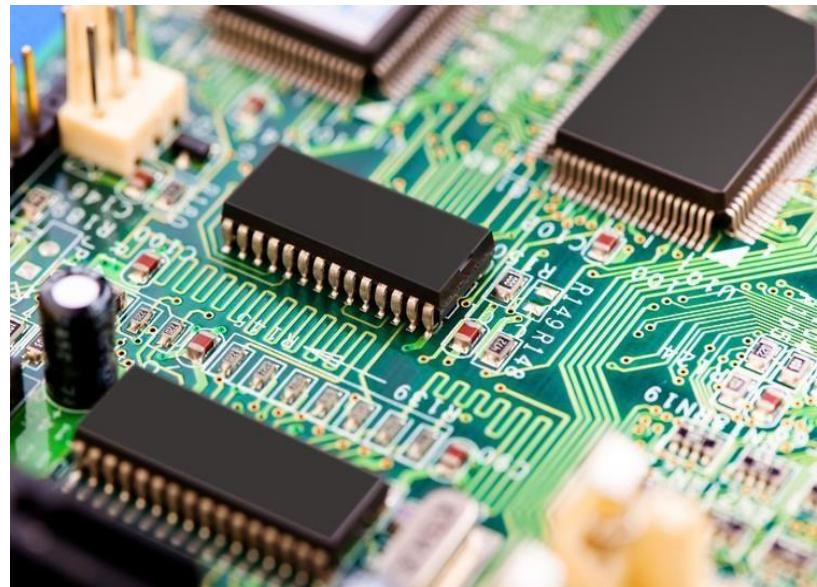
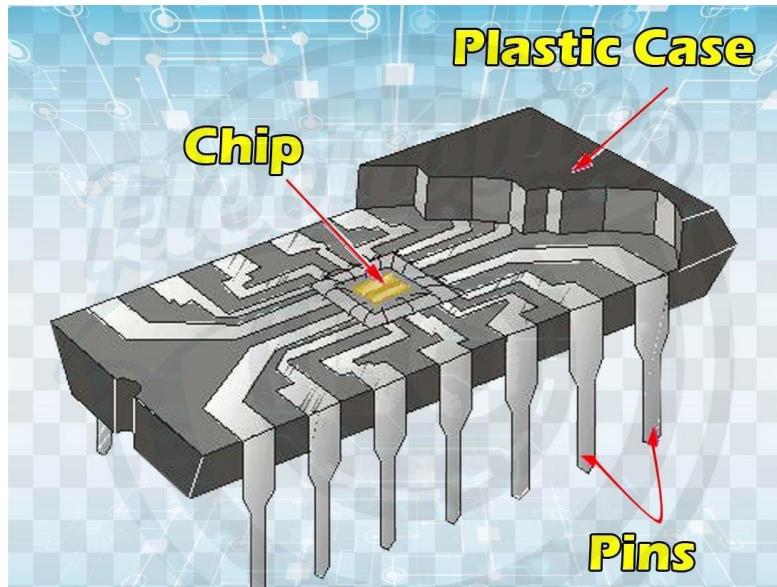
Various gates are interconnected inside the chip to form the required circuit. The chip is then mounted in a plastic or ceramic container, and the connections are welded by thin gold wires to external pins to form the integrated circuits.

Integrated Circuits



www.chipdip.ru

ICs



Types of Integration Devices

Small Scale Integration (SSI)

less than 10 gates in a single package

Medium Scale Integration (MSI)

10-200 gates in a single package

Large Scale Integration (LSI)

between 200 to a few thousand gates
in a single package (*processors, microchips*)

Very Large Scale Integration (VLSI)

thousands of gates in a single package
(*large memory arrays, complex microcomputer chips*)

[Skip to Next Component](#)

Logic Families of Integrated Circuits

TTL?

ECL?

MOS?

CMOS?

[Skip to Next Component](#)

Decoders

(Binary) Decoders

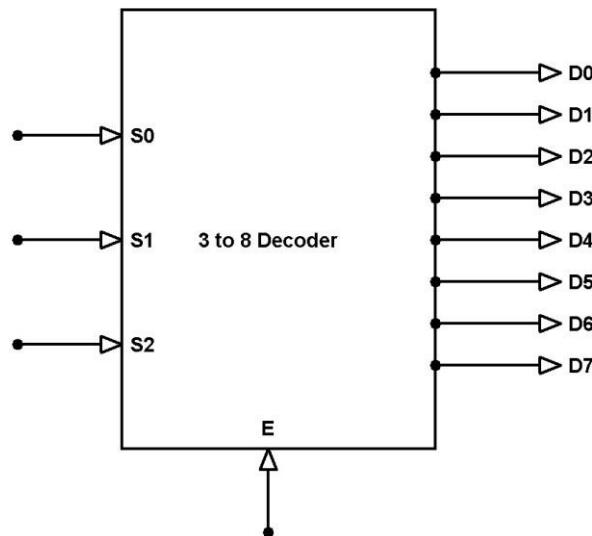
Information is represented in digital computers using binary codes; a code of n bits can represent 2^n distinct elements of information.



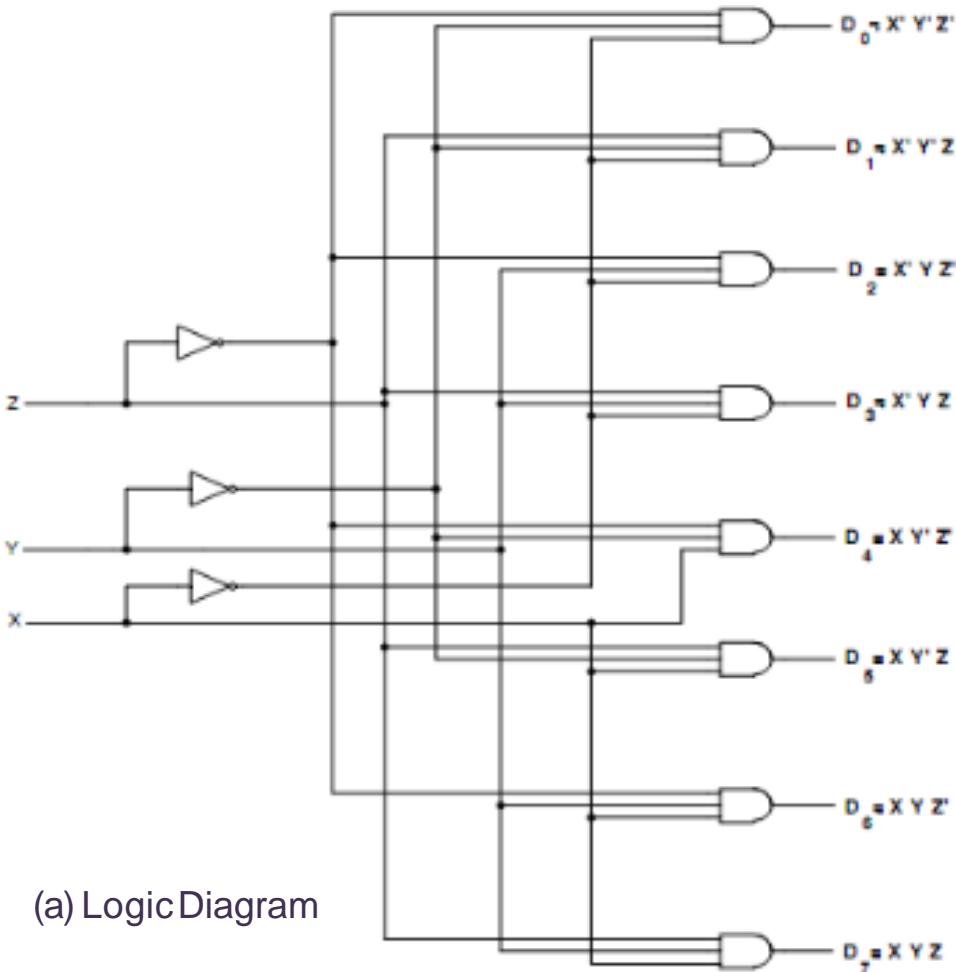
A binary decoder is a combinational circuit that converts binary information from n coded inputs to a maximum of 2^n unique outputs.

n-to-m Decoder

n-to-m line decoders (where $m \leq 2^n$) have n inputs and they generate 2^n (or fewer) outputs.



n-to-m Decoder

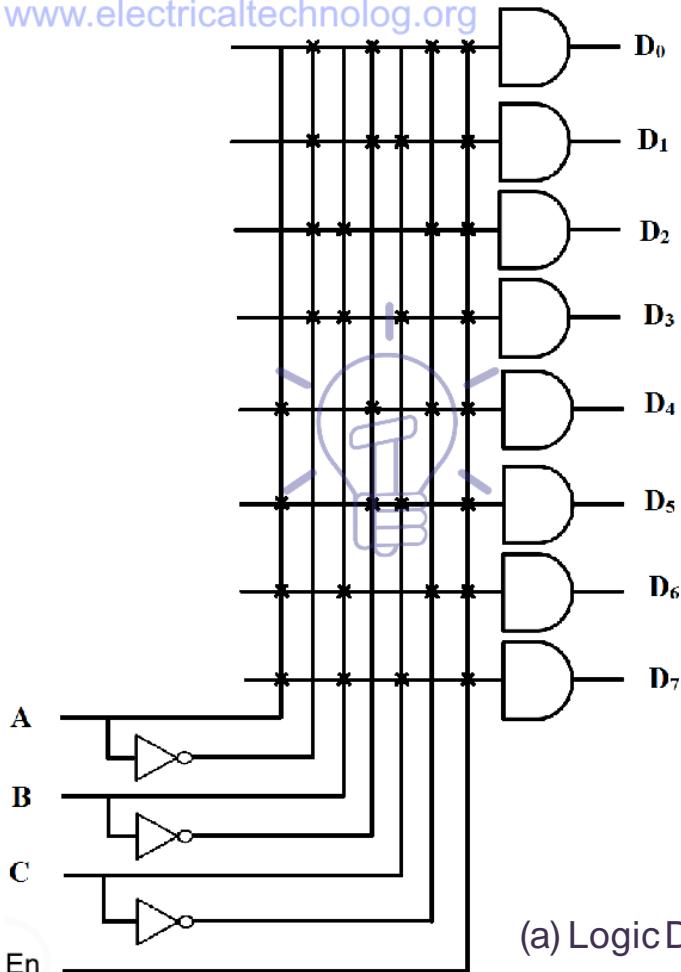


(a) Logic Diagram

Inputs	Outputs							
	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

(b) Truth Table

n-to-m Decoder with 'enable'

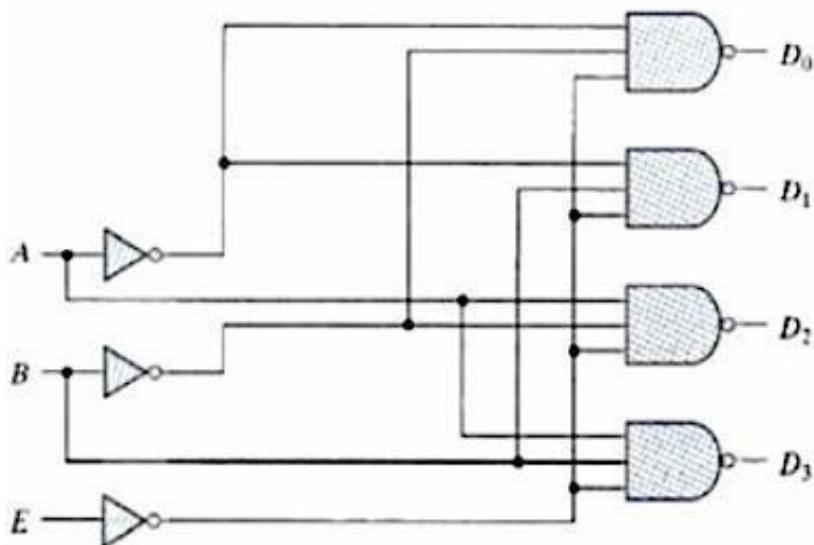


(a) Logic Diagram

INPUT				OUTPUT							
En	A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

(b) Truth Table

2-to-4 Line Decoder with NAND Gates



(a) Logic Diagram

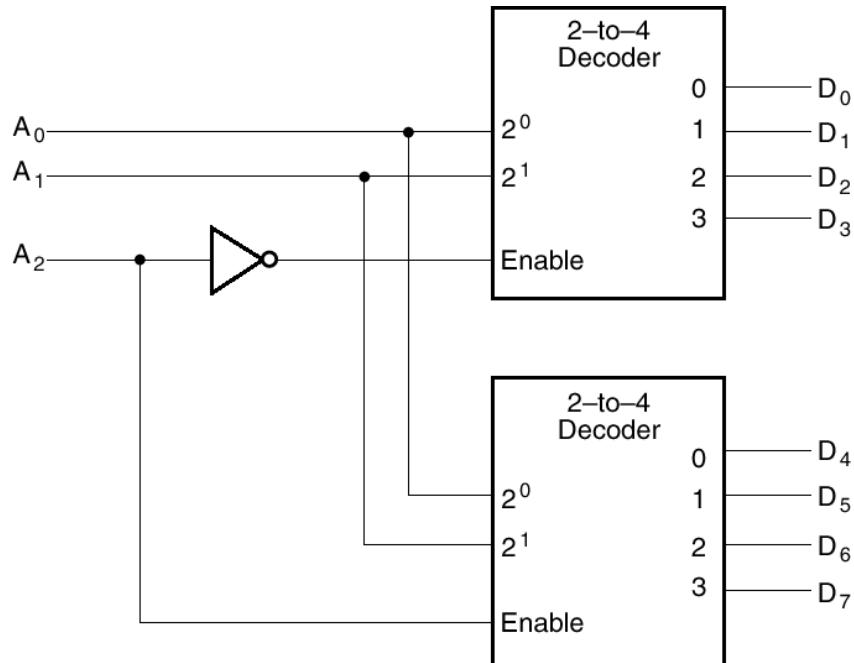
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth Table

Decoder Expansion

What do we do when we need a certain-sized decoder but only smaller decoders are available?

We combine several decoders with enable input to from a larger decoder.



A 3×8 decoder constructed using two 2×4 decoders.

Encoders

(Binary) Encoders

An encoder performs the inverse operation of a decoder.

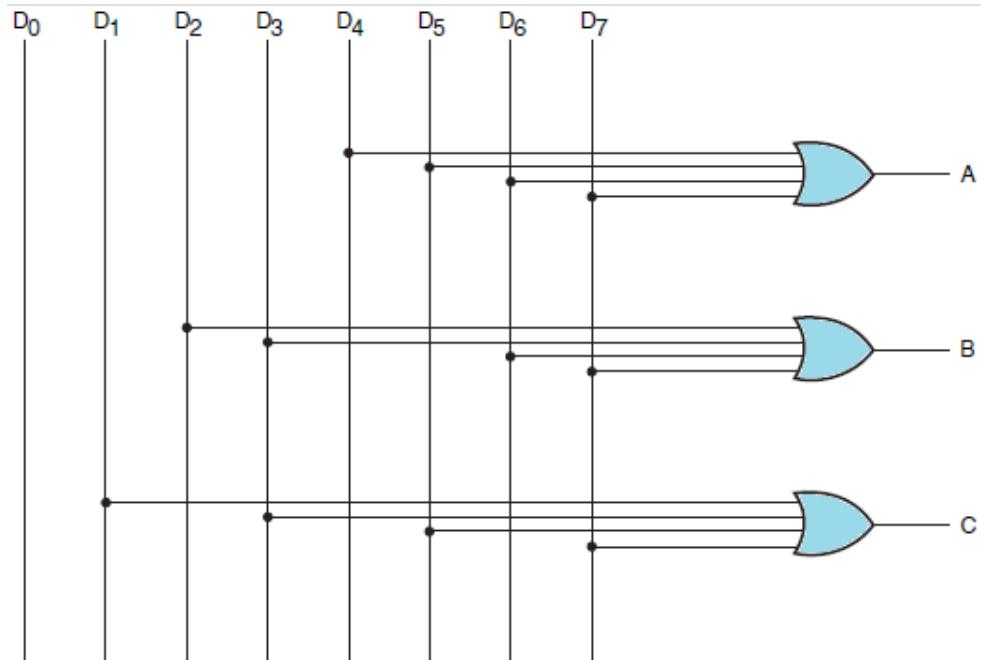


An encoder has 2^n (or less) inputs and n output lines.

Octal to Binary Encoder

This encoder has eight inputs, one for each of the octal digits, and three outputs that generate the corresponding binary number.

Only one of the inputs has the value of 1 at a time; otherwise, the circuit has no meaning.



(a) Logic circuit

Octal to Binary Encoder

$$A = D_4 + D_5 + D_6 + D_7$$

$$B = D_2 + D_3 + D_6 + D_7$$

$$C = D_1 + D_3 + D_5 + D_7$$

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

(b) Truthtable

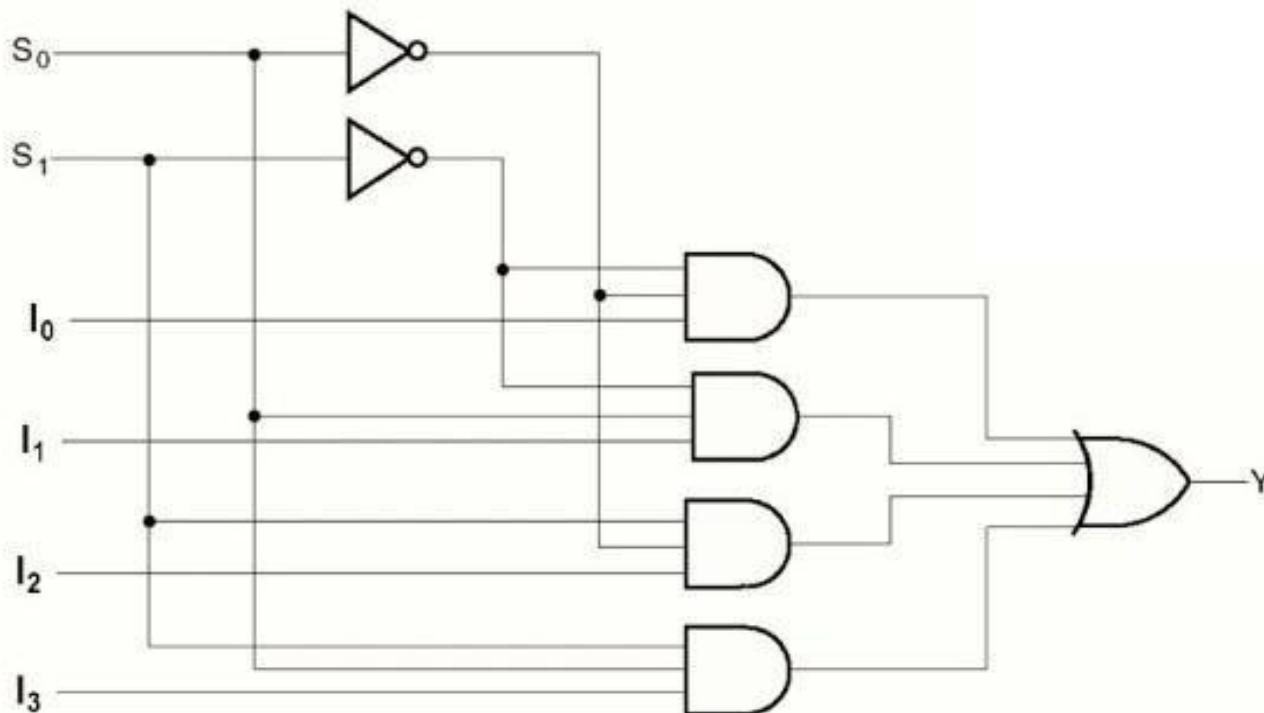
(a) Boolean functions

Multiplexers

Multiplexers

A multiplexer is a combinational circuit that receives binary information from one of 2^n input data lines and directs it to a single output line.

4-to-1 Line Multiplexer

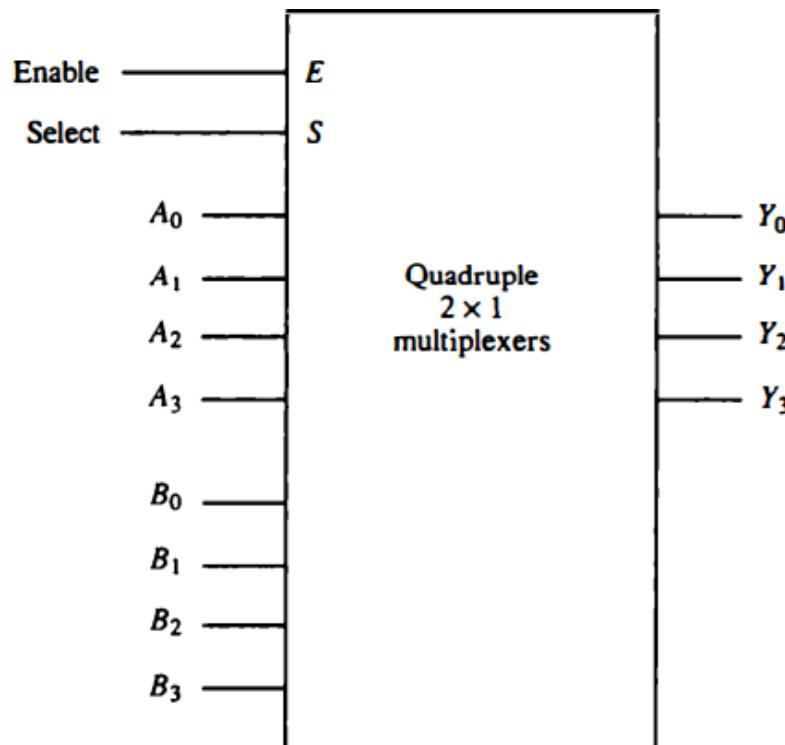


(a) Logic circuit

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

Quadruple 2-to-1 Line Mux



(a) Block Diagram

E	S	Y
0	x	All 0's
1	0	A
1	1	B

(b) Function Table

Registers

Registers

A register is a group of flip-flops, where each flip-flop holds one bit of binary information.

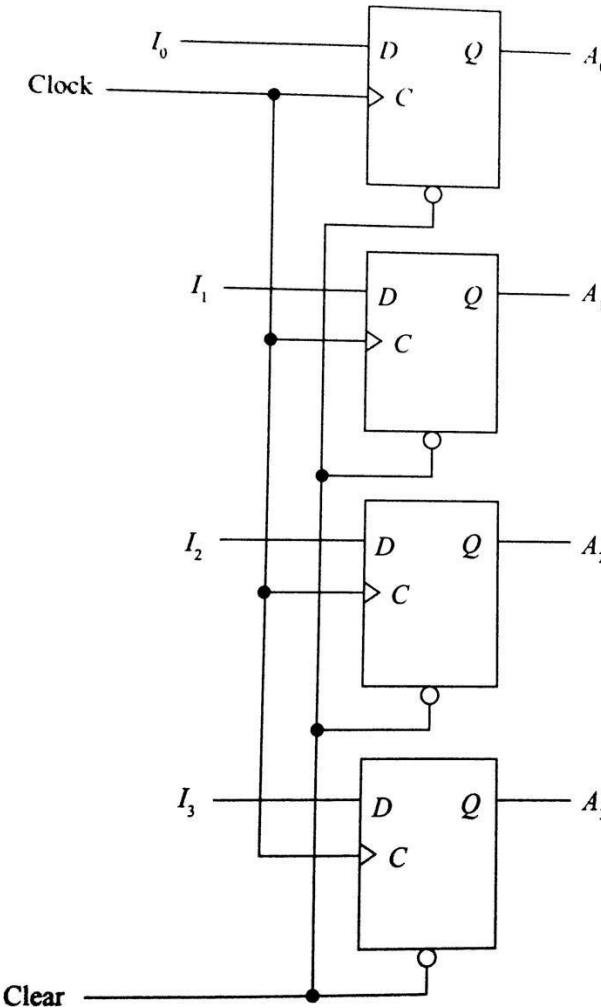
A register may also contain additional combinational gates.

The flip-flops **hold the binary information**, while the gates **control when and how the new information is transferred into the register**.

A Simple 4-Bit Register

This register is constructed with four D flip-flops.
It has no gates.

A common *clock* triggers all the flip-flops, and the binary information available at the four inputs is transferred into the 4-bit register.



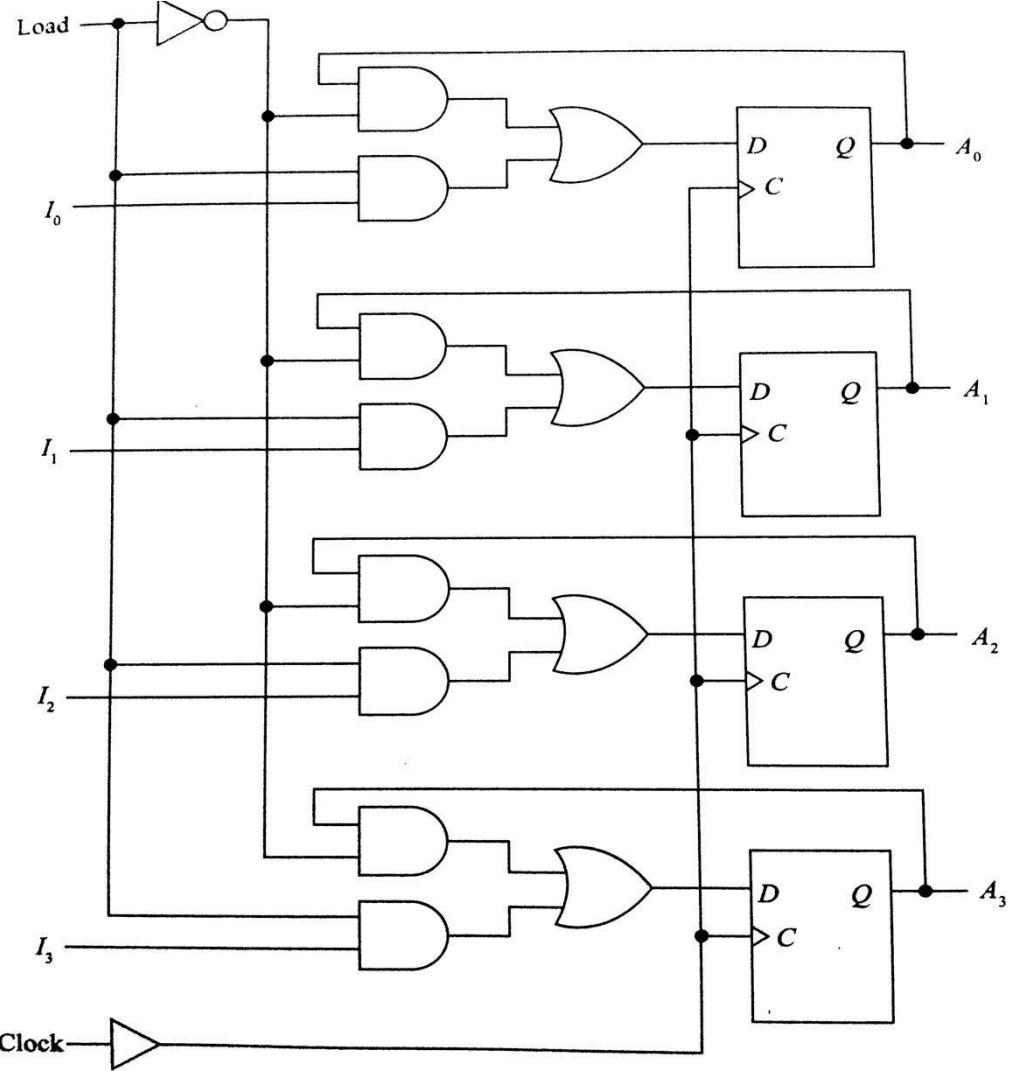
The *clear* input is used to clear the register.

Register Load

The transfer of new information into the register is referred to as the ‘loading of register’.

If all the bits of the register are loaded simultaneously with a common clock pulse, we say that the loading is ‘*done in parallel*’.

4-Bit Register with Parallel Load



Shift Registers

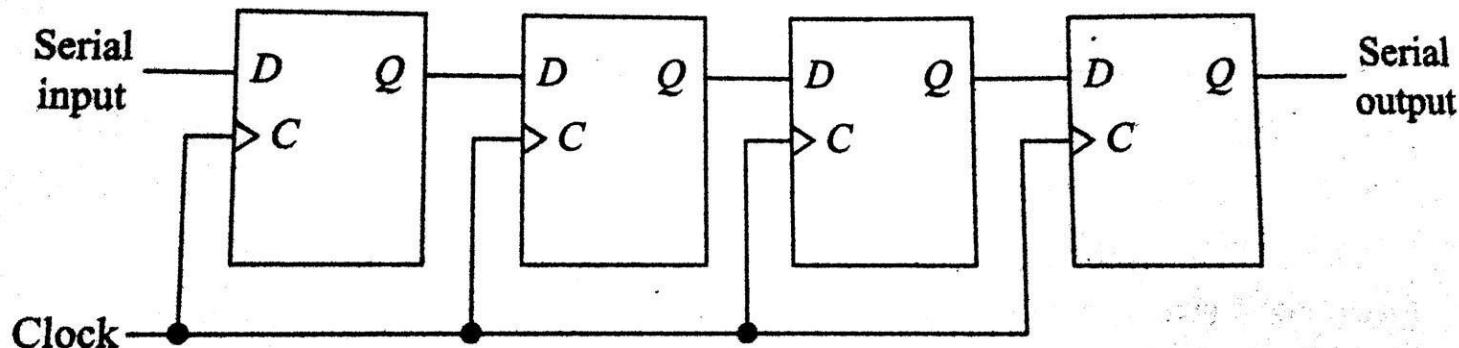
Shift Register

A register capable of shifting its binary information in one or more directions is known as a shift register.

Shift registers consist of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next.

All flip-flops receive a common clock pulse that initiates the shift from one stage to the next.

A 4-Bit Shift Register



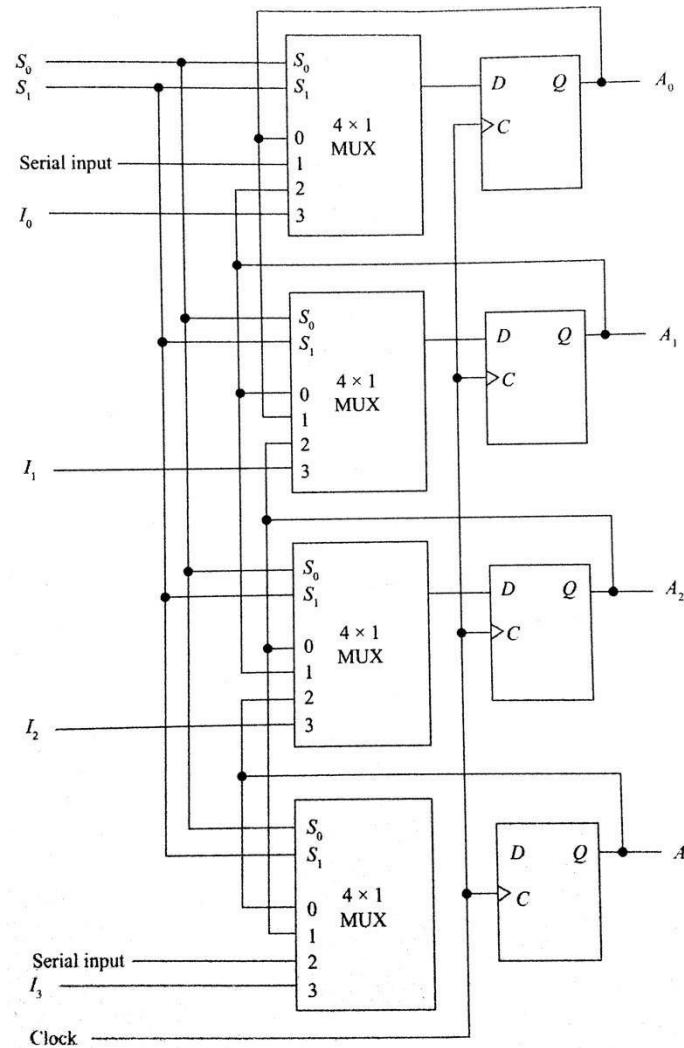
A simple 4-bit shift register with four cascading D flip-flops and a single clock.

Bidirectional Shift Register With Parallel Load

Mode control

S_1	S_0	Register operation
0	0	No change
0	1	Shift right (down)
1	0	Shift left (up)
1	1	Parallel load

Function Table



Binary Counters

Counter

A register that goes through a predetermined sequence of states upon the application of input pulses is called a counter.

The input pulse may be a clock or may come from an external source, and may occur at uniform intervals of time or at random.

Counters are useful for *counting the number of occurrences of an event* or for *generating timing signals to control the sequence of operations in a computer*.

Types of Counters

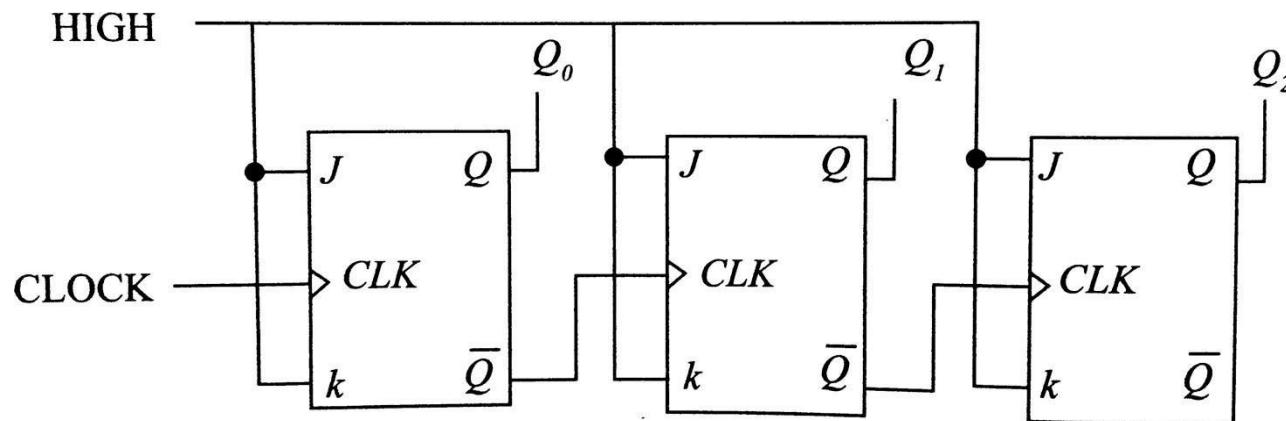
Asynchronous Counters (aka Ripple Counters)

Clock pulses are applied to the first flip-flop only; all subsequent flip-flops are clocked by the output produced by the preceding flip-flop.

Synchronous Counters

All flip-flops receive a common clock pulse.

4-Bit Asynchronous Counter (Ripple Counter)





Memory Unit

Memory

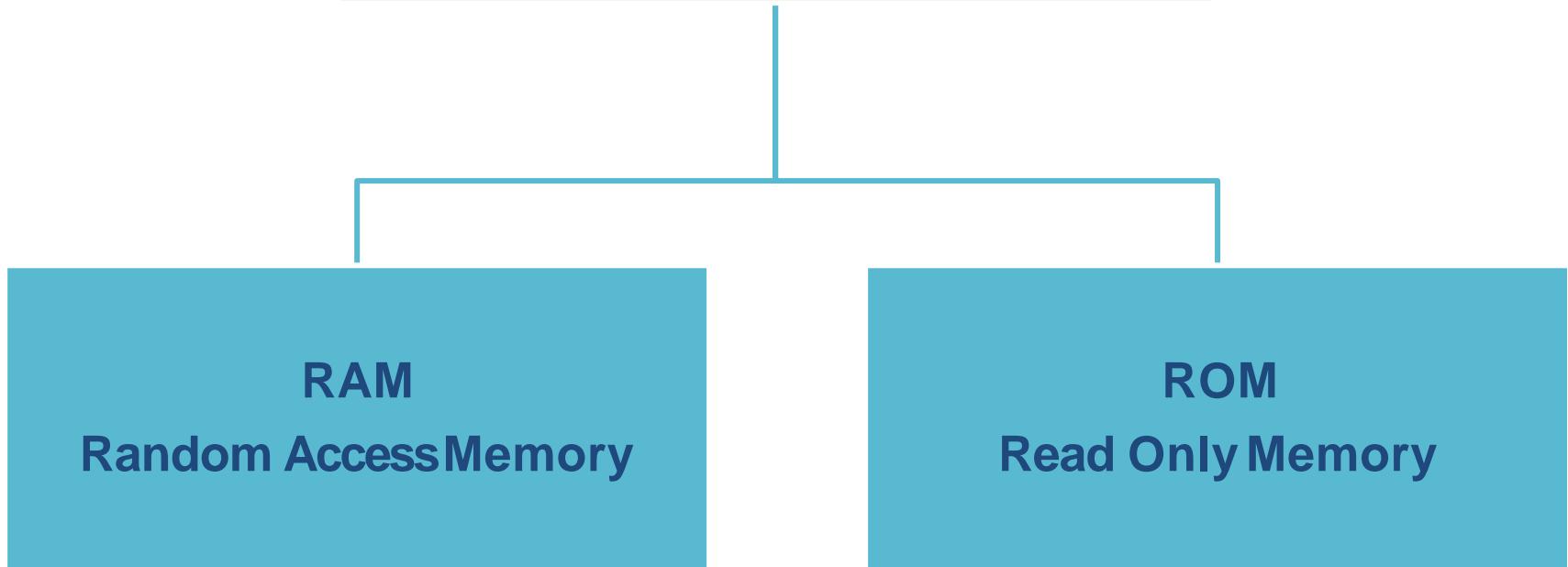
A memory unit is a collection of **storage cells**, together with the **associated circuits** that are needed to transfer information in and out of the storage cells.

Memory stores information in groups of bits called WORDS.

(1 word = ?Bytes)

A word is the **smallest entity** that can be moved in/out of the memory.

Major Types of Memory used in Computers



RAM

In RAM, any random memory cell can be accessed directly.

That is, the process of locating a word in the memory is the same and takes the same amount of time, regardless of the physical location of the memory cell that holds the desired word.

RAM is volatile.

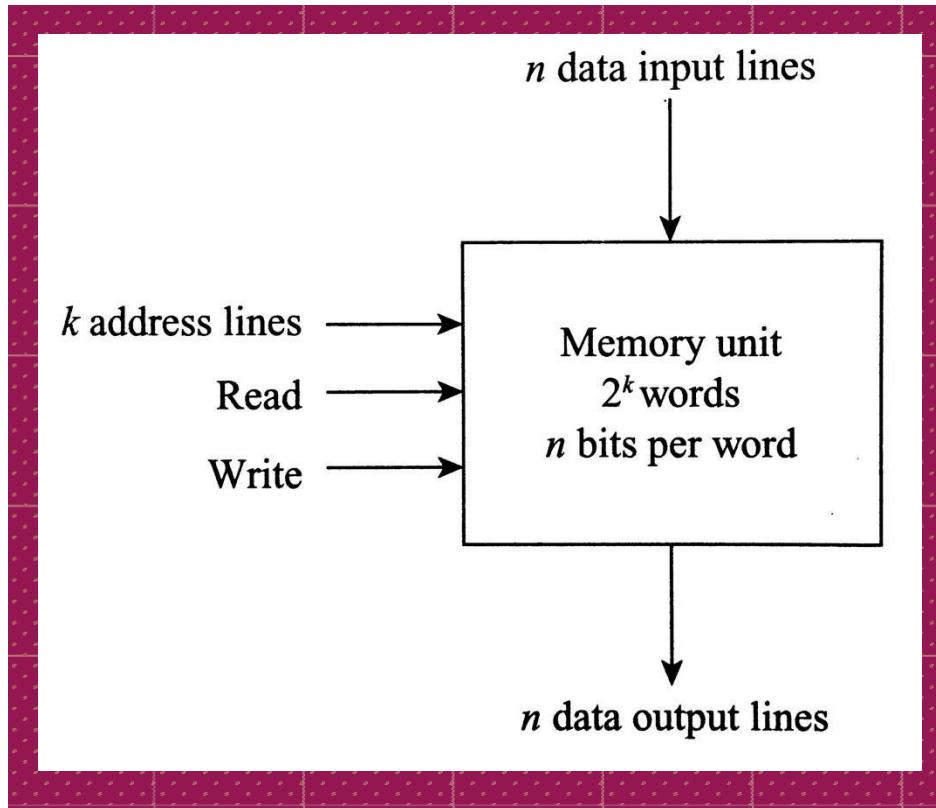
RAM

The two operations RAM can perform are READ and WRITE.

The ‘write’ signal specifies a **transfer-in** operation.

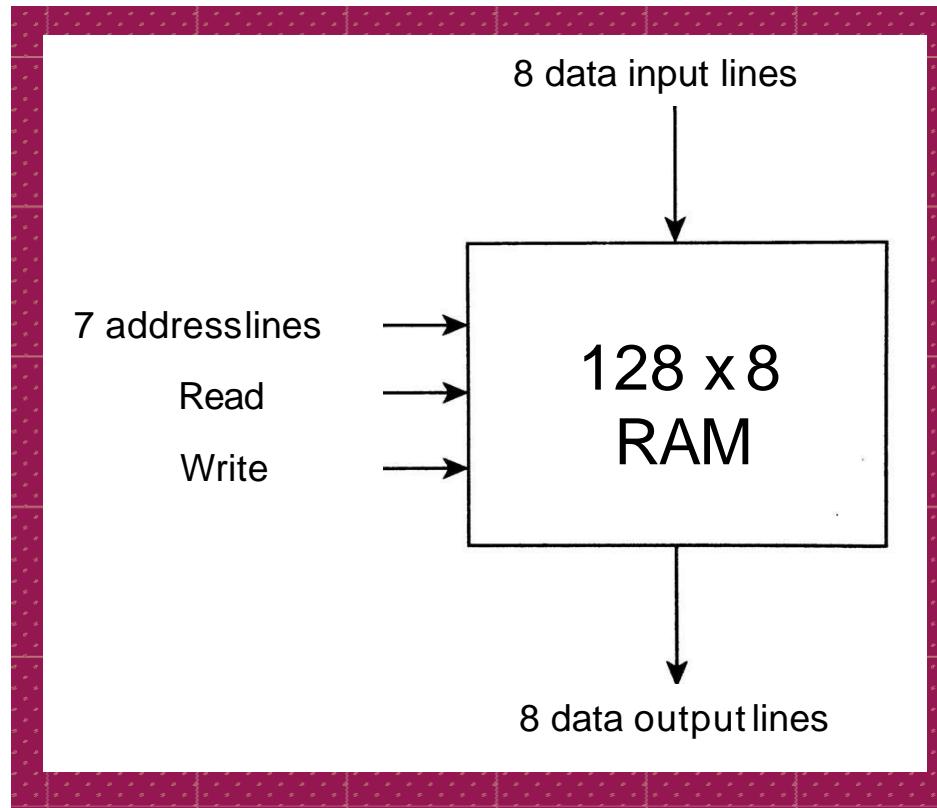
A ‘read’ operation specifies a **transfer-out** operation.

Block Diagram of RAM



Block Diagram of RAM

In an $m \times n$ RAM chip, 'n' is the number of data lines and 'k' is the number of address lines, where $2^k = m$



Steps for transferring a word in or out of the memory

The ‘Transfer-In’ Process:

- i. Apply the binary address of the desired word into the address lines.
- ii. Apply the data bits that must be stored in memory into the data input lines.
- iii. Activate the ‘write’ input.

The ‘Transfer-Out’ Process:

- i. Apply the binary address of the desired word into the address lines.
- ii. Activate the ‘read’ input.

ROM

ROM is a memory unit that only performs the *read* operation; it does not have a *write* capability.

The binary information stored in a ROM is made permanent when the hardware is produced, and this information cannot* be altered afterwards.

Unlike RAM, ROM is non-volatile.

* Atleast not without special equipment.

Block Diagram of ROM

