Building a Logic Circuit Using NANDs/NORs

Digital Design. M. Morris Mano

NANDs and NORs

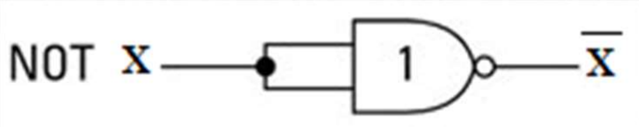
A NAND (NOR) gate is said to be a universal gate because any logic circuit can be implemented with it only.

The simplest way to build a digital circuit using only NANDs (or NORs) is to first obtain the simplest boolean function of it in terms of elementary boolean gates (Not – And – Or), then convert it into NANDs (or NORs).

Building Simple Gates Using NAND(s)

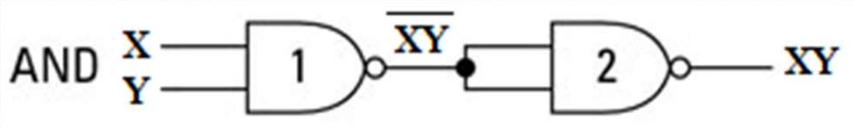
Not gate:

$$\overline{X} = \overline{X.X}$$
$$= NAND(X, X)$$



And gate:

$$X.Y = \overline{\overline{X.Y}}$$
$$= NOT(NAND(X, Y))$$

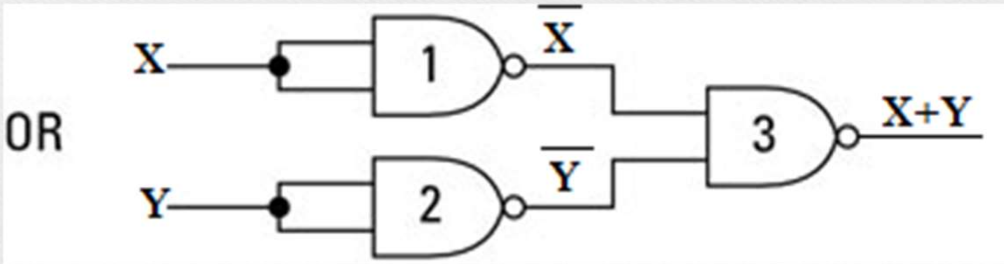


Building Simple Gates Using NAND(s)

Or gate:

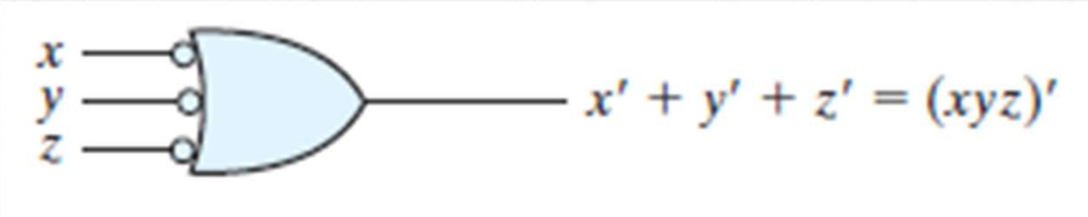
$$X + Y = \overline{\overline{X}.\overline{Y}} = NAND(NOT(X), NOT(Y))$$

De Morgan law



Building a digital circuit Using NAND(s)

Using this circuit help in transform OR into NANDs.

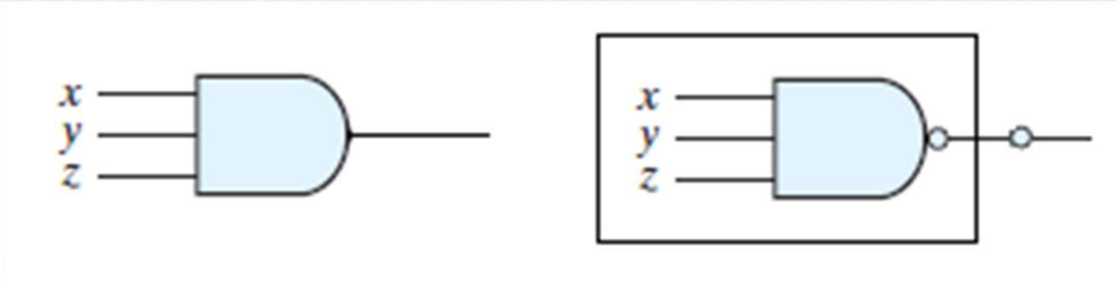


Building a digital circuit Using NAND(s)

1. Obtain the simplest boolean function of the circuit it in terms of elementary boolean gates (Not – And – Or). If you have any other gates, replace it with simple gates, e.g. NORs is replaced it by OR + NOT.

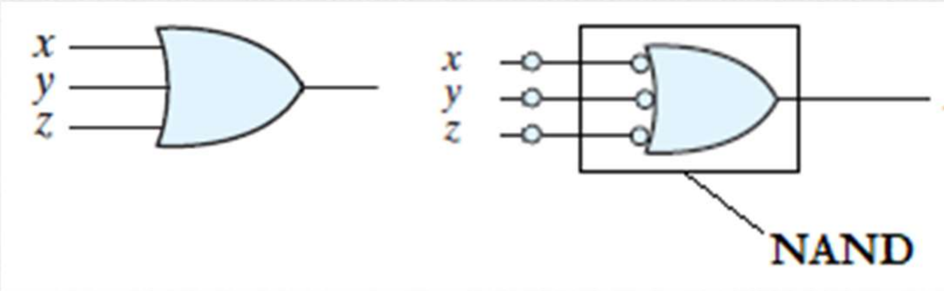
Building a digital circuit Using NAND(s)

2. To transform AND; add two bubbles (representing two inverters) after its output, one is pasted to the AND and the other after it but within the same line. A NAND is formed. The rested bubble is kept for now.



Building a digital circuit Using NAND(s)

3. To transform an OR; add two bubbles in each line fed in it (representing two inverters), one is pasted to the Or and the other before it but within the same line. A NAND is formed. The rested bubbles are kept for now.



Building a digital circuit Using NAND(s)

4. Now the unchanged gates left are the inverters and bubbles (also representing inverters). Successive inverters or bubbles are cancelling each other.
5. Left inverters and bubbles are replaced by 2-input NANDs where the two inputs are the same input of the original inverter or bubble.

Building a digital circuit Using NAND(s)

(Example-1)

<u>Example 1:</u>

Implement the following function using only NANDs.

$F(X,Y,Z)= \sum_m(1,2,3,4,5,7)$

Using k-map

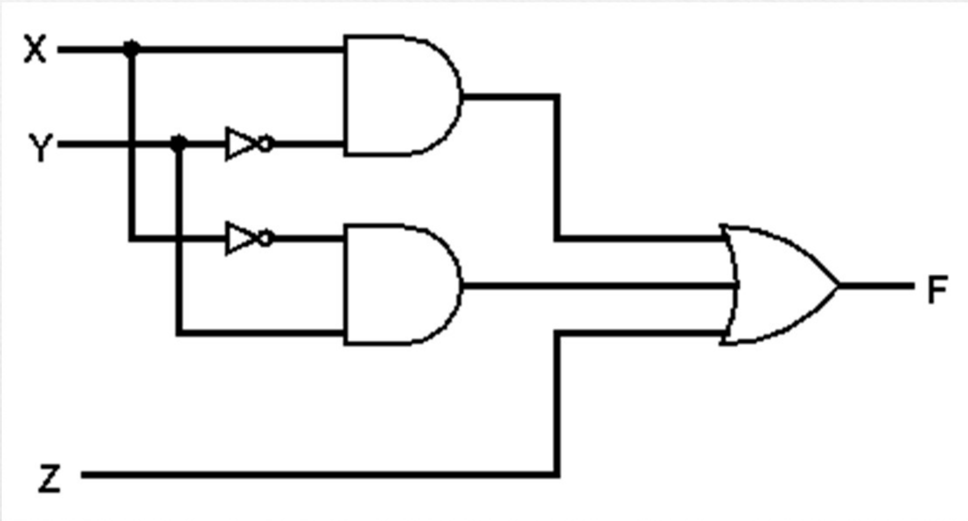
$F(X, Y, Z) = Z + X\bar{Y} + \bar{X}Y$

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

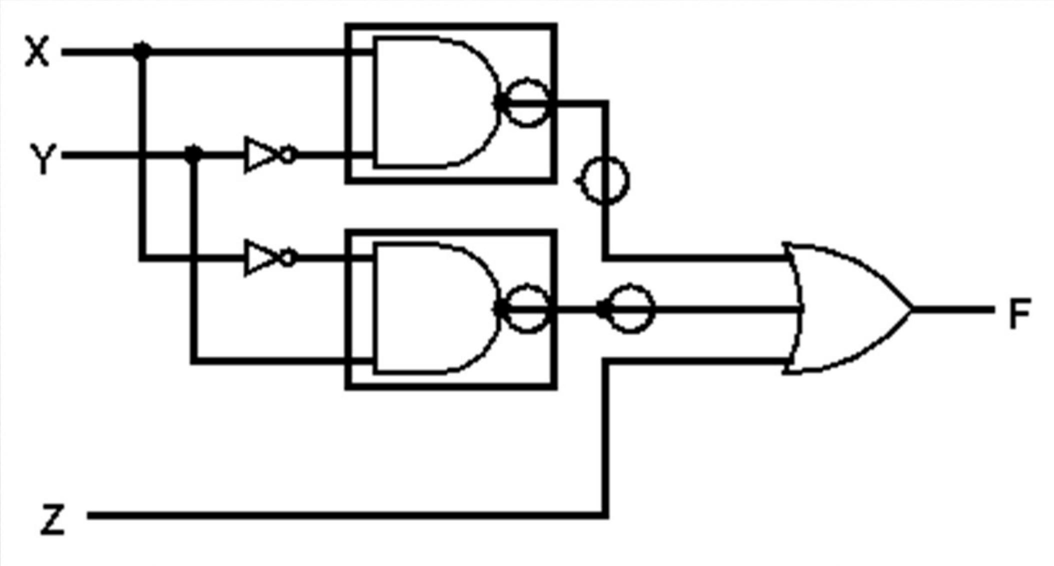
Building a digital circuit Using NAND(s) (Example-1)

Using k-map

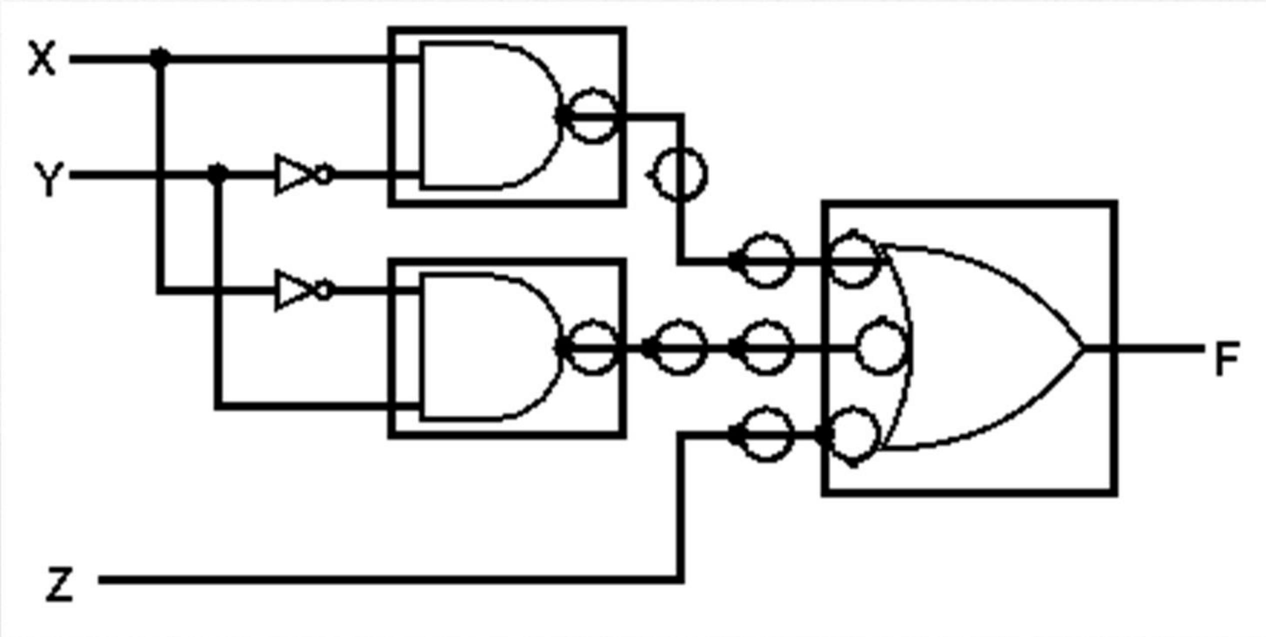
$$F(X, Y, Z) = Z + X\bar{Y} + \bar{X}Y$$



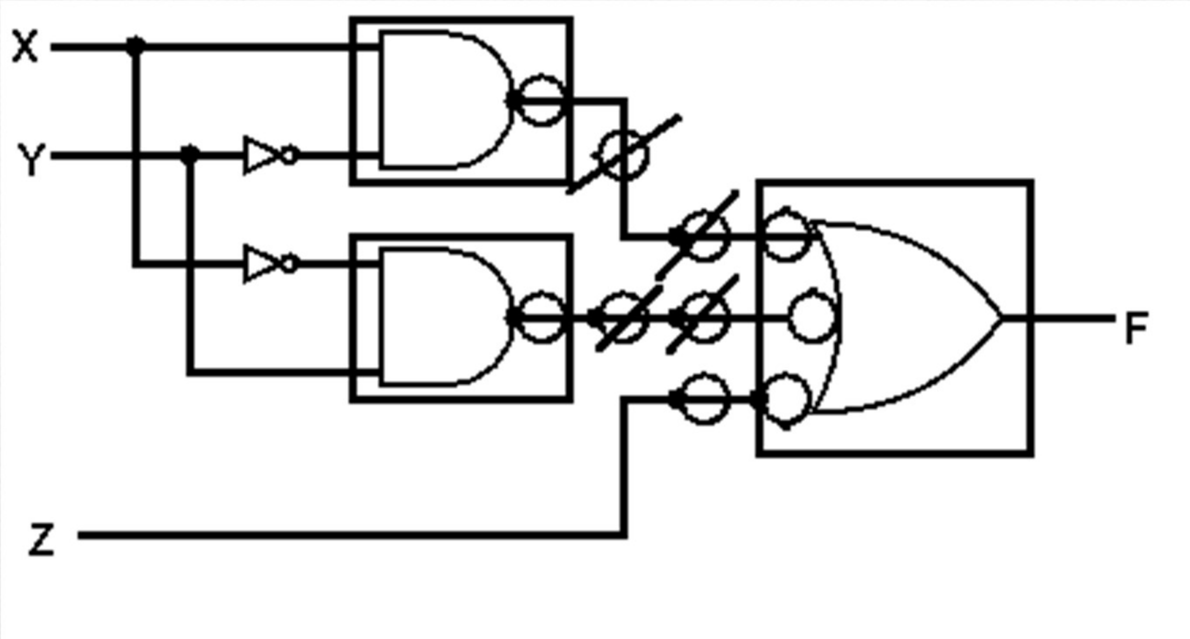
Building a digital circuit Using NAND(s) (Example-1)



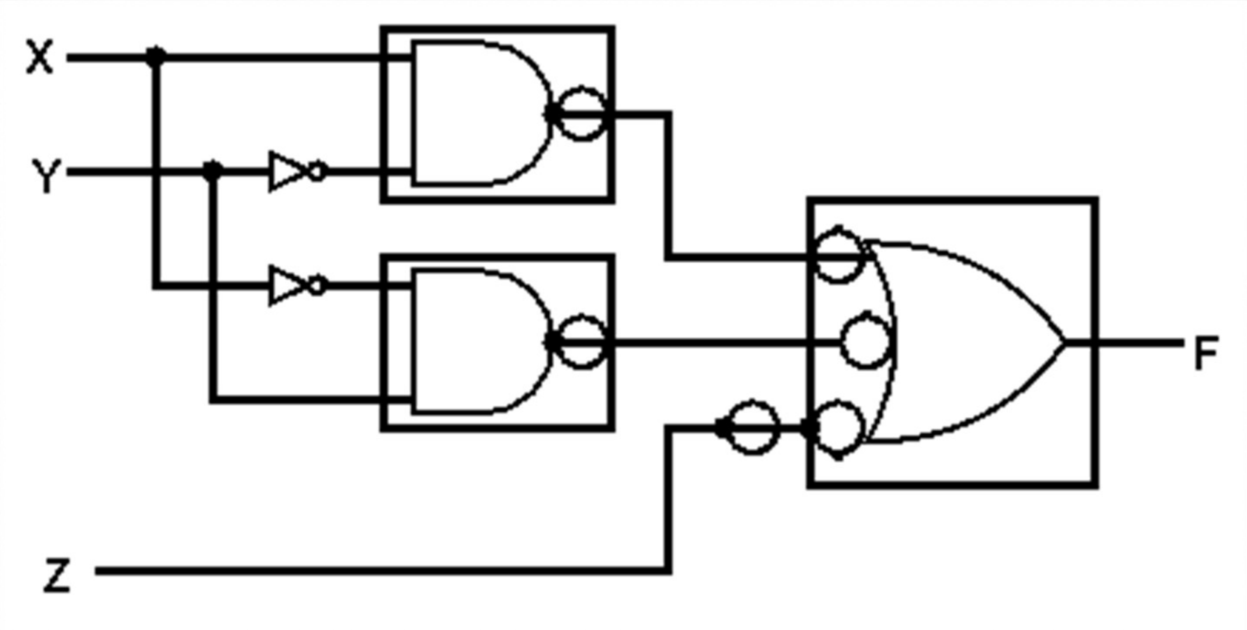
Building a digital circuit Using NAND(s) (Example-1)



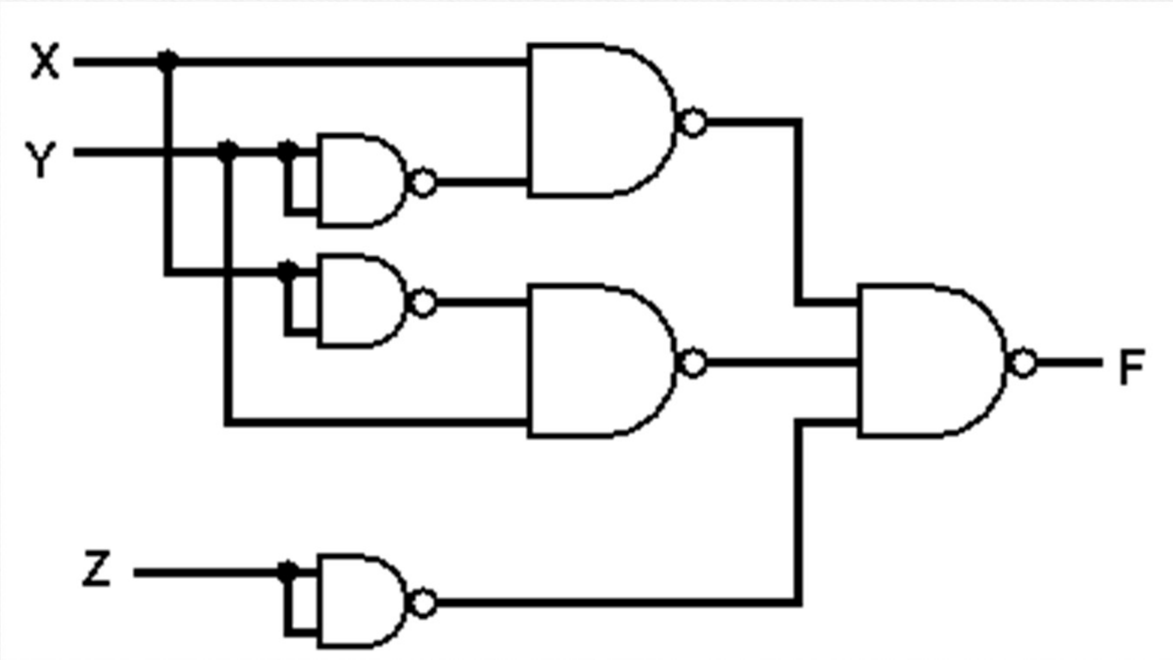
Building a digital circuit Using NAND(s)
(Example-1)



Building a digital circuit Using NAND(s) (Example-1)



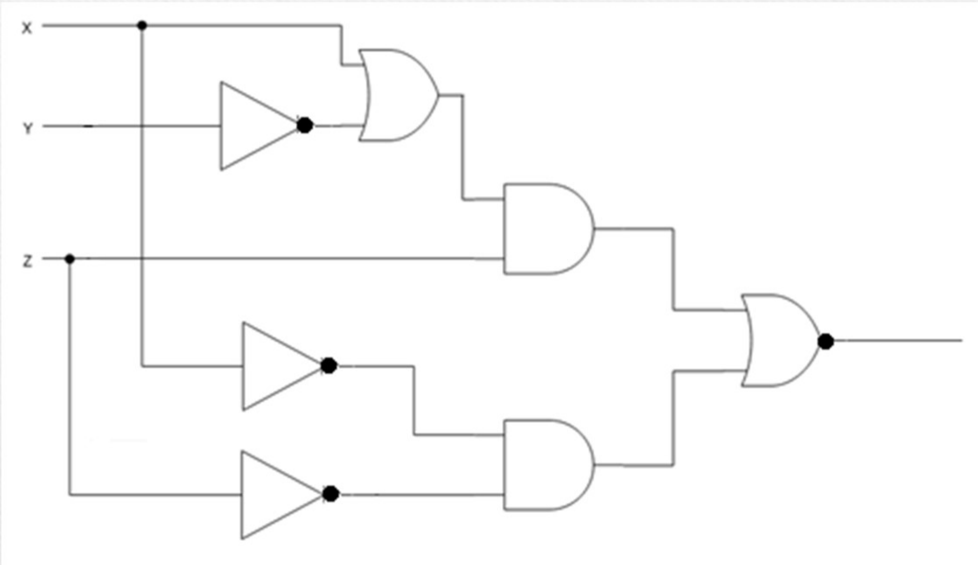
Building a digital circuit Using NAND(s) (Example-1)

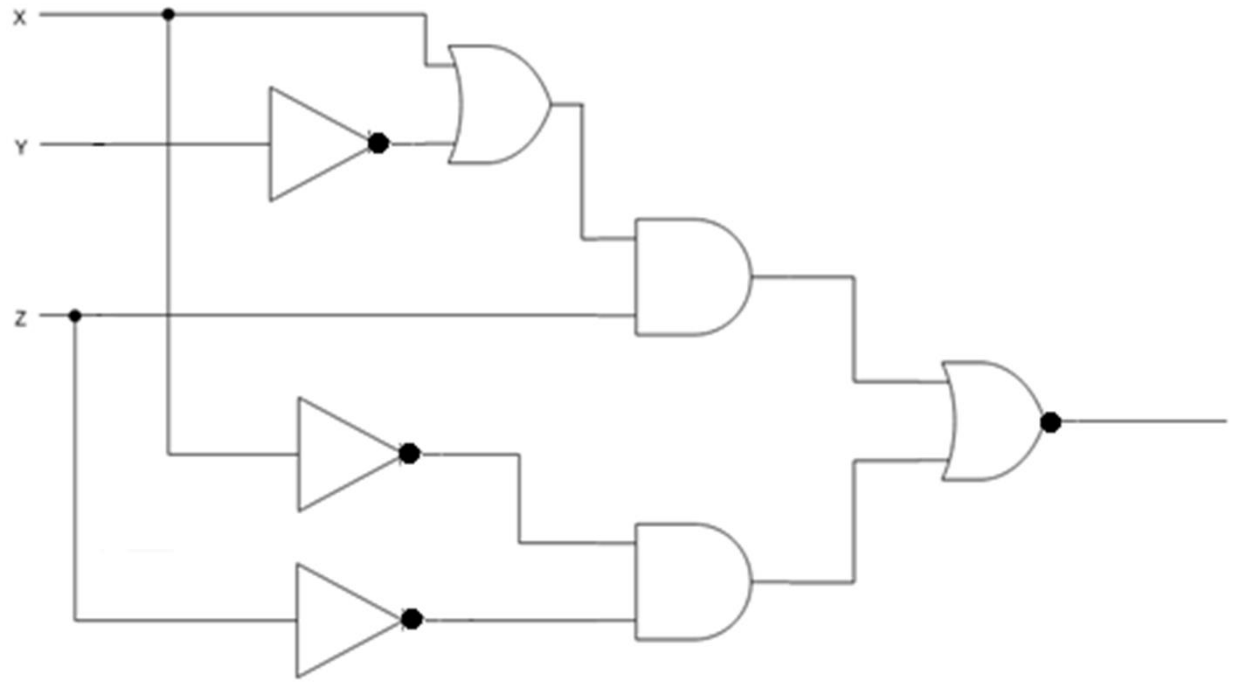


Building a digital circuit Using NAND(s)

<u>**Exercise:**</u>

Build the following function using NANDs only




X
Y
Z

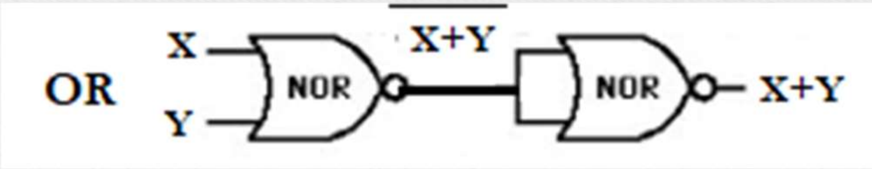
Building Simple Gates Using NOR(s)

Not gate:

$$\bar{X} = \overline{X + X}$$
$$= NOR(X, X)$$

Or gate:

$$X + Y = \overline{\overline{X + Y}}$$
$$= NOT(NOR(X, Y))$$



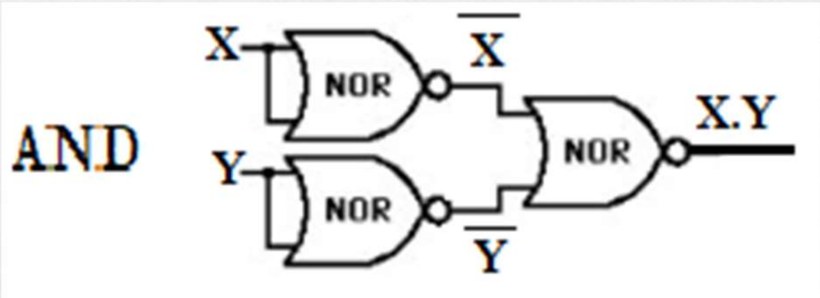
Building Simple Gates Using NOR(s)

And gate:

$$X.Y = \overline{\overline{X} + \overline{Y}}$$

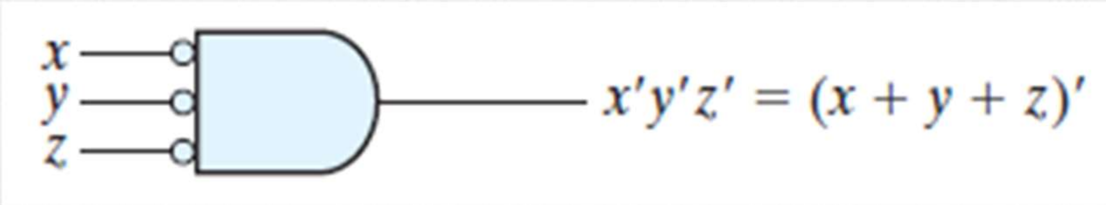
De Morgan law

$$= NOR(NOT(X), NOT(Y))$$



Building a digital circuit Using NOR(s)

Using this circuit help in transform And into NORs.



Building a digital circuit Using NOR(s)

1. **Obtain the simplest boolean function of the circuit it in terms of elementary boolean gates (Not – And – Or). If you have any other gates, replace it with simple gates, e.g. a NAND is replaced it by AND + NOT.**

Building a digital circuit Using NOR(s)

2. To transform an AND; add two bubbles (representing two inverters) after its output, one is pasted to the OR and the other after it but within the same line. A NOR is formed. The rested bubble is kept for now.

Building a digital circuit Using NOR(s)

3. To transform an AND; add two bubbles in each line fed in it (representing two inverters), one is pasted to the Or and the other before it but within the same line. A NOR is formed. The rested bubbles are kept for now.

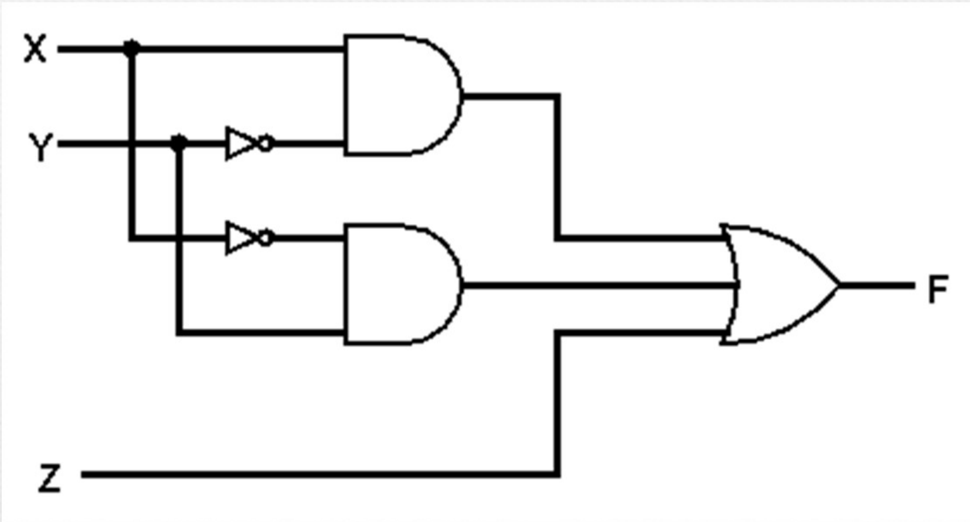
Building a digital circuit Using NOR(s)

4. Now the unchanged gates left are the inverters and bubbles (also representing inverters). Successive inverters or bubbles are cancelling each other.
5. Left inverters and bubbles are replaced by 2-input NORs where the two inputs are the same input of the original inverter or bubble.

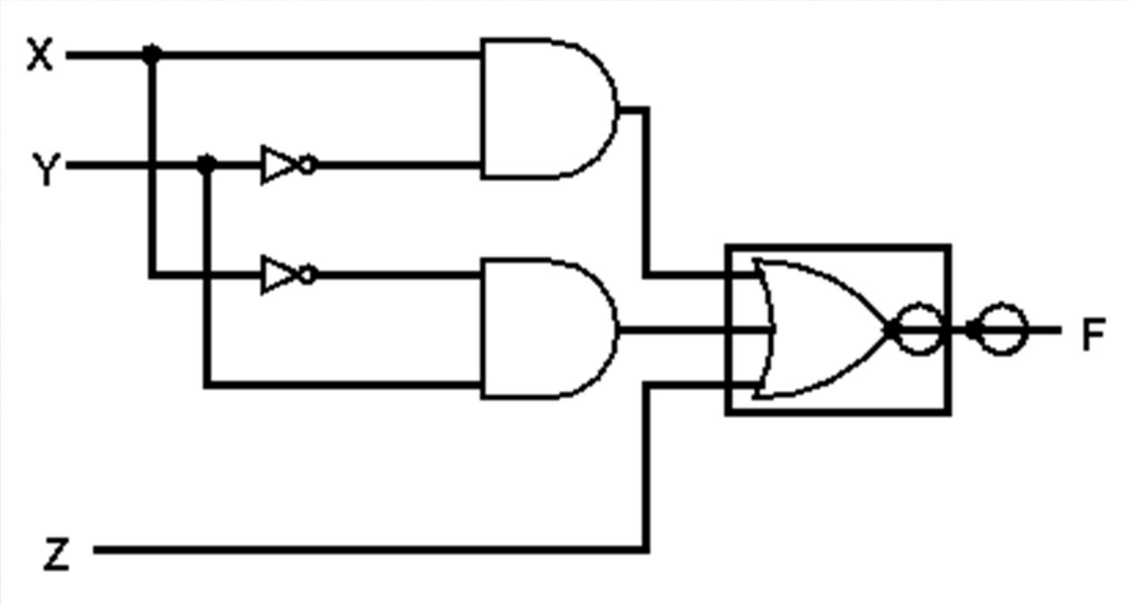
Building a digital circuit Using NOR(s) (Example-2)

Example 2:

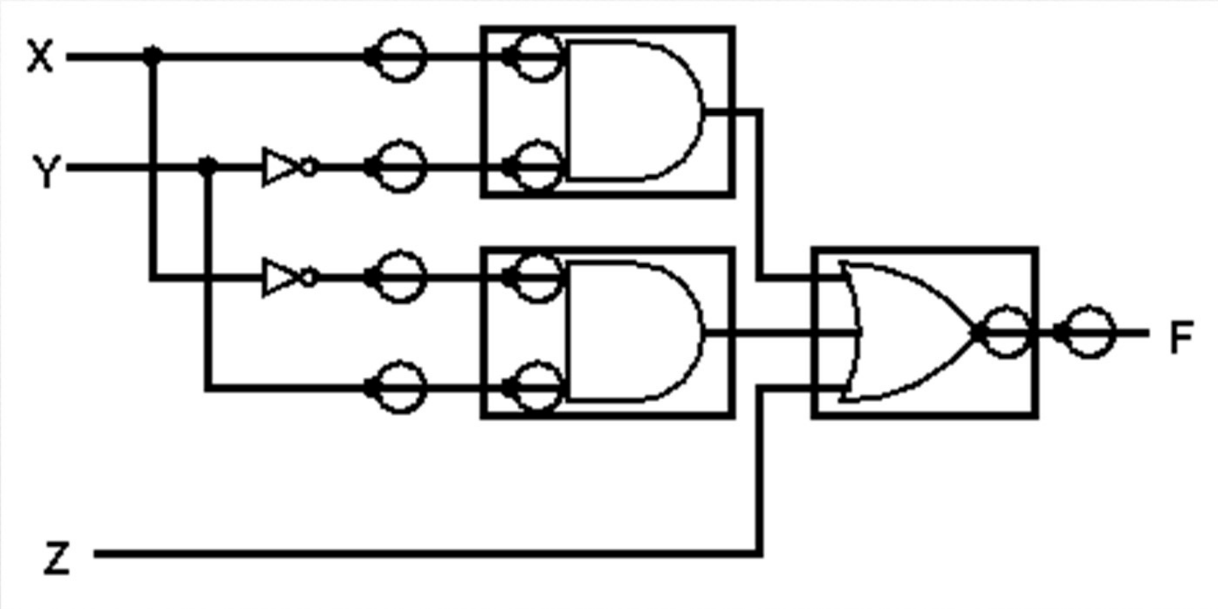
Implement the following function using only NORs.



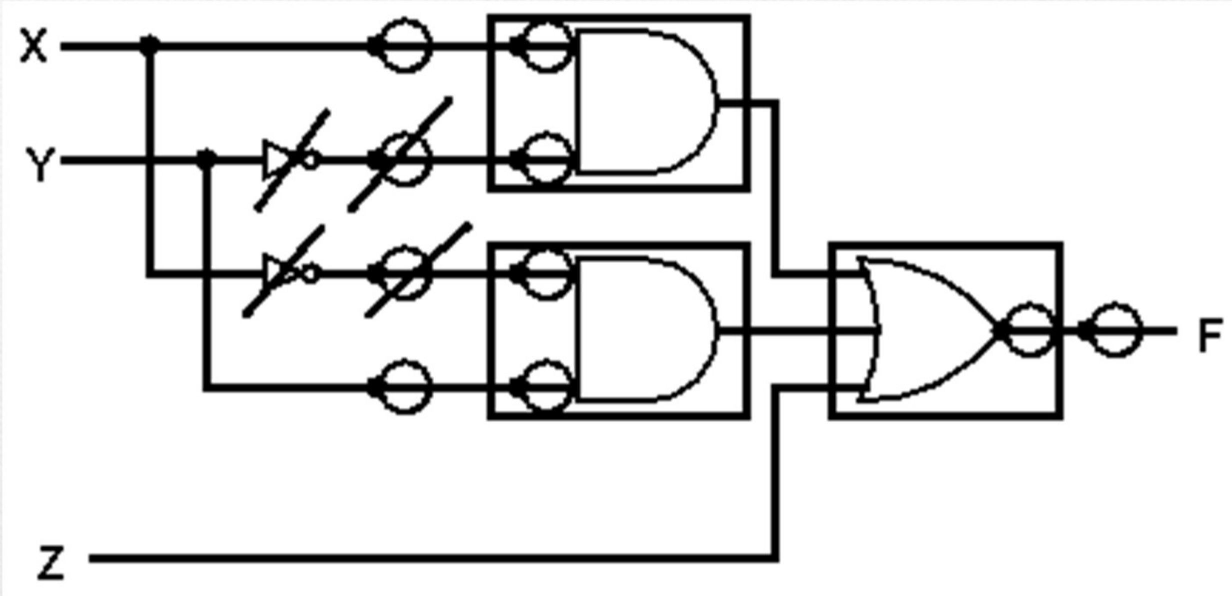
Building a digital circuit Using NOR(s) (Example-2)



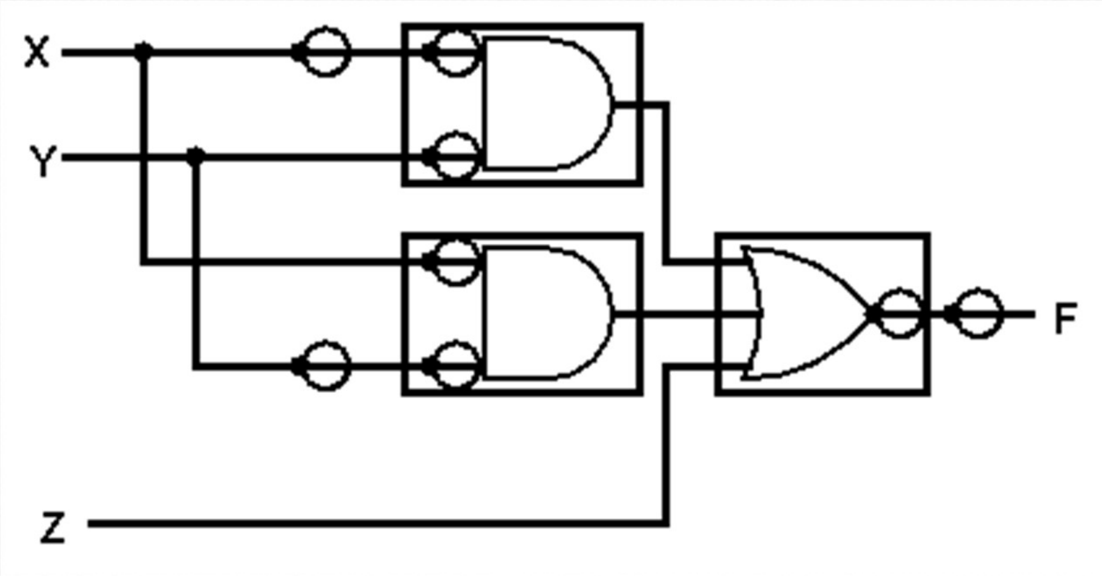
Building a digital circuit Using NOR(s)
(Example-2)



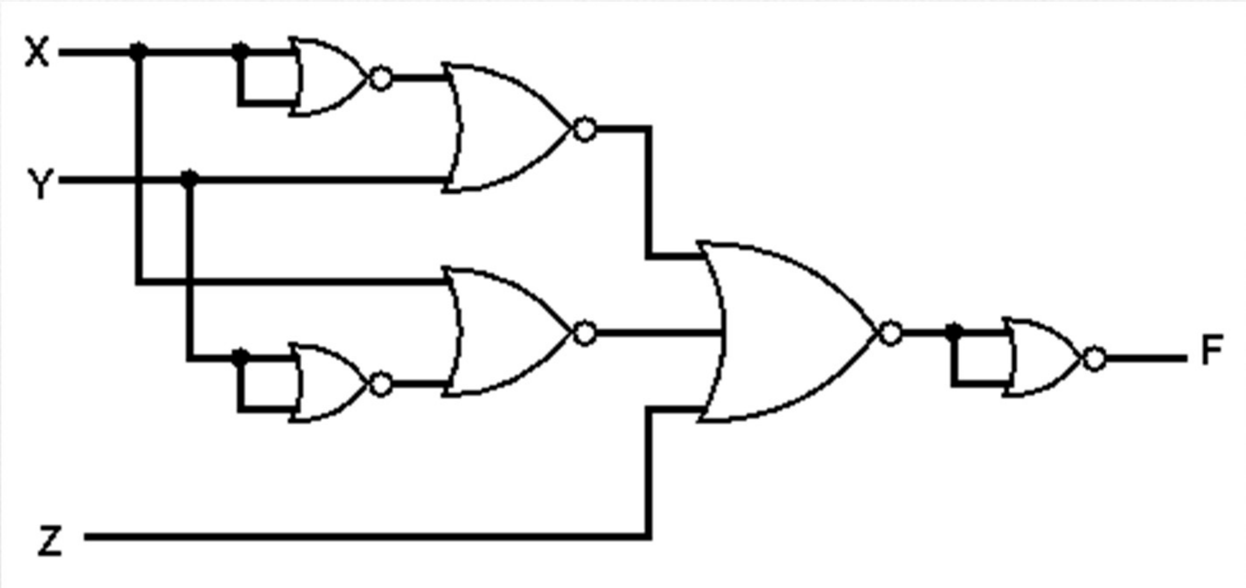
Building a digital circuit Using NOR(s)
(Example-2)



Building a digital circuit Using NOR(s)
(Example-2)



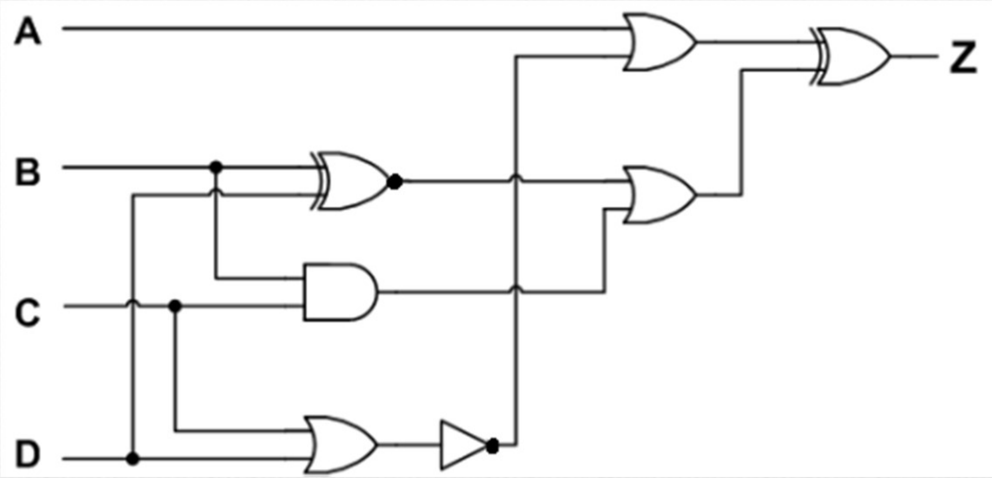
Building a digital circuit Using NOR(s) (Example-2)

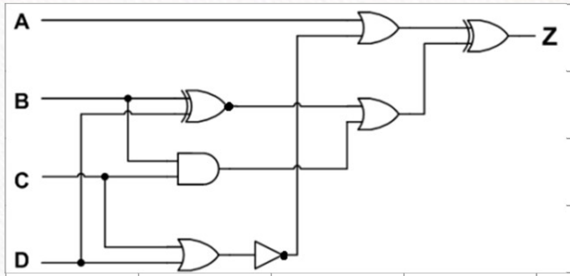


Building a digital circuit Using NOR(s)

<u>Exercise:</u>

Build the following function using NORs only




A
B
C
D
Z