CHAPTER IV

GATE DESIGN

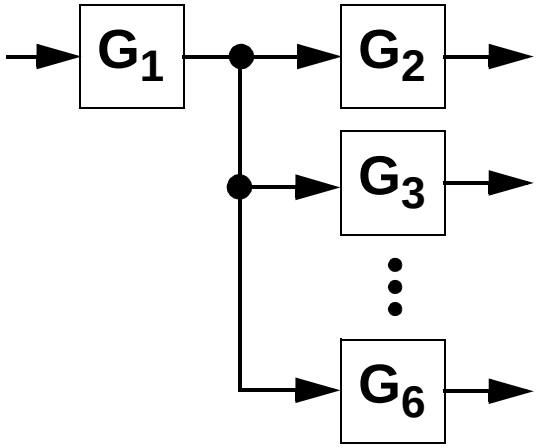
GATE NETWORKS

INTRODUCTION

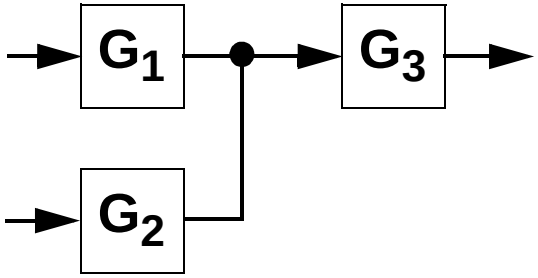
- Gate network consists of
  - Gates
  - External inputs and outputs
  - Connections
- Gate inputs
  - Only one connection to input is allowed (unless tri-state device is used)
    - Connected to constant value (**0** or **1**)
    - Connected to an external input
    - Connected to a gate output
- Gate outputs
  - Output load should not be greater then the fanout factor for the gate and technology being used.

GATE NETWORKS

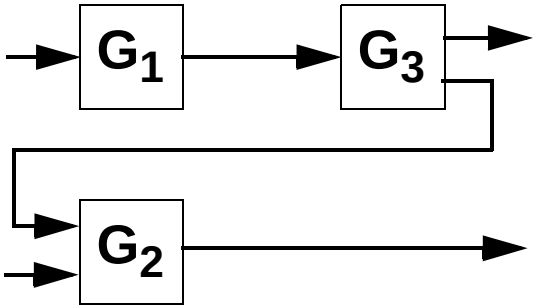
VALID/INVALID NETWORKS



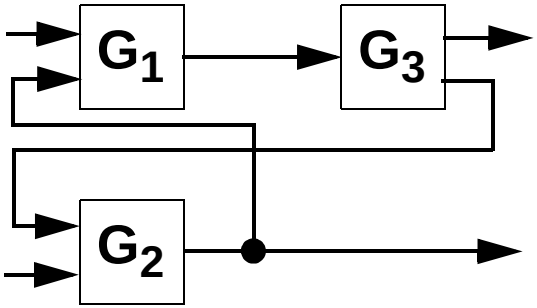
Valid or Invalid?



Valid or Invalid?



Valid or Invalid?

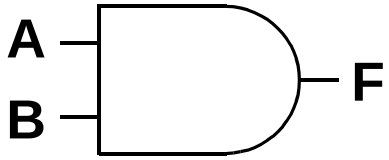


Valid or Invalid?

LOGIC GATES

NON-INVERTING OPERATORS

AND

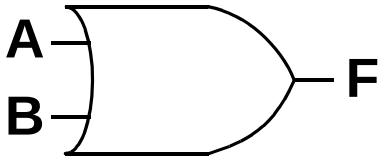


A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

$F = AB$

6 transistors

OR

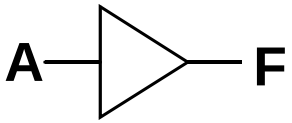


A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

$F = A + B$

6 transistors

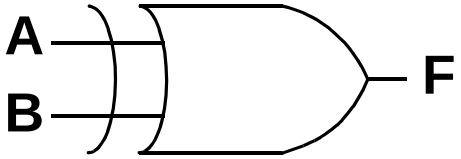
BUFFER



A	F
0	0
1	1

$F = A$

XOR



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

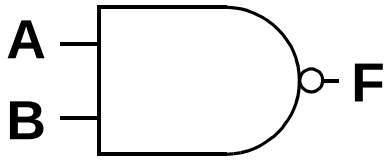
$F = A\bar{B} + \bar{A}B = A \oplus B$

8 transistors

LOGIC GATES

INVERTING OPERATORS

NAND

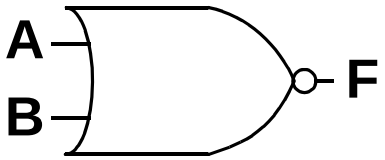


A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

$F = \overline{AB}$

4 transistors

NOR

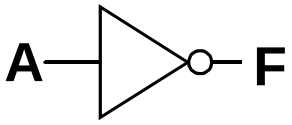


A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

$F = \overline{A + B}$

4 transistors

NOT

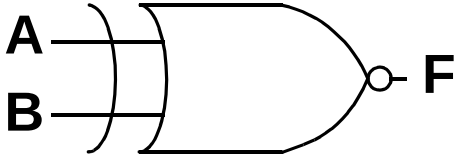


A	F
0	1
1	0

$F = \overline{A}$

2 transistors

XNOR



A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$F = AB + \overline{A}\,\overline{B} = \overline{A \oplus B}$

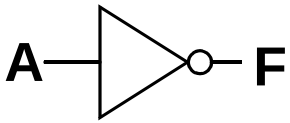
8 transistors

LOGIC GATES

INVERTERS

- Inverters can also be implemented with a **NAND** or with a **NOR** gate.

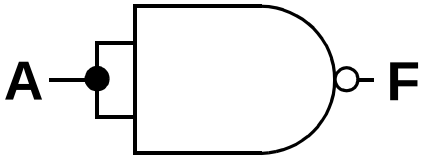
NOT



A	F
0	1
1	0

$F = \overline{A}$

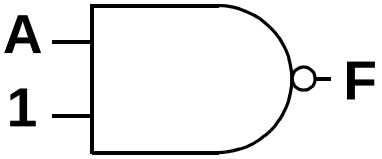
NAND



A	A	F
0	0	1
1	1	0

$F = \overline{AA} = \overline{A}$

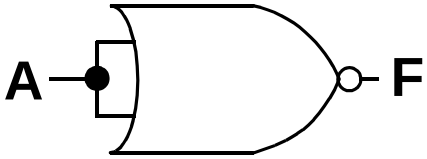
NAND



A	1	F
0	1	1
1	1	0

$F = \overline{A \cdot 1} = \overline{A}$

NOR



A	A	F
0	0	1
1	1	0

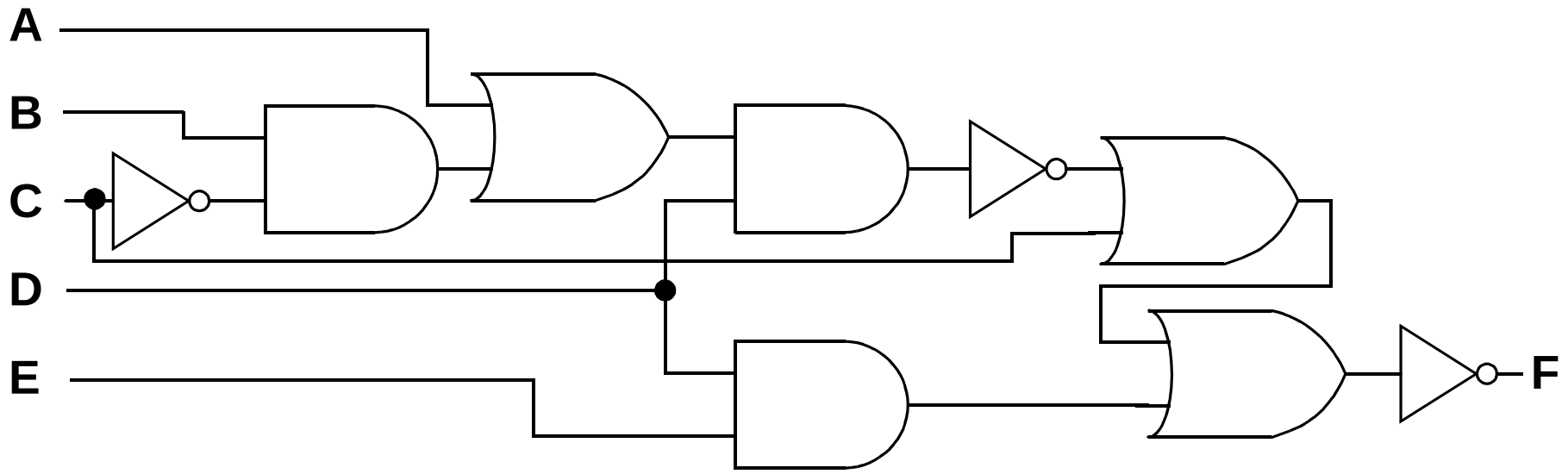
$F = \overline{A + A} = \overline{A}$

LOGIC NETWORKS
FROM BOOLEAN FUNCTIONS

- Implement the following Boolean function using logic gates

$$F = \overline{((A + B\bar{C})D)} + C + DE$$

- Possible solution:



- $3 \times 6_{AND} + 3 \times 6_{OR} + 3 \times 2_{NOT} = 42$ transistors for CMOS technology.

LOGIC NETWORKS

USING SPECIFIC GATES

- Because of various implementation reasons, it may be desired to use only specific sorts of logic gates in an implementation.
  - For instance, many CMOS implementations use only **NAND** gates. Some implementations use on **NOR** gates.
  - This can be done in a number of manners. One is to rework the Boolean functions so that only the specific gates desired are used.
  - May reduce the physical number of transistors required if the appropriate types of gates are used.

LOGIC NETWORKS

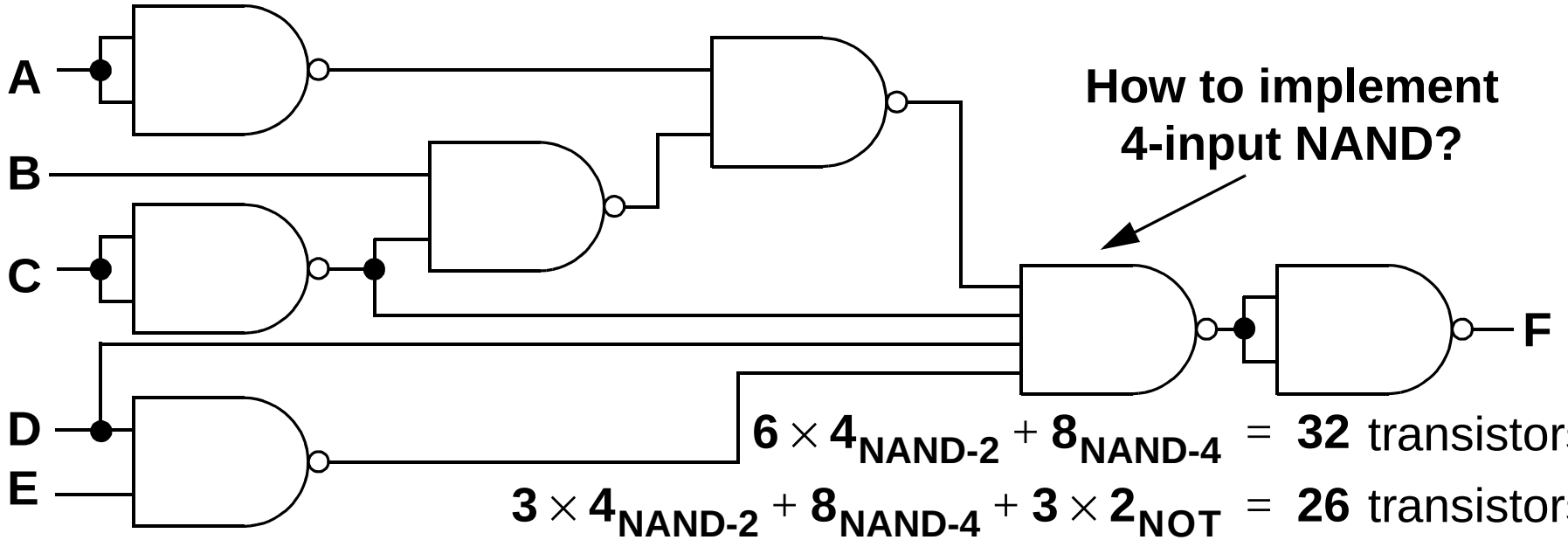
EXAMPLE USING NAND GATES

- Implement the following Boolean function using NAND gates

$$F = \overline{\overline{((A + B\bar{C})D) + C + DE}}$$

  - This Boolean function can be expressed as

$$F = \overline{\overline{A}\,\overline{B\bar{C}D}\,\overline{\bar{C}}\,\overline{DE}} = (((A'(BC')')'D)(C')(DE)')'$$



$6 \times 4_{\text{NAND-2}} + 8_{\text{NAND-4}} = 32$ transistors

$3 \times 4_{\text{NAND-2}} + 8_{\text{NAND-4}} + 3 \times 2_{\text{NOT}} = 26$ transistors

MIXED LOGIC

INTRODUCTION

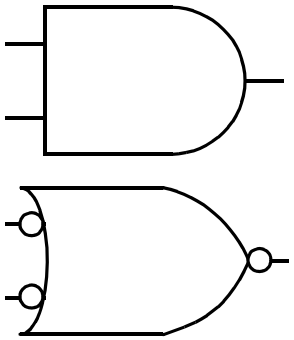
- Mixed logic is one approach that makes it easier to **redesign** a logic network to use **desired types of gates**.
- Mixed logic is also **self-documenting**
  - This means that you can see what the original designer started with and see how the logic network was changed for the implementation.
- The idea behind mixed logic is to diagram out the logic network from the Boolean equations given, and then make small changes to the logic network to achieve desired results for implementation.

MIXED LOGIC

DEMORGAN'S SQUARE

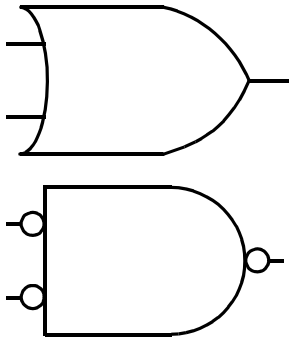
- DeMorgan's Square

AND



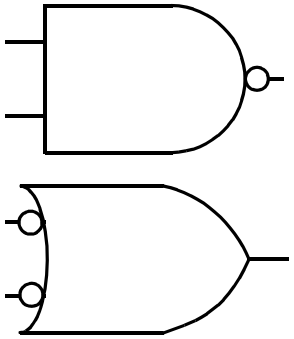
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

OR



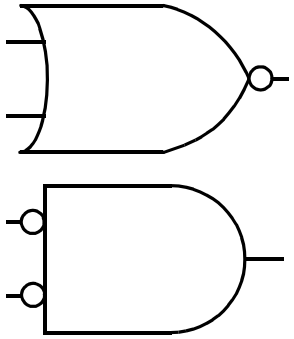
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

NAND



A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

NOR



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

MIXED LOGIC

MIXED LOGIC PROCEDURE

- The procedure for performing mixed logic conversions is as follows:
  - Draw the logic network for the given Boolean equation.
    - Use only **AND** and **OR** gates.
    - Replace all complements with a bar (no bubbles or inverters yet!)
  - Once the initial Boolean equation is drawn with **AND** gates, **OR** gates and bars, the **self-documenting redesign** begins:
    - **Add complement bubbles** and **NOT** gates within the network to appropriately convert logic gates to desired gate sets.
    - The rules in adding complement bubbles and **NOT** gates
      - **All bubbles must cancel each other out**
      - **Exactly one and only one bubble needed on each bar**

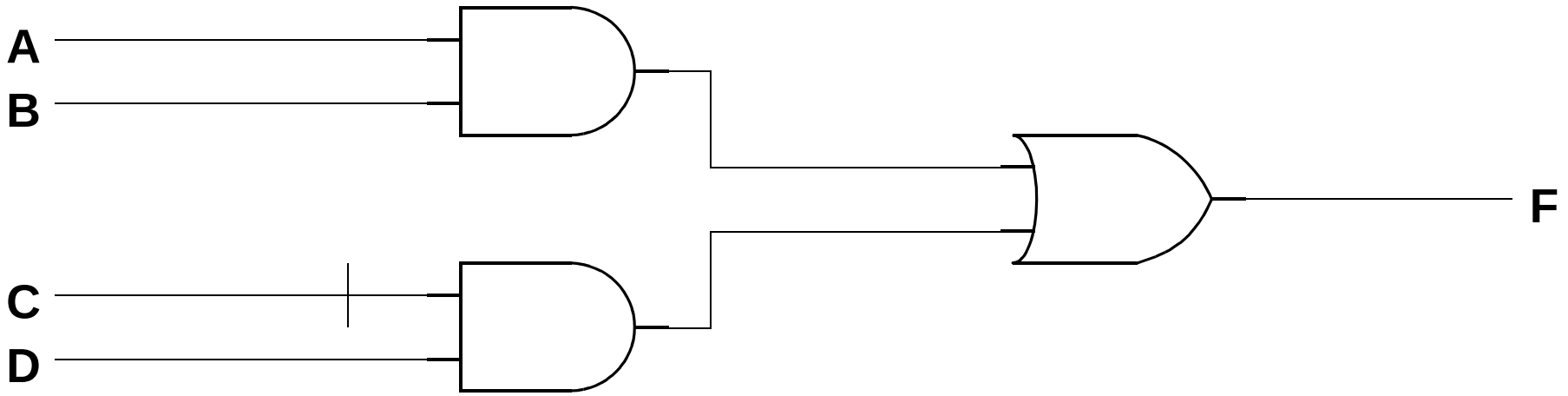
MIXED LOGIC

EXAMPLE #1 (1)

- Implement the following Boolean function using NAND gates and then also using NOR gates.

$$F = AB + \overline{C}D$$

- Solution: Start by drawing the logic network for the Boolean function with the complements as bars.

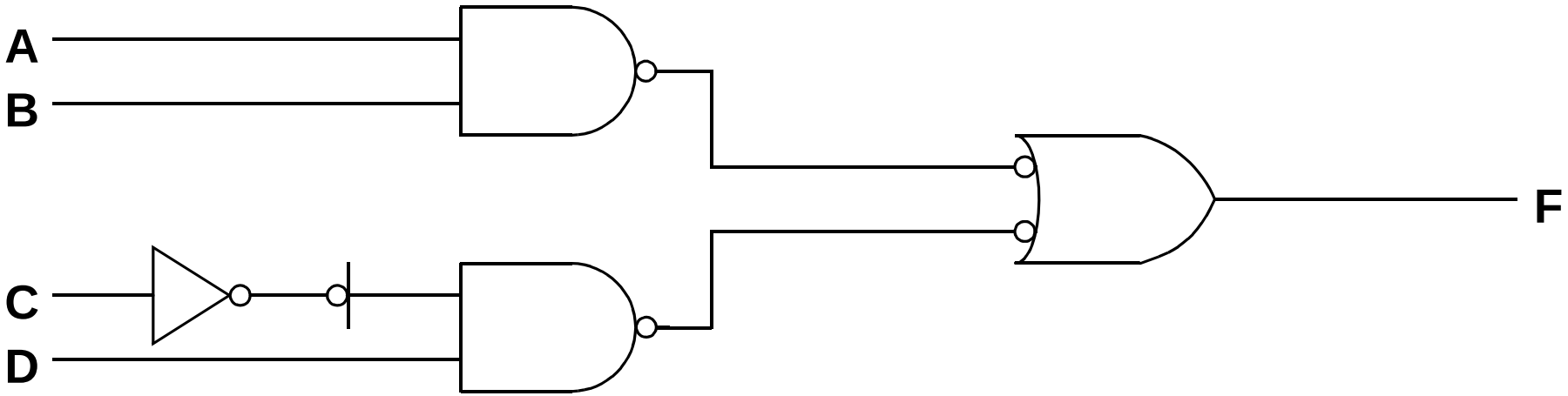


- This completes the information needed to get back original equation.

MIXED LOGIC

EXAMPLE #1 (2)

continued... using **NAND** gates



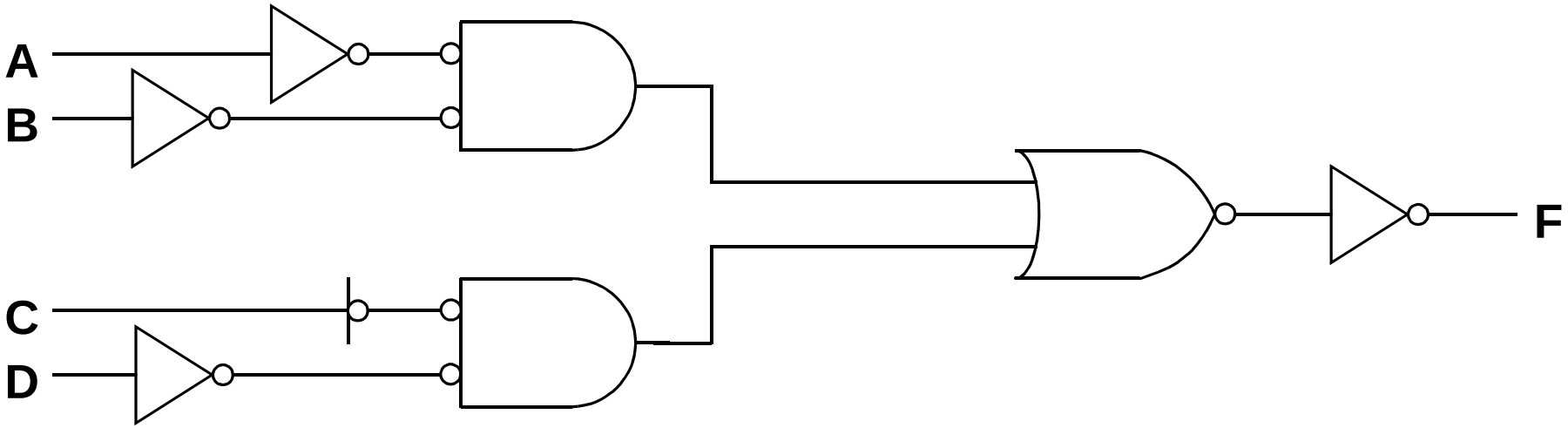
- This logic network now only uses **NAND** gates and inverters.
- This results in the following Boolean function, as obtained previously

$$F(A, B, C, D, E) = \overline{\overline{AB}\,\overline{CD}}$$

MIXED LOGIC

EXAMPLE #1 (3)

continued... using **NOR** gates



- This logic network now only uses **NOR** gates and inverters.
- This results in the following Boolean function, as obtained previously

$$\mathbf{F}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}) = \overline{\overline{\mathbf{A}} + \overline{\mathbf{B}} + \overline{\mathbf{C}} + \overline{\mathbf{D}}}$$

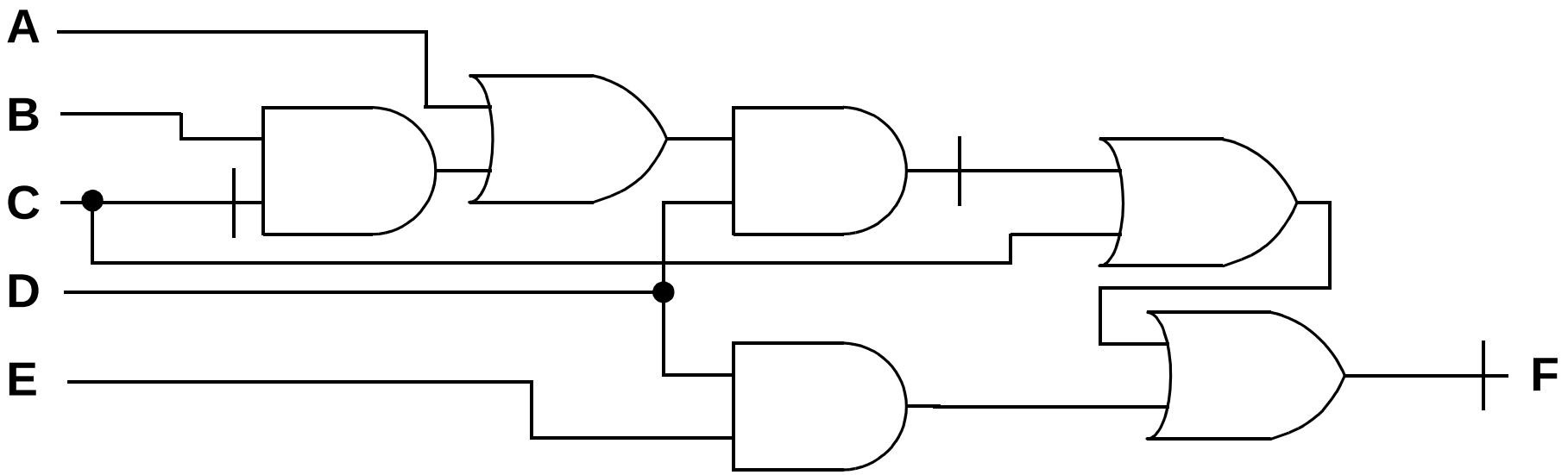
MIXED LOGIC

EXAMPLE #2 (1)

- Implement the following Boolean function using **NAND** gates

$$F = \overline{((A + B\bar{C})D)} + C + DE$$

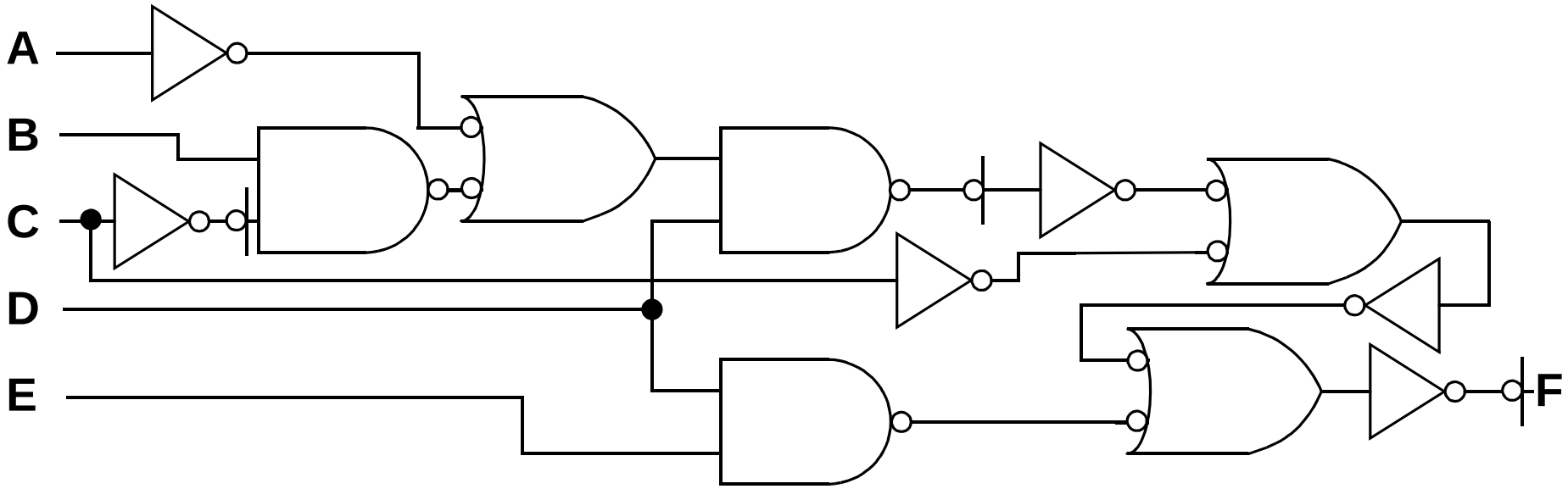
- Solution: Start by drawing the logic network for the Boolean function with the complements as bars.



MIXED LOGIC

EXAMPLE #2 (2)

continued...



- This logic network now only uses **NAND** gates and inverters.
- This results in the following Boolean function, as obtained previously

$$F(A, B, C, D, E) = \overline{\overline{A}\,\overline{B}\,\overline{C}\,D\,\overline{C}\,D\,E} = \overline{\overline{A}\,\overline{B}\,\overline{C}\,D\,\overline{C}\,D\,E}$$