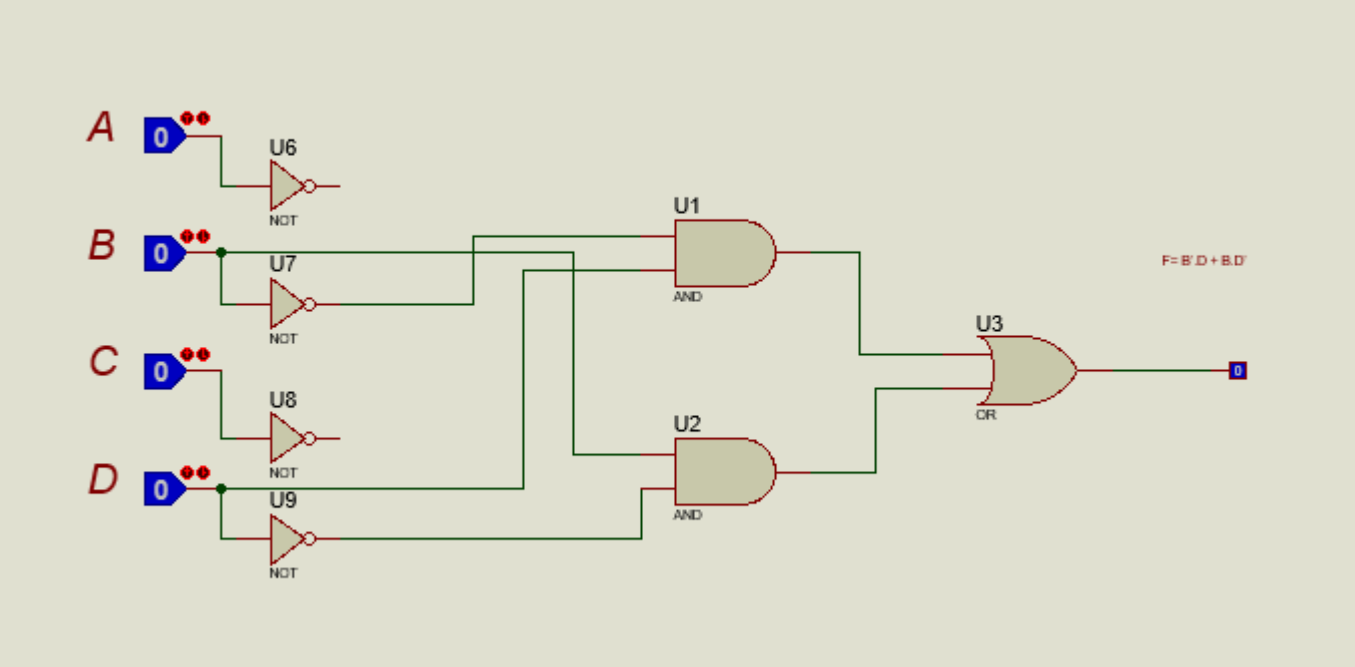
Ejercicio 1: Análisis de circuitos combinacionales Dado el siguiente circuito combinacional que utiliza puertas AND, OR y NOT, analizar la función lógica que representa y expresarla en forma de minitérminos y maxitérminos.

Entradas										Salidas			
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	S_0	S_1	S_2	S_3
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Ejercicio 2: **Síntesis de circuitos combinacionales Dada las siguientes funciones lógicas, simplificarlas utilizando el método de Karnaugh y construir un circuito combinacional que la represente utilizando puertas lógicas básicas.**

a) F(A, B, C, D) = Σm(1, 3, 5, 6, 9, 11, 12, 14)



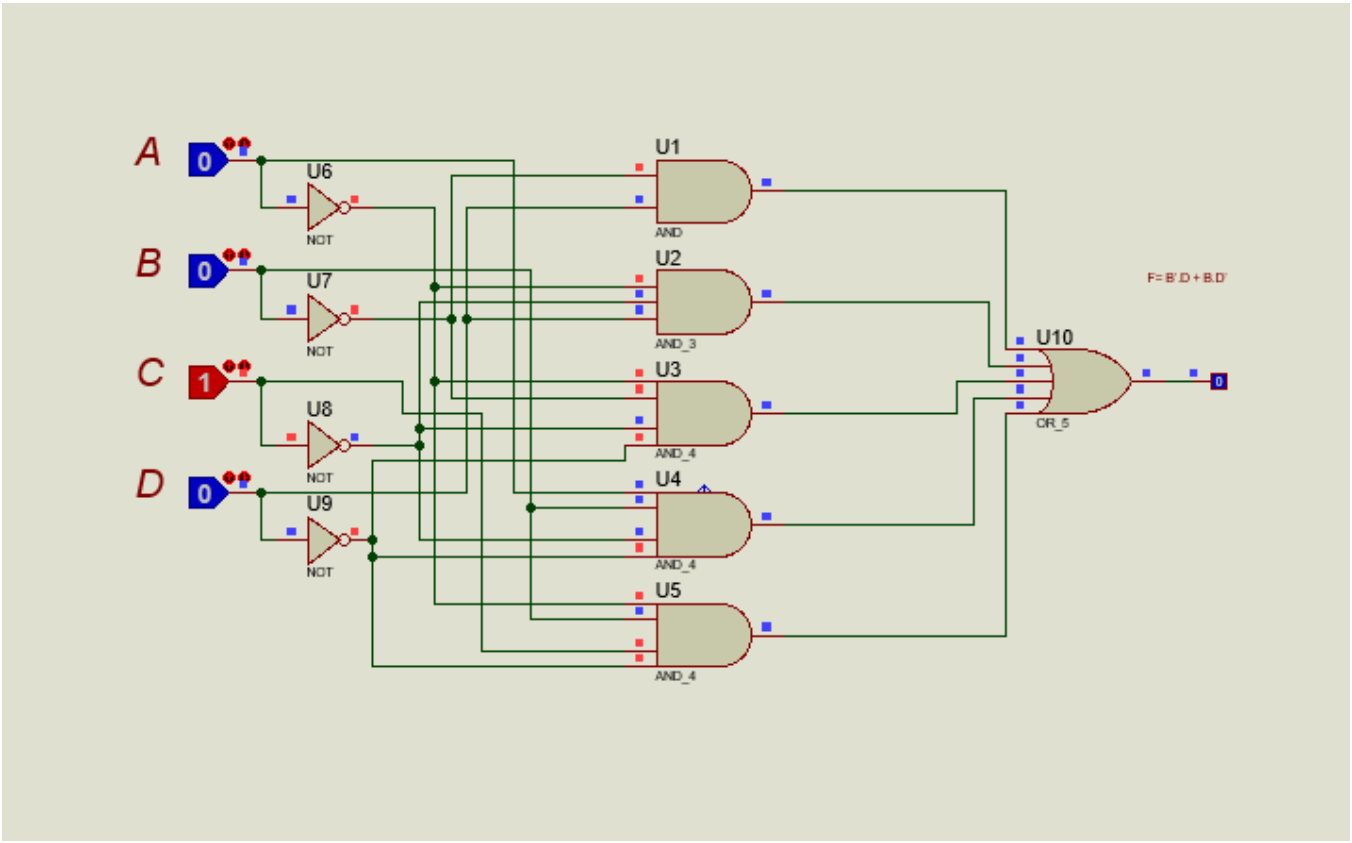
A	B	C	D	F	
0	0	0	0		
0	0	0	1	1	A'.B'.C'.D
0	0	1	0		
0	0	1	1	1	A'.B'.C.D
0	1	0	0		
0	1	0	1	1	A'.B.C'.D
0	1	1	0	1	A'.B.C.D'
0	1	1	1		
1	0	0	0		
1	0	0	1	1	A.B'.C'.D
1	0	1	0		
1	0	1	1	1	A.B'.C.D
1	1	0	0	1	A.B.C'.D'
1	1	0	1		
1	1	1	0	1	A.B.C.D'
1	1	1	1		

CD / AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

F= B'.D + B.D'

A'.B'.C.D+ A'.B.C'.D+ A'.B.C.D'+ A.B'.C'.D+ A.B'.C.D+ A.B.C'.D'+ A.B.C.D'

b) F(A, B, C, D) = ΠM(2, 4, 7, 8, 10, 13, 14, 15)



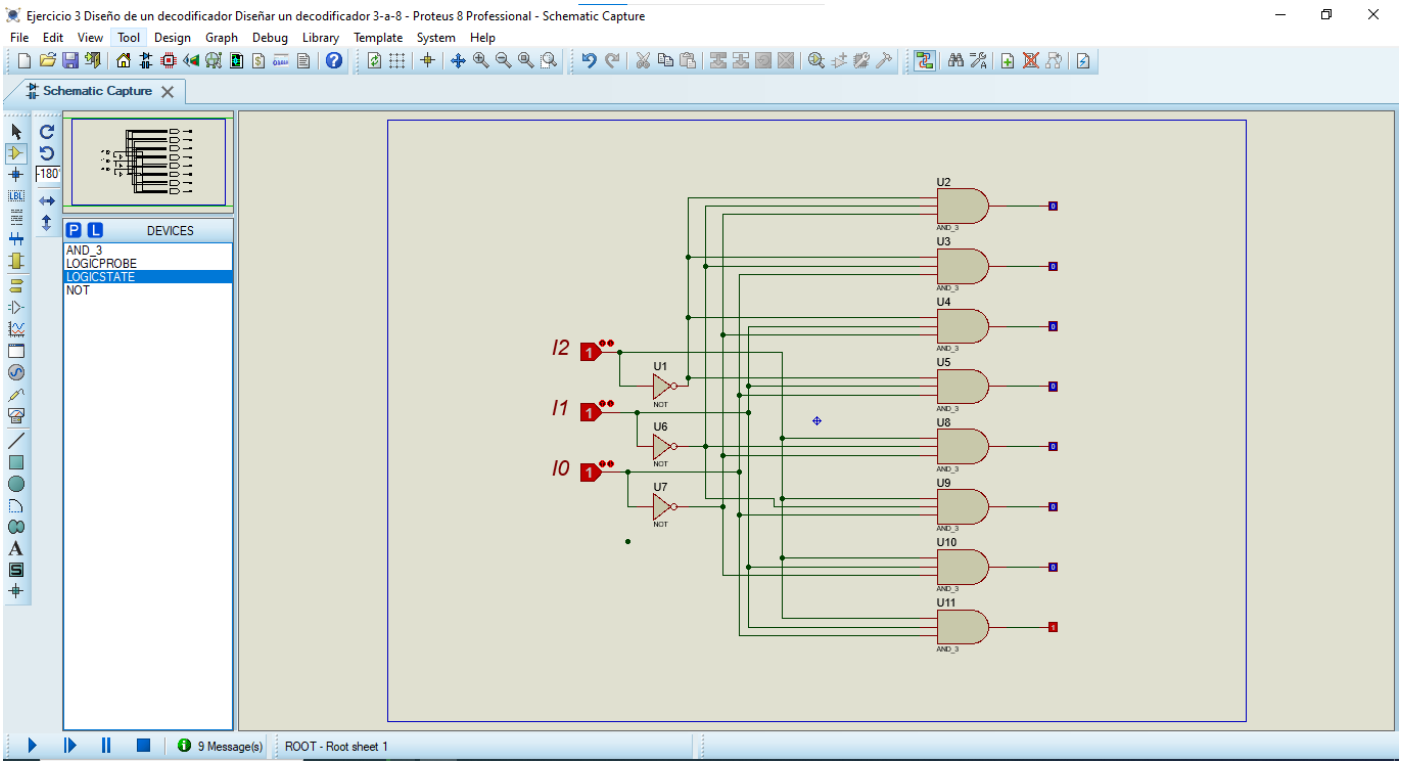
A	B	C	D	F	
0	0	0	0		
0	0	0	1		
0	0	1	0	0	A'.B'.C.D'
0	0	1	1		
0	1	0	0	0	A'.B.C'.D'
0	1	0	1		
0	1	1	0		
0	1	1	1	0	A'.B.C.D
1	0	0	0	0	A.B'.C'.D'
1	0	0	1		
1	0	1	0	0	A.B'.C.D'
1	0	1	1		
1	1	0	0		
1	1	0	1	0	A.B.C'.D
1	1	1	0	0	A.B.C.D'
1	1	1	1	0	A.B.C.D

CD / AB	00	01	11	10
00	1	1	1	
01	0	1	0	1
11	1	0	0	0
10	0	1	1	0

F= B'D+A'C'D+A'B'C'D'+ABC'D'+A'BCD'

(A'.B'.C.D')*(A'.B.C'.D')*(A'.B.C.D)*(A.B'.C'.D')*(A.B'.C.D')*(A.B.C'.D)*(A.B.C.D')*(A.B.C.D)

3: Diseño de un decodificador Diseñar un decodificador 3-a-8 utilizando puertas lógicas AND, OR y NOT. El decodificador debe convertir un código binario de 3 bits en 8 salidas, activando una salida única correspondiente al valor binario de entrada.



I0	I1	I2	S0	S1	S2	S3	S4	S5	S6	S7	F
0	0	0	1	0	0	0	0	0	0	0	**I0'.I1'.I2'**
0	0	1	0	1	0	0	0	0	0	0	**I0'.I1'.I2**
0	1	0	0	0	1	0	0	0	0	0	**I0'.I1.I2'**
0	1	1	0	0	0	1	0	0	0	0	**I0'.I1.I2**
1	0	0	0	0	0	0	1	0	0	0	**I0.I1'.I2'**
1	0	1	0	0	0	0	0	1	0	0	**I0.I1'.I2**
1	1	0	0	0	0	0	0	0	1	0	**I0.I1.I2'**
1	1	1	0	0	0	0	0	0	0	1	**I0.I1.I2**

Ejercicio 4: Introducción a circuitos secuenciales y flip-flops Investigar y describir brevemente las características y tipos de flip-flops (SR, D, JK, T). Explicar cómo estos elementos de memoria pueden ser utilizados en circuitos secuenciales.

Bistable: Mantienen un estado estable (0 o 1) hasta recibir una señal de control.

Sincronizados: Cambian de estado solo en flancos específicos del reloj.

Memoria: Almacenan un bit de información.

Versátiles: Se utilizan en una amplia gama de circuitos secuenciales.

Tipos de flip-flops:

Flip-flop SR (Set-Reset):
Entradas: S (Set) y R (Reset).

Función:
S = 1: Establece el estado a 1.
R = 1: Establece el estado a 0.
S = R = 1: Estado indeterminado.
***Aplicaciones**: Sincronización de señales, almacenamiento temporal de datos.*

Entrada: D (Data).

Función:
D = 1: Establece el estado a 1 en el siguiente flanco de subida del reloj.
D = 0: Establece el estado a 0 en el siguiente flanco de subida del reloj.
Aplicaciones: Registros de desplazamiento, contadores.

Flip-flop JK (J-K):

Entradas: J y K.
Función:
J = 1, K = 0: Establece el estado a 1 en el siguiente flanco de subida del reloj.
J = 0, K = 1: Establece el estado a 0 en el siguiente flanco de subida del reloj.
J = 1, K = 1: Conmuta el estado (1 a 0 o 0 a 1) en el siguiente flanco de subida del reloj.
J = K = 0: No cambia el estado.
***Aplicaciones:** Registros de desplazamiento, contadores, divisores de frecuencia*.

Flip-flop T (Toggle):

Entrada: T (Toggle).
Función:
T = 1: Conmuta el estado (1 a 0 o 0 a 1) en el siguiente flanco de subida del reloj.
T = 0: No cambia el estado.
Aplicaciones: Contadores asíncronos, divisores de frecuencia.

Aplicaciones de los flip-flops:

Los flip-flops son componentes esenciales en la construcción de circuitos secuenciales, como:
Máquinas de estado finito: Almacenan el estado actual de la máquina y determinan el siguiente estado en función de las entradas y el estado actual.
Contadores: Cuentan eventos o pulsos de entrada.
Registros: Almacenan datos binarios en microprocesadores y otros sistemas digitales.
Divisores de frecuencia: Dividen la frecuencia de una señal de reloj.