Department of Electrical & Computer Engineering

COEN 212:
DIGITAL SYSTEMS DESIGN
Lecture 4: Gate-Level Minimization

Instructor: Dr. Reza Soleymani, Office: EV-5.125,
Telephone: 848-2424 ext.: 4103.

Lecture 4:
Objectives of this lecture

UNIVERSITÉ Concordia UNIVERSITY
Department of Electrical & Computer Engineering

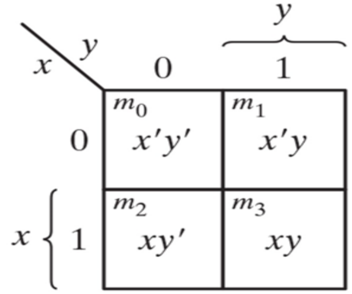
- K-Maps.
- Logic simplification using K-Maps.
- Incompletely specified circuits ("Don't care condition).
- NAND-only implementation.
- NOR-only implementation.

Lecture 4:
Reading for this lecture

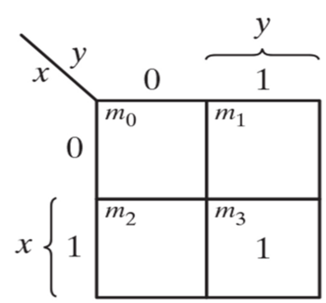
- Digital Design by M. Morris R. Mano and Michael D. Ciletti, 6th Edition, Pearson, 2018:
  - Chapter 3 (3.1 to 3.6)

Lecture 4: Two-value K-maps

- A K-map for an n variables circuit has 2^n squares.
- Each square represents a row of the truth table (a minterm).
- For two variables x and y , we have:



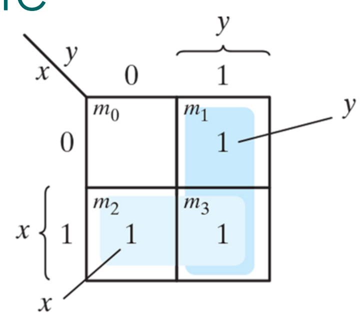
- Example: the AND Gate:



Université Concordia University
Department of Electrical & Computer Engineering

Lecture 4: Two-value K-Maps

- Example: the OR Gate



- The function is $m_1 + m_2 + m_3 = x'y + xy' + xy = x + y$.
- Using Boolean Algebra:

$$x'y + xy' + xy = x'y + xy + xy' + xy$$
$$= (x' + x)y + x(y + y') = x + y$$

- Using K-map: Group together
  - m_1 and m_2
  - m_2 and m_3

Lecture 4:

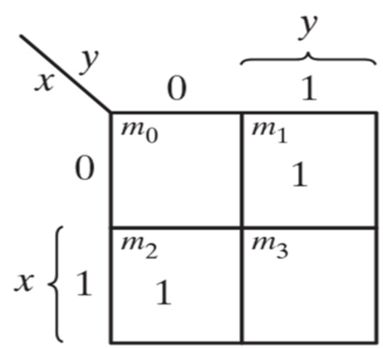
Two-value K-Maps

Concordia University
Department of Electrical & Computer Engineering

- Example: for a function with truth table

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

- The k-map is:

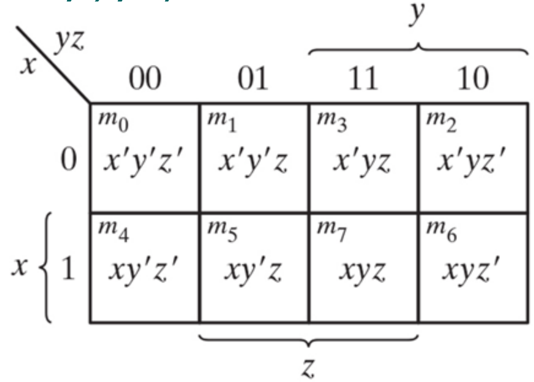


- And the Boolean expression is $m_1 + m_2 = x'y + xy'$.

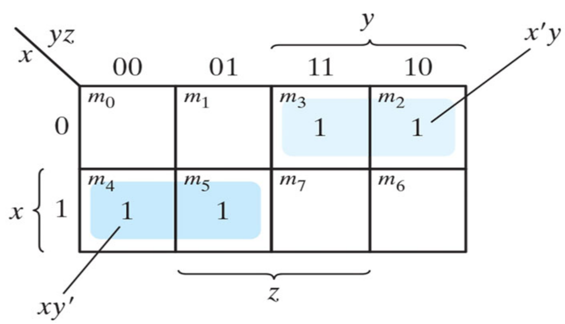
Department of Electrical & Computer Engineering

Lecture 4:
Three-value K-Maps

- For three variables x, y, z, we need $2^3 = 8$ squares.
- The k-map is:



- Example: simplify the function $F(x, y, z) = \sum(2,3,4,5)$.

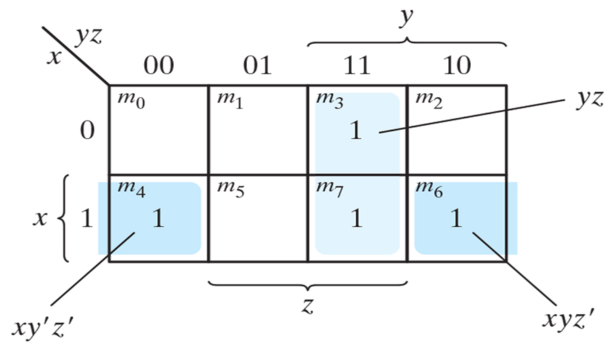


$$F(x, y, z) = xy' + x'y$$

Lecture 4: Three-value K-Maps

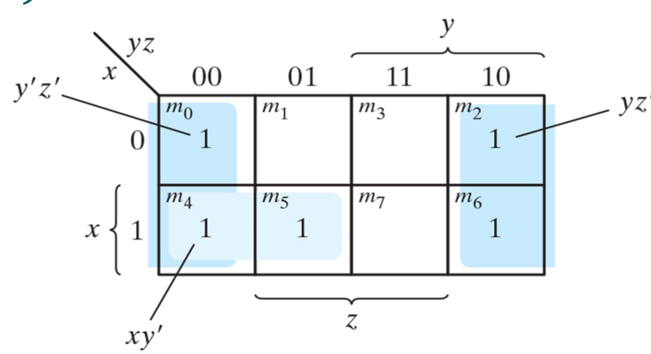
Concordia University
Department of Electrical & Computer Engineering

- Example: $F(x, y, z) = \sum(3,4,6,7)$.
- The k-map is:



$$F(x, y, z) = yz + xz'$$

- Example: $(x, y, z) = \sum(0,2,4,5,6)$.
- K-map:

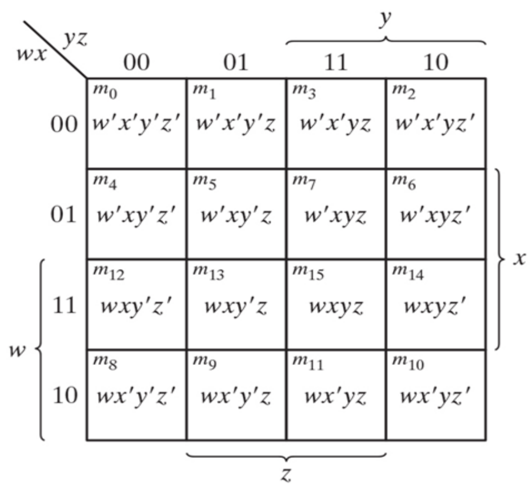


The function: $F(x, y, z) = z' + xy'$.

Université Concordia University
Department of Electrical & Computer Engineering

Lecture 4: Four-value K-Maps

- It has 16 squares

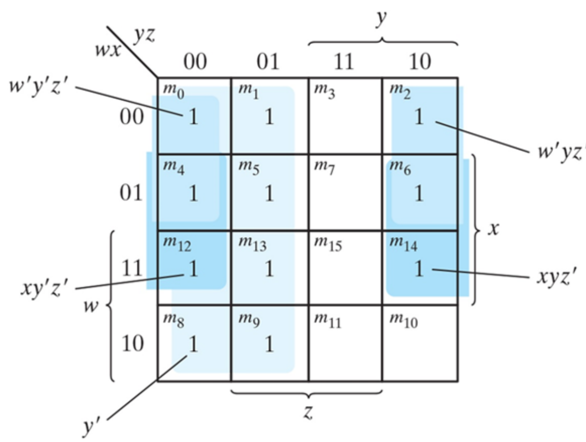


- In a 4-variable map:
  - One square represents one minterm with 4 literals.
  - Two adjacent squares represent a term with three literals.
  - Four adjacent squares represent a term with 2 variables.
  - Eight adjacent squares represent a term with one literal.

Lecture 4:
Four-value K-Maps

Concordia University
Department of Electrical & Computer Engineering

- Example: simplify the function: $F(w,x,y,z) = \sum(0,1,2,4,5,6,8,9,12,13,14)$.



- The expression is:

$$F(w,x,y,z) = y' + w'z' + xz'$$

Lecture 4:
K-Maps: Prime Implicants

Université Concordia University
Department of Electrical & Computer Engineering

- A prime implicant is a product term formed by combining the maximum possible number of squares in a K-map. Number of squares are a power of two: 1, 2, 4, 8, …
- A single square that cannot be combined with any other square forms a prime implicant.
- Any two adjacent squares that cannot be part of a group of 4 adjacent cells form a prime implicant.
- 4 adjacent squares that cannot be part of a group of 8 adjacent cells form a prime implicant.
- A prime implicant that has a square that is not part of any other prime implicant is called an essential prime implicant.

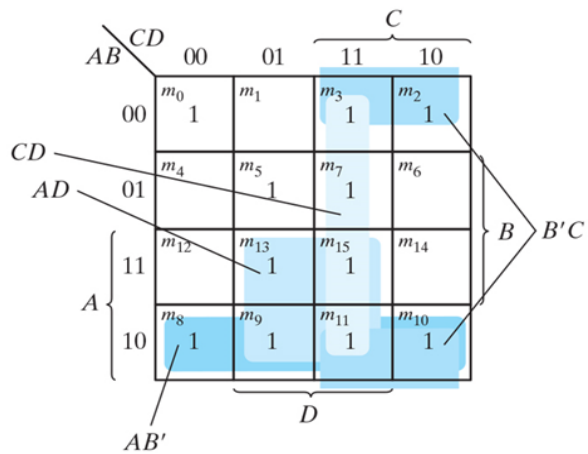
Lecture 4:
K-Maps: General Procedure

- Draw the Truth Table if one is not provided.
- Draw the K-map for the circuit using the Truth Table.
- Find all essential prime implicants and specify the associated terms.
- Form the simplified expression by logical sum (OR) of:
  - those terms and,
  - the minterms remaining.

Lecture 4:
K-Maps: General Procedure

Université Concordia University
Department of Electrical & Computer Engineering

- Example:
- $F(A,B,C,D) = \sum(0,2,3,5,7,8,9,10,11,13,15)$.



$$F = BD + B'D' + CD + AB'$$

Other possibilities:

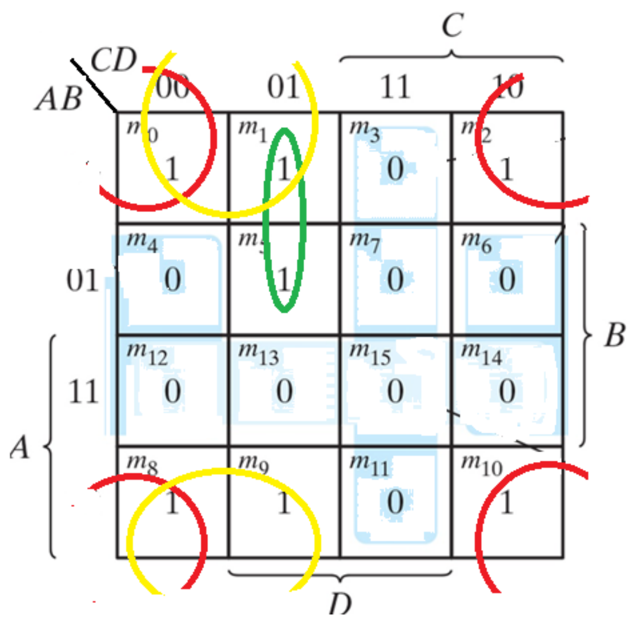
$$F = BD + B'D' + B'C + AD, \quad BD + B'D' + B'C + AB',$$

$$F = BD + B'D' + CD + AD.$$

Lecture 4:
K-Maps: Product of Sums

Concordia University
Department of Electrical & Computer Engineering

$$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$$

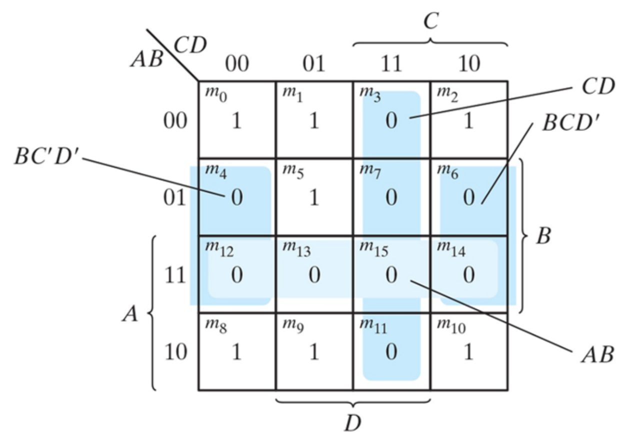


$$F = B'D' + B'C' + A'C'D$$

Lecture 4:
K-Maps: Product of Sums

Université Concordia University
Department of Electrical & Computer Engineering

$$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$$



So: $F' = AB + CD + BD'$

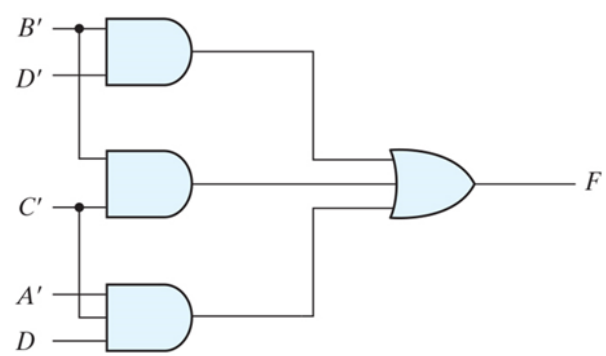
And using De Morgan's we get:

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

Lecture 4: Implementation: Sum of Products

Université Concordia University
Department of Electrical & Computer Engineering

$$F = B'D' + B'C' + A'C'D$$

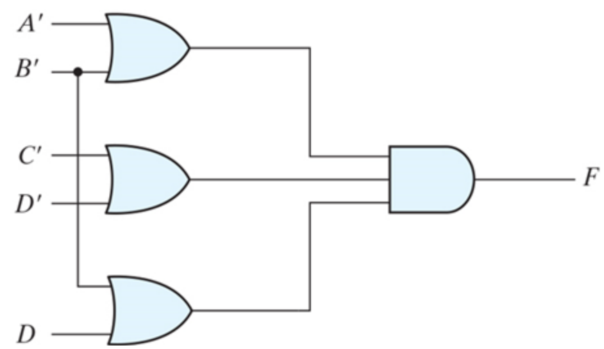


- AND-OR implementation

Lecture 4:
Implementation: Product of Sums

Université Concordia University
Department of Electrical & Computer Engineering

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



- OR-AND implementation

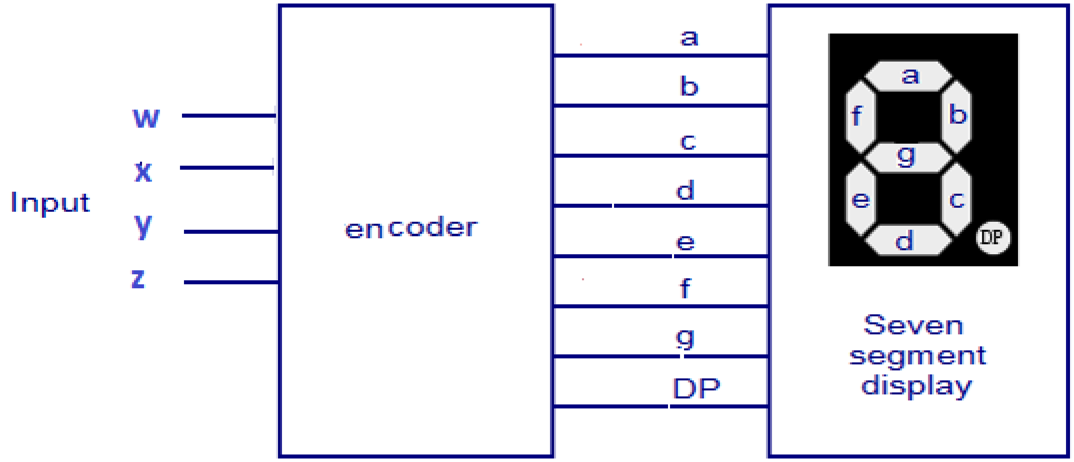
Lecture 4:
Don't' care condition

- When we don't care about the value of the logic for a certain combination of variables, we put a X instead of a 0 or a 1 in the square.
- A Don't care square may be considered as a 1 or a 0 square and combined with other squares of similar content when doing simplification.
- The choice is made such that the number of gates is minimized.

Lecture 4:
Don't' care condition: 7-segment example

Department of Electrical & Computer Engineering

- Input: Digits 0 to 9 (in binary).
- The output: Digits on the LED Display.



- Input has 4 bits. So, 16 possibilities.
- But we only need 10 of 16 and don't care for the rest.

Lecture 4:
Don't' care condition: 7-segment example

UNIVERSITÉ Concordia UNIVERSITY
Department of Electrical & Computer Engineering

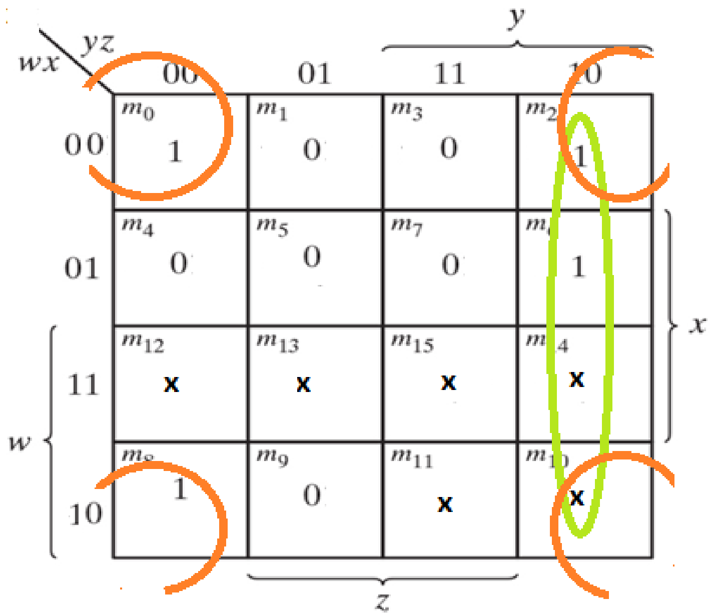
- Truth Table for the 7-segment encoder.

w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	0
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	1	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	0	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

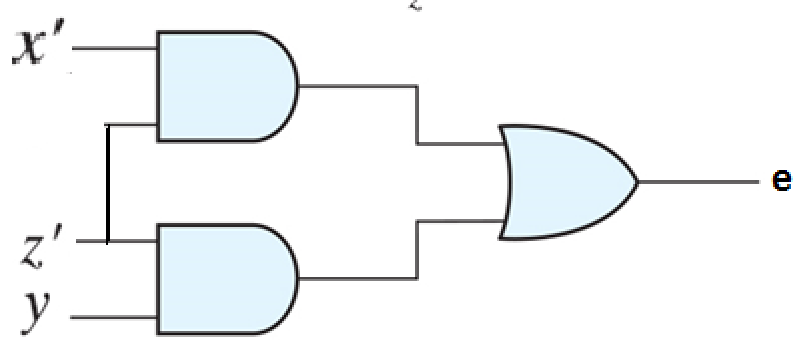
Université Concordia University
Department of Electrical & Computer Engineering

Lecture 4:
Don't' care condition: 7-segment example

- K-map for pin *e* of the LED:

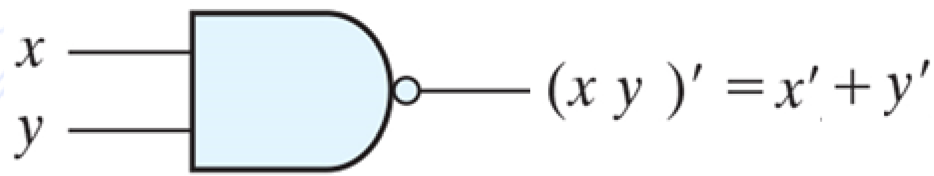


- $e = x'z' + yz'.$



Lecture 4: NAND gate

- NAND gate:



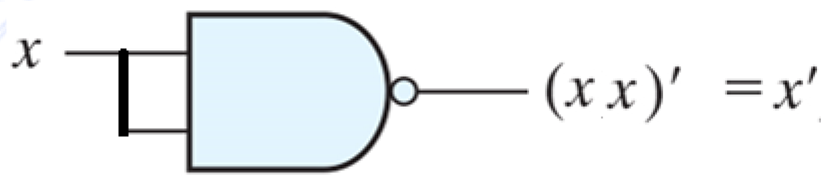
- NAND can be implemented using an AND and a NOT:

- AND and not can also be implemented using NAND

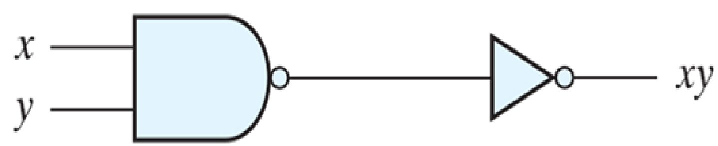
Lecture 4: NAND gate

Université Concordia University
Department of Electrical & Computer Engineering

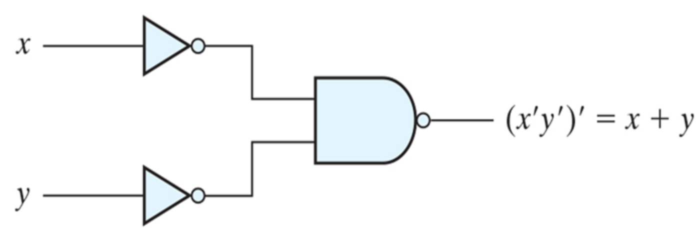
- NOT gate using NAND:



- AND gate using NAND:



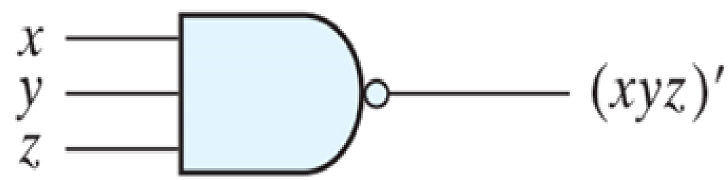
- OR gate using NAND:



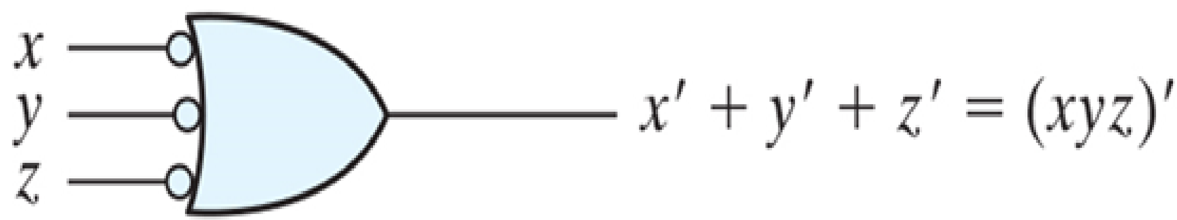
Lecture 4: NAND gate

Université Concordia University
Department of Electrical & Computer Engineering

- AND-invert implementation:



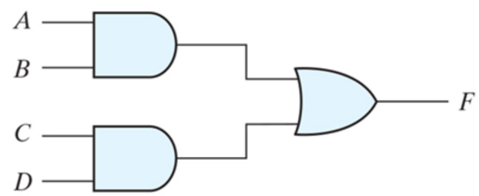
- Invert-OR implementation:



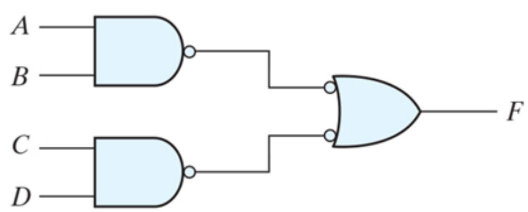
Lecture 4:
NAND only implementation

Université Concordia University
Department of Electrical & Computer Engineering

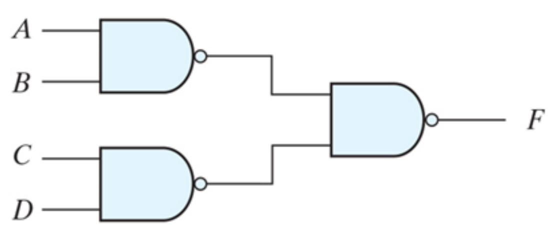
- Example: implement $F = AB + CD$ using only NAND gates;



- Invert the output of AND's and inputs of the OR:



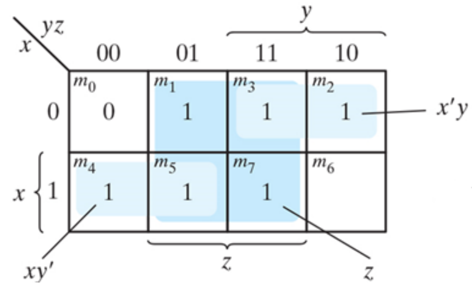
Or



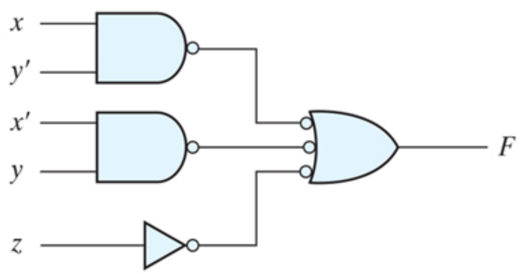
Lecture 4:
NAND only implementation

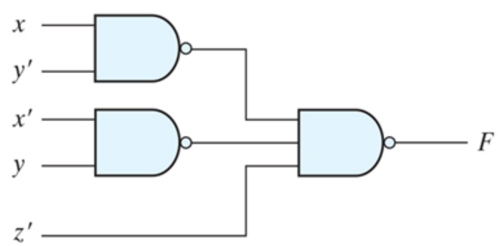
Université Concordia University
Department of Electrical & Computer Engineering

- Example: $F(x,y,z) = \sum(1,2,3,4,5,7)$



- $F = xy' + x'y + z$

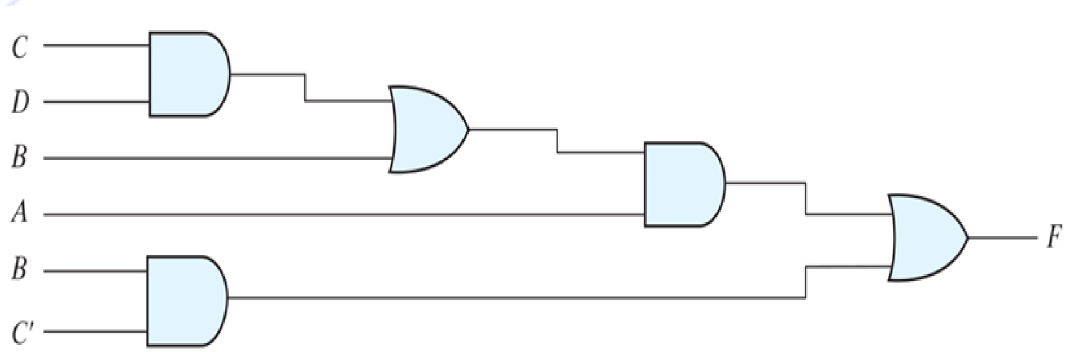




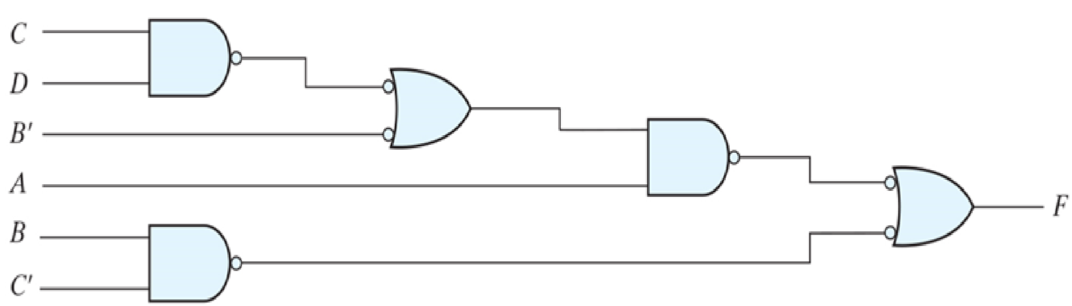
Lecture 4: NAND only implementation

Université Concordia University
Department of Electrical & Computer Engineering

- **Example:** implement $F = A(CD + B) + BC'$ using NAN only.
- Sum of product form:



- Use the procedure discussed (AND-invert and invert-AND):

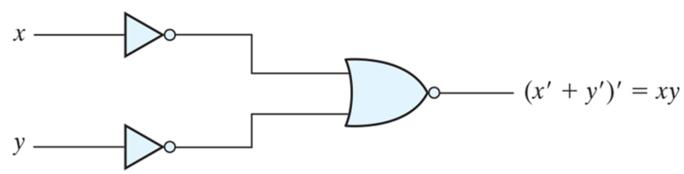


Lecture 4:
NOR implementation

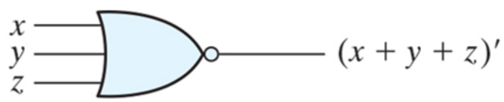
Concordia University
Department of Electrical & Computer Engineering

- The implementation of an OR gate using NOR is:

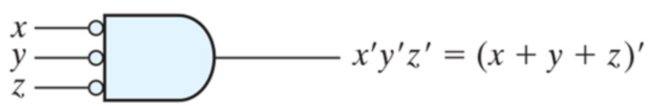
- AND gate implementation using NOR:



- OR-invert



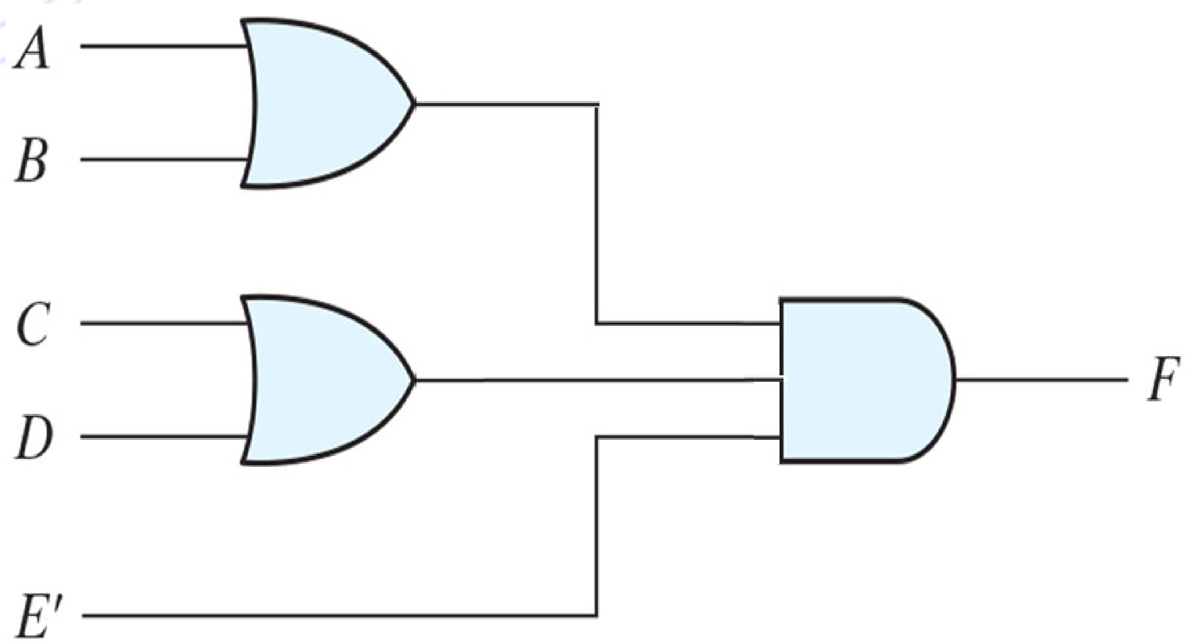
- Invert-AND



Lecture 4:
NOR implementation

Université Concordia University
Department of Electrical & Computer Engineering

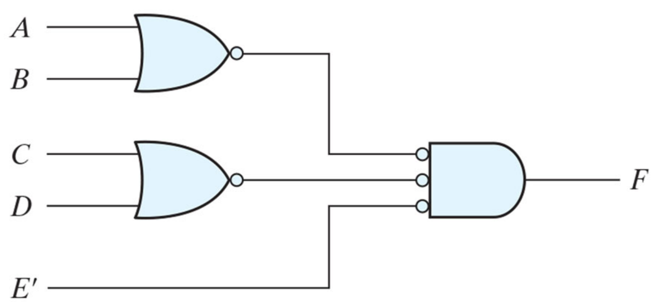
- Implement $F = (A + B)(C + D)E$ using NOR gates only.



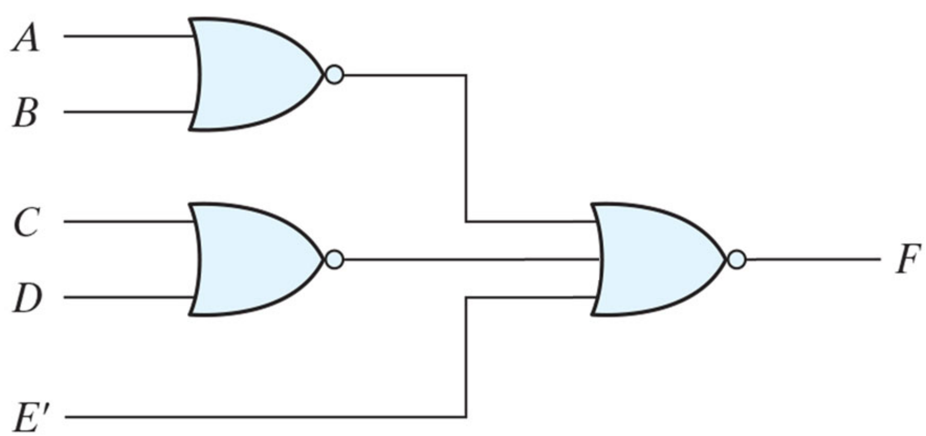
Department of Electrical & Computer Engineering

Lecture 4:
NOR implementation

- Do OR-invert and invert-AND to get:



- Or:



Lecture 4: Knowledge Check

Department of Electrical & Computer Engineering

- **Question 1:** The expression for the function for segment c of the 7-segment is:
- a) $c = w + x + y' + z'$ b) $c = x'z' + w'x + y'z$
- c) both a and b   ad) neither a not b

- **Question 2:** Implement $\mathrm{F} = x'z' + yz'$ using NAN gate only.

- **Question 23:** Implement $\mathrm{F} = xy + z$ using NOR gate only.