German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Aysha ElSafty

Introduction to Computer Science, Winter Semester 2017
Practice Assignment 11

Discussion: 30.12.2017 - 04.01.2018

Exercise 11-1 To be Discussed in Tutorial

Simplify the Boolean expressions to a minimum number of literals using the Boolean algebra. Please mention the applied rules.

$x + 0 = x$	$x * 1 = x$	
$x + 1 = 1$	$x * 0 = 0$	
$x + x = x$	$x * x = x$	
$x + x' = 1$	$x * x' = 0$	
$(x')' = x$		
$x + y = y + x$	$xy = yx$	Commutativity
$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$	Associativity
$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$	Distributivity
$(x + y)' = x'y'$	$(xy)' = x' + y'$	DeMorgan's Law

a) $ABC + ABC' + A'B$

Solution:

$$ABC + ABC' + A'B$$

$=$	$AB(C + C') + A'B$	**(Distributivity)**
$=$	$AB * 1 + A'B$	$[x + x' = 1]$
$=$	$AB + A'B$	$[x * 1 = x]$
$=$	$BA + BA'$	**(Commutativity)**
$=$	$B(A + A')$	**(Distributivity)**
$=$	$B * 1$	$[x + x' = 1]$
$=$	B	$[x * 1 = x]$

b) $(A + B)'(A' + B')$

Solution:

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

$= (A'B')(A' + B')$	$[(x + y)' = x'y']$
$= A'B'A' + A'B'B'$	**(Distributivity)**
$= A'A'B' + A'B'B'$	**(Commutativity)**
$= A'B' + A'B'$	$[x * x = x]$
$= A'B'$	$[x + x = x]$
$= (A + B)'$	**(DeMorgan's Law)**

c) $(A + B' + AB')(AB + A'C + BC)$

Solution:

$$\begin{aligned}
& (A + B' + AB')(AB + A'C + BC) & \\
&= (A + B'(1 + A))(AB + BC + A'C) & \textbf{(Distributivity)} \\
&= (A + B'(A + 1))(AB + BC + A'C) & \textbf{(Commutativity)} \\
&= (A + B' * 1)(AB + BC + A'C) & [(x + 1) = 1] \\
&= (A + B')(AB + BC + A'C) & [(x * 1) = x] \\
&= (A + B') * AB + (A + B') * BC + (A + B') * A'C & \textbf{(Distributivity)} \\
&= AB * (A + B') + BC * (A + B') + A'C * (A + B') & \textbf{(Commutativity)} \\
&= ABA + ABB' + BCA + BCB' + A'CA + A'CB' & \textbf{(Distributivity)} \\
&= AAB + ABB' + BCA + BB'C + AA'C + A'CB' & \textbf{(Commutativity)} \\
&= AB + ABB' + BCA + BB'C + AA'C + A'CB' & [(x * x) = x] \\
&= AB + 0 + BCA + 0 + 0 + A'CB' & [(x * x') = 0] \\
&= AB + BCA + A'CB' & [(x + 0) = x] \\
&= AB + ABC + A'B'C & \textbf{(Commutativity)} \\
&= AB(1 + C) + A'B'C & \textbf{(Distributivity)} \\
&= AB(C + 1) + A'B'C & \textbf{(Commutativity)} \\
&= AB * 1 + A'B'C & [(x + 1) = 1] \\
&= AB + A'B'C & [(x * 1) = x]
\end{aligned}$$

d) $P'XY + PX'Y + PXY' + PXY$

Solution:

$$\begin{aligned}
& P'XY + PX'Y + PXY' + PXY & \\
&= PXY + P'XY + PX'Y + PXY' & \textbf{(Commutativity)} \\
&= XYP + XYP' + PX'Y + PXY' & \textbf{(Commutativity)} \\
&= XY(P + P') + PX'Y + PXY' & \textbf{(Distributivity)} \\
&= XY * 1 + PX'Y + PXY' & [(x + x') = 1] \\
&= XY + PX'Y + PXY' & [(x * 1) = x] \\
&= XY + P(X'Y + XY') & \textbf{(Distributivity)}
\end{aligned}$$

e) $(AB)'(A + B)$

Solution:

$$\begin{aligned}
& (AB)'(A + B) & \\
&= (A' + B')(A + B) & [(xy)' = x' + y'] \\
&= (A' + B')A + (A' + B')B & \textbf{(Distributivity)} \\
&= A(A' + B') + B(A' + B') & \textbf{(Commutativity)} \\
&= AA' + AB' + BA' + BB' & \textbf{(Commutativity)} \\
&= 0 + AB' + BA' + 0 & [(x * x' = 0)] \\
&= AB' + 0 + BA' + 0 & \textbf{(Commutativity)} \\
&= AB' + BA' & [(x + 0 = x)]
\end{aligned}$$

To construct a circuit for $(AB)'(A+B)$, we will need 4 gates. To construct a circuit for $AB' + BA'$, we need 5 gates. Thus $(AB)'(A+B)$ is simpler than $AB' + BA'$.

f) $B + A'C + AB'$

Solution:

$$\begin{aligned}
& B + A'C + AB' \\
&= B + AB' + A'C && \textbf{(Commutativity)} \\
&= B + B'A + A'C && \textbf{(Commutativity)} \\
&= (B + B')(B + A) + A'C && [(x + yz) = (x + y)(x + z)] \\
&= 1 * (B + A) + A'C && [(x + x' = 1)] \\
&= (B + A) * 1 + A'C && \textbf{(Commutativity)} \\
&= (B + A) + A'C && [(x * 1 = x)] \\
&= A + A'C + B && \textbf{(Commutativity)} \\
&= (A + A')(A + C) + B && [(x + yz) = (x + y)(x + z)] \\
&= 1 * (A + C) + B && [(x + x' = 1)] \\
&= (A + C) * 1 + B && \textbf{(Commutativity)} \\
&= A + C + B && [(x * 1 = x)]
\end{aligned}$$

g) $AB + A'C + BC$

Solution:

$$\begin{aligned}
& AB + A'C + BC \\
&= AB + A'C + BC * 1 && [(x * 1 = x)] \\
&= AB + A'C + BC(A + A') && [(x + x' = 1)] \\
&= AB + A'C + BCA + BCA' && \textbf{(Distributivity)} \\
&= AB + A'C + ABC + A'CB && \textbf{(Commutativity)} \\
&= AB + ABC + A'C + A'CB && \textbf{(Commutativity)} \\
&= AB(1 + C) + A'C(1 + B) && \textbf{(Distributivity)} \\
&= AB(C + 1) + A'C(B + 1) && \textbf{(Commutativity)} \\
&= AB * 1 + A'C * 1 && [(x + 1 = 1)] \\
&= AB + A'C
\end{aligned}$$

Exercise 11-2

Given the following Boolean expression, simplify it to a minimum number of literals using the Boolean algebra. Please mention the applied rules.

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

Hint: The circuit of the simplified expression consists of zero gates.

Solution:

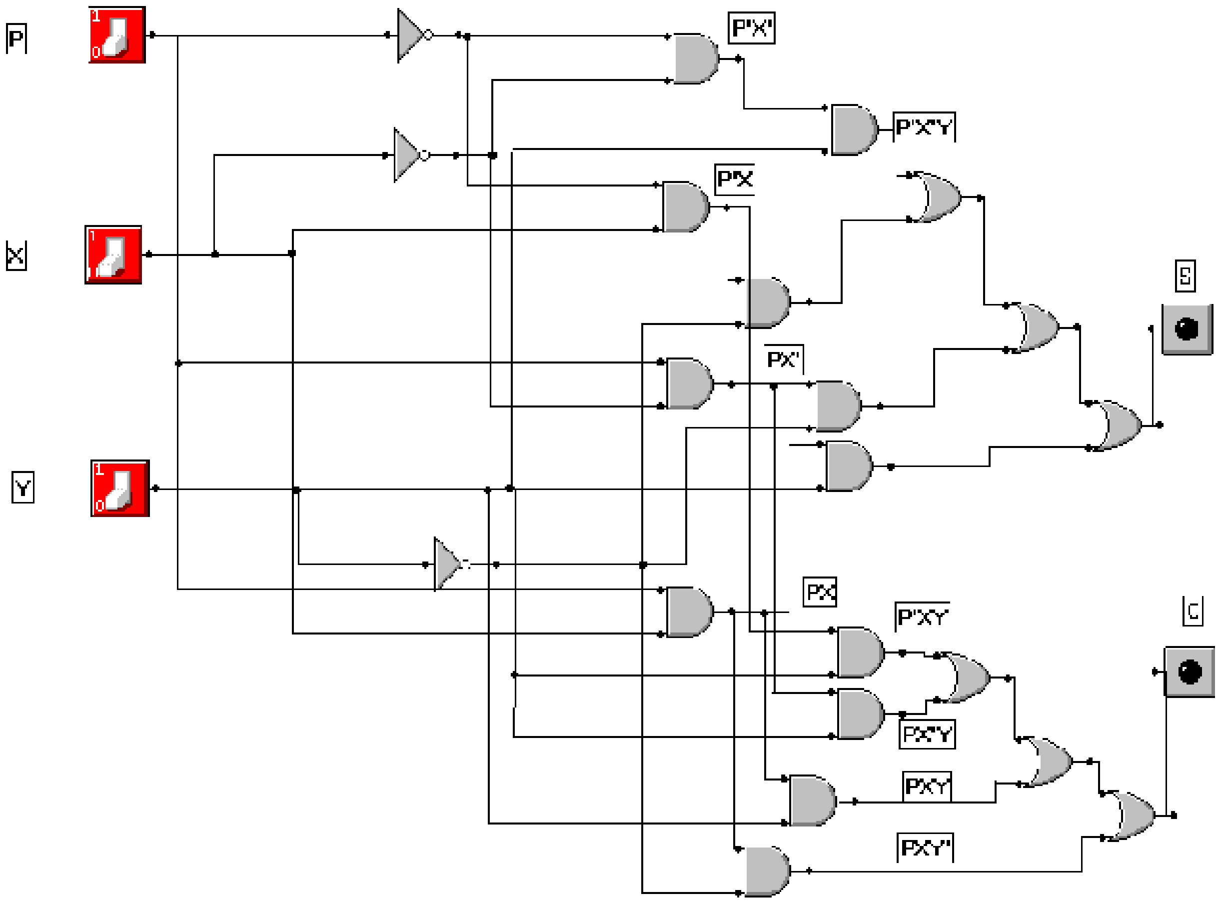
$$
\begin{array}{llr}
= & AB' + AC' + AD' + BB' + BC' + BD' + AB'C' + B'B'C' + B'C'C + A'C + D & (Distributivity) \\
= & AB' + AC' + BB' + BC' + BD' + AB'C' + B'C' + B'C'C + A'C + D + AD' & (Associativity) \\
= & AB' + AC' + BB' + BC' + BD' + AB'C' + B'C' + B'C'C + A'C + (D + A)(D + D') & (Distributivity) \\
= & AB' + AC' + BB' + BC' + BD' + AB'C' + B'C' + B'C'C + A'C + (D + A)(1) & (x + x' = 1) \\
= & AB' + AC' + BB' + BC' + BD' + AB'C' + B'C' + B'C'C + A'C + D + A & (x * 1 = x) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + A'C + D + A + BC' + B'C' & (Associativity) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + A'C + D + A + C'(B + B') & (Distributivity) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + A'C + D + A + C'(1) & (x + x' = 1) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + A'C + D + A + C' & (x * 1 = x) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + A + C' + A'C & (Associativity) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + A + (C' + A')(C' + C) & (Distributivity) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + A + (C' + A')(1) & (x + x' = 1) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + A + C' + A' & (x * 1 = x) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + C' + (A + A') & (Associativity) \\
= & AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + C' + 1 & (x + x' = 1) \\
= & (AB' + AC' + BB' + BD' + AB'C' + B'C'C + D + C') + 1 & (x + 1 = 1) \\
= & 1 &
\end{array}
$$

Exercise 11-3

Use AND, OR and NOT gates to implement the circuits represented by the following two expressions:

$$
\begin{aligned}
S &= P'X'Y + P'XY' + PX'Y' + PXY \\
C &= P'XY + PX'Y + PXY' + PXY
\end{aligned}
$$

Solution:

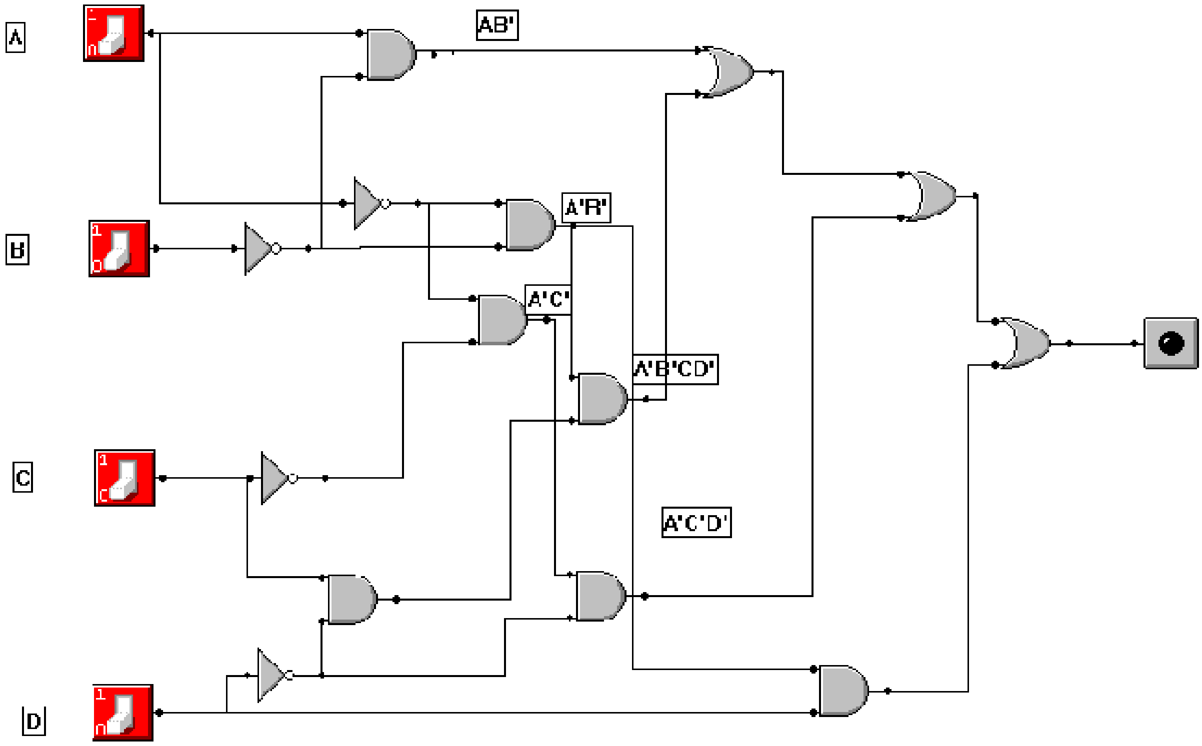

P
X
Y
P'X'
P'X'Y
P'X
PX'
S
PX
P'XY
PX'Y
PXY
PXY'
C

Exercise 11-4 To be Discussed in Tutorial

Draw a logic circuit that corresponds to each of the expressions shown below:

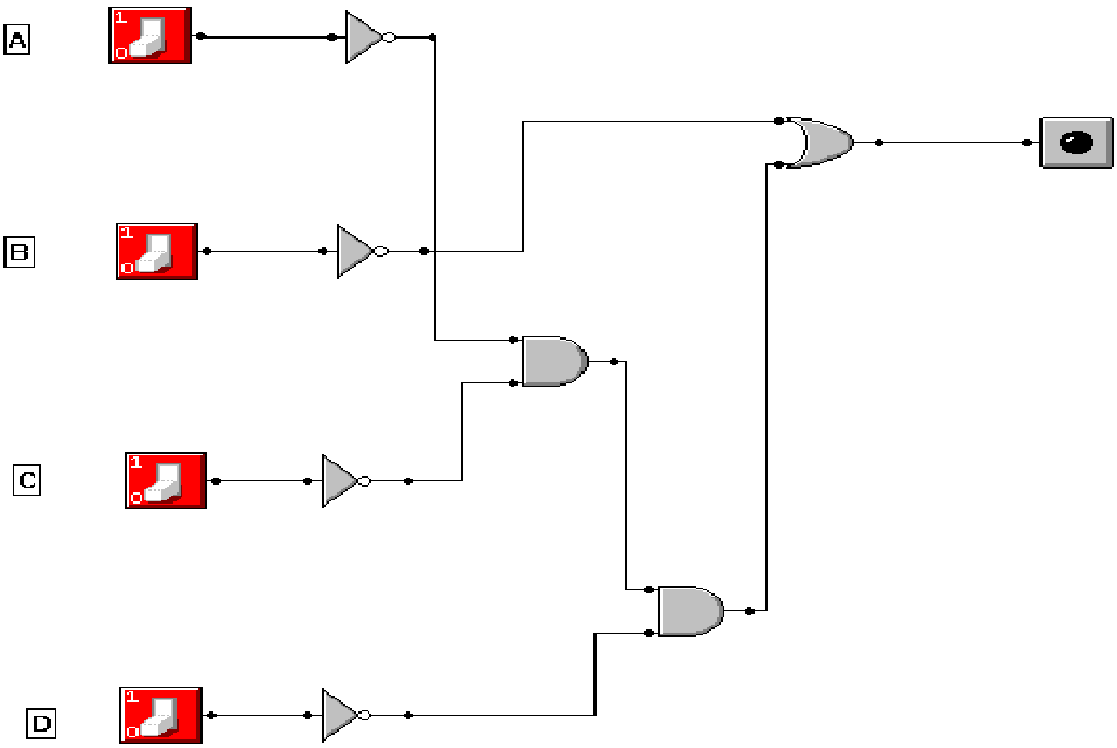
a) $AB' + A'C'D' + A'B'D + A'B'CD'$

Solution:



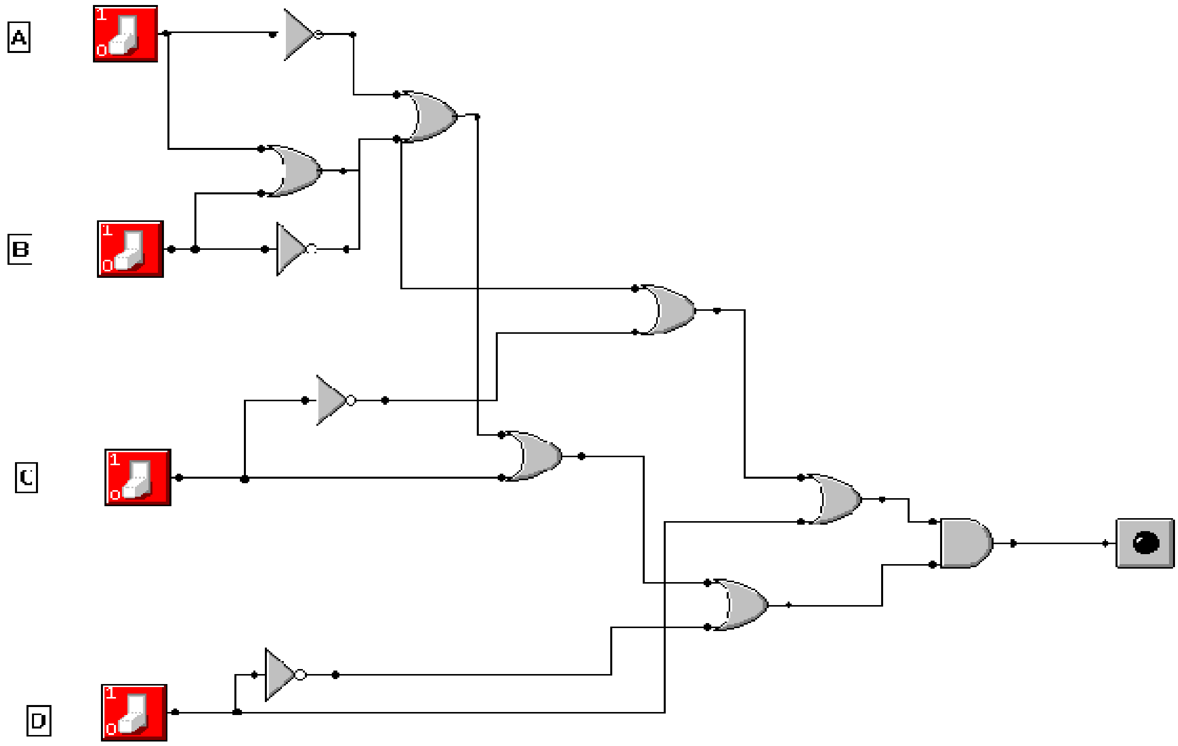
b) $B' + A'C'D'$

Solution:



c) $(A' + B' + C + D')(A + B + C' + D)$

Solution:



Exercise 11-5

Given the following the following truth table, where **A**, **B** and **C** are the input variables and **X** is the output variable.

A	**B**	**C**	**X**
0	0	0	1
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	0

a) Use the sum-of-products algorithm to find the Boolean expression that describes the output of the truth table.

Solution:

$$x = A'B'C' + A'BC + AB'C + ABC'$$

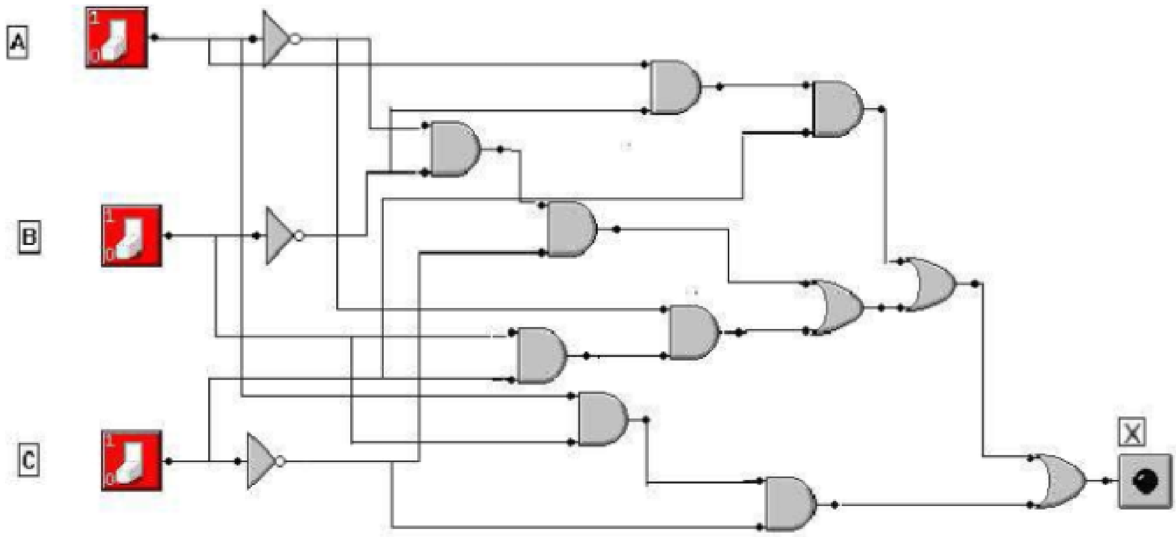
b) What is the functionality of the circuit?

Solution:

The circuit computes the parity of a number. The parity bit is equal to 1 if the number of ones is even otherwise, the parity bit is equal to 0.

c) Draw the Boolean circuit. **Note** that each gate can have only two inputs.

Solution:



Exercise 11-6 To be Discussed in Tutorial
Comparator

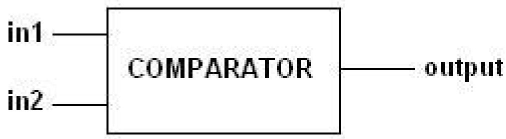
A one-bit comparator is a circuit that takes two numbers consisting of one bit each and outputs 1 if the numbers are equal, 0 otherwise.

a) Construct a truth table for a one bit equality comparator.

Solution:

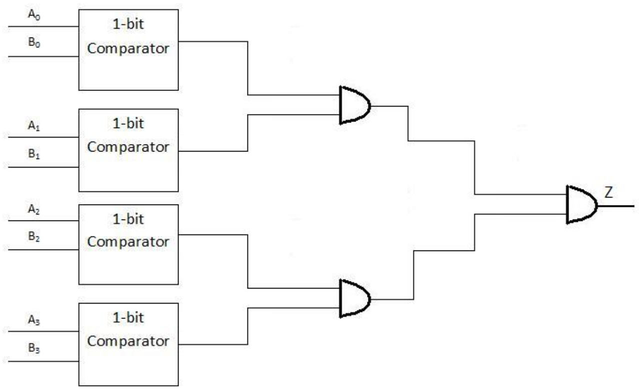
x	y	Output
0	0	1
0	1	0
1	0	0
1	1	1

b) Assume that you have already manufactured one-bit comparators.

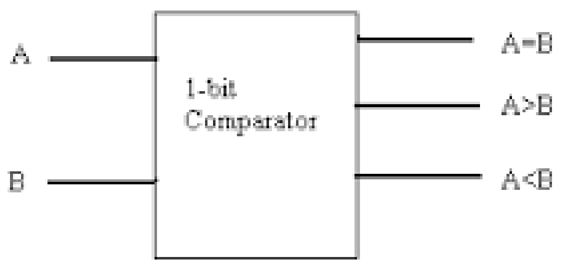


Design a circuit that uses one-bit comparators and AND-gates to check the equality of two numbers consisting of 4 bits each.

Solution:

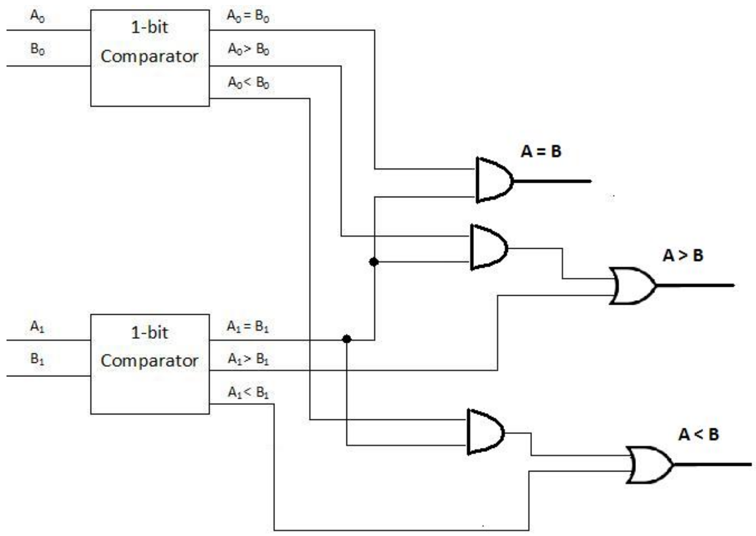


c) Assume that our one-bit comparator was modified to have two input variables A, B and three output variables (one checking for $A = B$, one checking for $A > B$ and one checking for $A < B$).



Design a circuit that uses the modified one-bit comparators with other gates to compare two numbers consisting of 2 bits each. **Do not draw the truth table.**

Solution:



Exercise 11-7 To be Discussed in Tutorial

A circuit should be designed to perform the operation $(A - 1)$ where A represents a number in sign/magnitude notation consisting of 2 bits.

a) How many output variables are needed? Justify your answer.

Solution:

3 output variables are needed. 2 bits in sign magnitude can represent a range of $[-1, 1]$. Thus the output of calculating $-1 - 1$, which is equal to -2, needs 3 bits to represent it in sign/magnitude notation (110).

b) Construct the truth table for this circuit.

Solution:

A1	A2	O1	O2	O3
0	0	1	0	1
0	1	0	0	0
1	0	1	0	1
1	1	1	1	0

c) Use the sum-of-products algorithm to find the Boolean expressions that corresponds to the truth table.

Solution:

$$\begin{aligned} O1 = &\ A1'A2' \quad +A1A2' + A1A2 \\ O2 = &\ A1A2 \\ O3 = &\ A1'A2' \quad +A1A2' \end{aligned}$$

d) Simplify the Boolean expressions that you got in c) to a minimum number of literals using the Boolean algebra. Please mention the applied rules.

Solution:

$$
\begin{aligned}
O1 &= A1'A2' + A1A2' + A1A2 & \\
O1 &= A2'A1' + A2'A1 + A1A2 & (Commutativity)\\
O1 &= A2'(A1' + A1) + A1A2 & (Distributivity)\\
O1 &= A2' * 1 + A1A2 & (x' + x = 1)\\
O1 &= A2' + A1A2 & (x * 1 = x)\\
O1 &= (A2' + A1)(A2' + A2) & (Distributivity)\\
O1 &= (A2' + A1) * 1 & (x' + x = 1)\\
O1 &= (A2' + A1) & (x * 1 = x)
\end{aligned}
$$

$$O2 = A1A2$$

$$
\begin{aligned}
O3 &= A1'A2' + A1A2' & \\
O3 &= A2'A1' + A2'A1 & (Commutativity)\\
O3 &= A2'(A1' + A1) & (Distributivity)\\
O3 &= A2' * 1 & (x' + x = 1)\\
O3 &= A2' & (x * 1 = x)
\end{aligned}
$$

e) Draw a logic circuit that corresponds to the simplified expressions you got in d).

Solution: