Ejercicio 1

Simplificar las siguientes funciones booleanas a un número mínimo de literales.

a) $x \cdot y + x \cdot y'$

b) $(x+y) \cdot (x+y')$

c) $x \cdot y \cdot z + x' \cdot y + x \cdot y \cdot z'$

d) $z \cdot x + z \cdot x' \cdot y$

e) $(A+B)' \cdot (A'+B')'$

f) $y \cdot (w \cdot z' + w \cdot z) + x \cdot y$

Solución: **Punto a)**
La función $x \cdot y + x \cdot y'$ se puede simplificar de la siguiente manera:

$$\begin{aligned} x \cdot y + x \cdot y' &= x \cdot (y + y')(\text{postulado 4.}) \\ &= x \cdot 1(\text{postulado 5.}) \\ &= x(\text{postulado 2.}) \end{aligned}$$

Punto b)
La función $(x+y) \cdot (x+y')$ se puede simplificar de la siguiente manera:

$$\begin{aligned} (x+y) \cdot (x+y') &= x + y \cdot y'(\text{postulado 4.}) \\ &= x + 0(\text{postulado 5.}) \\ &= x(\text{postulado 2.}) \end{aligned}$$

Punto c)
La función $x \cdot y \cdot z + x' \cdot y + x \cdot y \cdot z'$ se puede simplificar de la siguiente manera:

$$\begin{aligned} x \cdot y \cdot z + x' \cdot y + x \cdot y \cdot z' &= y \cdot (x \cdot z + x' + x \cdot z')(\text{postulado 4.}) \\ &= y \cdot (x \cdot z + x \cdot z' + x')(\text{postulado 3.}) \\ &= y \cdot (x \cdot (z + z') + x')(\text{postulado 4.}) \\ &= y \cdot (x \cdot (1) + x')(\text{postulado 5.}) \\ &= y \cdot (x + x')(\text{postulado 2.}) \\ &= y \cdot 1(\text{postulado 5.}) \\ &= y(\text{postulado 2.}) \end{aligned}$$

Punto d)
La función $z \cdot x + z \cdot x' \cdot y$ se puede simplificar de la siguiente manera:

$$z \cdot x + z \cdot x' \cdot y = z \cdot (x + x' \cdot y)(\text{postulado 4.}) \qquad = z \cdot (x + y)(\text{redundancia})$$

Punto e)
La función $(A+B)' \cdot (A'+B')'$ se puede simplificar de la siguiente manera:

$$\begin{aligned} (A+B)' \cdot (A'+B')' &= (A' \cdot B') \cdot (A \cdot B)(\text{teorema 5.}) \\ &= (A' \cdot A) \cdot (B' \cdot B)(\text{postulado 3.}) \\ &= 0 \cdot 0(\text{postulado 5.}) \\ &= 0(\text{teorema 2.}) \end{aligned}$$

Punto f)
La función $y \cdot (w \cdot z' + w \cdot z) + x \cdot y$ se puede simplificar de la siguiente manera:

$$\begin{aligned} y \cdot (w \cdot z' + w \cdot z) + x \cdot y &= y \cdot (w \cdot (z' + z)) + x \cdot y (\text{postulado 4.}) \\ &= y \cdot (w \cdot 1) + x \cdot y (\text{postulado 5.}) \\ &= y \cdot w + x \cdot y (\text{postulado 2.}) \\ &= y \cdot (w + x) (\text{postulado 4.}) \end{aligned}$$

Ejercicio 2

Reducir a un número mínimo de literales las siguientes funciones booleanas:

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

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

c) $[(A.B)'.A].[(A.B)'.B]$

d) $A.B' + C'.D'$

a) Graficar las expresiones encontradas en "b" y "d" mediante cualquier tipo de compuertas del número de entradas necesarias.

b) Encontrar expresiones equivalentes a las funciones "b" y "d", pero utilizando sólo compuertas NAND del número de entradas necesarias.

c) Graficar las expresiones encontradas en el punto anterior.

Solución: **Reducción de funciones booleanas**

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

$$\begin{aligned} &(B.C' + A'.D).(A.B' + C.D') \\ &= (B.C' + A'.D).A.B' + (B.C' + A'.D).C.D' (\text{postulado 4}) \\ &= (B.C').(A.B') + (A'.D).(A.B') + (B.C').(C.D') + (A'.D).(C.D') (\text{postulado 4}) \\ &= (B.B').(C'.A) + (A.A').(D.B') + (C.C').(B.D') + (D.D').(A'.C) (\text{teorema 4}) \\ &= 0.(C'.A) + 0.(D.B') + 0.(B.D') + 0.(A'.C) (\text{postulado 5}) \\ &= 0 (\text{teorema 2}) \end{aligned}$$

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

$$
\begin{aligned}
&= B'.D + A'.B.C' + A'.B.C + A.C.D (\text{postulado } 3)\\
&= B'.D + A'.B.(C' + C) + A.C.D (\text{teorema } 4)\\
&= B'.D + A'.B.C' + A.C.D + A'.B.C\\
&= B'.D + A'.B + A.C.D (\text{teorema } 6)\\
&= B'.D + B'.D.A'.C + A'.B + A'.B.C.D + A.C.D (\text{postulado } 2 \text{ y } 5)\\
&= B'.D + B'.D.A'.C + A'.B + A'.B.C.D + A.C.D.(B + B') (\text{postulado } 4)\\
&= B'.D + B'.D.A'.C + A'.B + A'.B.C.D + A.C.D.B + A.C.D.B' (\text{postulado } 3\text{, teorema } 4)\\
&= B'.D + A'.B + (A'.B.C.D + A.C.D.B) + (A.C.D.B' + B'.D.A'.C) (\text{postulado } 3)\\
&= B'.D + A'.B + (A'.B.C.D + A.B.C.D) + (A.B'.C.D + A'.B'.C.D) (\text{postulado } 4)\\
&= B'.D + A'.B + B.C.D.(A' + A) + B'.C.D.(A' + A) (\text{postulados } 5{,}2 \text{ yteorema } 4)\\
&= B'.D + A'.B + (B.C.D + B'.C.D) (\text{postulado } 4)\\
&= B'.D + A'.B + C.D.(B + B') (\text{postulado } 5 \text{ y teorema } 2)\\
&= B'D + A'B + CD
\end{aligned}
$$

c) $[(A.B)'.A].[(A.B)'.B]$

$$
\begin{aligned}
&[(A.B)'.A].[(A.B)'.B]\\
&= [(A' + B').A].[(A' + B').B] (\text{teorema } 5)\\
&= (A'.A + B'.A).(A'.B + B'.B) (\text{postulado } 4)\\
&= (0 + B'.A).(A'.B + 0) (\text{postulado } 5)\\
&= B'.A.A'.B (\text{postulado } 2)\\
&= A.A'.B'.B (\text{teorema } 4)\\
&= 0 (\text{postulado } 5)
\end{aligned}
$$

d) $A.B' + C'.D'$
No se puede simplificar más, ya que en sus términos no hay ninguna relación.

Punto a)

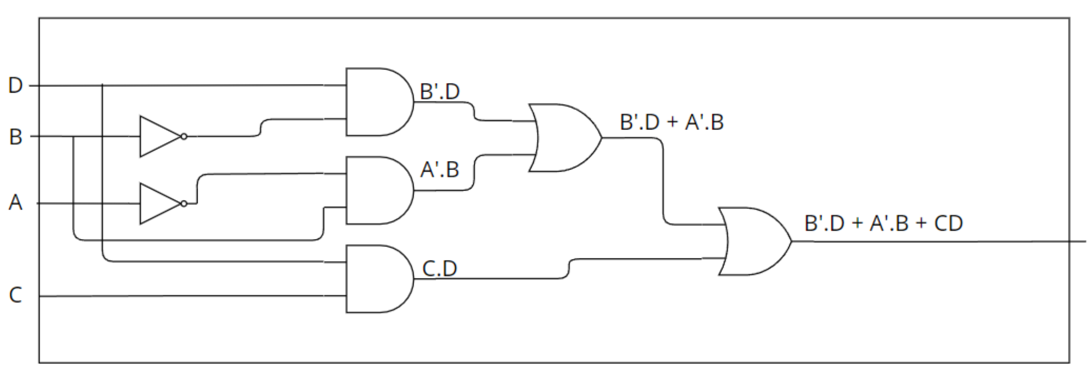

Figure 1: Circuito para la función $B'D + A'B + CD$

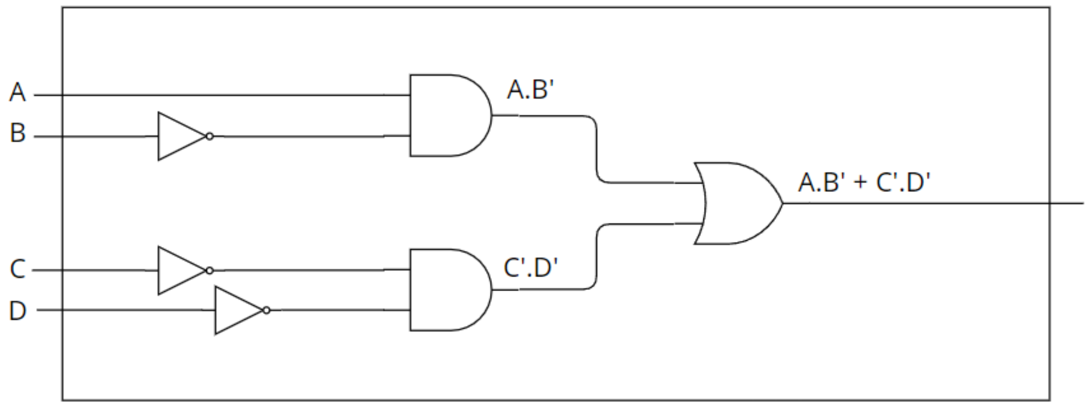

Figure 2: Circuito para la función $A.B' + C'.D'$

Punto b)

b) $B'.D + A'.B + CD$ para buscar el equivalente con compuertas NAND.

$$\begin{aligned}
&B'.D + A'.B + CD \\
&= (B'.D + A'.B + CD)'' \\
&= ((B'.D)' \cdot (A'.B)' \cdot (C.D)')'
\end{aligned}$$

d) $A.B' + C'.D'$ para buscar el equivalente con compuertas NAND.

$$\begin{aligned}
&A.B' + C'.D' \\
&= (A.B' + C'.D')'' \\
&= ((A.B')' \cdot (C'.D')')'
\end{aligned}$$

Punto c)

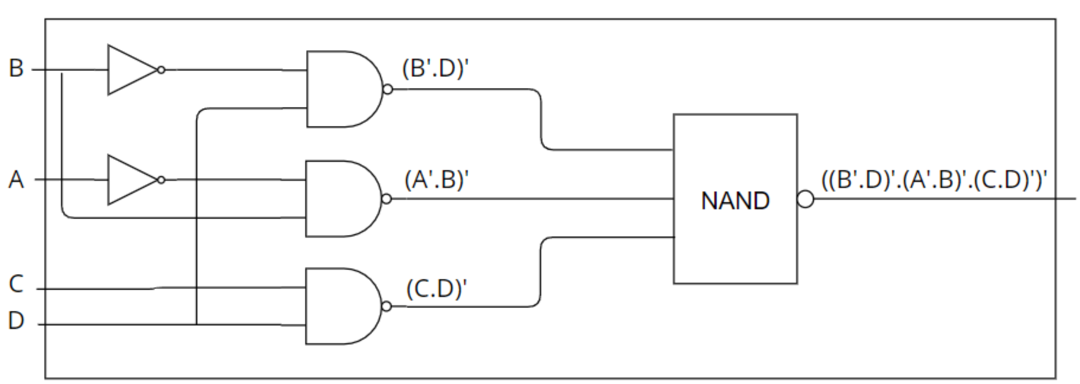

Figure 3: Circuito para la función $B'.D + A'.B + CD$ con compuertas NAND

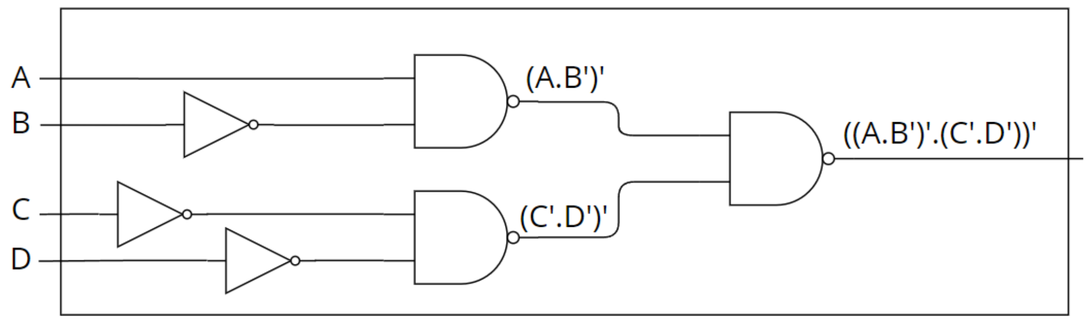

Figure 4: Circuito para la función $A.B' + C'.D'$ con compuertas NAND

Ejercicio 3

La función OR-exclusiva, denotada por ˆ tiene dos entradas y una salida. Si a y b son las entradas y c es la salida, entonces c es 1 sólo cuando exactamente una de las entradas vale 1. En el resto de los casos es 0.

a) Hacer una tabla de verdad de la función OR-exclusiva.

b) Encontrar la expresión equivalente a la función OR-exclusiva utilizando sólo suma de productos y graficar con compuertas.

c) Implementar una OR-exclusiva de 3 entradas usando OR-exclusivas de 2 entradas.

Solución: **Punto a)**
La tabla de verdad de la función OR-exclusiva es la siguiente:

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

Punto b)
La expresión equivalente a la función OR-exclusiva utilizando sólo suma de productos es:

$$c = (a \cdot b') + (a' \cdot b)$$

El circuito para la función OR-exclusiva es el siguiente:

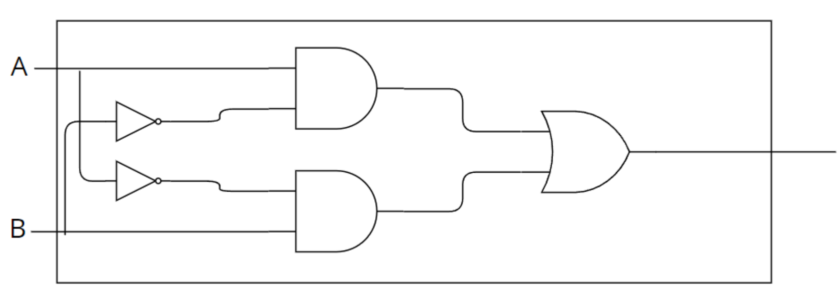

Figure 5: Circuito para la función OR-exclusiva

Punto c)
La función OR-exclusiva de 3 entradas se puede implementar de la siguiente manera:

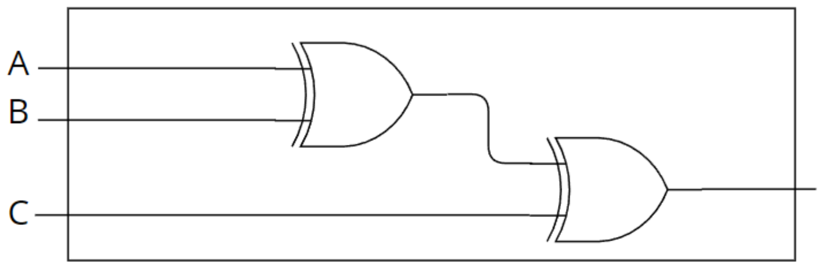

Figure 6: Circuito para la función OR-exclusiva de 3 entradas

Ejercicio 4

Mostrar que la función NAND (Not AND) es universal en el sentido de que las funciones NOT, AND, OR y NOR se pueden expresar como productos negados. Graficar las implementaciones de las compuertas NOT, AND, OR y NOR con compuertas NAND.
Solución: La función NAND es universal en el sentido de que las funciones NOT, AND, OR y NOR se pueden expresar como productos negados.

NOT: La función NOT se puede expresar como un producto negado de la siguiente manera:

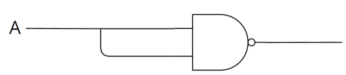

Figure 7: Circuito para la función NOT con compuertas NAND

La tabla de verdad de la función NOT es la siguiente:

a	c
0	1
1	0

AND: La función AND se puede expresar como un producto negado de la siguiente manera:

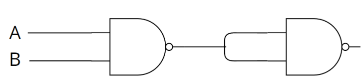

Figure 8: Circuito para la función AND con compuertas NAND

La tabla de verdad de la función AND es la siguiente:

a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

OR: La función OR se puede expresar como un producto negado de la siguiente manera:

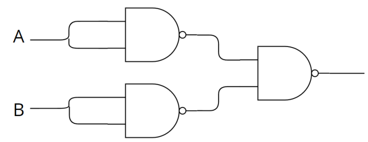

Figure 9: Circuito para la función OR con compuertas NAND

La tabla de verdad de la función OR es la siguiente:

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

NOR: La función NOR se puede expresar como un producto negado de la siguiente manera:

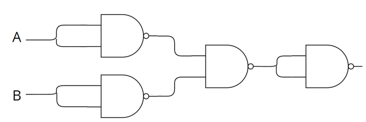

Figure 10: Circuito para la función NOR con compuertas NAND

La tabla de verdad de la función NOR es la siguiente:

a	b	c
0	0	1
0	1	0
1	0	0
1	1	0

Ejercicio 5

Mostrar que la función NOR (Not OR) es universal en el sentido de que las funciones NOT, OR, AND y NAND se pueden expresar como sumas negadas. Graficar las implementaciones de las compuertas NOT, OR, AND y NAND con compuertas NOR.
Solución: La función NOR es universal en el sentido de que las funciones NOT, OR, AND y NAND se pueden expresar como sumas negadas.

NOT: La función NOT se puede expresar como una suma negada de la siguiente manera:

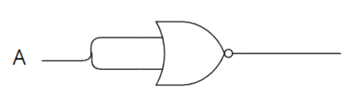

Figure 11: Circuito para la función NOT con compuertas NOR

AND: La función AND se puede expresar como una suma negada de la siguiente manera:

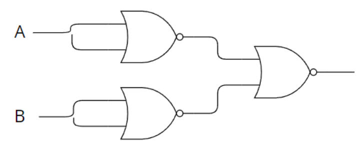

Figure 12: Circuito para la función AND con compuertas NOR

OR: La función OR se puede expresar como una suma negada de la siguiente manera:

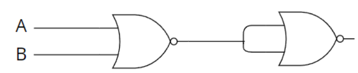

Figure 13: Circuito para la función OR con compuertas NOR

NAND: La función NAND se puede expresar como una suma negada de la siguiente manera:

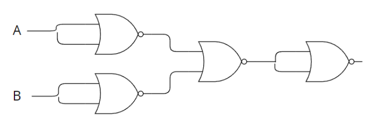

Figure 14: Circuito para la función NAND con compuertas NOR