Basic Logic Gates

Basic Logic Gates and Basic Digital Design

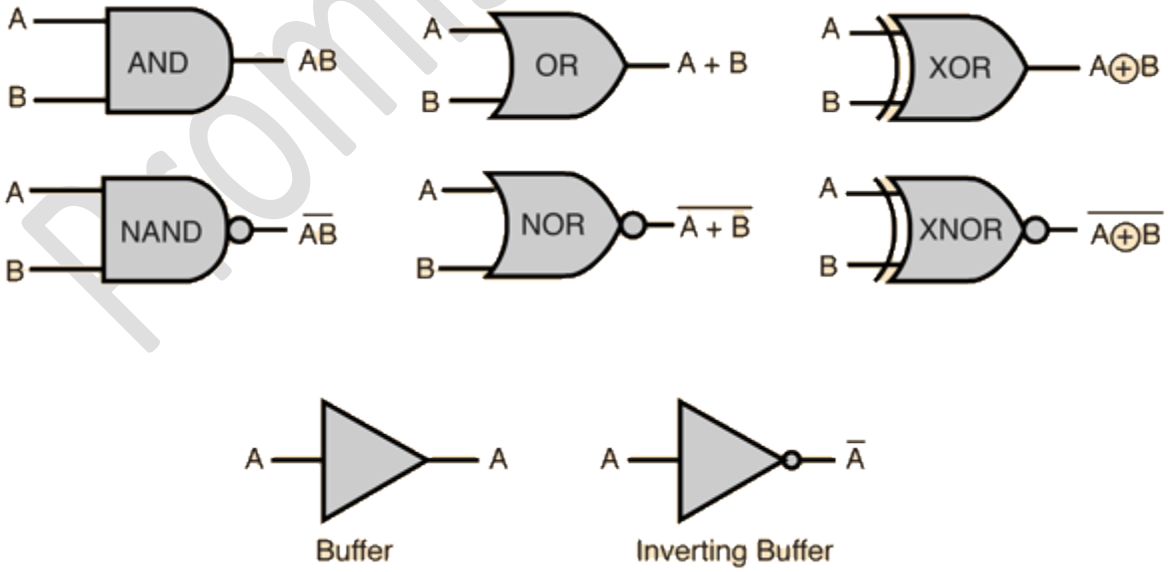
- Boolean (Logic) functions may be practically implemented by using electronic gates. The following points are important to understand.
- Electronic gates require a power supply.

Truth Tables

- Truth tables are used to help show the function of a logic gate. Truth tables are made using 0 and 1. 0 means “False” and 1 means “True”.

There are different types of gates as sown below:

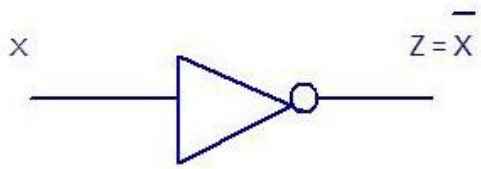
- Basic Gates : NOT, AND, and OR Gates
- NAND and NOR Gates
- DeMorgan’s Theorem
- Exclusive-OR (XOR) Gate
- Multiple-input Gates



NOT, AND, and OR Gates

1. NOT gate

NOT Gate

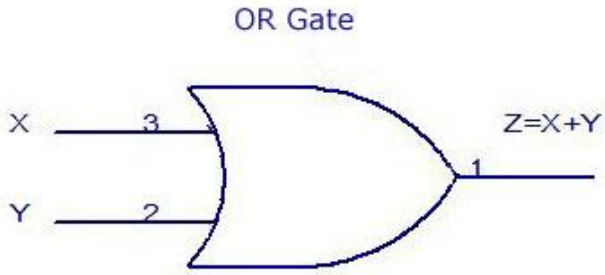


TRUTH TABLE

INPUT	OUTPUT
X	Z
0	1
1	0

2. OR gate

2 Input OR Gate

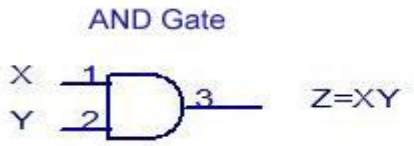


TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

3. **AND gate**

2 Input AND Gate

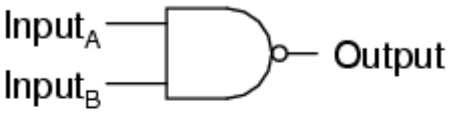


TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

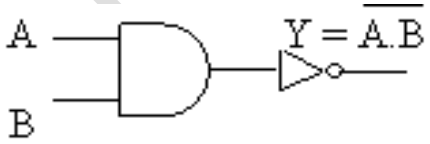
4. NAND gate

NAND gate



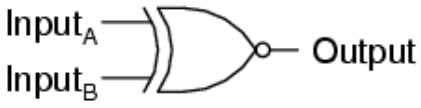
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Evaluated Circuit



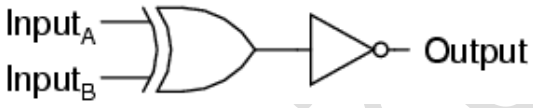
5. NOR gate

Exclusive-NOR gate

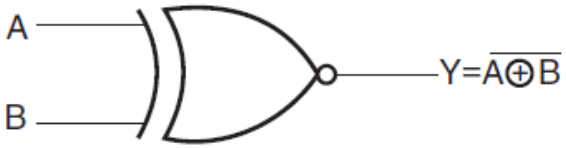


A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

Equivalent gate circuit



6. X-OR Gate

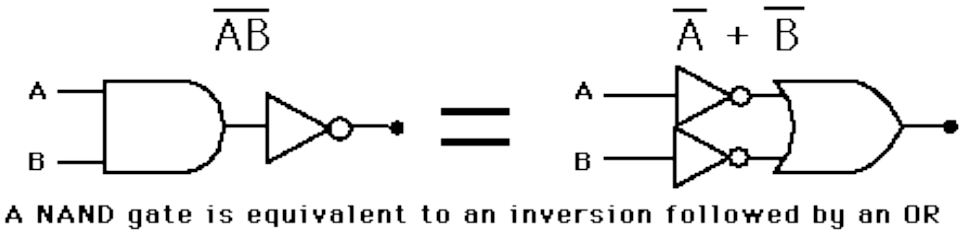


$$Y = (\overline{A \oplus B}) = (A.B + \overline{A}.\overline{B})$$

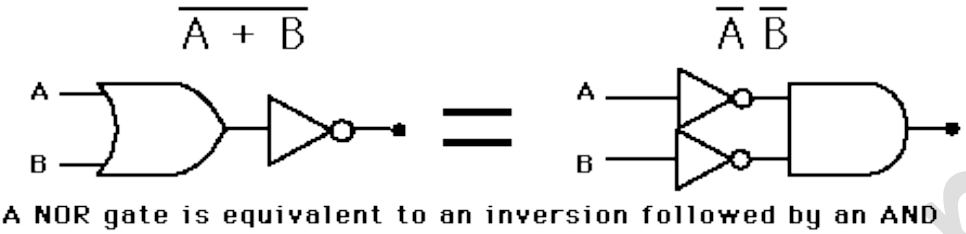
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

De-morgens theorem

1. ~(X & Y) = ~X | ~Y
2. ~(X | Y) = ~X & ~Y



A NAND gate is equivalent to an inversion followed by an OR



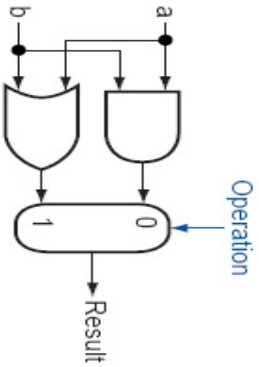
A NOR gate is equivalent to an inversion followed by an AND

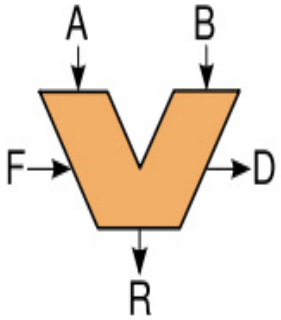
ALU (Arithmetic Logic Unit)

- ALU stands for: Arithmetic Logic Unit
- ALU is a digital circuit that performs Arithmetic (Add, Sub, etc.) and Logical (AND, OR, NOT) operations.
- John Von Neumann proposed the ALU in 1945 when he was working on EDVAC.

Construction of ALU

- This is a one-bit ALU which can do Logical AND & Logical OR operation.
- Result = a AND b when operation = 0
- Result = a OR b when operation = 1
- The operation line is the input.

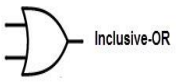




Half Adder

BINARY Addition circuit

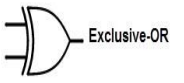
- Logic gates are used to accomplish the arithmetic operation of binary addition in digital circuits.
- A two input logic gate is required to accomplish the addition of two binary numbers. The exclusive-OR gate is used to achieve binary addition which is slightly different from basic OR gate.
- An inclusive-OR gate or basic OR gate adds integers together and produces an output 1 when both or either inputs are high.



A	B	Z	
0	0	0	
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	1	AB

$Z = \bar{A}B + A\bar{B} + AB$

$Z = A + B$

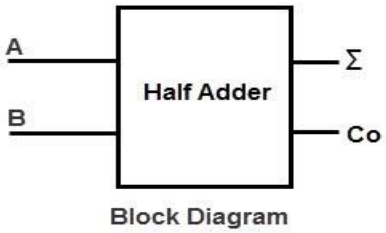


A	B	Z	
0	0	0	
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	

$Z = \bar{A}B + A\bar{B}$

HALF ADDER

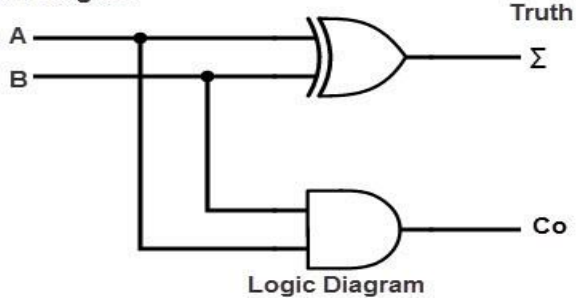
- A logic circuit block used for adding two one bit numbers or simply two bits is called as a half adder circuit. This circuit has two inputs which accept the two bits and two outputs, with one producing sum output and other produce carry output.
- As we discussed above that binary addition is commonly performed by Ex-OR gate, but for the first three rules, it performs the binary addition and when the two inputs are logic 1, it does not develop any carry.
- To accomplish the binary addition with Ex-OR gate, there is need of additional circuitry to perform the carry operation. Hence, a half adder is formed by connecting AND gate to the input terminals of the Ex-OR gate so as to produce the carry as shown in this figure.



Block Diagram

A	B	Sum (∑)	Carry Out (Co)
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**0**	**1**

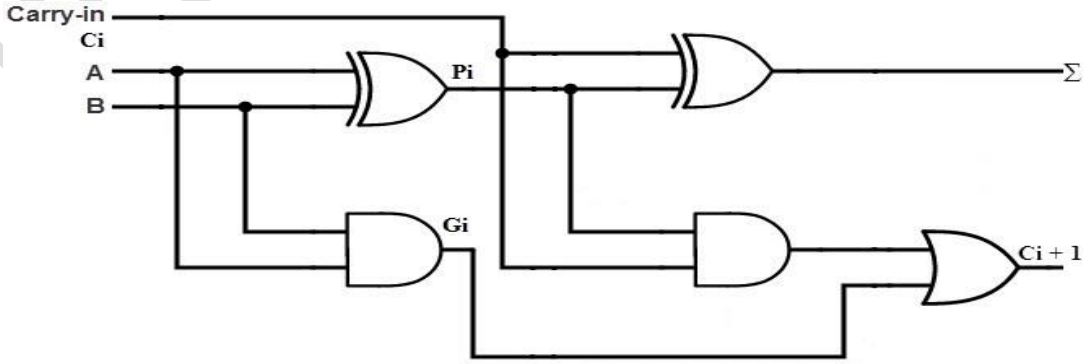
Truth Table

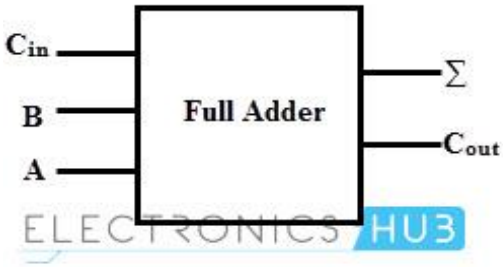


Logic Diagram

Full Adder

- A binary full adder is a multiple output combinational logic network that performs the arithmetic sum of three input bits.
- As we have seen that the half adder cannot respond to the three inputs and hence the full adder is used to add three digits at a time.
- It consists of three inputs, in which two are input variables represent the two significant bits to be added, labeled as A and B, whereas the third input terminal is the carry from the previous lower significant position and labeled as Cin. The two outputs are a sum and a carry outputs which are labeled as **?** and Cout respectively.





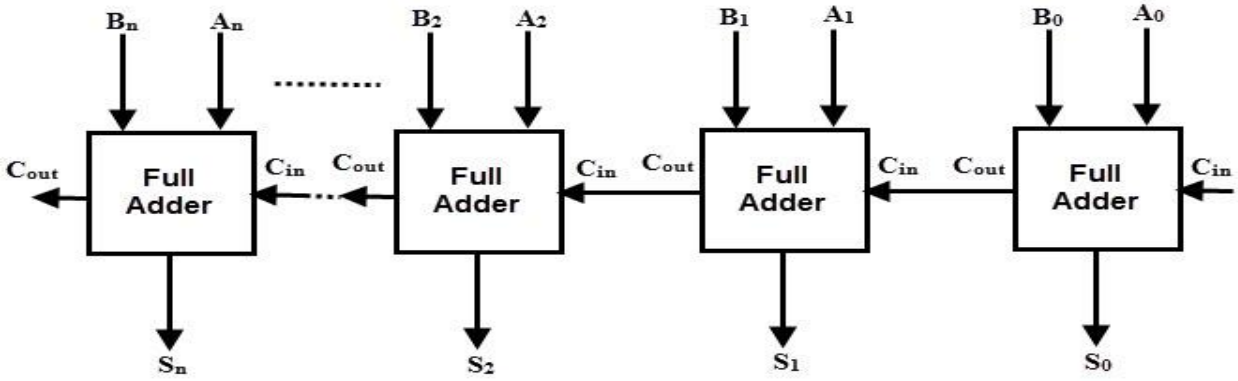
Block Diagram of Full Adder

C_{in}	B	A	∑	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**

Truth Table

Parallel Adder

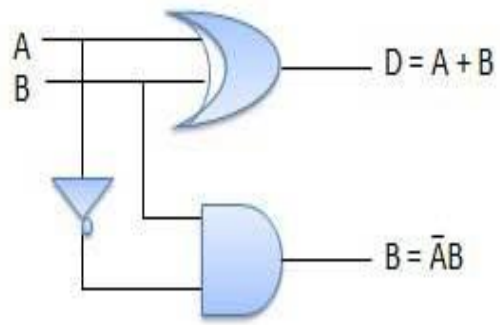
By connecting a number of full adders in parallel, n-bit parallel adder is constructed. From the below figure, it is to be noted that the output carry of one full adder is given to the another full adder as carry input.



Half Subtractor

- Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow).
- It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed.

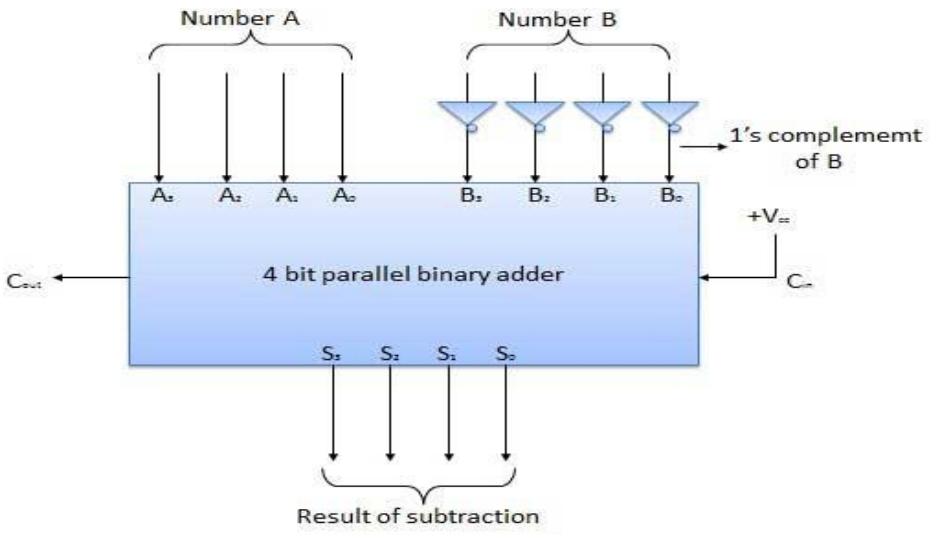
- In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.



Inputs		Output	
A	B	(A – B)	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full Subtractor

- The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,C and two output D and C'.
- A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and C' is the borrow output.
- The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction.
- S_3 S_2 S_1 S_0 represents the result of binary subtraction (A-B) and carry output C_{out} represents the polarity of the result.
- If A > B then C_{out} = 0 and the result of binary form (A-B) then C_{out} = 1 and the result is in the 2's complement form.

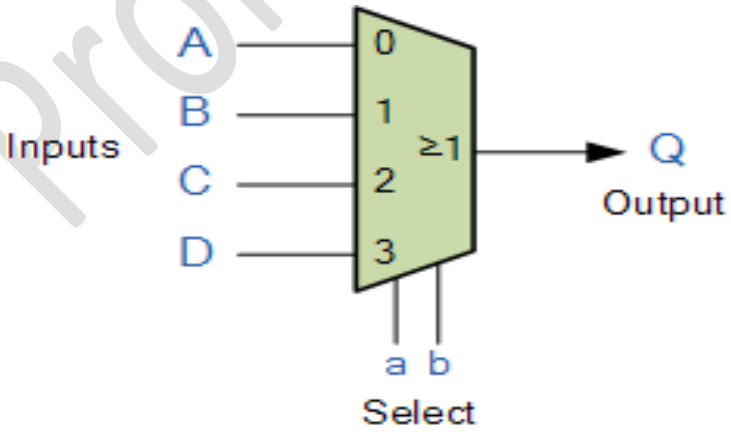


Digital component

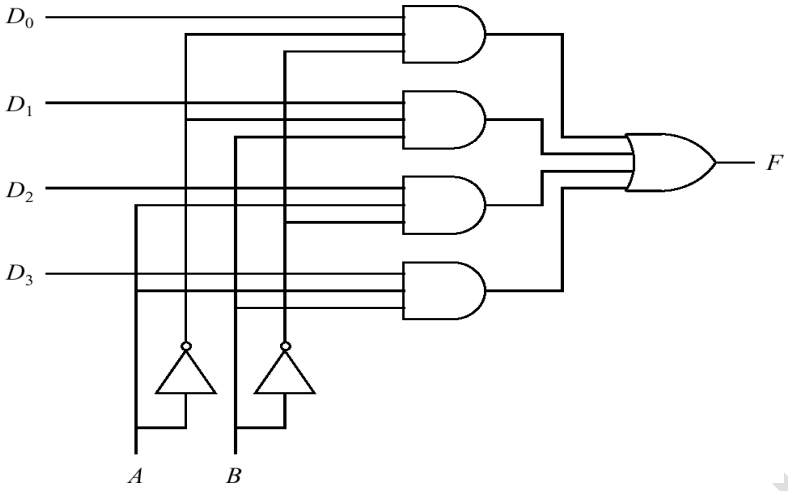
- High level digital circuit designs are normally made using collections of logic gates referred to as components, rather than using individual logic gates. The majority function can be viewed as a component.

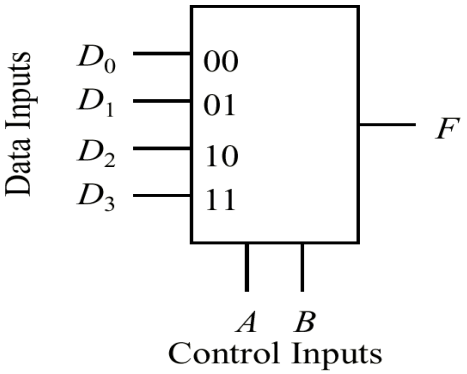
Multiplexer

- In electronics, a **multiplexer** (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line.



Gate level layout of logic gates

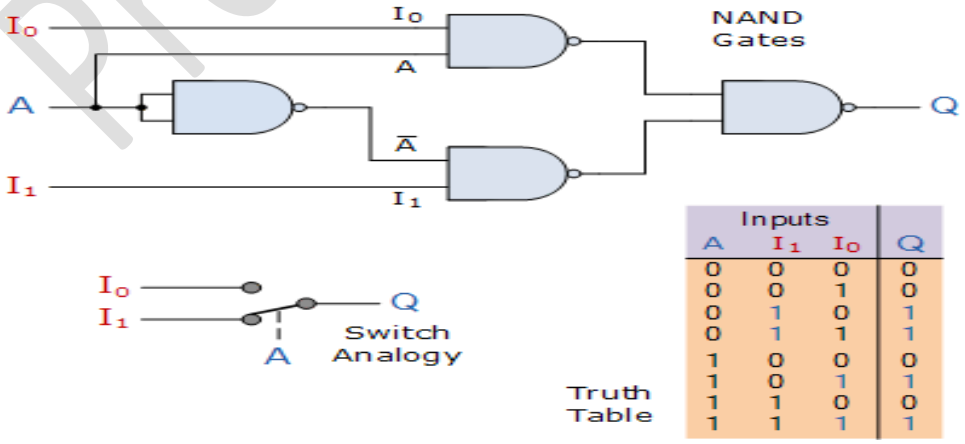




A	B	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$F = \overline{A}\,\overline{B}\,D_0 + \overline{A}\,B\,D_1 + A\,\overline{B}\,D_2 + A\,B\,D_3$$

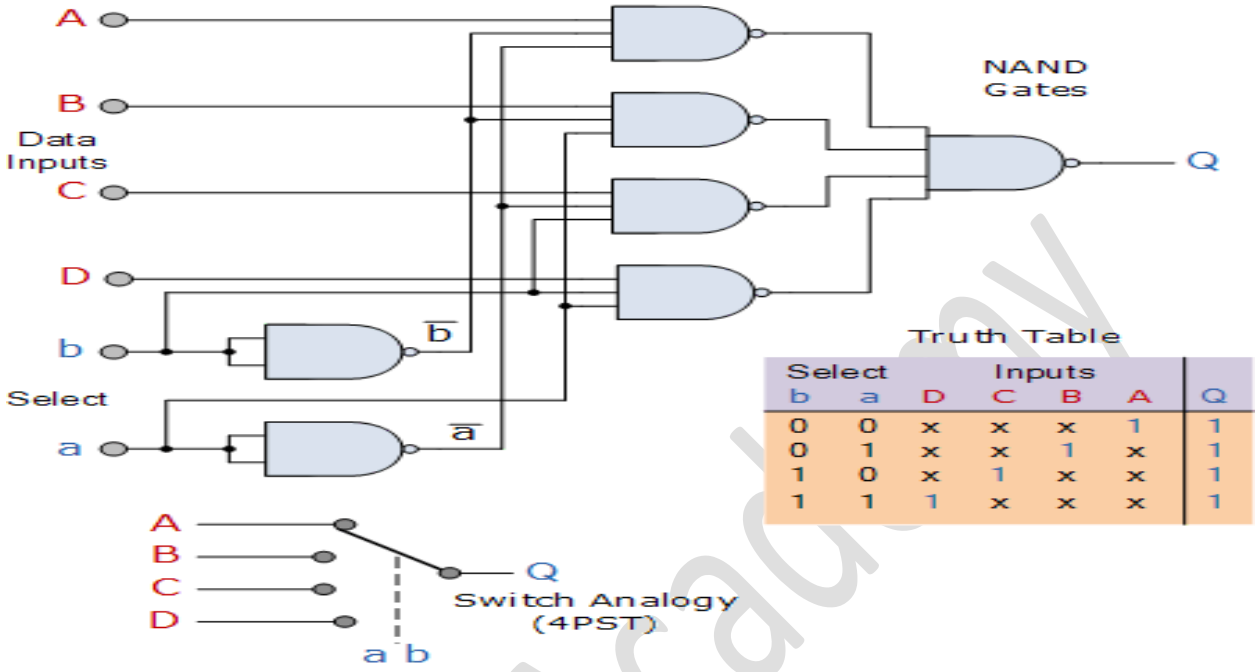
2-input Multiplexer Design



Inputs			
A	I_1	I_0	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

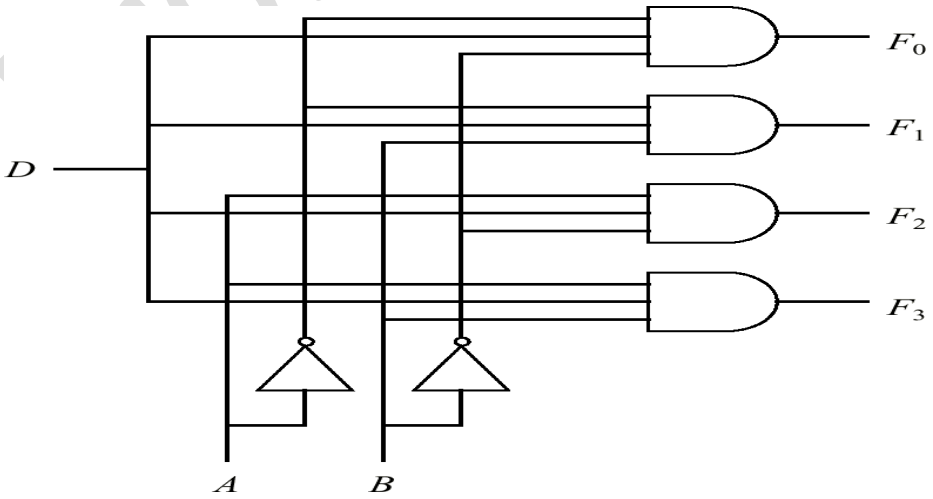
Truth Table

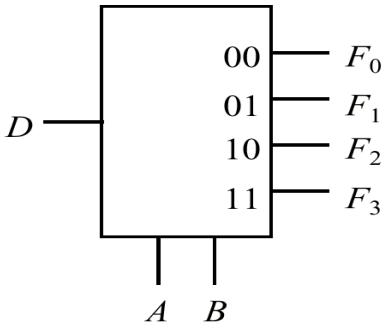
4-to-1 Channel Multiplexer



Demultiplexer

- The process of getting information from one input and transmitting the same over one of many outputs is called demultiplexing. A demultiplexer is a combinational logic circuit that receives the information on a single input and transmits the same information over one of 2n possible output lines.





D	A	B	F_0	F_1	F_2	F_3
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	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$F_0 = D\overline{A}\,\overline{B} \qquad F_2 = DA\overline{B}$$

$$F_1 = D\overline{A}B \qquad F_3 = DAB$$