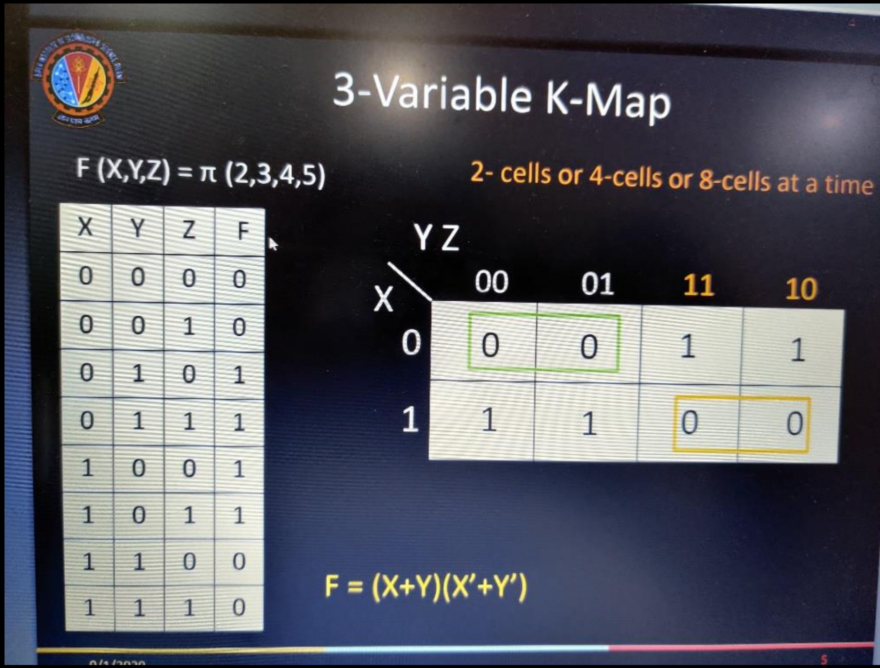
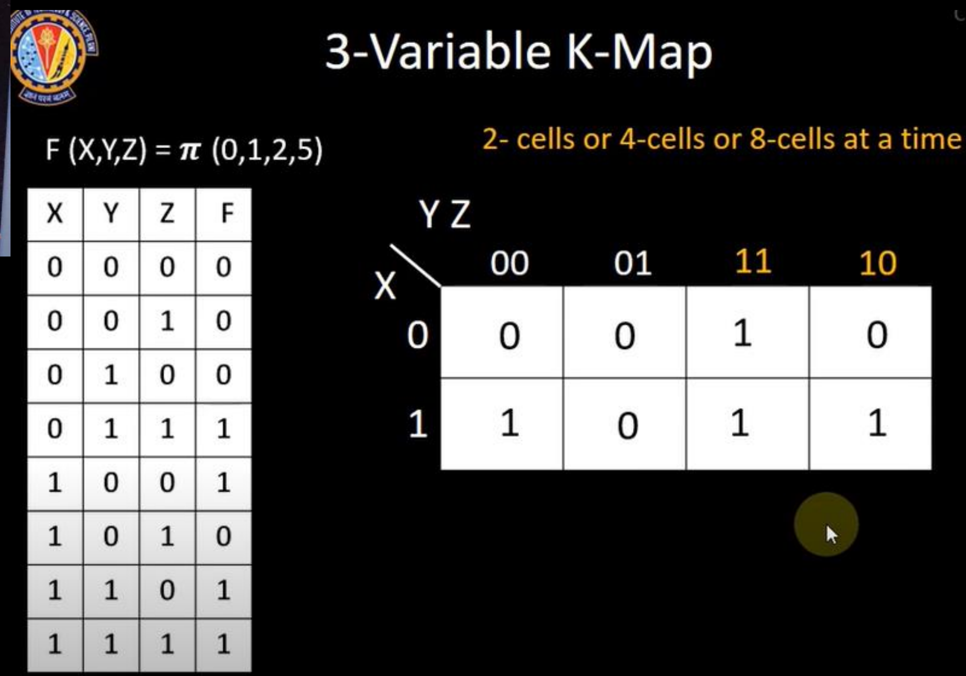
Digital Design

Lecture 7: Logic Gate Realization and Design





Shreya Guda 2019A7PS1202H

Eva Tiwari 2018B5A70816H

NAND-NOR Implementations

NAND gate and NOR gates are called Universal gates

Any logic can be implemented using NAND/NOR gates

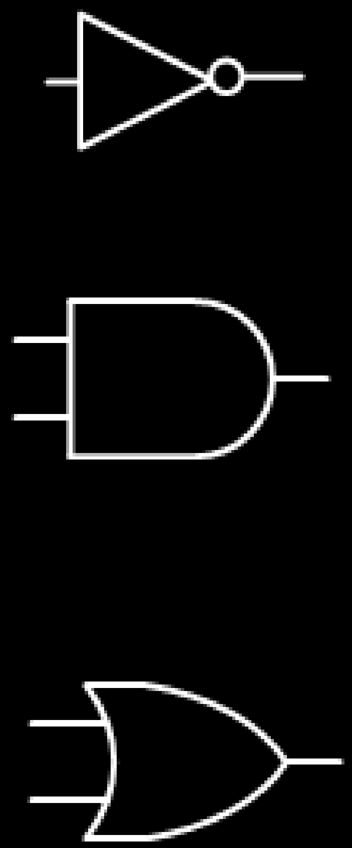
In CMOS NAND and NOR gates are basic gates

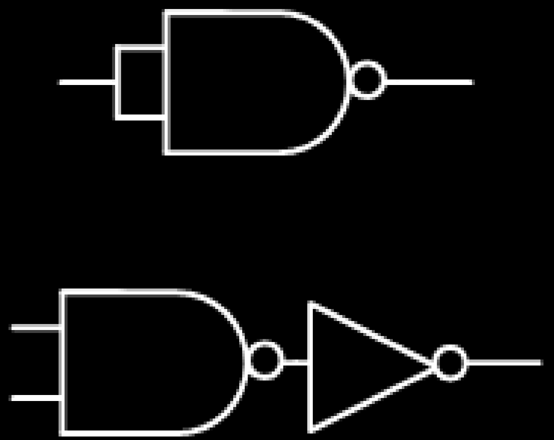
AND gate is realized by using NAND gate with Inverter

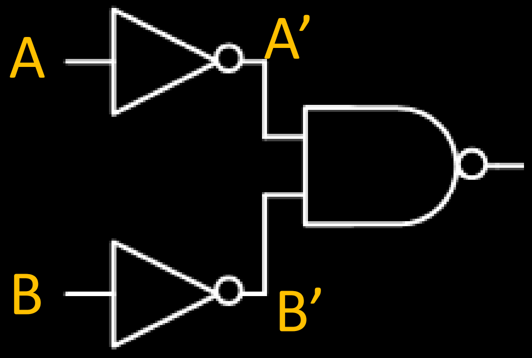
OR gate is realized by using NOR gate with Inverter

NAND-NOR Implementations

Implementation of NAND Gate







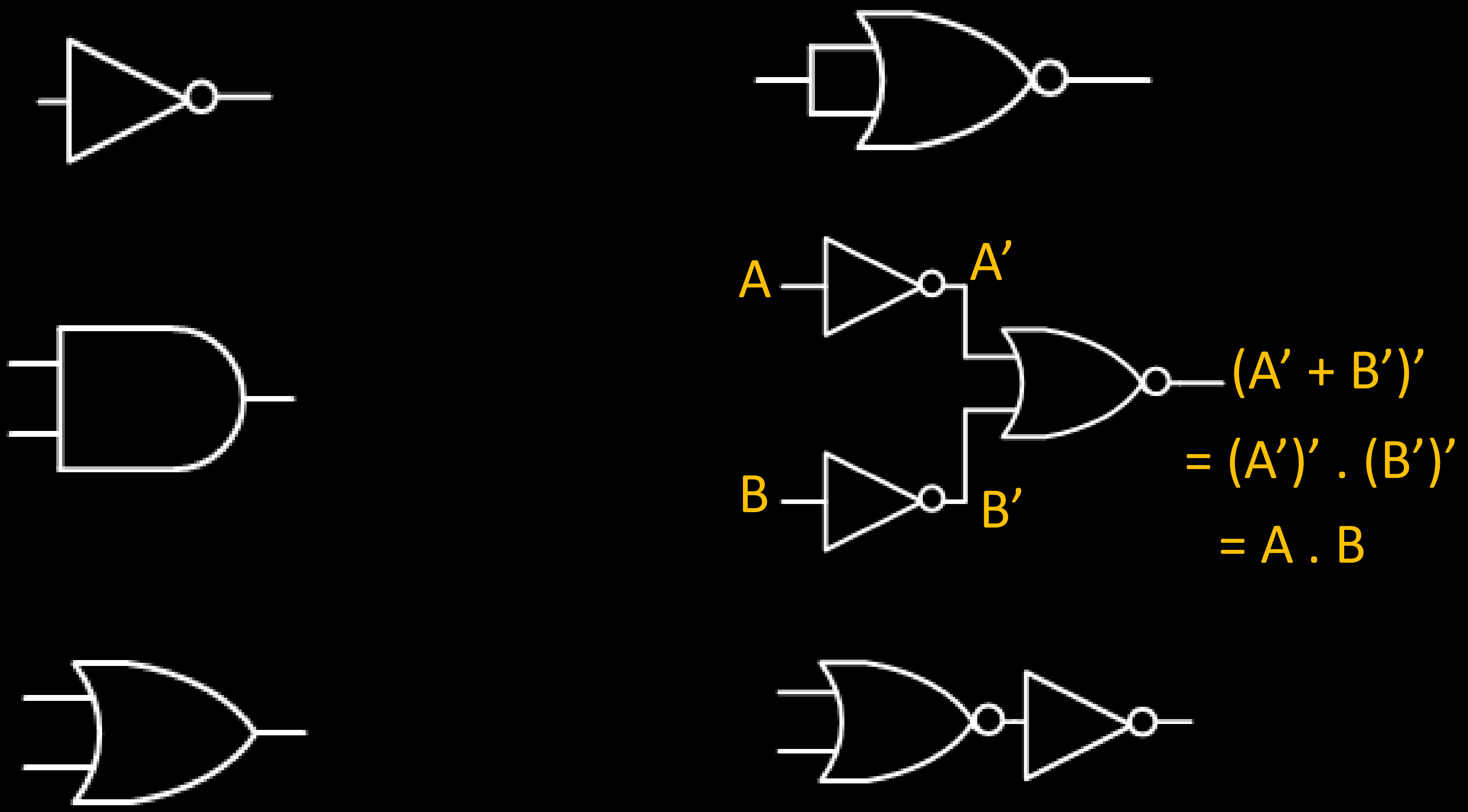
$(A'B')'$

$= (A')' + (B')'$

$= A + B$

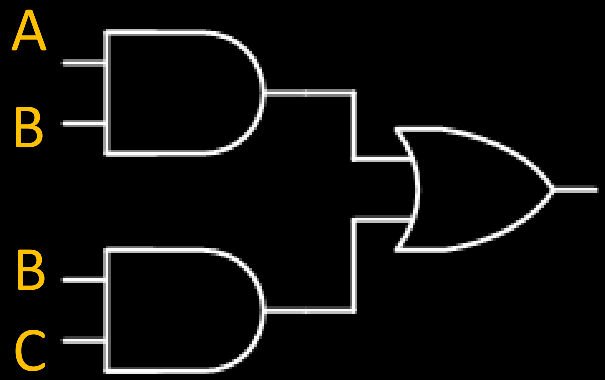
NAND-NOR Implementations

Implementation of NOR Gate

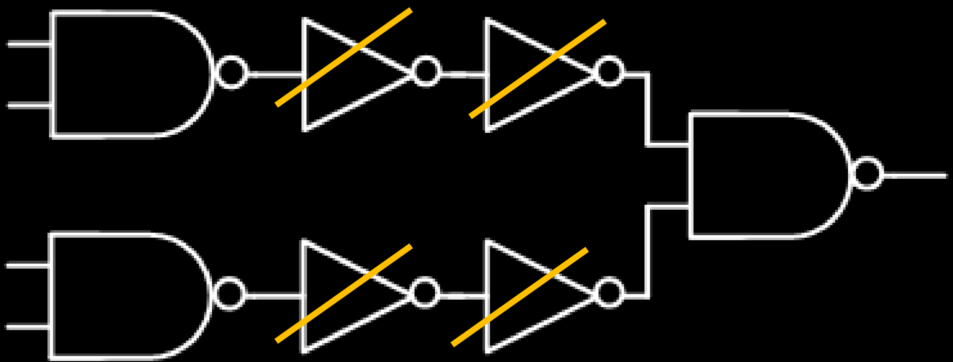


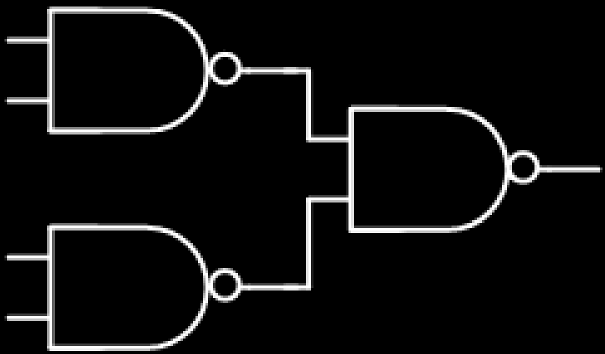
NAND Implementations

NAND gate realization AB + BC



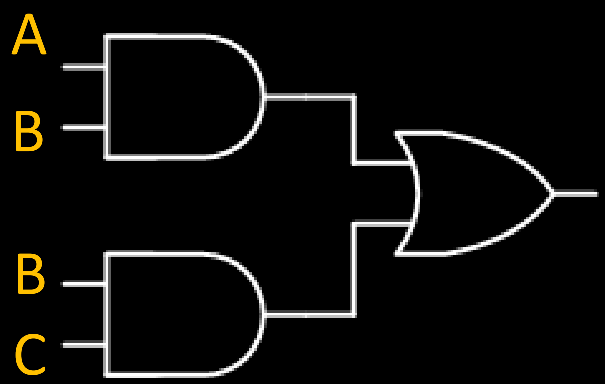
Replace each gate by its corresponding NAND realization



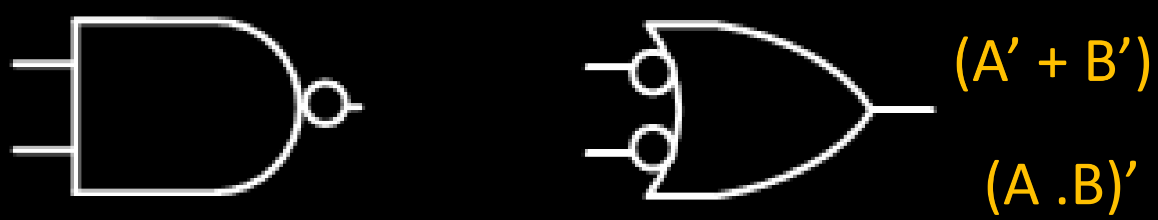


NAND Implementations

NAND gate realization   AB + BC



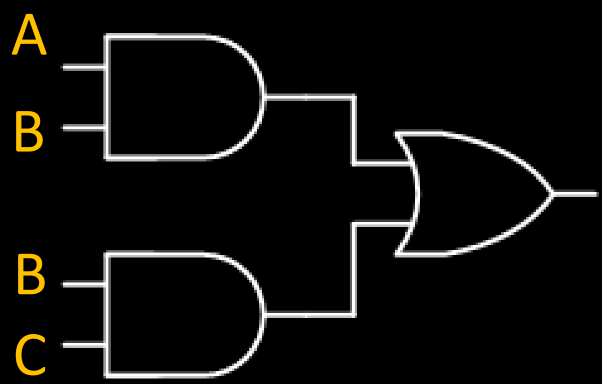
Two representations of NAND gate



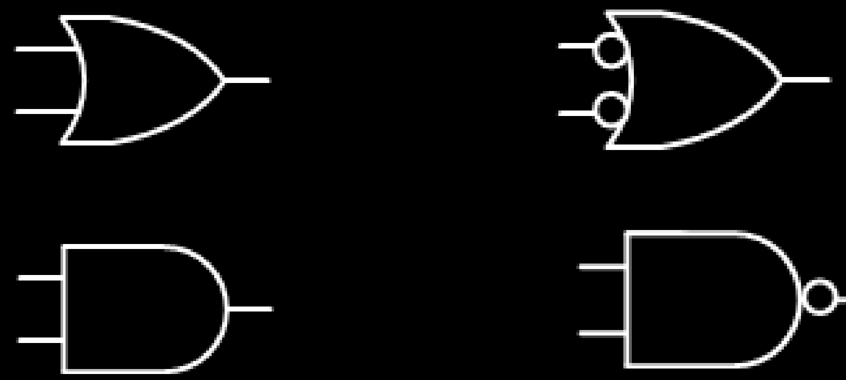
Bubbles indicate invert operation

NAND Implementation

NAND gate realization AB + BC



Replace all



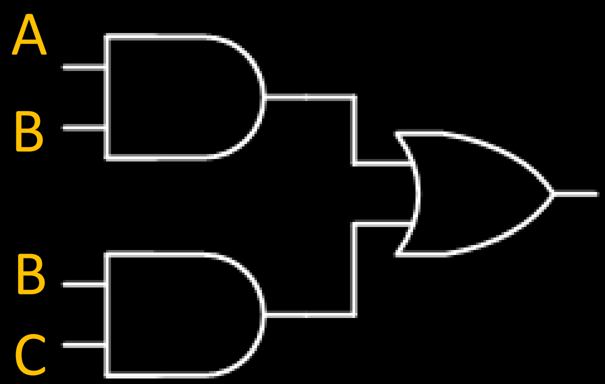
After replacing check if all bubbles are canceling out

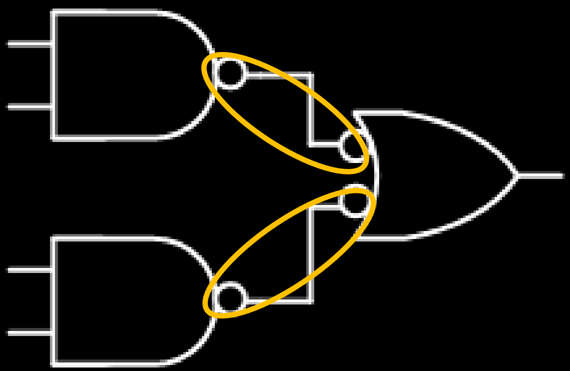
If not then add inverter gates to compensate bubbles

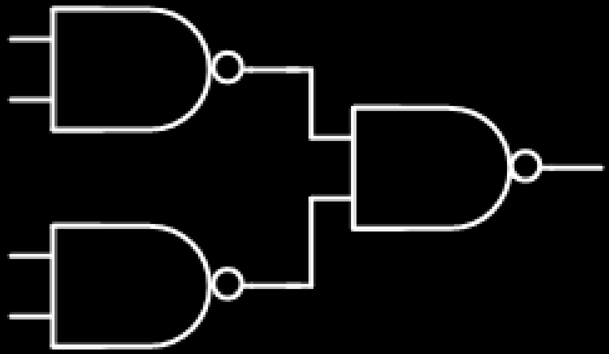
Replace all gates by NAND gates

NAND Implementation

NAND gate realization AB + BC

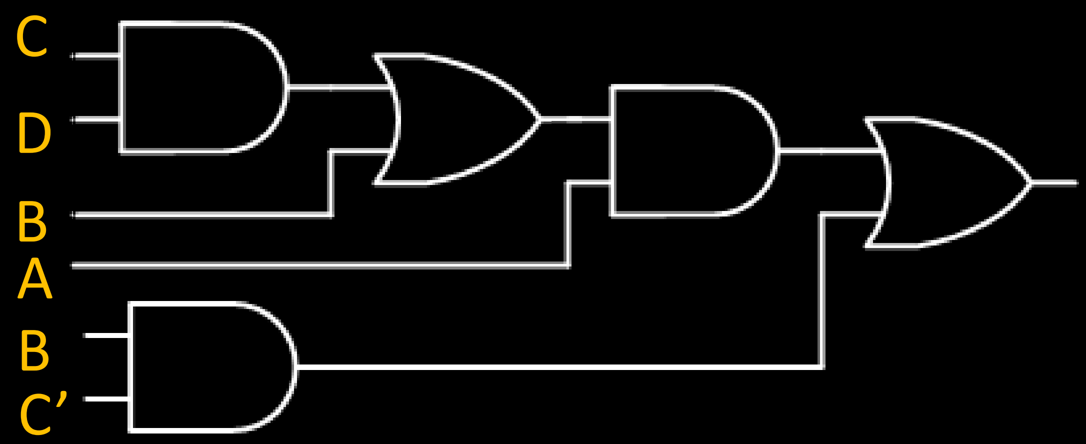






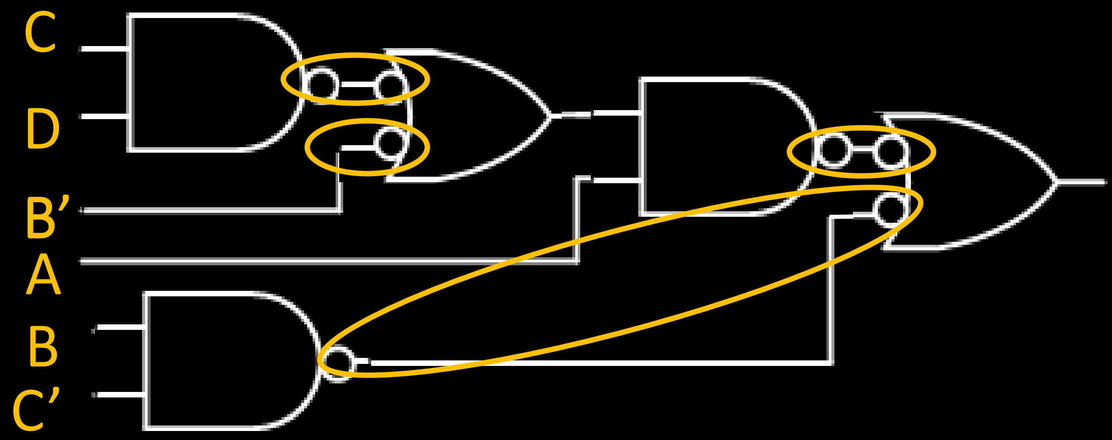
NAND Implementation

NAND gate realization   A(CD+B) + BC’



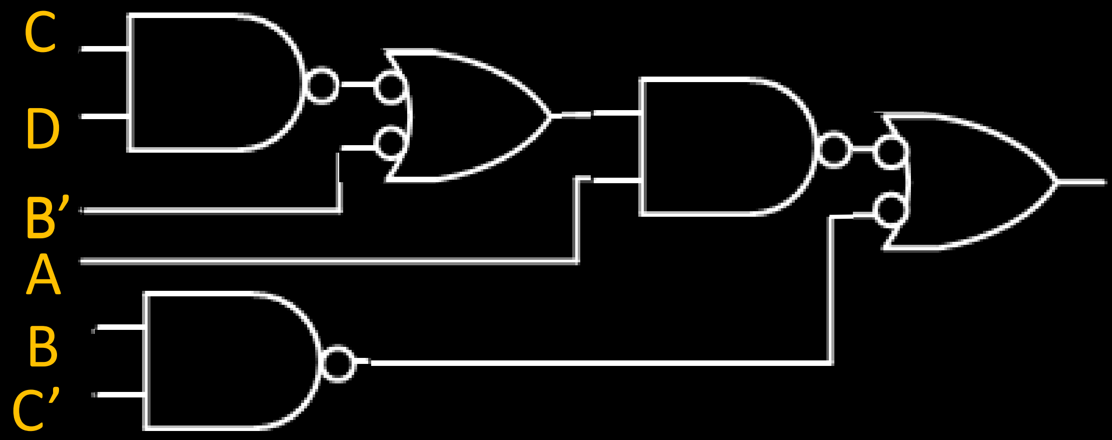
NAND Implementation

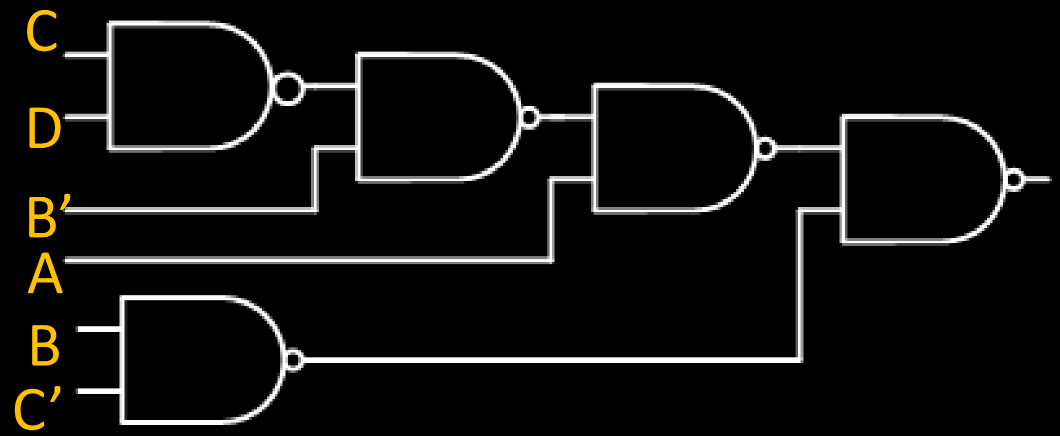
NAND gate realization   A(CD+B) + BC’



NAND Implementation

NAND gate realization   A(CD+B) + BC’





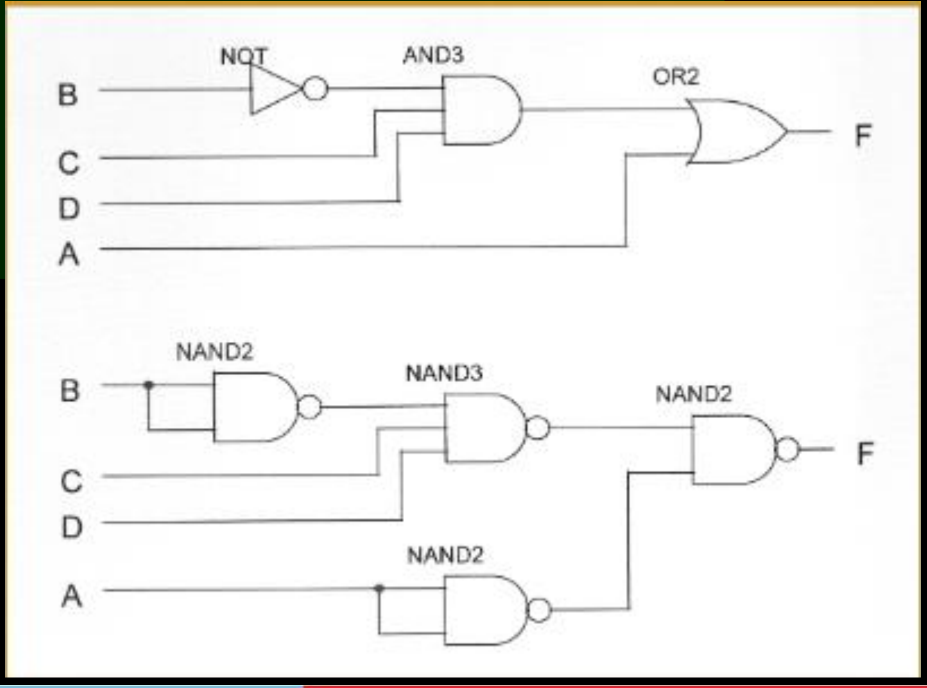
NAND Implementation

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

B'CD

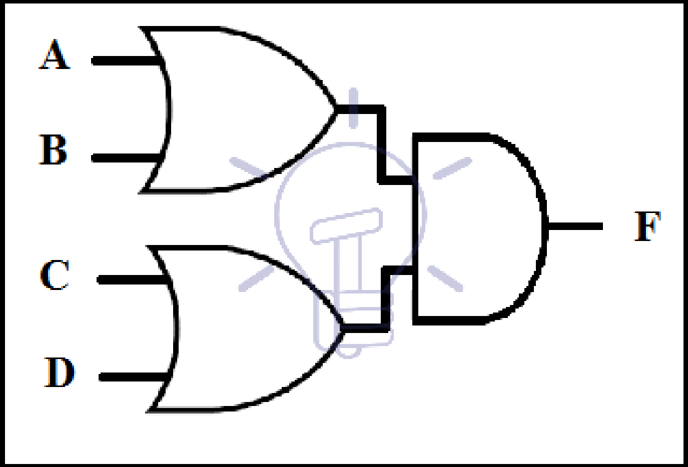
A

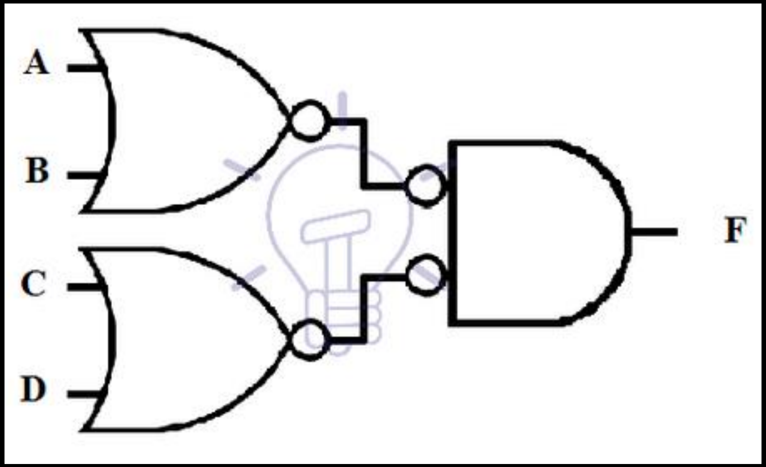
$F = A + B'CD$

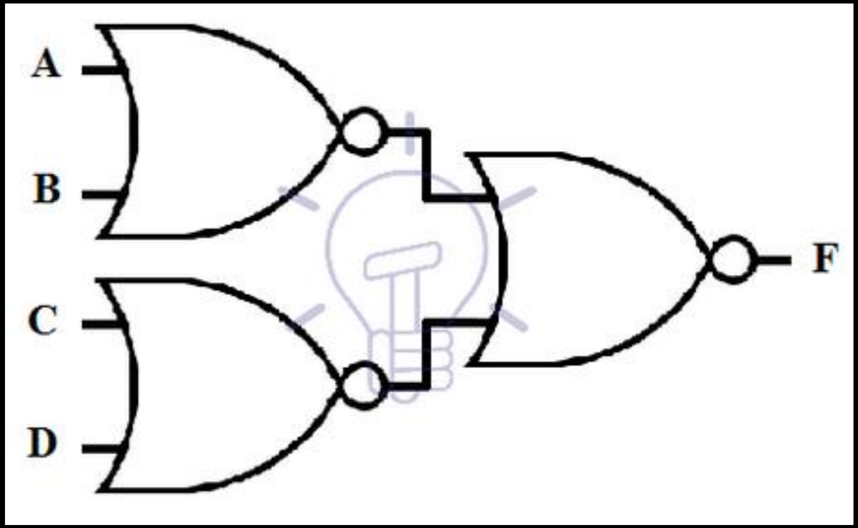


NOR Implementation

F = (A + B) (C + D)







NOR Implementation

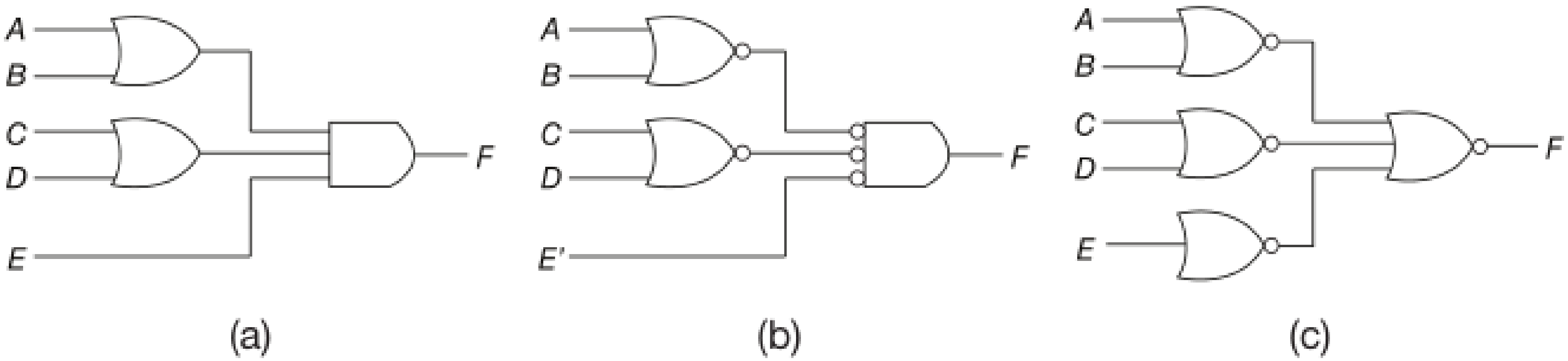


Figure 3.20 Three ways to implement $F = (A + B)(C + D)E$

The Gray Code

Imp. Features:

1. Only one bit ever changes between two successive numbers in the sequence.

   → unit distance code

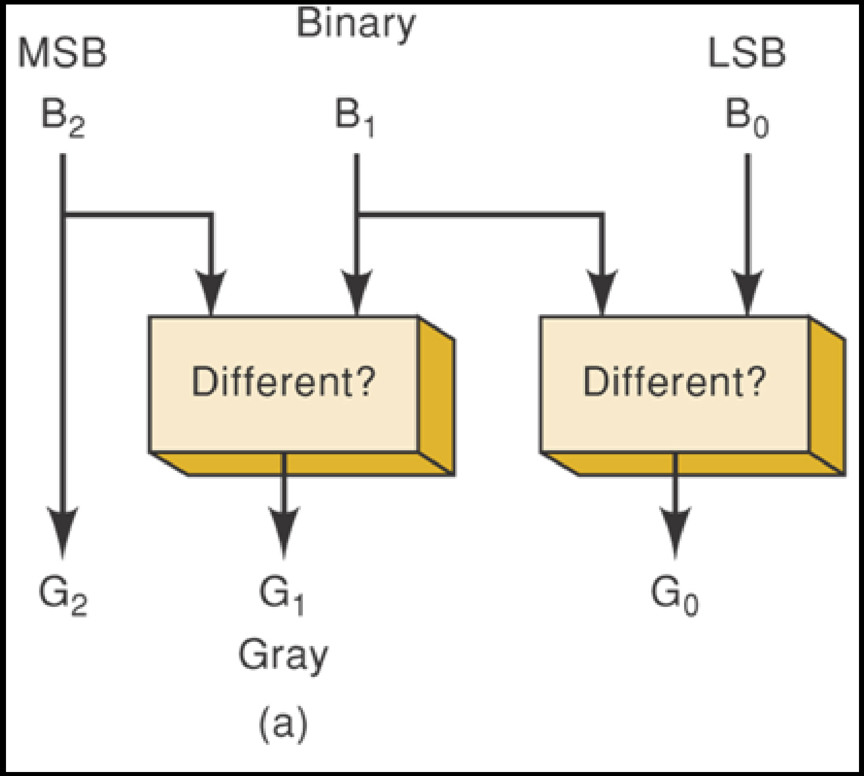
2. It is a non-weighted code.

   → not suitable for arithmetic operations.

3. Gray code is a reflective code.

i.e., the *n-1* least significant bits for 2^{n-1} through $2^n -1$ are the mirror images of those for 0 through $2^{n-1} -1$.

Binary to Gray conversion:



$$G_n = B_n$$

$$G_{n-1} = B_n \oplus B_{n-1}$$

$$G_{n-2} = B_{n-1} \oplus B_{n-2}$$

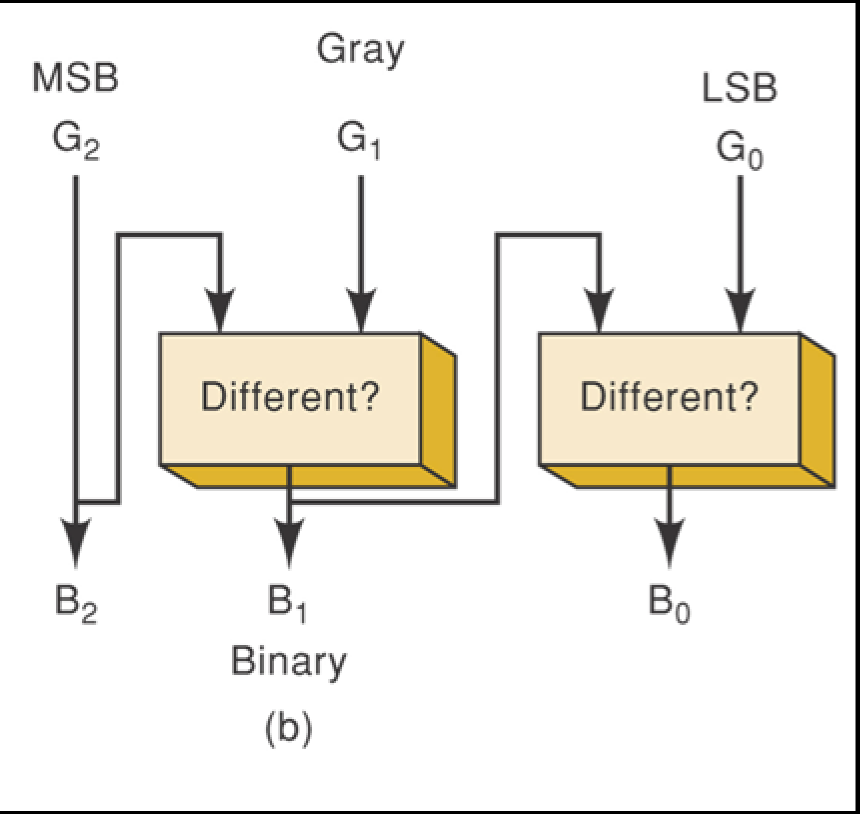
.

.

.

$$G_1 = B_2 \oplus B_1$$

Gray to binary conversion:



$$B_n = G_n$$

$$B_{n-1} = B_n \oplus G_{n-1}$$

$$B_{n-2} = B_{n-1} \oplus G_{n-2}$$

.

.

.

$$B_1 = B_2 \oplus G_1$$

Example:

code.

Binary	1	0	0	1
Gray	1	1	0	1

Gray to Binary conversion

Gray	1	1	0	1
Binary	1	0	0	1

Gray code → Reflective - code

Gray Code				Decimal	Binary
1- bit	2-bit	3-bit	4-bit		4-bit
0	00	000	0000	0	0000
1	01	001	0001	1	0001
	11	011	0011	2	0010
	10	010	0010	3	0011
		110	0110	4	0100
		111	0111	5	0101
		101	0101	6	0110
		100	0100	7	0111
			1100	8	1000
			1101	9	1001
			1111	10	1010
			1110	11	1011
			1010	12	1100
			1011	13	1101
			1001	14	1110
			1000	15	1111

Reading Assignment

5-Variable k-Map

Next Class

Quine-McCluskey (QM) Technique

Thank You