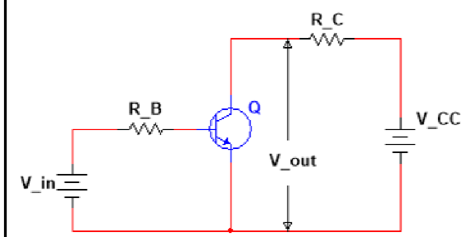
COMBINATIONAL LOGIC

Satish Chandra
Assistant Professor
Department of Physics
P P N College, Kanpur
www.satish0402.weebly.com

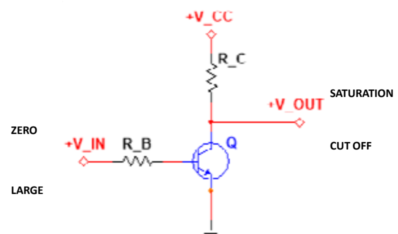
LOGIC GATES

- A circuit with an output signal that is logical function of the inputs.
- A circuit with one or more input voltages but only one output voltage.
- They are of two types
  1. Basic gates – NOT, OR & AND.
  2. Combinational gates
     a) Universal gates – NOR & NAND
     b) Arithmatic gates – Ex-OR & Ex-NOR.

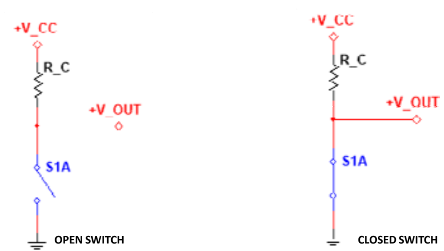
CE NPN Transistor Circuit



Equivalent CE Circuit



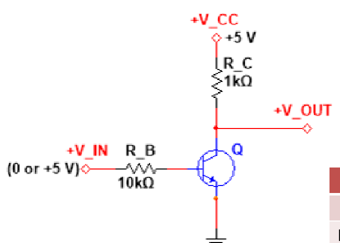
Transistor as a switch



Logic Circuits

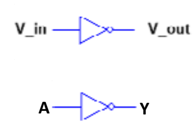
BASIC GATES

Transistor Inverter



V_{in}	V_{out}
LOW	HIGH
HIGH	LOW

NOT Gate



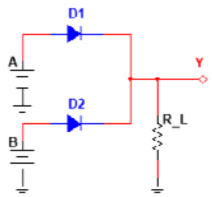
A	Y
0	1
1	0

$Y = \overline{A}$

NOT Gate

- A gate with only one input and one output.
- Output state is always opposite the input state
- An INVERTER.

OR Gates



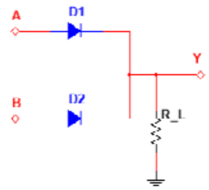
CASE 1: A is low and B is low.

CASE 2: A is low and B is high.

CASE 3: A is high and B is low.

CASE 4: A is high and B is high.

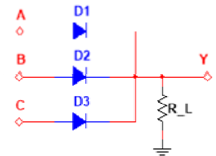
Two-input OR gate



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

Y = A + B

Three-input OR gate



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

Y = A + B + C

OR Gate Symbols

Two-input  Y = A + B

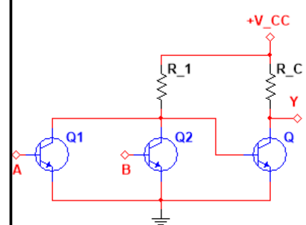
Three-input  Y = A + B + C

Four-input  Y = A + B + C + D

OR Gate

- Two or more input signals but only one output signal.
- Output is high if **any or all** of the inputs are high.
- Logical Symbol 
- OR gate performs logical addition
- Boolean expression Y = A + B

OR Gate



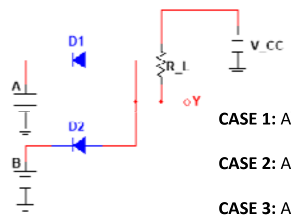
CASE 1:
A is low and B is low.

CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

CASE 4:
A is high and B is high

AND Gates



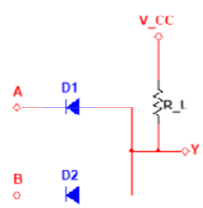
CASE 1: A is low and B is low.

CASE 2: A is low and B is high.

CASE 3: A is high and B is low.

CASE 4: A is high and B is high.

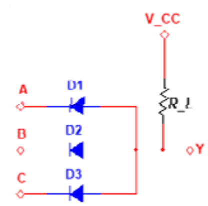
Two-input AND Gate



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

Y = A.B

Three-input AND Gate



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

Y = A.B.C

AND Gate Symbols

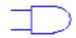
Two-input  $Y = A.B$

Three-input  $Y = A.B.C$

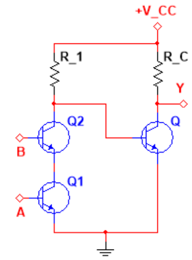
Four-input  $Y = A.B.C.D$

AND Gate

- Two or more input signals but only one output signal.
- Output is high if **all** of the inputs are high.
- Logical Symbol
- AND gate performs logical multiplication
- Boolean expression $Y = A.B$



AND Gate



CASE 1:
A is low and B is low.

CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

CASE 4:
A is high and B is high.

Logic Circuits

BOOLEAN ALGEBRA

Boolean Algebra

- **OR Addition Laws**

$A + 0 = A$

$A + 1 = 1$

$A + A = A$

$A + \bar{A} = 1$

- **AND Multiplication Laws**

$A.0 = 0$

$A.1 = A$

$A.A = A$

$A.\bar{A} = 0$

Boolean Algebra

- **Commutative Laws**

$A + B = B + A$

$A.B = B.A$

- **Associative Laws**

$A + (B + C) = (A + B) + C$

$A.(B.C) = (A.B).C$

Boolean Algebra

- **Distributive Laws**

$A(B + C) = A.B + A.C$

$A + B.C = (A + B).(A + C)$

- **Absorptive Laws**

$A. + A.B = A$

$A.(A + B) = A$

$A.(\bar{A} + B) = AB$

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

Boolean Algebra

- **Complement Theorem**

$\bar{\bar{A}} = A$

- **De Morgan's Theorem**

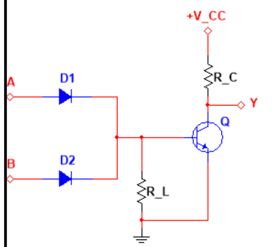
$\overline{A + B} = \bar{A}.\bar{B}$

$\overline{A.B} = \bar{A} + \bar{B}$

Logic Circuits

UNIVERSAL GATES & ARITHMETIC GATES

NOR Gate



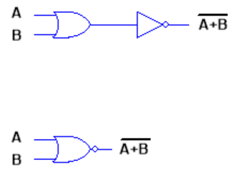
CASE 1:
A is low and B is low.

CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

CASE 4:
A is high and B is high.

NOR Gate



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

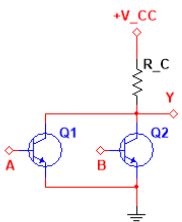
$Y = \overline{A+B}$

NOR Gate

- Two or more input signals but only one output signal.
- Output is low if **any or all** of the inputs are high.
- Logical Symbol
- Boolean expression

$Y = \overline{A+B}$

NOR Gate



CASE 1:
A is low and B is low.

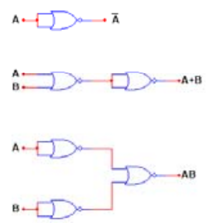
CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

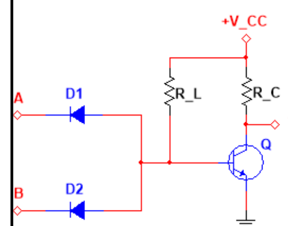
CASE 4:
A is high and B is high.

NOR as a Universal gate …..

- NOT Gate
- OR Gate
- AND Gate



NAND Gate



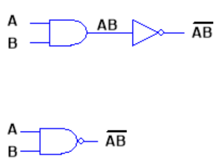
CASE 1:
A is low and B is low.

CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

CASE 4:
A is high and B is high.

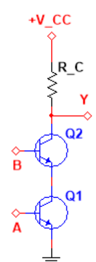
NAND Gate



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

$Y = \overline{AB}$

NAND Gate



CASE 1:
A is low and B is low.

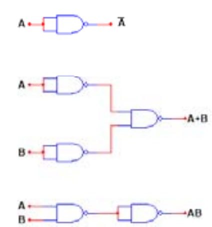
CASE 2:
A is low and B is high.

CASE 3:
A is high and B is low.

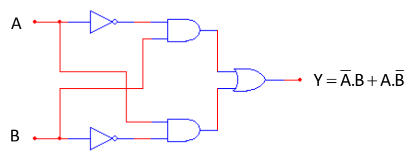
CASE 4:
A is high and B is high.

NAND as a Universal gate …..

- NOT Gate
- OR Gate
- AND Gate



Ex-OR Gate

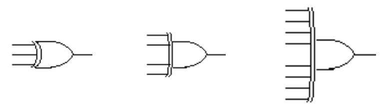


Ex-OR Gate

$Y = \bar{A}.B + A.\bar{B}$

$Y = A \oplus B$

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

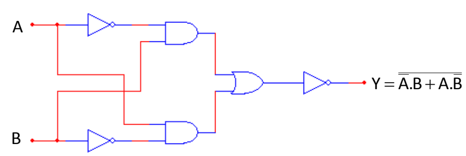


Ex-OR Gate

- Two or more input signals but only one output signal.
- Output is high if the inputs are **different**.
- Logical Symbol
- Boolean expression

$Y = A \oplus B$

Ex-NOR Gate

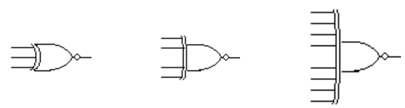


Ex-NOR Gate

$Y = \overline{\bar{A}.B + A.\bar{B}}$

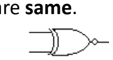
$Y = \overline{A \oplus B}$

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



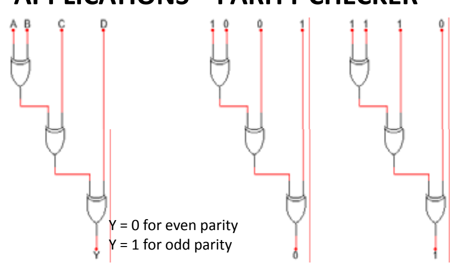
Ex-NOR Gate

- Two or more input signals but only one output signal.
- Output is high if the inputs are **same**.
- Logical Symbol
- Boolean expression

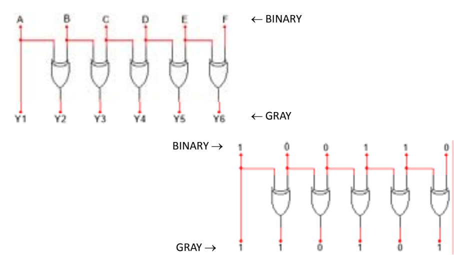


$Y = \overline{A \oplus B}$

APPLICATIONS – PARITY CHECKER



APPLICATIONS – BINARY TO GRAY CODE CONVERTER



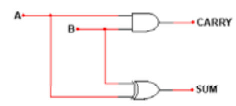
Logic Circuits

ARITHMETIC CIRCUITS

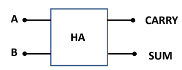
HALF ADDER

- An elementary circuit.
- Used to adds two binary digits at a time.
- It performs binary addition
- Consists of an **XOR gate** whose output gives the **sum** and an **AND gate** whose output gives the **carry** of addition.

HALF ADDER



A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



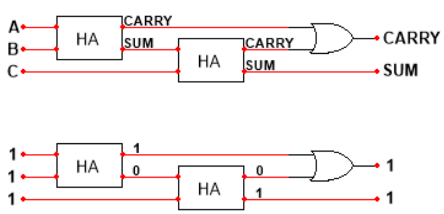
CARRY = A.B

SUM = A ⊕ B

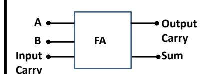
FULL ADDER

- Used to add **three binary digits** at a time.
- It performs binary addition of two bits and a carry.
- Consists of two half adders and an OR gate.

FULL ADDER



FULL ADDER



C_{in}	A	B	C_{out}	SUM
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

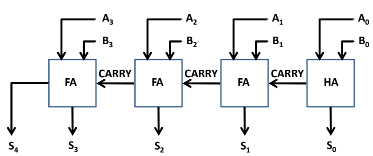
PARALLEL BINARY ADDER

- Used to add two binary numbers. For example

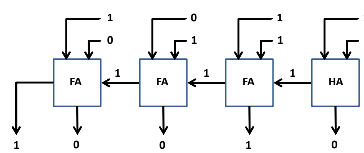
$$\begin{array}{r} A_3A_2A_1A_0 \\ \underline{B_3B_2B_1B_0} \\ S_4S_3S_2S_1S_0 \end{array}$$

- First column needs only a Half Adder. For any other column (except First), use Full Adder.

4 BIT PARALLEL BINARY ADDER



4 BIT PARALLEL BINARY ADDER

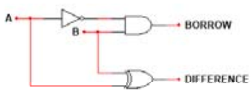


$$\begin{array}{r} 1011 \\ \underline{0111} \\ 10010 \end{array}$$

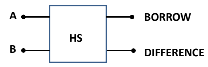
HALF SUBTRACTOR

- An elementary circuit used to subtract one binary digit from another.
- It performs binary subtraction handles two bits at a time and can be used for the LSB column.
- Consists of an **XOR gate** whose output gives the **difference** and an **AND gate** whose output gives the **borrow** of subtraction.
- One input of AND gate inverts itself using the **NOT gate**.

HALF SUBTRACTOR



A	B	Borrow	Difference
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



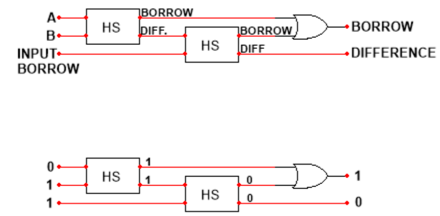
BORROW = $\bar{A}.B$

DIFFERENCE = $A \oplus B$

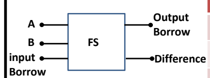
FULL SUBTRACTOR

- Used to subtract **two binary digits** with **one borrow digit** at a time.
- It performs binary subtraction of two bits and a borrow and can be used for higher order columns.
- Consists of two half subtractors and an OR gate.

FULL SUBTRACTOR



FULL SUBTRACTOR



Bo_{in}	A	B	Bo_{out}	Differ
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

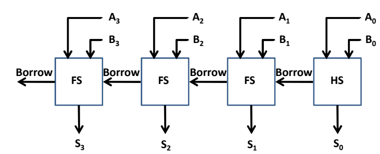
PARALLEL BINARY SUBTRACTOR

- Used to subtract two binary numbers. For example

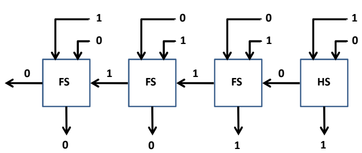
$$A_3A_2A_1A_0$$
$$\underline{B_3B_2B_1B_0}$$
$$S_3S_2S_1S_0$$

- First column needs only a Half Subtractor. For any other column (except First), use Full Subtractor.

4 BIT PARALLEL BINARY SUBTRACTOR



4 BIT PARALLEL BINARY SUBTRACTOR



1001
<u>0110</u>
0011

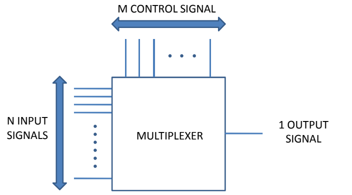
Logic Circuits

DATA PROCESSING CIRCUITS

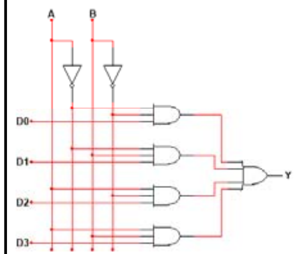
MULTIPLEXERS

- Multiplex means **many into one**.
- A circuit with **many inputs** but only **one output**.
- By applying **control signals**, only one selected input can be transmitted to the output.
- Also called a **data selector**, because the output bit depends on the input data bit that is selected.

MULTIPLEXERS



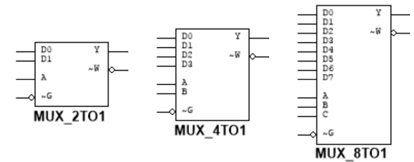
Four-to-one Multiplexer



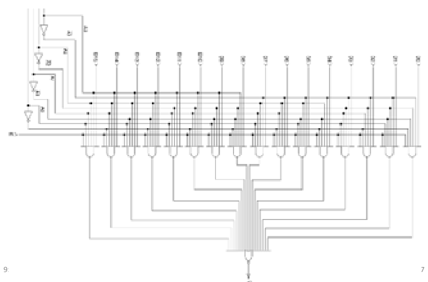
A	B	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$Y = D_n$
where n is decimal equivalent of AB

MULTIPLEXERS

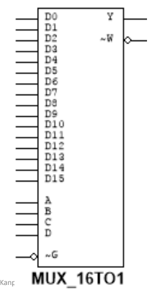


SIXTEEN-to-one Multiplexer



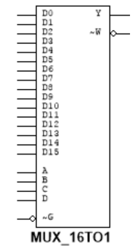
Sixteen-to-one Multiplexer

Strobe G	A	B	C	D	Y
0	0	0	0	0	D_0
0	0	0	0	1	D_1
0	0	0	1	0	D_2
0	0	0	1	1	D_3
0	0	1	0	0	D_4
0	0	1	0	1	D_5
0	0	1	1	0	D_6
0	0	1	1	1	D_7
0	1	0	0	0	D_8
0	1	0	0	1	D_9
0	1	0	1	0	D_{10}
0	1	0	1	1	D_{11}
0	1	1	0	0	D_{12}
0	1	1	0	1	D_{13}
0	1	1	1	0	D_{14}
0	1	1	1	1	D_{15}
1	x	x	x	x	H

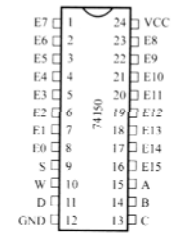


$Y = D_n$

SIXTEEN-to-one Multiplexer - 74150



$W = \overline{D}_n$



SIXTEEN-to-one Multiplexer - 74150

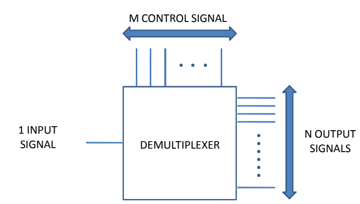
Inputs					Outputs
Select				Strobe	W
D	C	B	A	S	
X	X	X	X	H	H
L	L	L	L	L	$\overline{E0}$
L	L	L	H	L	$\overline{E1}$
L	L	H	L	L	$\overline{E2}$
L	L	H	H	L	$\overline{E3}$
L	H	L	L	L	$\overline{E4}$
L	H	L	H	L	$\overline{E5}$
L	H	H	L	L	$\overline{E6}$
L	H	H	H	L	$\overline{E7}$
H	L	L	L	L	$\overline{E8}$
H	L	L	H	L	$\overline{E9}$
H	L	H	L	L	$\overline{E10}$
H	L	H	H	L	$\overline{E11}$
H	H	L	L	L	$\overline{E12}$
H	H	L	H	L	$\overline{E13}$
H	H	H	L	L	$\overline{E14}$
H	H	H	H	L	$\overline{E15}$

$W = \overline{E}_n$

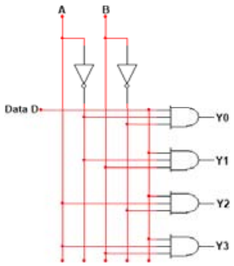
DEMULTIPLEXERS

- Demultiplex means **one into many**.
- A circuit with only **one input** and **many outputs**.
- By applying **control signals**, input signal can be transmitted to any one of the output lines.

DEMULTIPLEXERS



One-to-four Demultiplexer

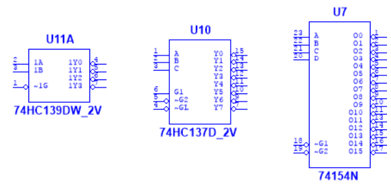


A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$Y_n = D$
where n is decimal equivalent of AB

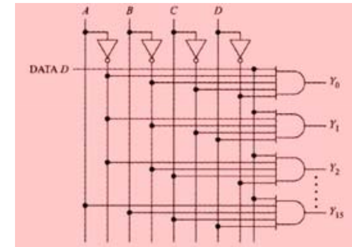
9:18:18 PM Satish Chandra, P P N College, Kanpur 73

DEMULTIPLEXERS



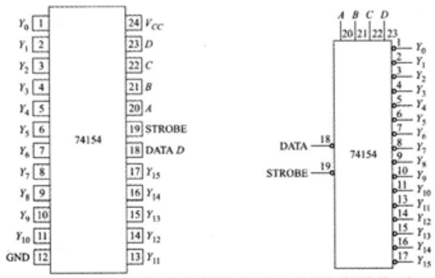
9:18:18 PM Satish Chandra, P P N College, Kanpur 74

One-to-sixteen Demultiplexer



9:18:18 PM Satish Chandra, P P N College, Kanpur 75

One-to-Sixteen Demultiplexer - 74154



9:18:18 PM Satish Chandra, P P N College, Kanpur 76

One-to-sixteen Demultiplexer

Strobe	Data	A	B	C	D	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

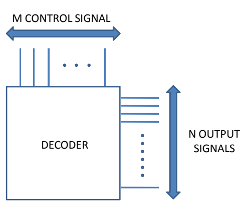
9:18:18 PM Satish Chandra, P P N College, Kanpur 77

DECODER

- Decoder is similar to demultiplexer, with one exception – there is no data input.
- Only inputs are the control bits.

9:18:18 PM Satish Chandra, P P N College, Kanpur 78

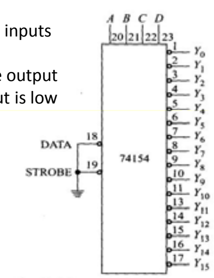
DECODER



9:18:18 PM Satish Chandra, P P N College, Kanpur 79

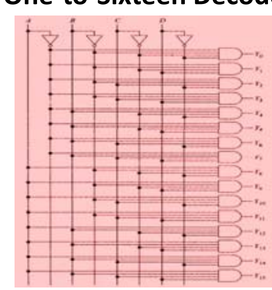
One-to-Sixteen Decoder

- DATA and STROBE inputs are grounded.
- The bubbles at the output shows that the output is low when it is active.



9:18:18 PM Sati

One-to-Sixteen Decoder



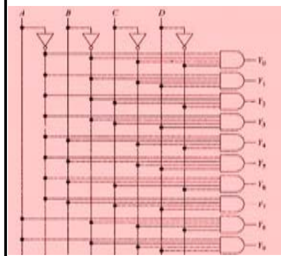
9:18:18 PM Satish Chandra, P P N College, Kanpur 81

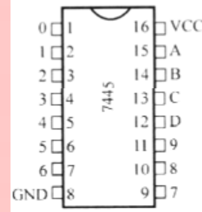
BCD TO DECIMAL DECODERS

- BCD stands for Binary-Coded-Decimal.
- A BCD number is a four-bit binary group that represents one of the ten decimal digits 0 through 9.
- Example: Decimal number 4926

Decimal	4	9	2	6
BCD	0100	1001	0010	0110

BCD TO DECIMAL DECODERS





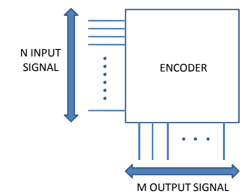
BCD TO DECIMAL DECODERS

No.	INPUTS				OUTPUTS									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
INVALID	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H

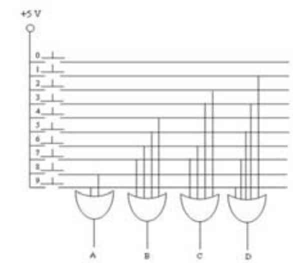
ENCODER

- An encoder converts non-digital data to digital data.
- An encoder has $n = 2^m$, input lines and m output lines.
- Only one of the input line will be active.
- The output lines generate a binary code corresponding to the that input line.

ENCODER



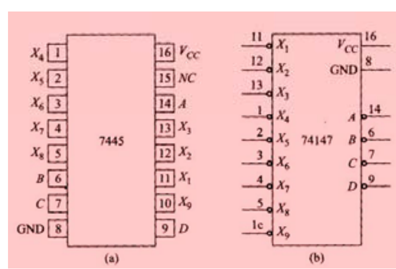
DECIMAL TO BCD ENCODER



DECIMAL TO BCD ENCODER

- The decimal to BCD encoder with 10 inputs and 4 outputs. The switches 0 to 9 are push button switches like a pocket calculator.
- When button 3 is pressed, the C and D OR gates have high inputs, therefore output is ABCD=0011 i.e. decimal 3 is converted into its equivalent binary number 0011.
- Similarly, if you press button 5, the output will be ABCD=0101 and for switch 9, we get ABCD=1001

DECIMAL TO BCD ENCODER - 74147



DECIMAL TO BCD ENCODER - 74147

TRUTH TABLE

Inputs									Outputs			
X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	A	B	C	D
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L