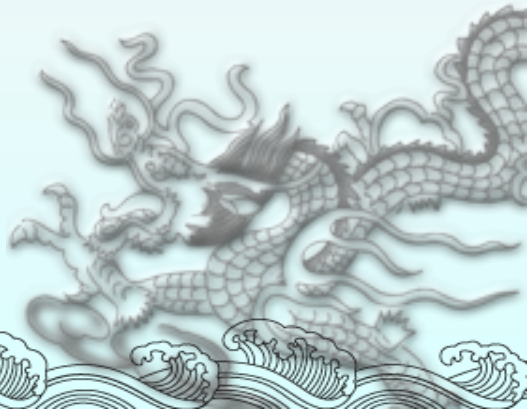
CPSC 855 Embedded Systems

Combinational Logic Circuits

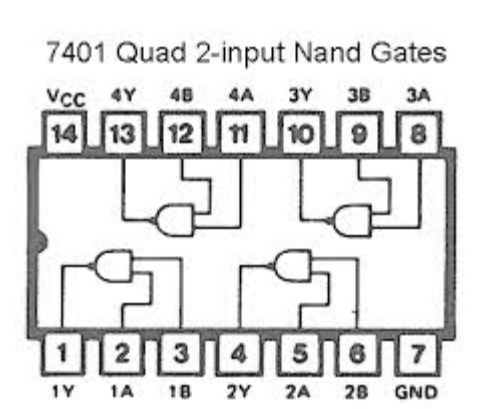
Fryad M. Rashid and Pei-Lin Chung

Outline

- Design Combinational Logic Circuit for scenario
- Adder
- Subtractor
- Comparator
- Multiplexer
- Demultiplexer
- Encoder
- Decoder
- Code Conversions
- Implementation

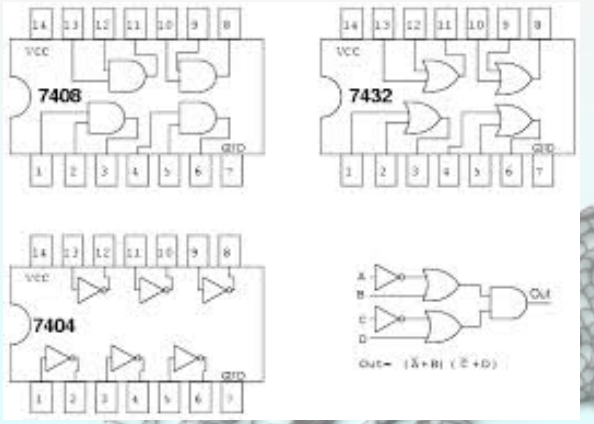


Design Methods

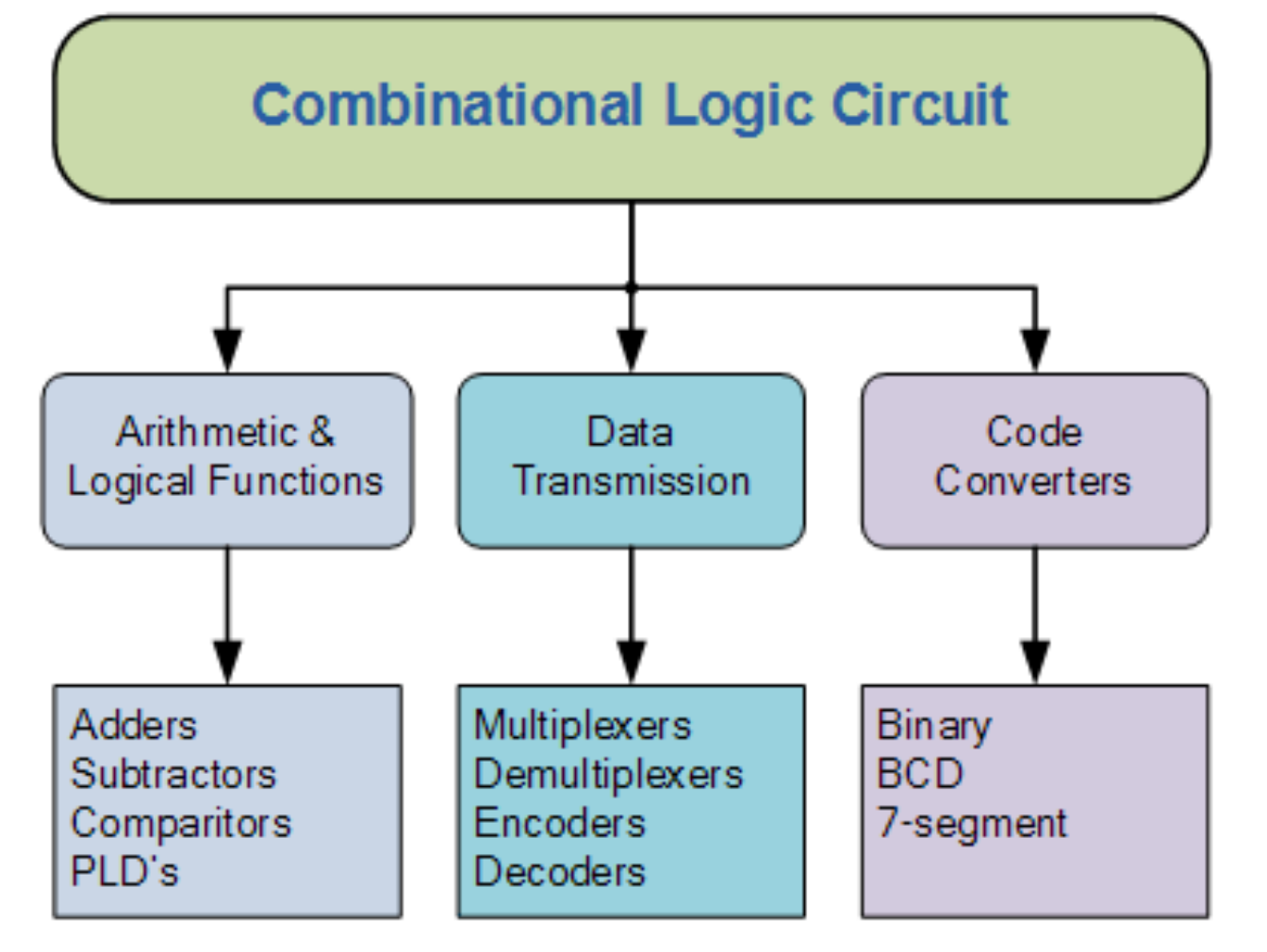


- Type of IC chips (based on packing density) :
  - Small-scale integration (SSI): up to 12 gates
  - Medium-scale integration (MSI): 12-99 gates
  - Large-scale integration (LSI): 100-9999 gates
  - Very large-scale integration (VLSI): 10,000-99,999 gates
  - Ultra large-scale integration (ULSI): > 100,000 gates

- Main objectives of circuit design:
  - (i) reduce cost
    - reduce number of gates (for SSI circuits)
    - reduce IC packages (for complex circuits)
  - (ii) increase speed
  - (iii) design simplicity (reuse blocks where possible)



Application of CLC



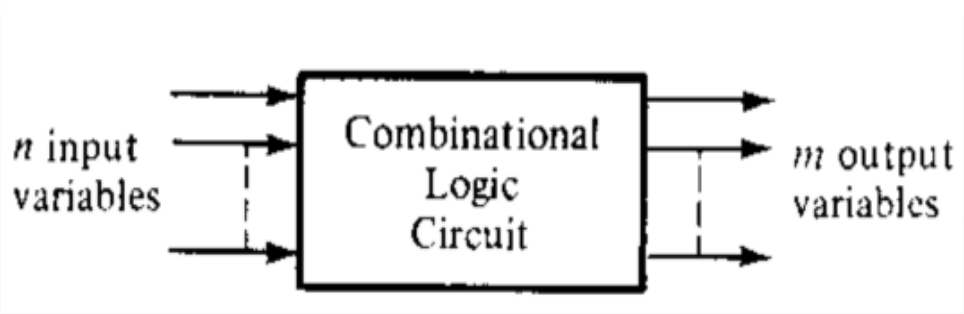
Introduction

LOGIC CIRCUITS:
1. Combinational
2. Sequential

Combinational Logic Circuits (Circuits without a memory): In this type of logic circuits outputs depend only on the current inputs.

Sequential Logic Circuits (Circuits with memory): In this type of logic circuits outputs depend on the current inputs and previous inputs. These circuits employ storage elements and logic gates.

Combinational Logic Circuits

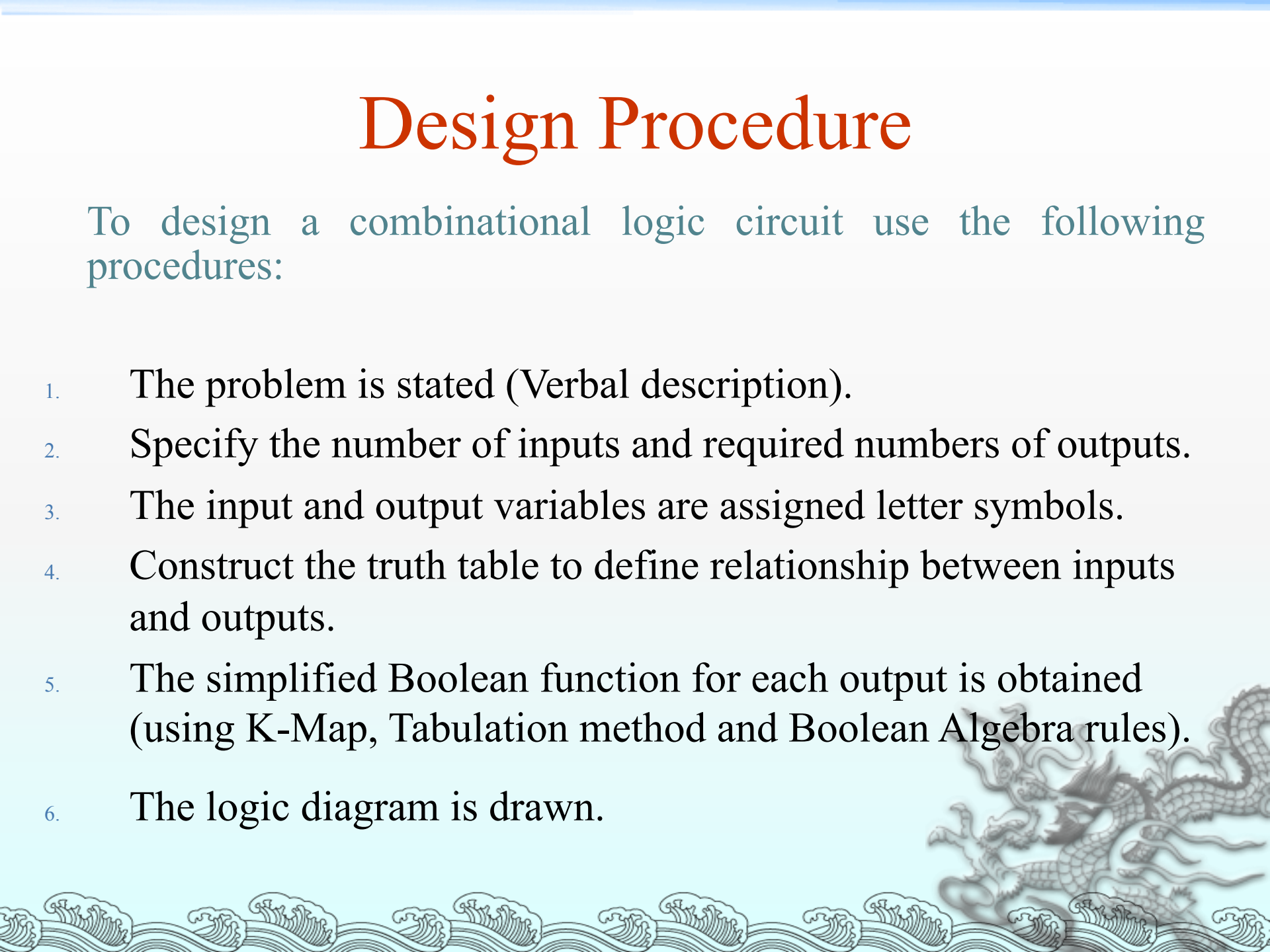


- A combinational circuit consists of input variables (n), logic gates, and output variables (m).
- For (n) input variables there are 2^n possible combinations of binary input values.
- For each possible input combination there is one and only one possible output combination, a combinational circuit can be describe by (m) Boolean functions one for each output variable.
- Each output function expressed in terms of the (n) input variables.

Design Procedure

To design a combinational logic circuit use the following procedures:

1. The problem is stated (Verbal description).
2. Specify the number of inputs and required numbers of outputs.
3. The input and output variables are assigned letter symbols.
4. Construct the truth table to define relationship between inputs and outputs.
5. The simplified Boolean function for each output is obtained (using K-Map, Tabulation method and Boolean Algebra rules).
6. The logic diagram is drawn.



Practical Design

A practical design method would have to consider such constrains as:

1. Min. no. of gates.
2. Min. no. of inputs to gates.
3. Min. no. of interconnections.
4. Min. propagation time of the signal throw the circuit.

Simple Design – Remind You

Example: Simplify two inputs OR gate truth table by using K-Map?

1.Problem is stated.

2.No. of inputs are two (X and Y) & No. of required output is (F)

3.Construct truth table (F= X+Y), inputs =2 ➔ 2^2=4 possibilities

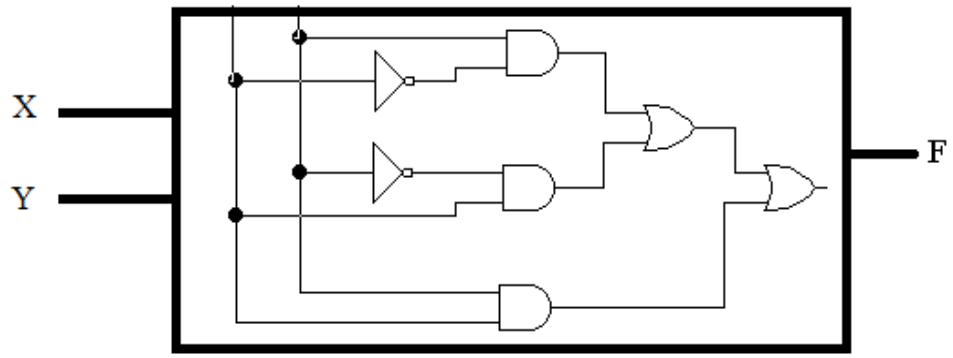
4.Using K-Map to simplify the circuit.

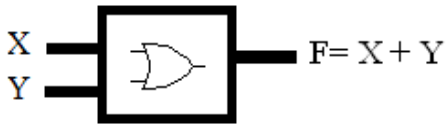
5.Final design.

F= X + Y .... (Two inputs OR gate equation)

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

X \ Y	0	1
0	0	1
1	1	1





$F = \bar{X}Y + X\bar{Y} + XY$

Design for scenario

- A committee of three individuals decide issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least two yes votes. Design a circuit that determines whether a proposal passes.

Solution

Inputs are three (x, y, z) , Output is proposal (F)

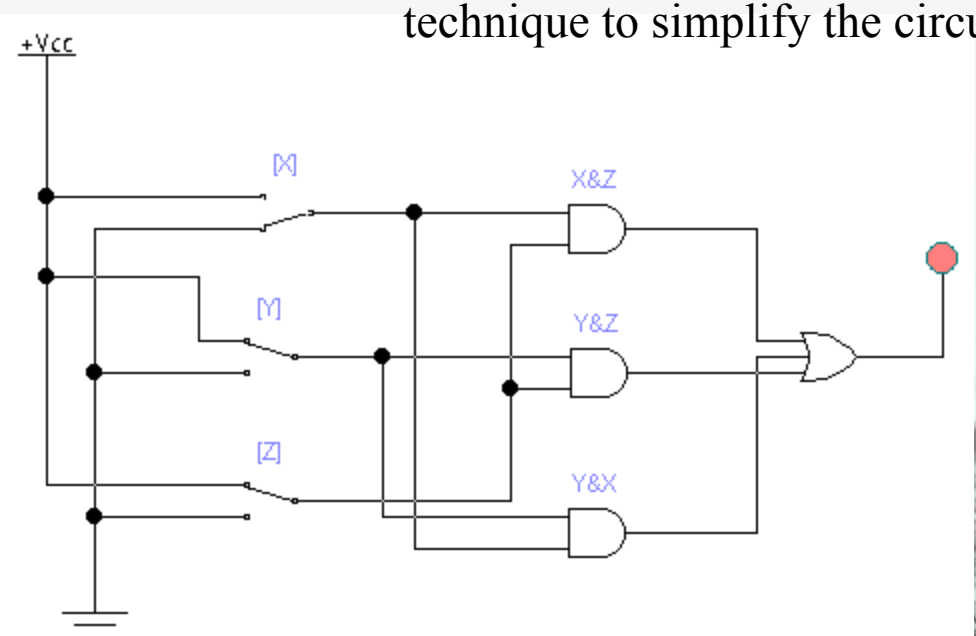
Individual1 (X)	Individual2 (Y)	Individual3 (Z)	Proposal(F)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

X \ YZ	00	01	11	10
0			1	
1		1	1	1

$$F = XY + XZ + YZ$$

We used K-Map minimization technique to simplify the circuit.

We used truth table to make a relationship between inputs and output.



Adders

- Digital computers perform a variety of information processing tasks. Among the basic functions encountered are the various *arithmetic operations (addition)*.

Binary Arithmetic

1. Addition: The rules of addition are:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 10$

$1 + 1 + 1 = 11$

Binary Adder -Half Adder

Q/Design a combinational logic circuit that performs arithmetic operation for adding two bits?

Answer: n=2 bit , $n=2^2=4$

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

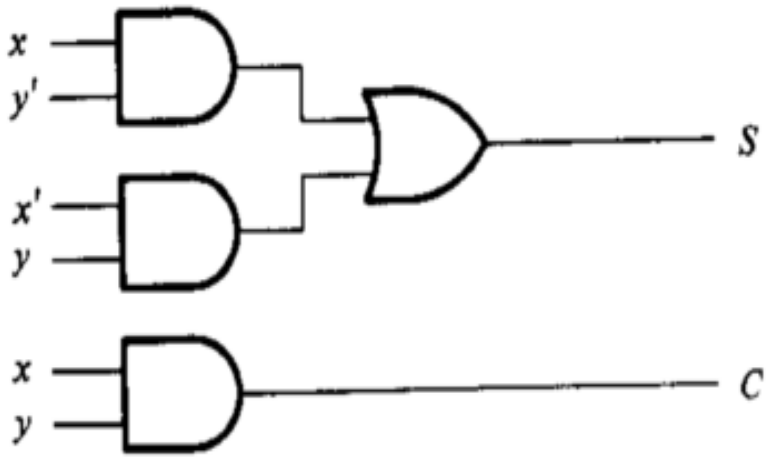
1 + 1 = 10

Inputs		Outputs	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

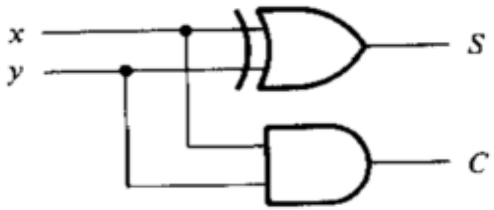
$$S = \bar{X}Y + X\bar{Y}$$

$$C = XY$$

The block diagram for the half adder is:



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

Binary Adder -Full Adder

Q/Design a combinational logic circuit that performs arithmetic operation for adding three bits?

Answer: n=3bit , $n=2^3=8$

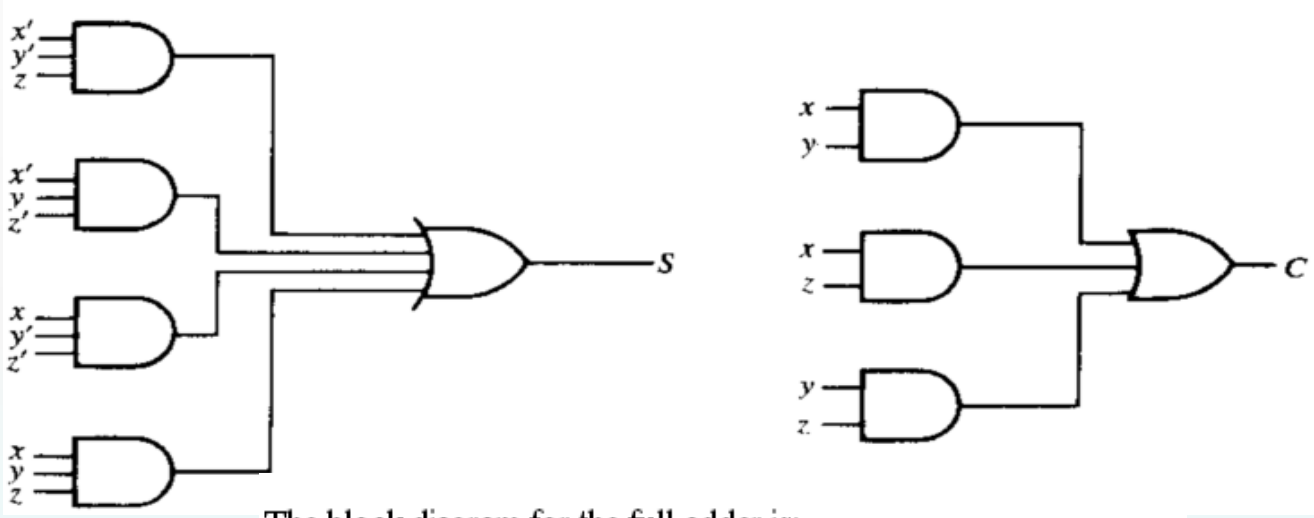
Inputs			Outputs	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

X \ YZ	00	01	11	10
0		1		1
1	1		1	

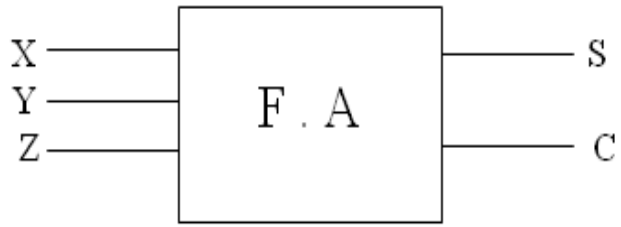
X \ YZ	00	01	11	10
0			1	
1		1	1	1

$S = \overline{X}\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ$

$C = XY + XZ + YZ$



The block diagram for the full adder is:



Subtractor

- Digital computers perform a variety of information processing tasks. Among the basic functions encountered are the various *arithmetic operations (Subtraction)*.

Binary Arithmetic

2. Subtraction: The rules of subtractions are:

$0 - 0 = 0$

$1 - 0 = 1$

$10 - 1 = 1$

$1 - 1 = 0$

Binary Subtractor- Half Subtractor

Q/Design a combinational logic circuit that performs arithmetic operation for subtracting two bits?

Answer: n= 2bit , $n=2^2=4$

0 – 0 =0

1 – 0 =1

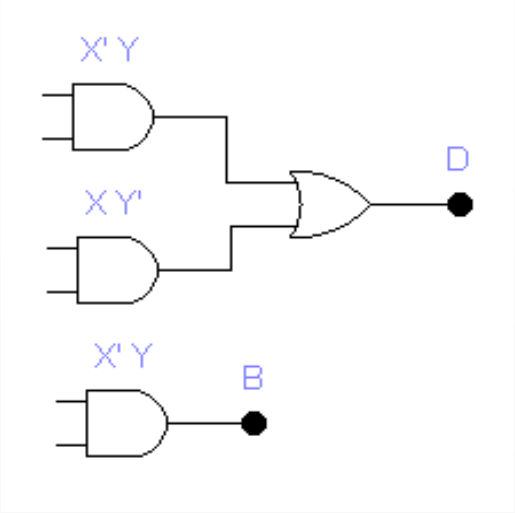
1 – 1 =0

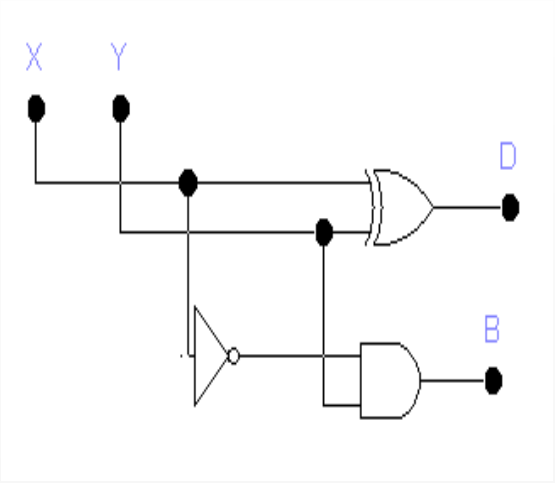
0 – 1 = 10 – 1 = 1 (The 1 borrowed from the next higher stage)

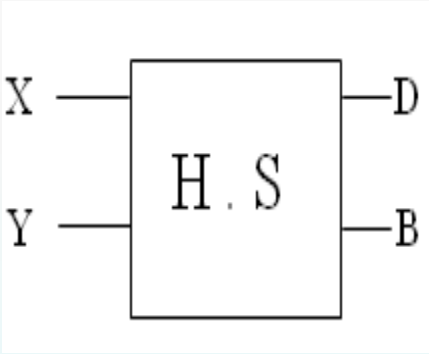
Inputs		Outputs	
X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$D = \bar{X}Y + X\bar{Y} = X \oplus Y$

$B = \bar{X}Y$


X' Y
D
X Y'
X' Y
B


X
Y
D
B

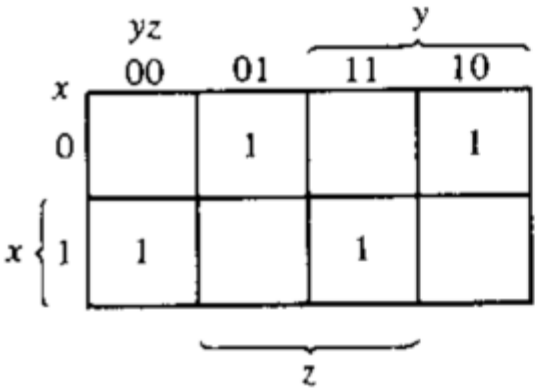

X
Y
H . S
D
B

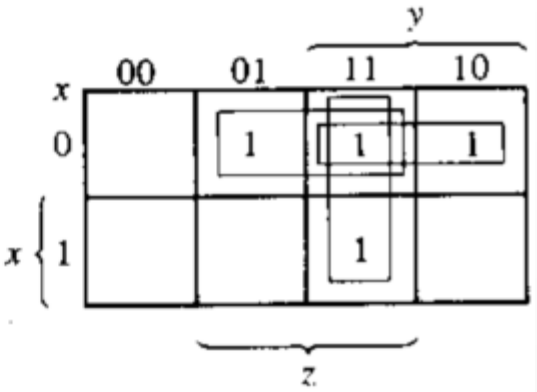
Binary Subtractor – Full Subtractor

Q/Design a combinational logic circuit that performs arithmetic operation for subtracting three bits?

Answer: n=3bits , $n=2^3=8$

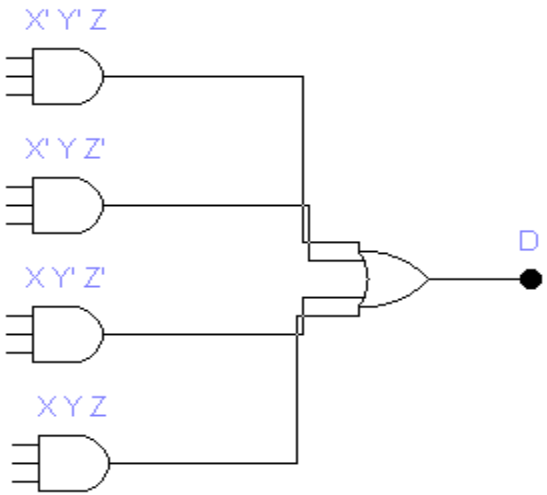
Inputs			Outputs	
X	Y	Z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

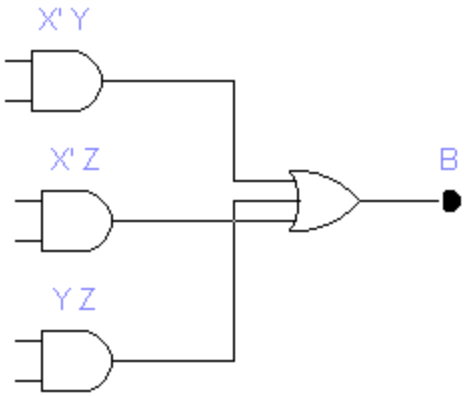


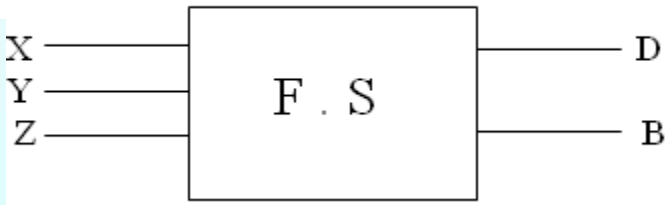


$D = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$

$B = \bar{X}Y + \bar{X}Z + YZ$







Comparator

The comparison of two numbers is an operation that determines if one number is greater than, less than, or equal to the other number. A comparator is a CLC that compares two numbers A, B, and determines their relative magnitudes. The outcome of the comparison is specified my three binary variables that indicate whether A>B, A=B, or A<B.

Example: Design a combinational logic circuit that compares two 1-bit numbers (A and B)?

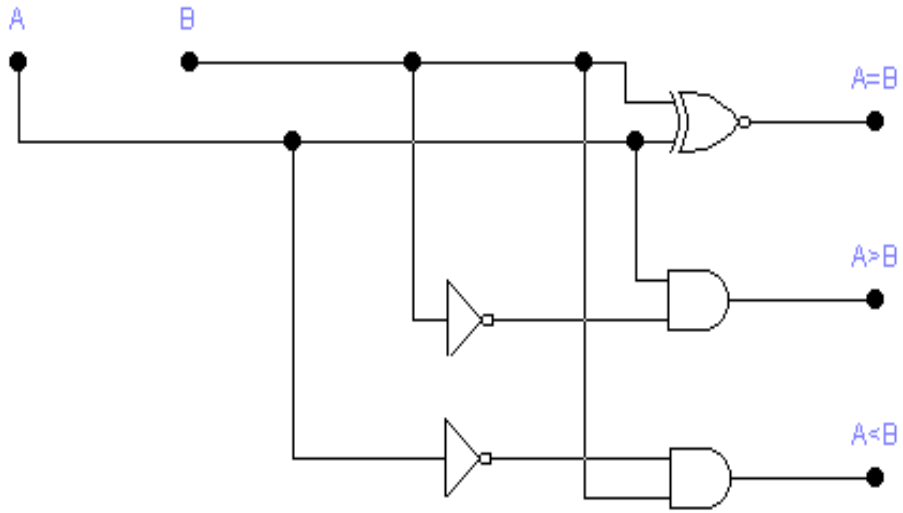
Solution: $A=A_0$ $B=B_0$

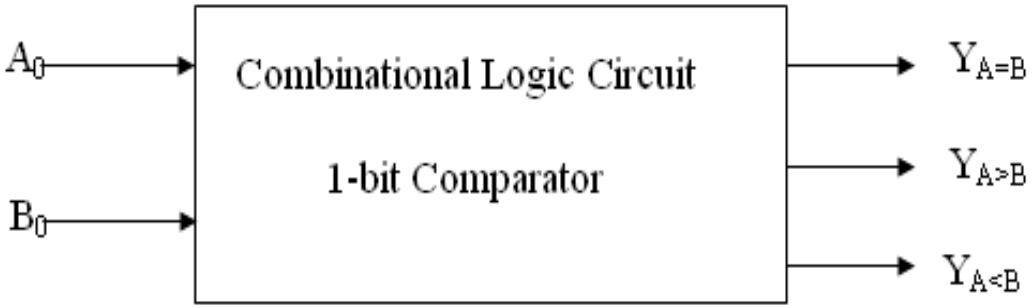
A_0	B_0	$Y_{A=B}$	$Y_{A>B}$	$Y_{A<B}$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$$Y_{A=B} = \overline{A}_0 \overline{B}_0 + A_0 B_0$$

$$Y_{A>B} = A_0 \overline{B}_0$$

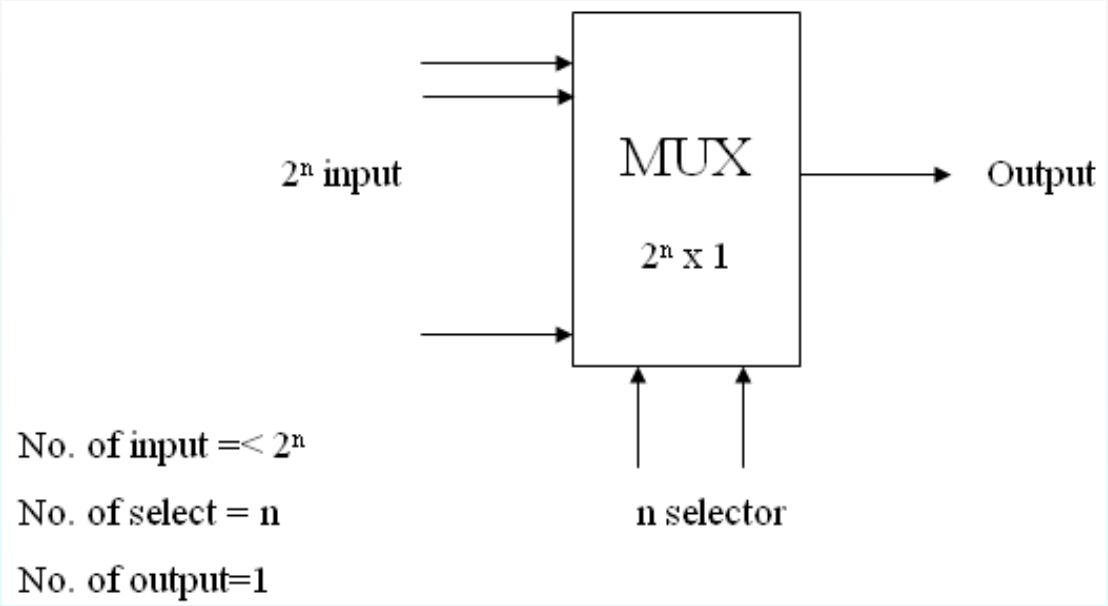
$$Y_{A<B} = \overline{A}_0 B_0$$





Multiplexer (Data Selector)

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is CLC that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by of a selection lines.



Design MUX:

AND gates used to represent inputs.

One **OR** gate only used to collect inputs.

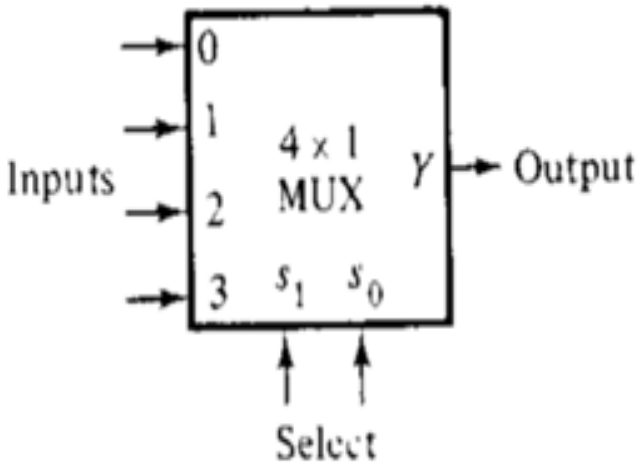
NOT gates as a selector to connect inputs to output.

Example: Design 4 x 1 multiplexer?

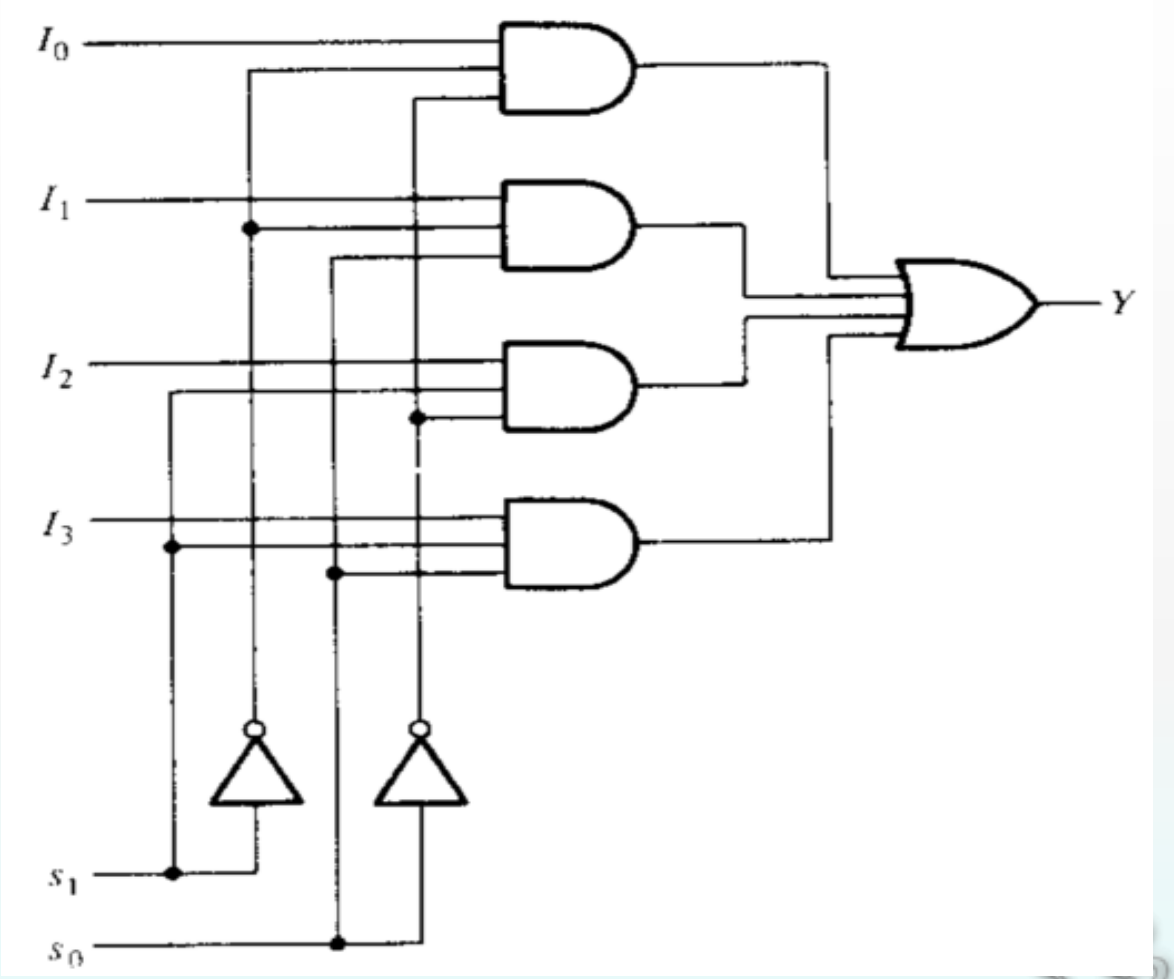
Solution: No. of inputs = $4 = 2^2$, No. of select=2, No. of output=1

Select		Output
s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Truth Table



Block diagram



4x1 Multiplexer Logic Diagram

E1 and T1 MUX/DMUX

E1:It is the European format for digital transmission. According to the ITU-T recommendations, it consists of 32 channels (2 channels are reserved for signaling and synchronization, 30 channels for carry voice calls and data communications, band width for each channel=64Kbps, data rate for E1=2048Kbps or 2.048Mbps).

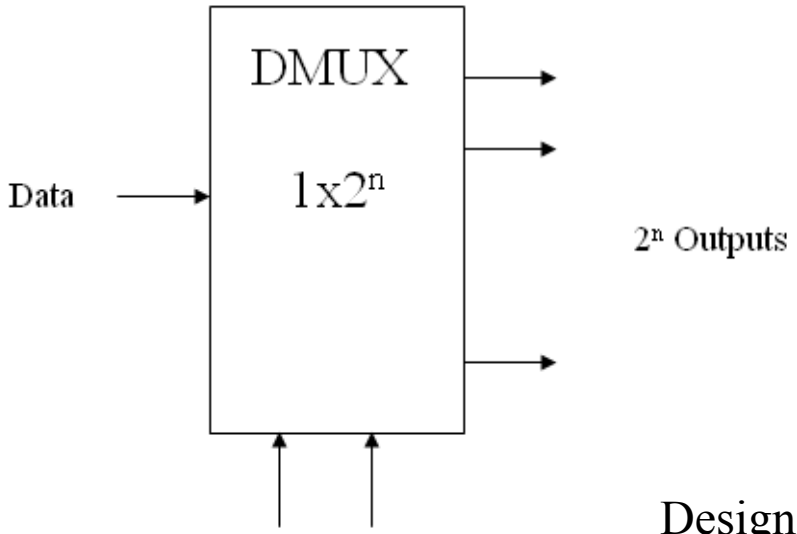
TDM is used for separate channels from each other. E1 is designed to send PCM voice signal (Sampling frequency= 8000 sample per second, E1 time frame= 1/8000 = 125µs, within this time frame we have 32 sample x 8 bit per sample = 256 bits. Therefore, Data Rate of E1= 2.048 Mbps(256bits/ 125µs).

E1 and T1 MUX/DMUX

T1 :T1 is the North American digital communication carrier standard that consists of 24 channels, which has 64Kbps bandwidth each. Initially each 64Kbps channel is designed to transfer pulse code modulated voice signals. T1 frame consists of 193 bits (24 samples x 8 bits per sample) that need to be transferred within 125µs. Therefore, data rate of T1 carrier is 1.544 Mbps (193 bits/125µs).

Demultiplexer

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs.



No. of inputs =1 (Data)

No. of outputs =<2^n

No. of select= n

Demultiplexer block diagram

Design DMUX:

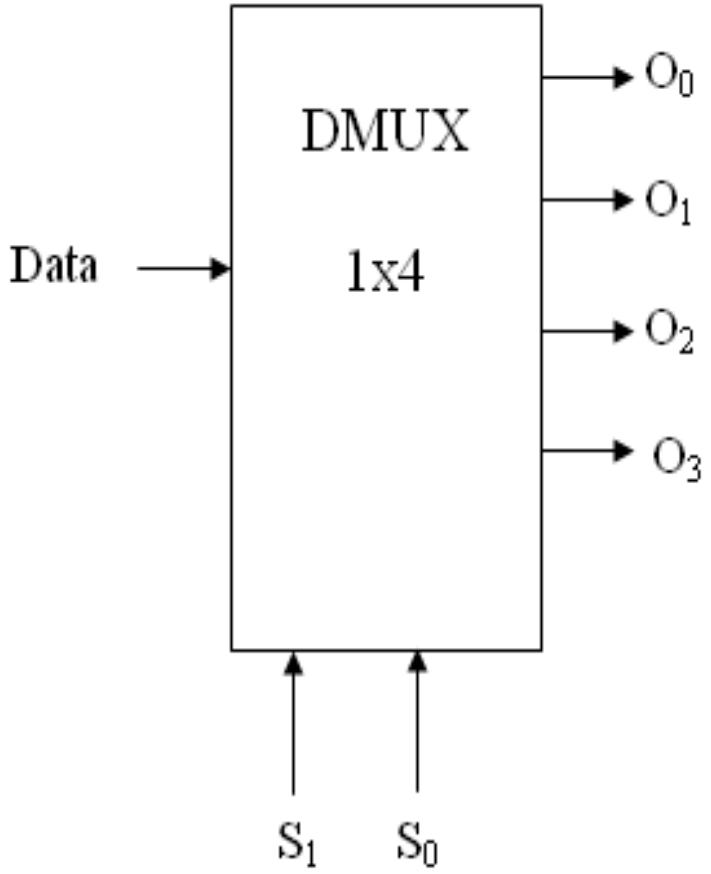
AND gates used to represent inputs.

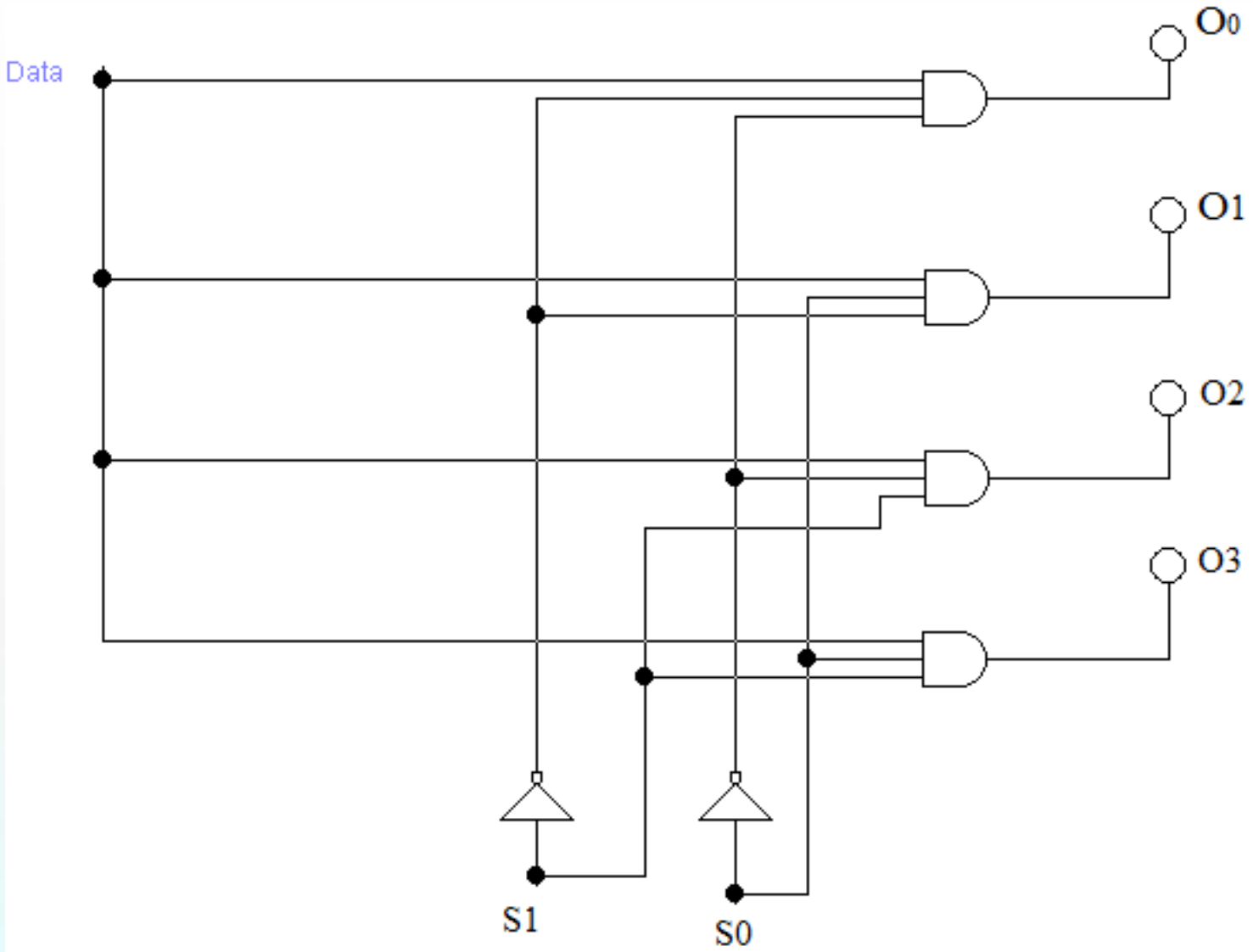
NOT gates as a selector to connect inputs to output.

Example: Design 1x4 demultiplexer?

Solution: No. of input=1, No. of output=4, No. of select=2

Select		Output			
S_1	S_0	O_0	O_1	O_2	O_3
0	0	Data	0	0	0
0	1	0	Data	0	0
1	0	0	0	Data	0
1	1	0	0	0	Data

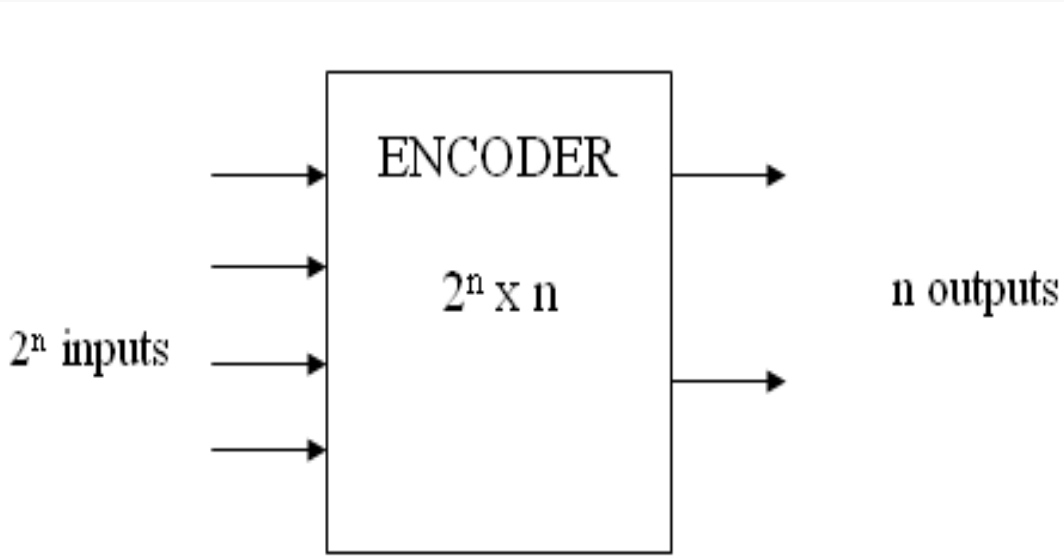




1x4 DMUX Logic Diagram

Encoder

An encoder is a device, circuit, software program, algorithm or person that converts information from one format or code to another. The purpose of encoder is standardization, speed, secrecy, security, or saving space by shrinking size. If a device output code has fewer bits than the input code has, the device is usually called an encoder.



Encoder Block diagram

No. of inputs =< 2^n
No. of outputs= n

Design ENC:

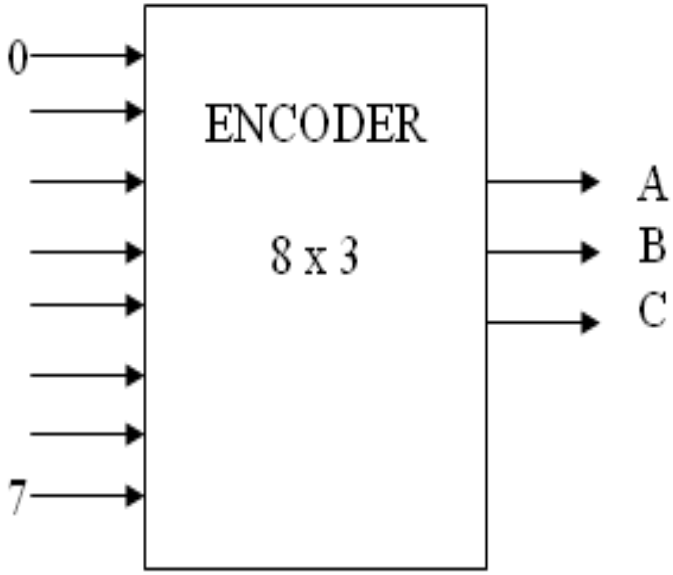
OR gates used to design encoder.

Example: Design 8x3 Encoder?

Solution: No. of inputs = $8 = 2^3$

No. of outputs = 3

Input	Output		
	A	B	C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

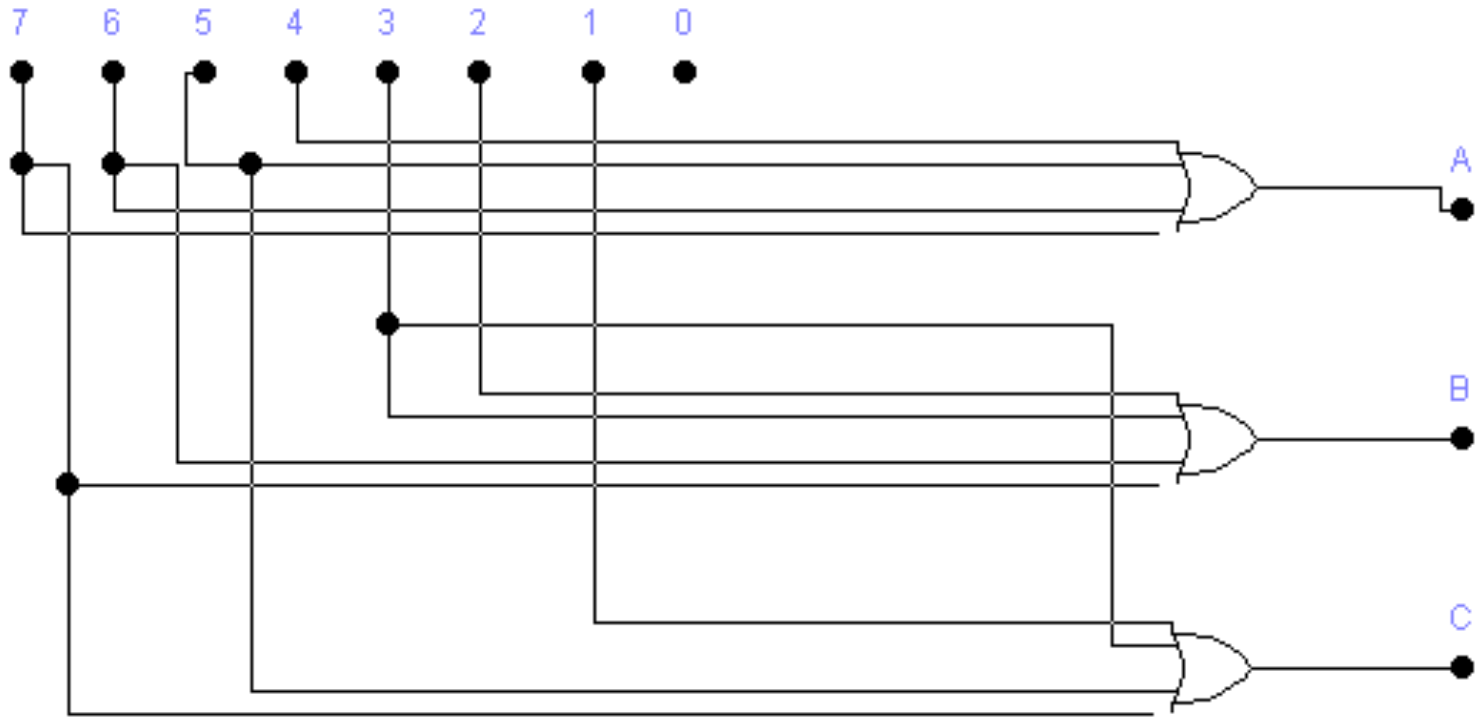


Octal to binary encoder

From the truth table: $A = \Sigma m\ (4, 5, 6, 7)$

$B = \Sigma m\ (2, 3, 6, 7)$

$C = \Sigma m\ (1, 3, 5, 7)$



8x3 Encoder logical diagram

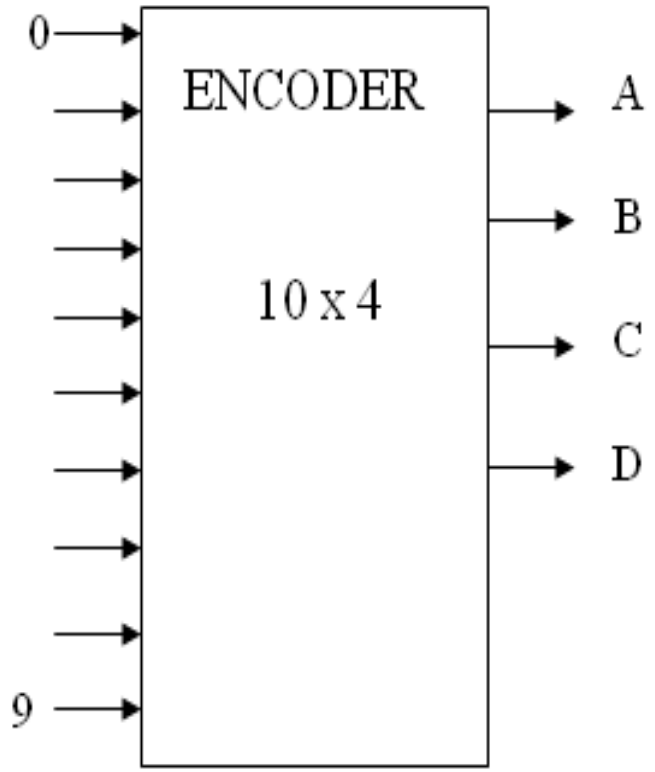
Example: Design a Decimal to BCD encoder?

Solution:

No. of inputs= 10

No. of outputs= 4

Input	Output			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

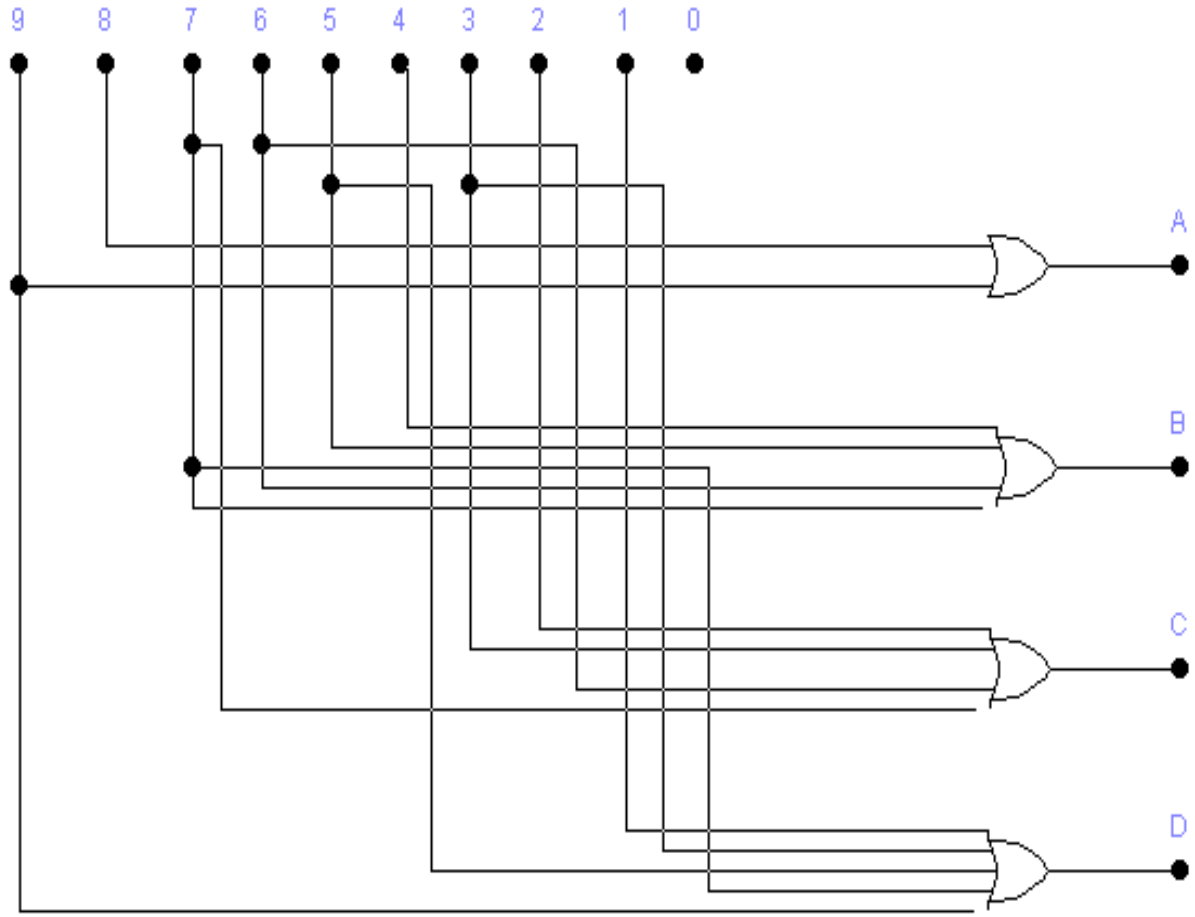


From the truth table: $A = \sum m\ (8, 9)$

$B = \sum m\ (4, 5, 6, 7)$

$C = \sum m\ (2, 3, 6, 7)$

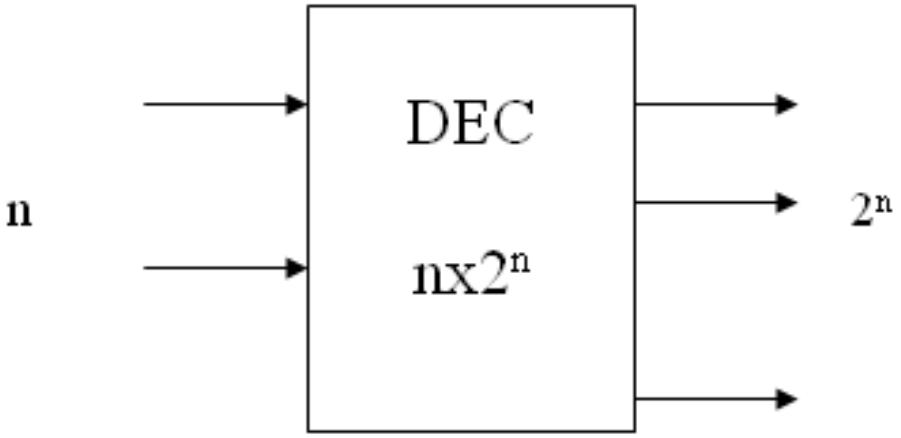
$D = \sum m\ (1, 3, 5, 7, 9)$



Decimal to BCD Encoder logical diagram

Decoder

A *decoder is* a combinational circuit that converts binary information from *n* input lines to a maximum of 2^n unique output lines.



Decoder Block Diagram

No. of inputs = n

No. of outputs =< 2^n

Design DEC:

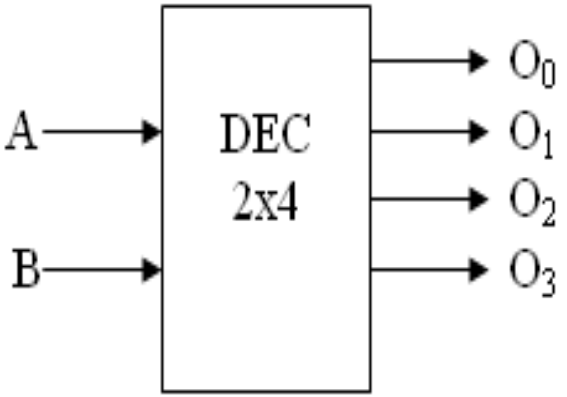
AND gates used to represent inputs.

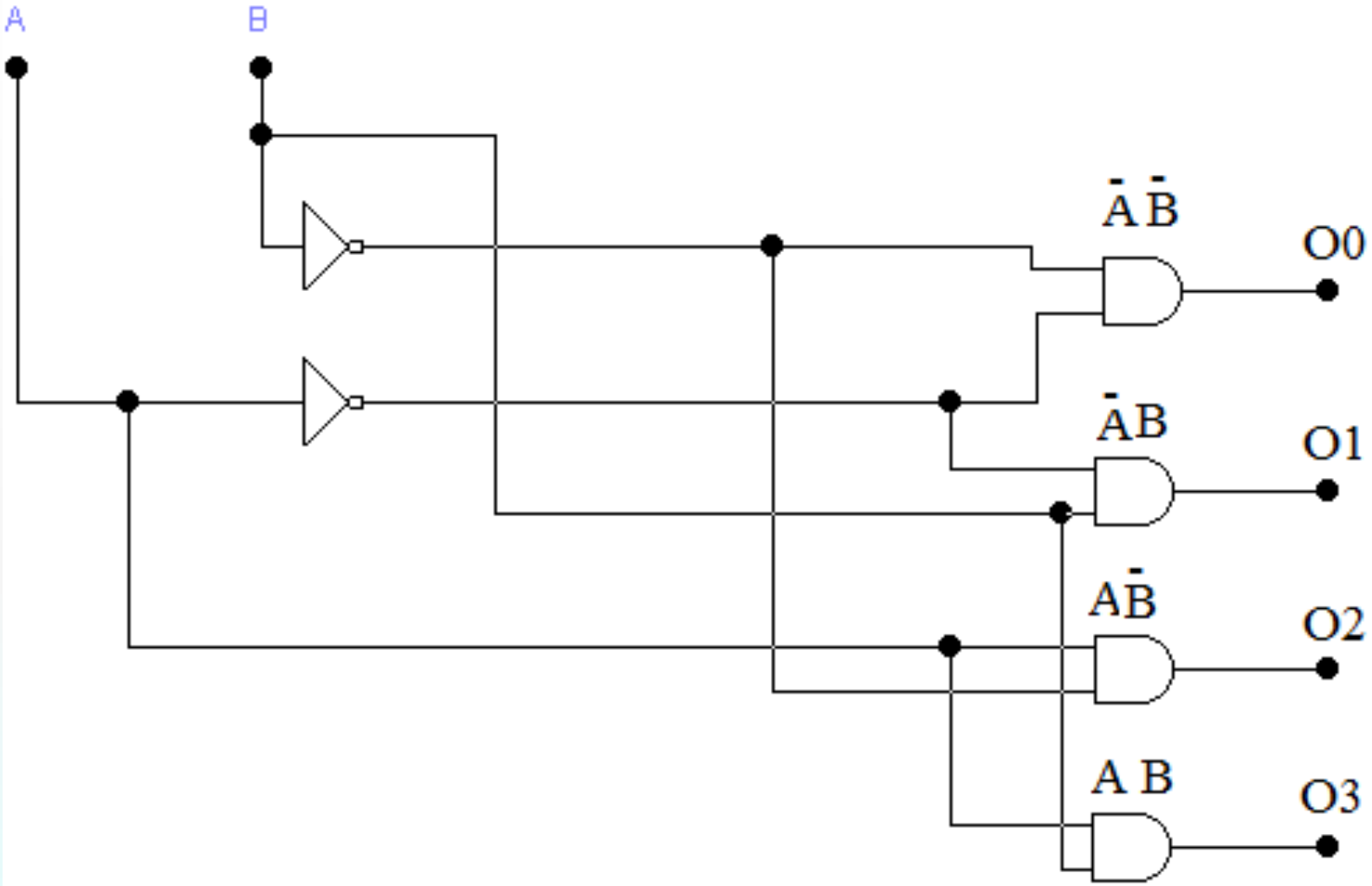
NOT gates to connect inputs to output.

Example: Design 2x4 decoder?

Solution: No. of inputs=2, No. of outputs=4

Inputs		Outputs			
A	B	O_0	O_1	O_2	O_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



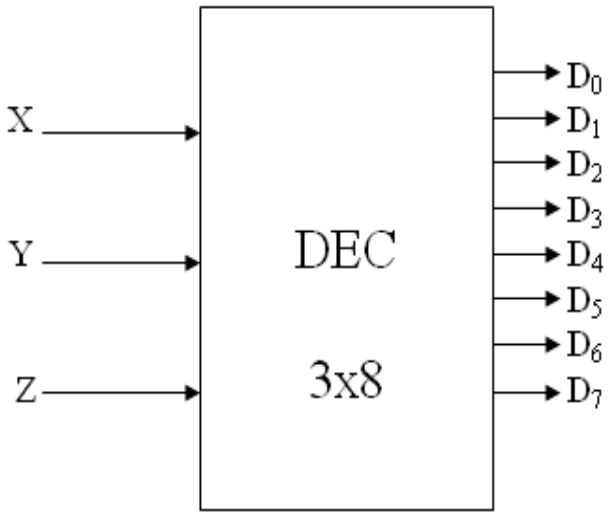


2x4 Decoder logical diagram

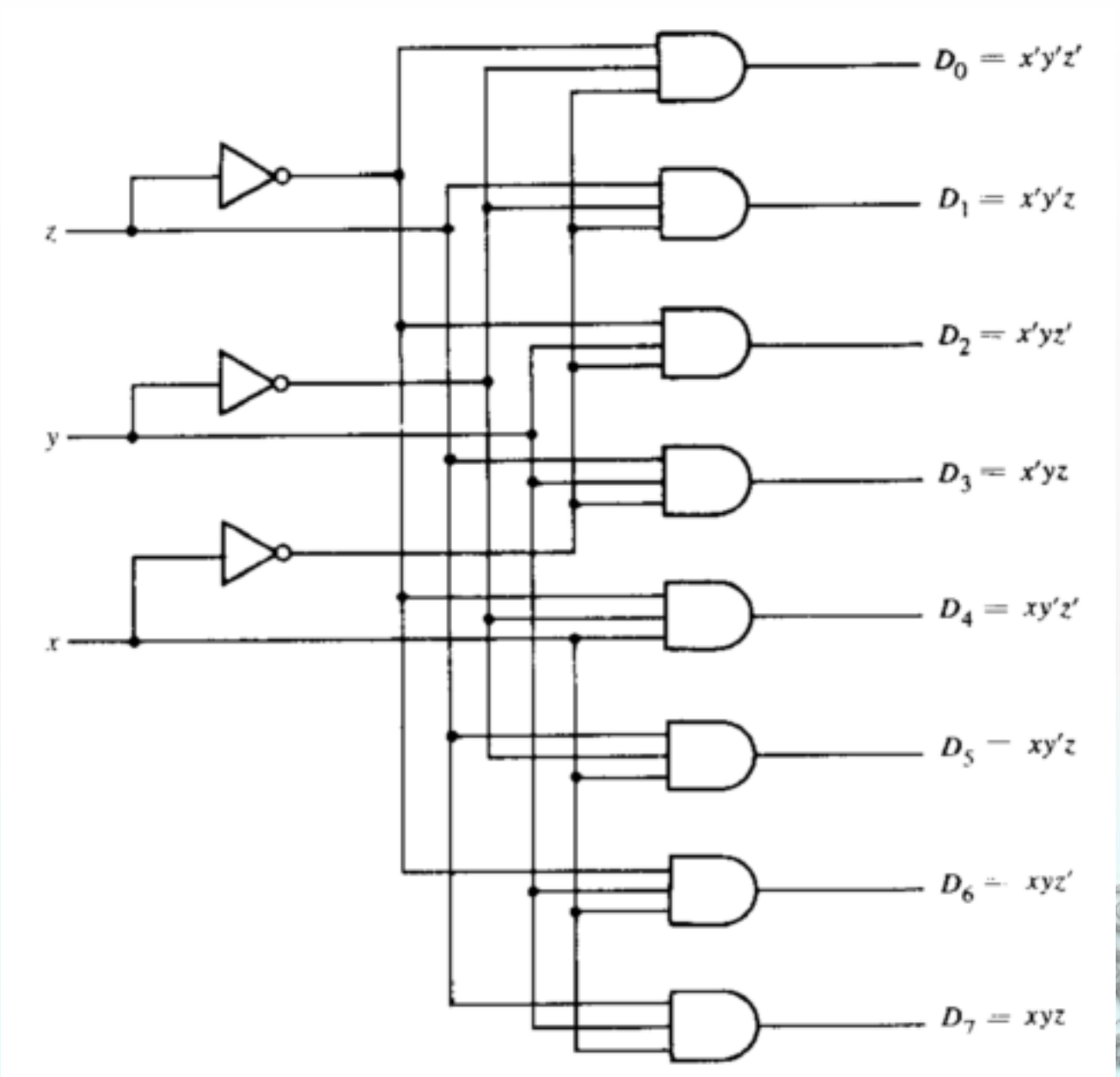
Example: Design 3x8 decoder?

Solution: No. of inputs= 3, No. of outputs= 8

Inputs			Outputs							
X	Y	Z	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



Binary to octal decoder



3x8 Decoder logical diagram

Code Converters :

Convert Binary to Gray Code

Example: Design a combinational logic circuit that converts 4bits binary to gray code?

Solution: n=4bits, $n=2^4=16$ possibilities

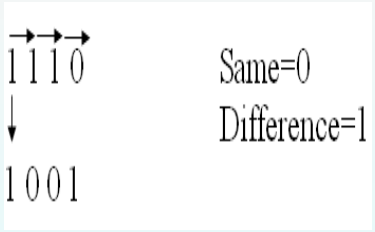
Input

Binary Code			
A	B	C	D
0	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	0	0	0
1	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

Output

Gray Code			
X	Y	Z	K
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

Advantage gray code over binary number?



CD / AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$X = A$

CD / AB	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

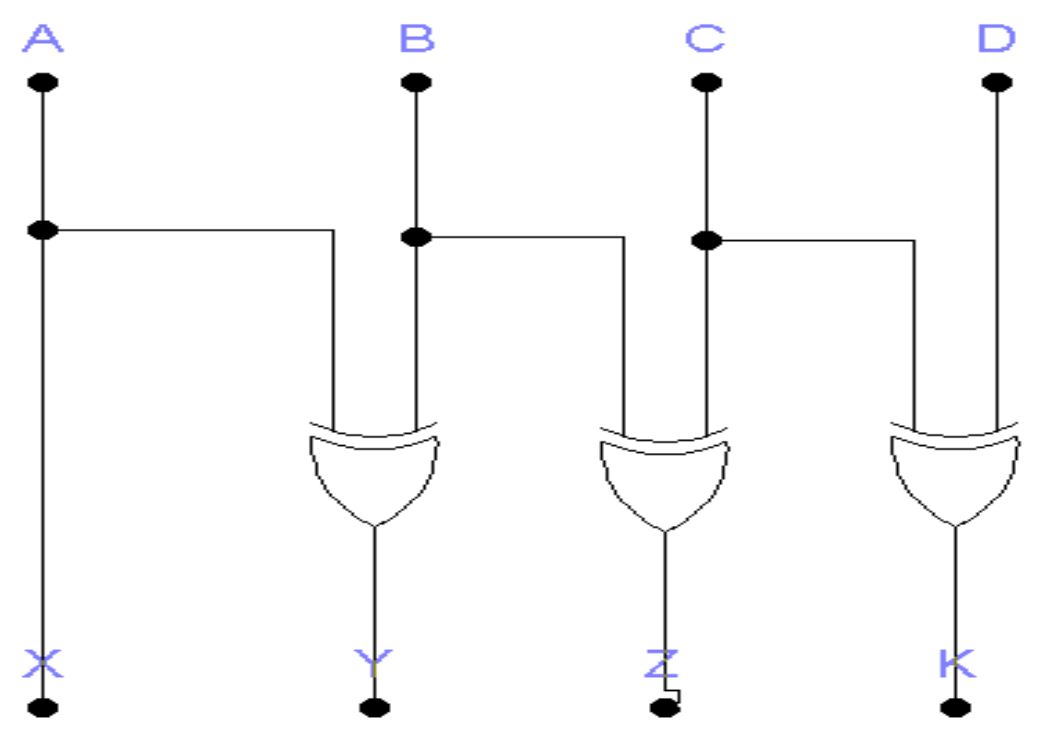
$Y = \overline{A}B + A\overline{B} = A \oplus B$

CD / AB	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$Z = B\overline{C} + \overline{B}C = B \oplus C$

CD / AB	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$K = \overline{C}D + C\overline{D} = C \oplus D$



4bit binary to gray code converter

Implementation

- Electronics Workbench Circuit Board Design and Simulation Software such as EWB, Multisim, ..etc

References

- http://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm
- http://www.differencebetween.com/difference-between-e1-and-vs-t1/
- http://coep.vlab.co.in/?sub=28&brch=81&sim=609&cnt=1