HNDIT1032 Computer and Network Systems

Week 05- Digital Circuits

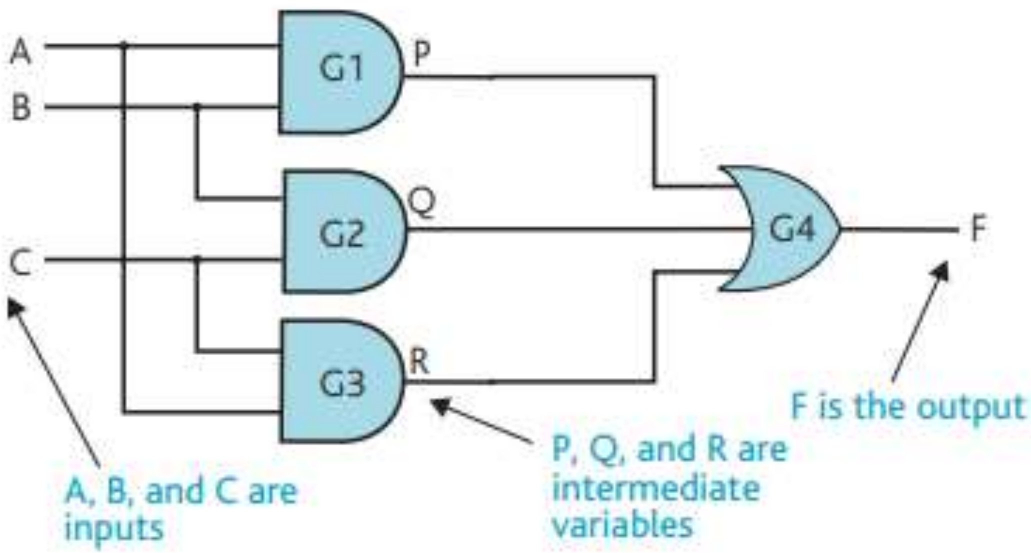
Introduction

- The digital computer consists of nothing more than the inter connection of three types of primitive elements called AND, OR, and NOT gates.
- Other gates called NAND, NOR, and EOR gates can be derived from these gates

Application of Gates

- Circuits are constructed by connecting gates together. The output from one gate can be connected (i.e. wired) to the input of one or more other gates.

Example 01



A
B
C
G1
P
G2
Q
G3
R
G4
F
A, B, and C are inputs
P, Q, and R are intermediate variables
F is the output

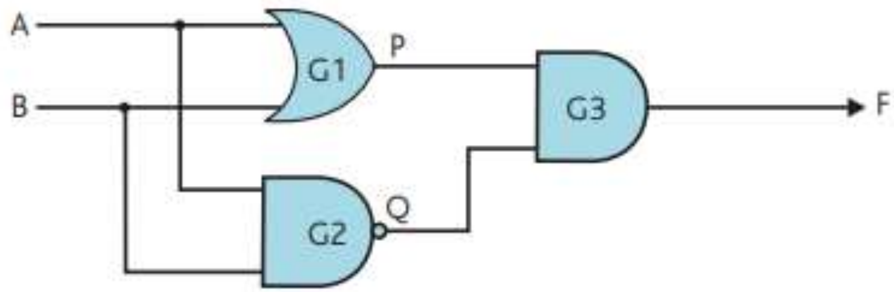
Example 01…

- Consider the circuit of Fig. 2.13 that uses three two-input AND gates labeled G1, G2, and G3, and a three input OR gate labeled G4.
- This circuit has three inputs A, B, and C, and an output F.
- One approach is to create a truth table that tabulates the output F for all the eight possible combinations of the three inputs A, B, and C.

Example…

Inputs			Intermediate values			Output
A	B	C	$P = A \cdot B$	$Q = B \cdot C$	$R = A \cdot C$	$F = P + Q + R$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

Example 02



Example 02...

$$F = P \cdot Q$$

$$P = A + B$$

$$Q = \overline{A \cdot B}$$

$$F = (A + B) \cdot \overline{A \cdot B}$$

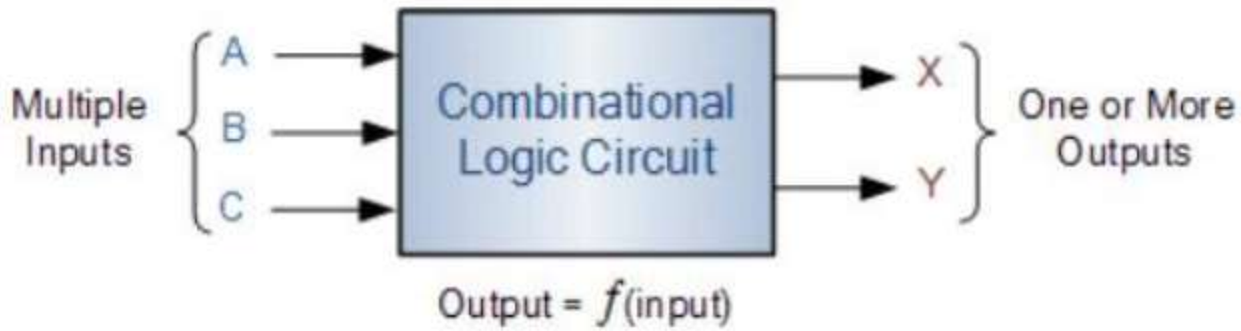
Inputs		Intermediate values		Output
A	B	$P = A + B$	$Q = \overline{A \cdot B}$	$F = P \cdot Q$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

Types of Digital Logic Circuits

- Digital logic circuits can be classified into
  - Combinational
  - Sequential

Combinational Circuits

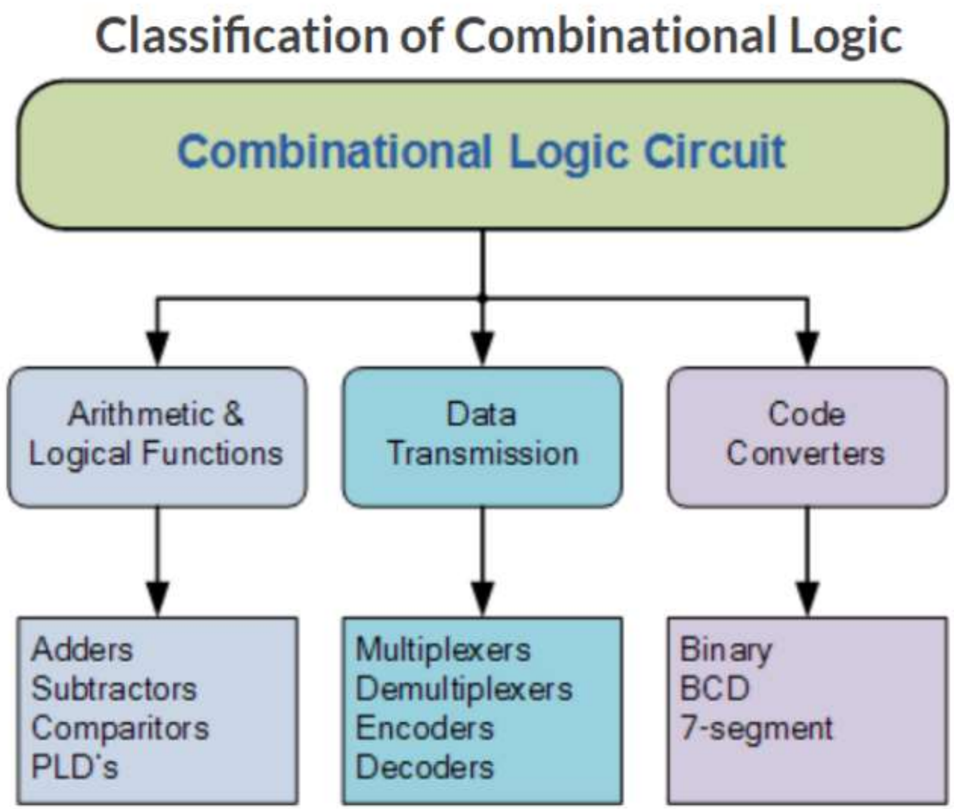
- Theses circuits are made up from logic gates that are “combined” or connected together to produce more complicated switching circuits.



Combinational Circuits…

- Combinational logic circuits have no feedback, and any changes to the signals being applied to their inputs will immediately have an effect at the output.
- Common combinational circuits made up from individual logic gates that carry out a desired application include Multiplexers, Demultiplexers, Encoders, Decoders, Full and Half Adders

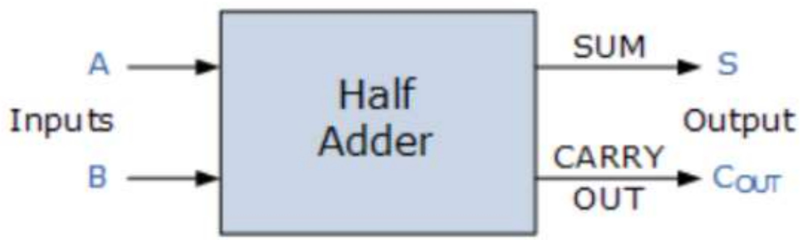
Classification



Half Adder

- Half adder is the simplest of all adder circuits. Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (s) and carry bit (c) both as output.
- The addition of 2 bits is done using a combination circuit called a Half adder.

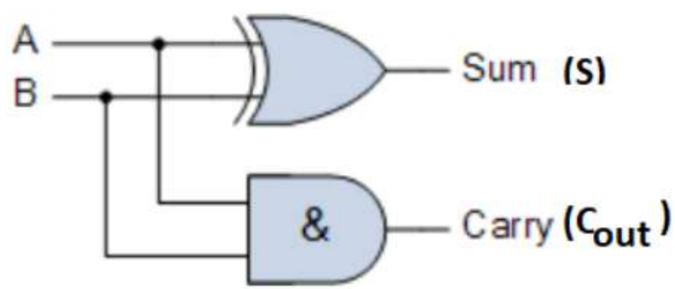
Half Adder…



Input		Output	
A	B	S	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A'B + AB'$ → A XOR B → $A \oplus B$

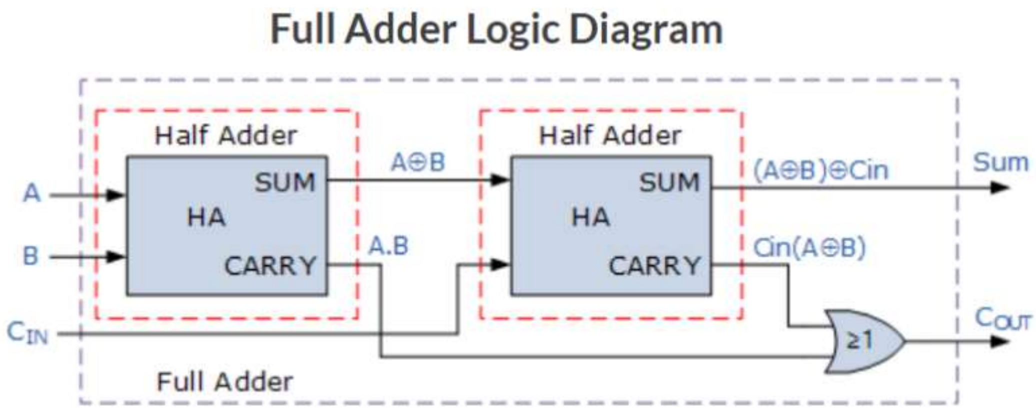
$C_{out} = AB$



Full Adder

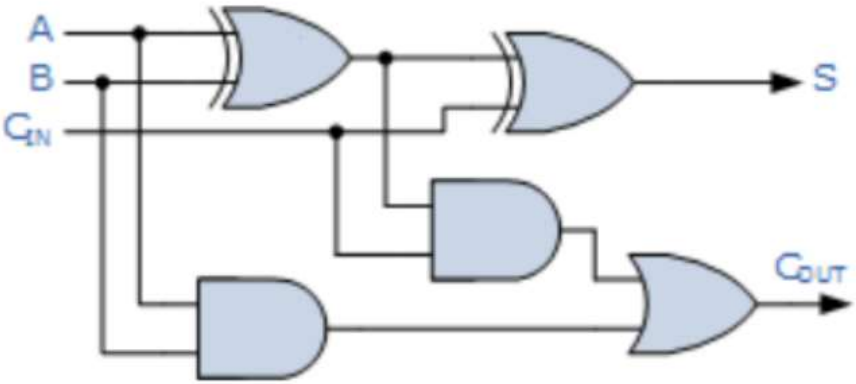
- Full Adder is the adder that adds three inputs and produces two outputs.
- The first two inputs are A and B and the third input is an input carry as C-IN.
- The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

Full Adder

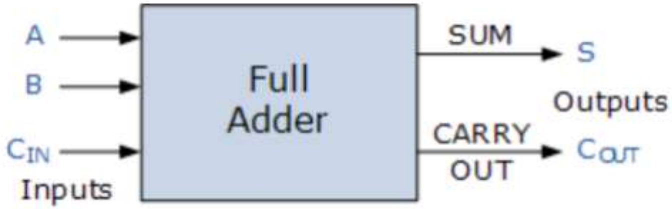


$S = A \oplus B \oplus C$

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



Full Adder



$$S = A'B'C_{in} + A'BC'_{in} + AB'C'_{in} + ABC_{in}$$

$$C_{out} = A'BC_{in} + AB'C_{in} + ABC'_{in} + ABC_{in}$$

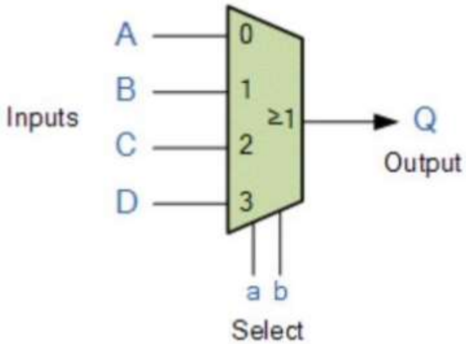
$$S = A \oplus B \oplus C$$

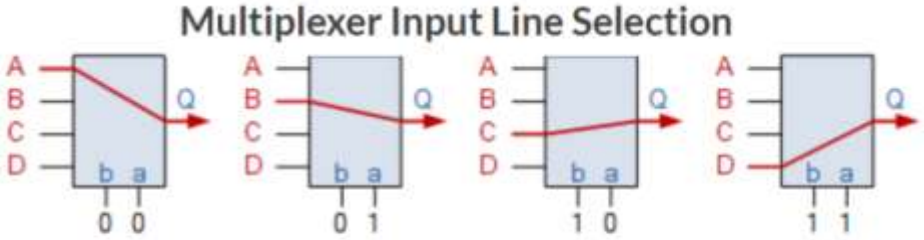
$$C_{out} = A.B + C_{in}(A \oplus B)$$

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

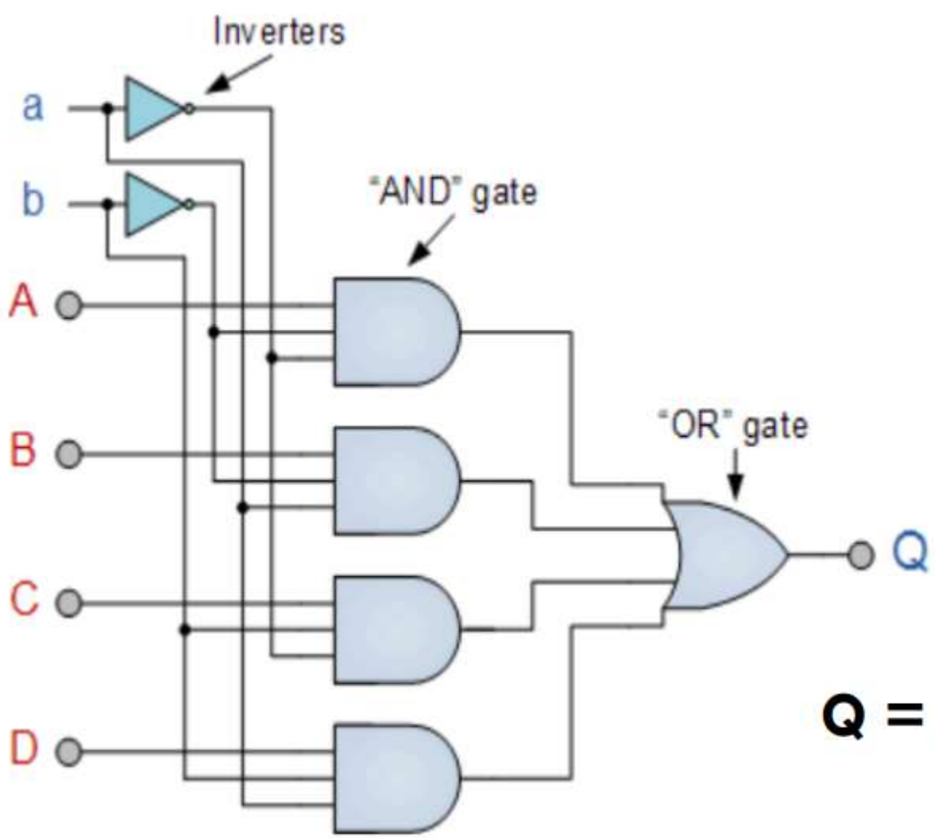
Multiplexer

- The multiplexer is a combinational logic circuit designed to switch one of several input lines to a single common output line.
- The multiplexer, shortened to “MUX” or “MPX”.





4X1 Multiplexer

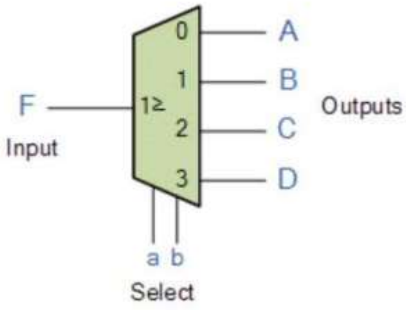


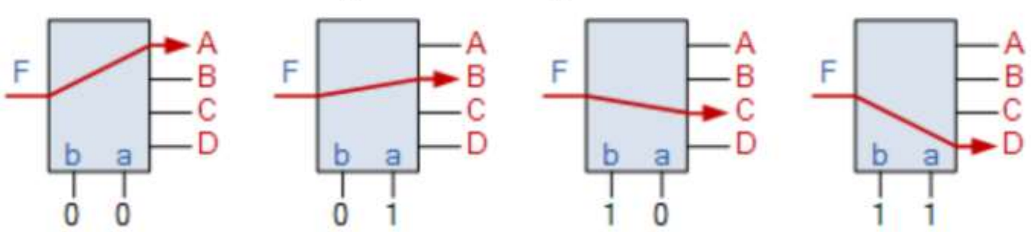
b	a	Q
0	0	A
0	1	B
1	0	C
1	1	D

Q = a'b'A + ab'B + a'bC + abD

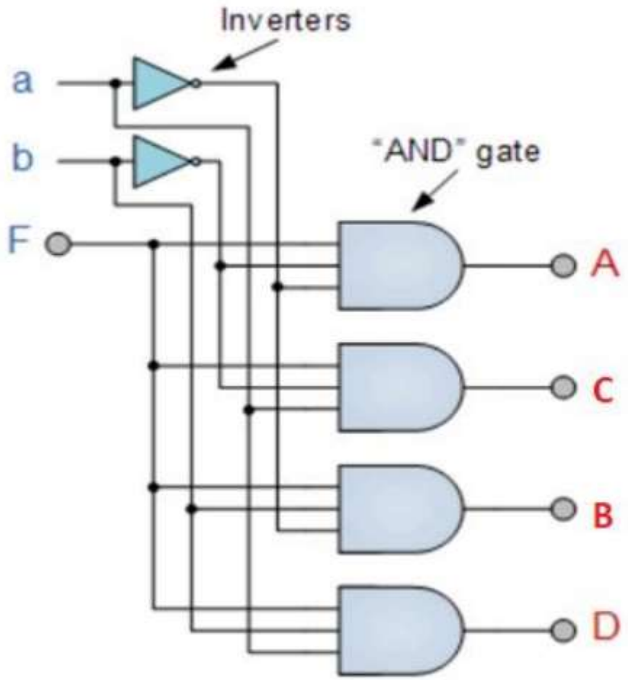
Demultiplexer

- The demultiplexer takes one single input data line and then switches it to any one of a number of individual output lines one at a time.





Demultiplexer

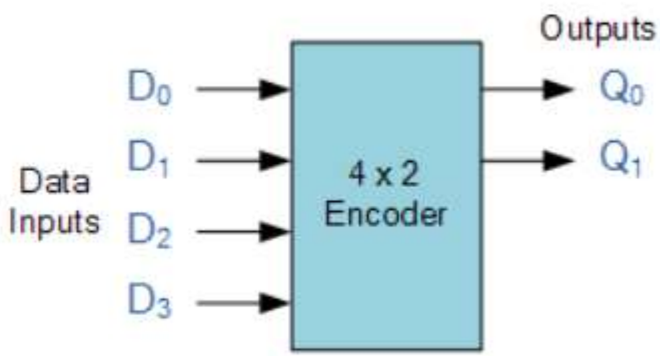


Output Select		Data Output Selected
a	b	
0	0	A
0	1	B
1	0	C
1	1	D

Encoder

- The Digital Encoder more commonly called a Binary Encoder takes ALL its data inputs one at a time and then converts them into a single encoded output.
- An “n-bit” binary encoder has 2^n input lines and n-bit output lines with common types that include 4-to-2, 8-to-3 and 16-to-4 line

4-to-2 Bit Binary Encoder

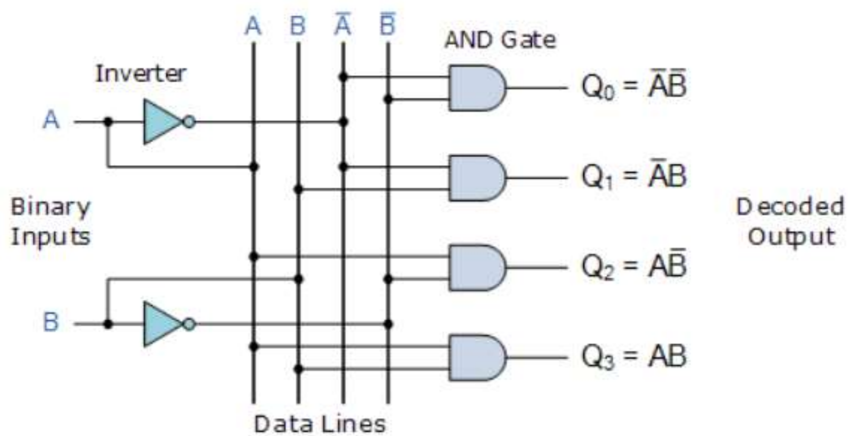


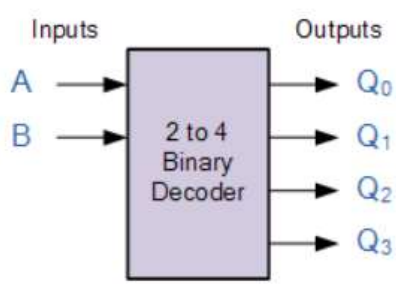
Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

Decoder

- The term “Decoder” means to translate or decode coded information from one format into another.
- Binary decoder transforms “n” binary input signals into an equivalent code using 2^n outputs.

2-to-4 Bit Binary Decoder



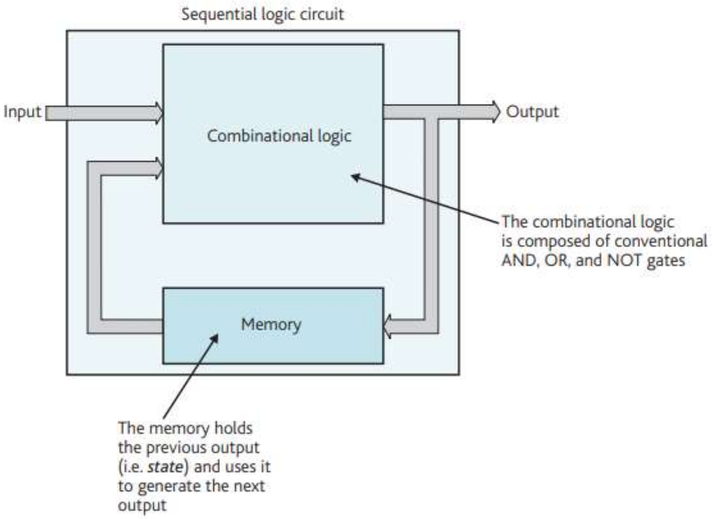


Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Sequential Circuit

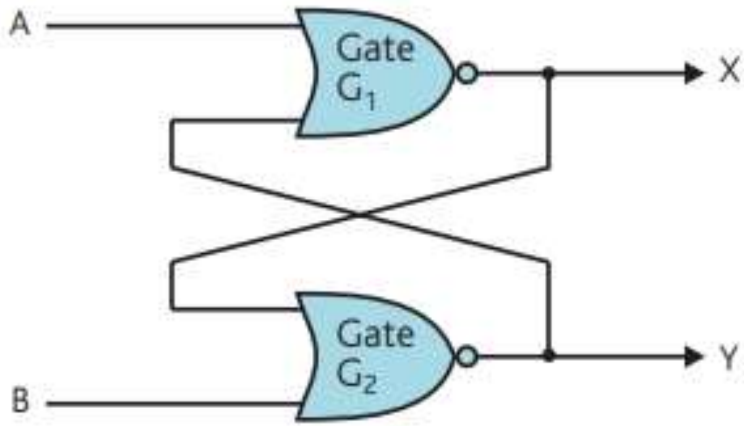
- The output of a sequential circuit depends not only on its current inputs, but also on its previous inputs.



Latch

- A latch is a 1-bit memory element.
- You can capture a single bit in a latch at one instant and then use it later; for example, when adding numbers you can capture the carry-out in a latch and use it as a carry-in in the next calculation

RS Flip-flop



A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

1. $X = \overline{A + Y}$
2. $Y = \overline{B + X}$

If we substitute the value for Y from equation (2) in equation (1), we get

3. $X = \overline{A + \overline{B + X}}$

$= \overline{A}\cdot\overline{\overline{B + X}}$ By de Morgan's theorem

$= \overline{A}\cdot(B + X)$ Two negations cancel

$= \overline{A}\cdot B + \overline{A}\cdot X$ Expand the expression

Other examples

- D flip-flop
- Clocked flip-flop
- Jk flip-flops

Sum of Product

- The *Sum of Product* (SOP) expression comes from the fact that two or more products (AND) are summed (OR) together.
- That is the outputs from two or more AND gates are connected to the input of an OR gate so that they are effectively OR’ed together to create the final AND-OR logical output.

Example 01 SOP

$$Q = A.B.\overline{C} + A.\overline{B}.C + \overline{A}.B.C$$

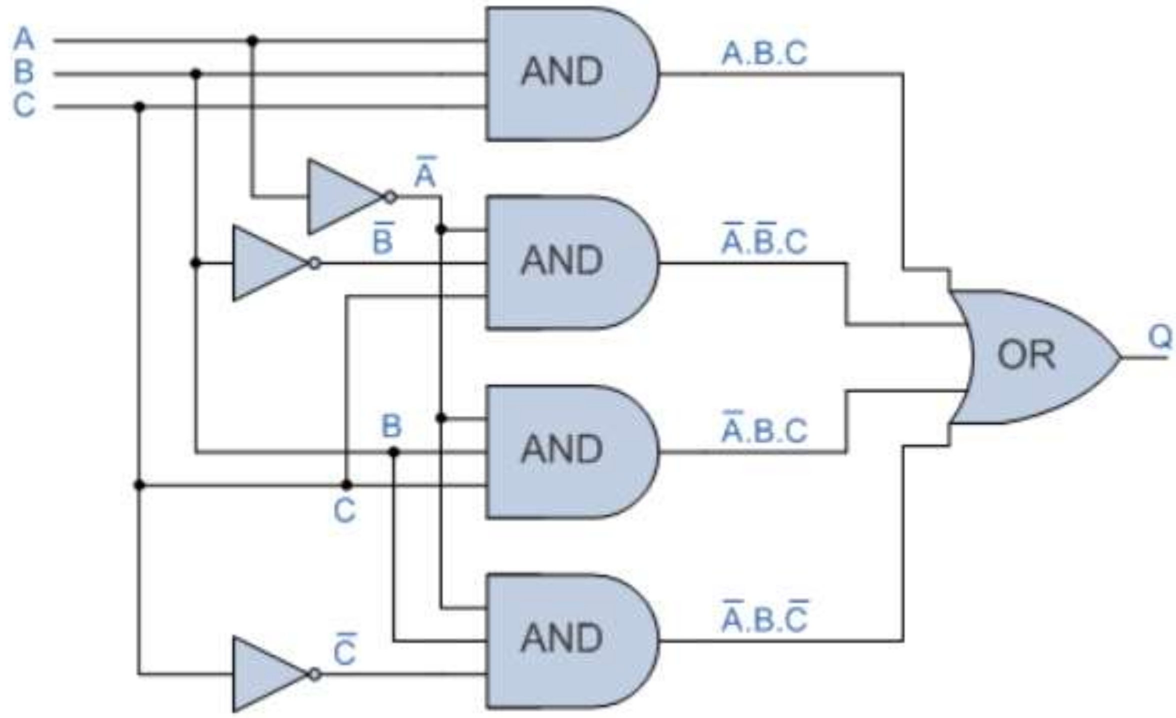
Inputs			Output	Product
C	B	A	Q	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$A.B.\overline{C}$
1	0	0	0	
1	0	1	1	$A.\overline{B}.C$
1	1	0	1	$\overline{A}.B.C$
1	1	1	0	

Example 02 SOP

$$Q = A.B.C + \overline{A}.\overline{B}.C + \overline{A}.B.C + \overline{A}.B.\overline{C}$$

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	
0	0	1	0	
0	1	0	**1**	$\overline{A}.B.\overline{C}$
0	1	1	0	
1	0	0	**1**	$\overline{A}.\overline{B}.C$
1	0	1	0	
1	1	0	**1**	$\overline{A}.B.C$
1	1	1	**1**	$A.B.C$

Example 02 SOP



Product of Sum

- The *Product of Sum* (POS) expression comes from the fact that two or more sums (OR’s) are added (AND’ed) together.
- That is the outputs from two or more OR gates are connected to the input of an AND gate so that they are effectively AND’ed together to create the final (OR AND) output.

Example 01 POS

$$Q = (A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$$

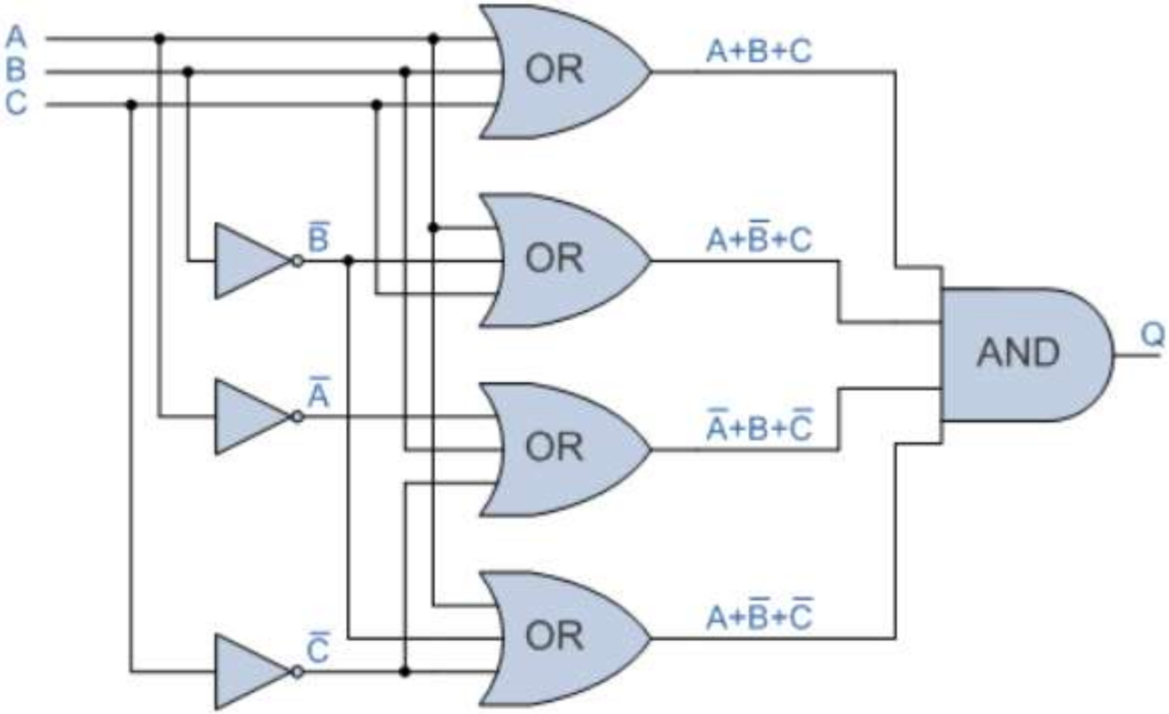
Inputs			Output	Product
C	B	A	Q	
0	0	0	0	$A+B+C$
0	0	1	1	
0	1	0	0	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	0	$A+\overline{B}+\overline{C}$
1	1	1	1	

Example 02 POS

$$Q = (A + B + C)(A + \overline{B} + C)(\overline{A} + B + \overline{C})(A + \overline{B} + \overline{C})$$

Inputs			Output	Product
C	B	A	Q	
0	0	0	**0**	$A+B+C$
0	0	1	1	
0	1	0	**0**	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	**0**	$\overline{A}+B+\overline{C}$
1	1	0	**0**	$A+\overline{B}+\overline{C}$
1	1	1	1	

Example 02 POS



References

- Clements, A., The Principles of Computer Hardware, Oxford University Press (4th Ed), 2006.
- https://www.electronics-tutorials.ws/logic