

Laboratorio de sistemas electrónicos digitales

(modalidad presencial)

Teoría y Práctica

Autores: Juan Angel Garza Garza, Dra. Norma Patricia Puente Ramírez e M. C. Jesús Daniel Garza Camarena.

Primera edición, febrero 2023

©Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Pedro de Alba S/N

Cd. Universitaria

San Nicolás de los Garza N.L.

México Cp 66450

<http://WWW.FIME.UANL.MX>

Tel. (0181) 83294020 Ext. 5921

Email: jagarza@uanl.mx

<http://jagarza.fime.uanl.mx/>

Impreso en FIME UANL México

ISBN: En trámite

Contenido

Sesión 5 HDL y Minimización de funciones booleanas	3
Objetivos particulares	3
Elementos de competencia	3
Fundamento Teórico.....	3
Operadores Lógicos en ABEL-HDL	4
Material a utilizar.....	5
Actividad de aprendizaje.....	6
Reporte	19
Reporte (lista de Cotejo, Check List)	19

Sesión 5

HDL y Minimización de funciones booleanas

Objetivos particulares

Durante el desarrollo de esta sesión el estudiante obtendrá las ecuaciones mínimas en las formas SOP (Suma de Productos And/Or) y POS (Productos de Suma Or/And) de dos funciones Booleanas, la primera F1 de la forma SOP y la segunda F2 en forma POS, para reducir así la complejidad del circuito a implementar.

Elementos de competencia

Construir circuito digital correspondiente a dos funciones booleanas de forma SOP y POS, en un Dispositivo Lógico Programable (PLD), utilizando el Lenguaje de Descripción de Hardware (HDL), ABEL (Advanced Boolean Expression Language), simulando su funcionamiento por medio del Test_Vectors, usando el programa IspLever, para que funcione en base a las funciones Booleanas.

Comprobar el funcionamiento del circuito construido, utilizando los resultados obtenidos en el archivo reporte del IspLever (las ecuaciones mínimas), por medio de Mapas de Karnaugh en las formas: SOP (agrupando unos) y POS (agrupando ceros) y utilizando el software LogicAid, para garantizar su correspondencia con las funciones Booleanas.

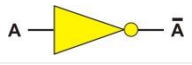






Aplicar el Teorema de D' Morgan para obtener las ocho formas estandar.

Fundamento Teórico

Por medio del Lenguaje de Descripción de Hardware (HDL) es posible fabricar un circuito integrado a la medida (ASIC), utilizando código para proporcionar las ecuaciones o la tabla de verdad en el lenguaje ABEL- HDL que fue desarrollado por Data I/O Corporation para la construcción física de circuitos correspondientes a las funciones booleanas en dispositivos lógicos programables (PLD).

ABEL-HDL necesita un procesador de lenguaje llamado compilador (IspLEVER), cuyo trabajo consiste en traducir el archivo de texto de ABEL a un mapa de fusibles (JEDEC) del dispositivo físico seleccionado, pasando por un proceso de validación de las instrucciones, así como de minimización de las funciones para ajustar, si es posible, la capacidad del dispositivo elegido.

Operadores Lógicos en ABEL-HDL

Descripción	Símbolo	Operador	Operador en Abel	
Not		A'	!	!A
And		AB	&	A&B
Or		A+B	#	A#B
Exor		A⊕B	\$	A\$B
Nand		(AB)'	!&	!(A&B)
Nor		(A+B)'	!#	!(A#B)
Exnor		(A⊕B)'	!\$!(A\$B)

Estructura del archivo ABEL-HDL Module

- 1 **Module** inicio del programa máximo 8 caracteres no números
- 2 " Comentarios: fecha, autor, propósito
- 3 **Declaration** a) asignación de terminales de entrada
- 4.- b) asignación de terminales de salida **istype 'com'** ;
- 4 **Descripción lógica** (ecuaciones, Tabla de verdad etc.)
- 5 **Test_vectors** (vectores de prueba)
- 6 **End** fin del programa

Nota: Para mayor información sobre el lenguaje ABEL, consultar los manuales disponibles en la página <http://jagarza.fime.uanl.mx/>.

Material a utilizar

Programas de aplicación (software):




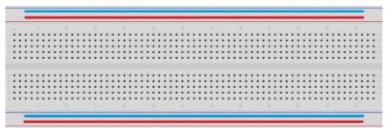

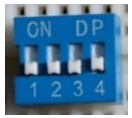




IspLEVER Classic o Projnav

Microsoft Word (reporte)

Proteus

ScreenToGif

Recortes (Windows XP o Windows 7)

Cant.	Descripción	imagen
1	Una fuente de alimentación de cinco volts	
6	Resistores de 330 Ω a 1/4 W	
6	Led's de 5 mm económico diferentes colores, ámbar, Azul, rojos y verdes.	
1	Tablilla de conexiones (Proto-Board) 1 Bloque 2 Tiras 830 puntos	
1	DIP Switch deslizable (4 o 8 interruptores deslizables) tipo TTL	 
4	Switch Push Micro NO (interruptor de no retención normalmente abierto)	
1	Cables diferentes tamaños ya listos para alambrear	
1	Dispositivo Logico Programable (PLD) 22V10	
		
		Microchip
		Lattice

Actividad de aprendizaje.

Trabajo solicitado

- 1.- Obtener la tabla de verdad de las ecuaciones que le fueron asignadas en la página <http://jagarza.fime.uanl.mx>, en la sección de laboratorio, sesión 5.
- 2.- Elabore el archivo en formato ABEL-HDL con las ecuaciones proporcionadas.
En un solo circuito integrado (PLD) construya el prototipo con las ecuaciones que le fueron asignadas en la página <http://jagarza.fime.uanl.mx>, en la sección de laboratorio, sesión 5, por medio del lenguaje ABEL-HDL, usando el comando **equations**, anexar en el mismo archivo la simulación (test_vectors).
- 3.- Con los valores de la tabla de verdad utilizar los mapas de Karnaugh para obtener las ecuaciones mínimas de las funciones asignadas, en la forma SOP (agrupando unos) y POS (agrupando ceros),
- 4.- Compruebe los resultados obtenidos en el Mapa de Karnaugh por medio del software LogicAid.
- 5.- Verificar los resultados anteriores con las ecuaciones mínimas obtenidas en el archivo reporte del IspLever.
- 6.- Si es necesario aplicar el teorema de D´Morgan para obtener la forma And/Nor.

Procedimiento:

- 1.- Partiendo de las ecuaciones proporcionadas obtenga la tabla de verdad.
- 2.- Cree una nueva carpeta en su computadora con el propósito de guardar ahí todos los archivos del proyecto.
- 2.- Con el programa IspLEVER, cree un nuevo proyecto en la opción de *File* y asigne un nombre, *Projet Name* (Recuerde que el nombre no debe de exceder de 8 caracteres).
- 3.- Sitúe la carpeta de este proyecto (*Location:*)
- 4.- Elija el tipo de diseño (*Design Entry Type*) *Schematic/ABEL*.
- 5.- Seleccione su dispositivo (*Select Device*) GAL 16, 18, 20, 22 o 26 con el tipo de empaque (*Package Type*) DIP (aquí no importa la marca, solo el tamaño y el tipo de empaque).
- 6.- En *Source* cree una nueva fuente (*New*) con la opción de *ABEL-HDL Module*.
- 7.- Asigne un nombre al archivo, máximo 8 caracteres de preferencia no números, por ejemplo Pcinco, la extensión por defecto de este archivo será ABL.
- 8.- Transforme sus ecuaciones F1 y F2 en el formato de lenguaje ABEL-HDL

Ejemplo: $F1_{(A,B,C,D)} = A B' + A B D' + A B' D + A' B' C' D$ (SOP, And/Or)

En ABEL-HDL: $F1 = A \& !B \# A \& B \& !D \# A \& !B \& D \# !A \& !B \& !C \& D;$

$F2_{(A, B, C, D)} = (B' + C + D)(B' + C' + D)(A' + B' + C' + D')(A' + B + C + D')$ (POS, Or/And)

En ABEL-HDL: $F2 = (!B \# C \# D) \& (!B \# !C \# D) \& (!A \# !B \# !C \# !D) \& (!A \# B \# C \# !D);$

9.- Capture el código con el formato como se muestra en la página siguiente con las ecuaciones que le fueron asignadas.

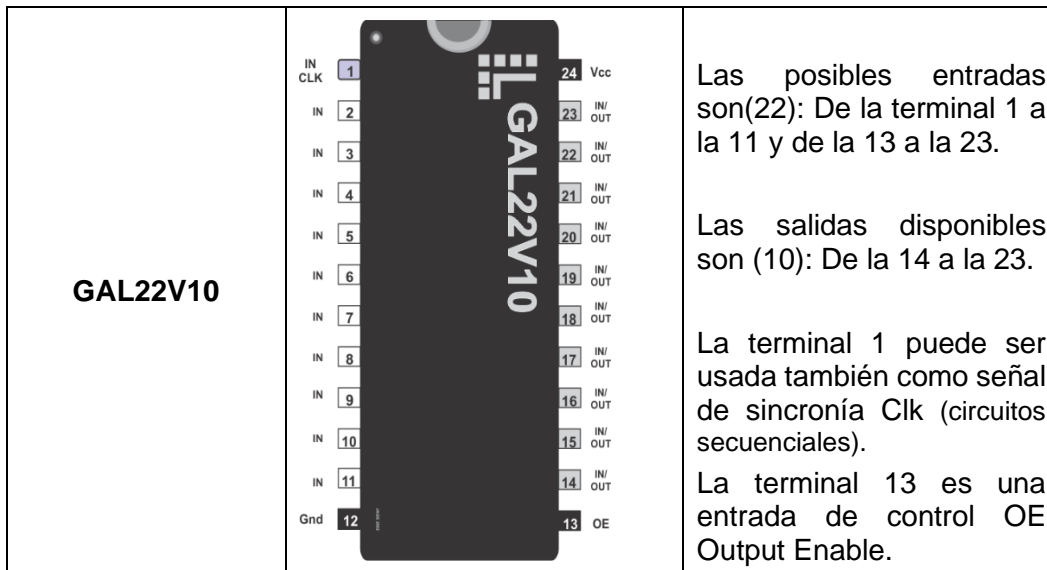
Código del archivo ABEL-HDL module

```
MODULE Pcinco
"Entradas
A, B, C, D PIN 1..4;
"Salidas
F1, F2 PIN 14, 15 ISTYPE 'COM';
EQUATIONS
"F1(A,B,C,D)=A B' +A B D' + A B' D + A' B'C'D
F1=A&!B # A&B&!D # A&!B&D # !A&!B&!C&D;
"F2(A,B,C,D)=(B'+C'+D)(B'+C'+D)(A'+B'+C'+D')(A'+B'+C'+D')
F2=(!B#C#D)&(!B#!C#D)&!A#!B#!C#!D)&(!A#B#C#!D);

TEST_VECTORS
([A,B,C,D]->[F1,F2])
[0,0,0,0]->[.X.,.X.];
[0,0,0,1]->[.X.,.X.];
[0,0,1,0]->[.X.,.X.];
[0,0,1,1]->[.X.,.X.];
[0,1,0,0]->[.X.,.X.];
[0,1,0,1]->[.X.,.X.];
[0,1,1,0]->[.X.,.X.];
[0,1,1,1]->[.X.,.X.];
[1,0,0,0]->[.X.,.X.];
[1,0,0,1]->[.X.,.X.];
[1,0,1,0]->[.X.,.X.];
[1,0,1,1]->[.X.,.X.];
[1,1,0,0]->[.X.,.X.];
[1,1,0,1]->[.X.,.X.];
[1,1,1,0]->[.X.,.X.];
[1,1,1,1]->[.X.,.X.];
END
```

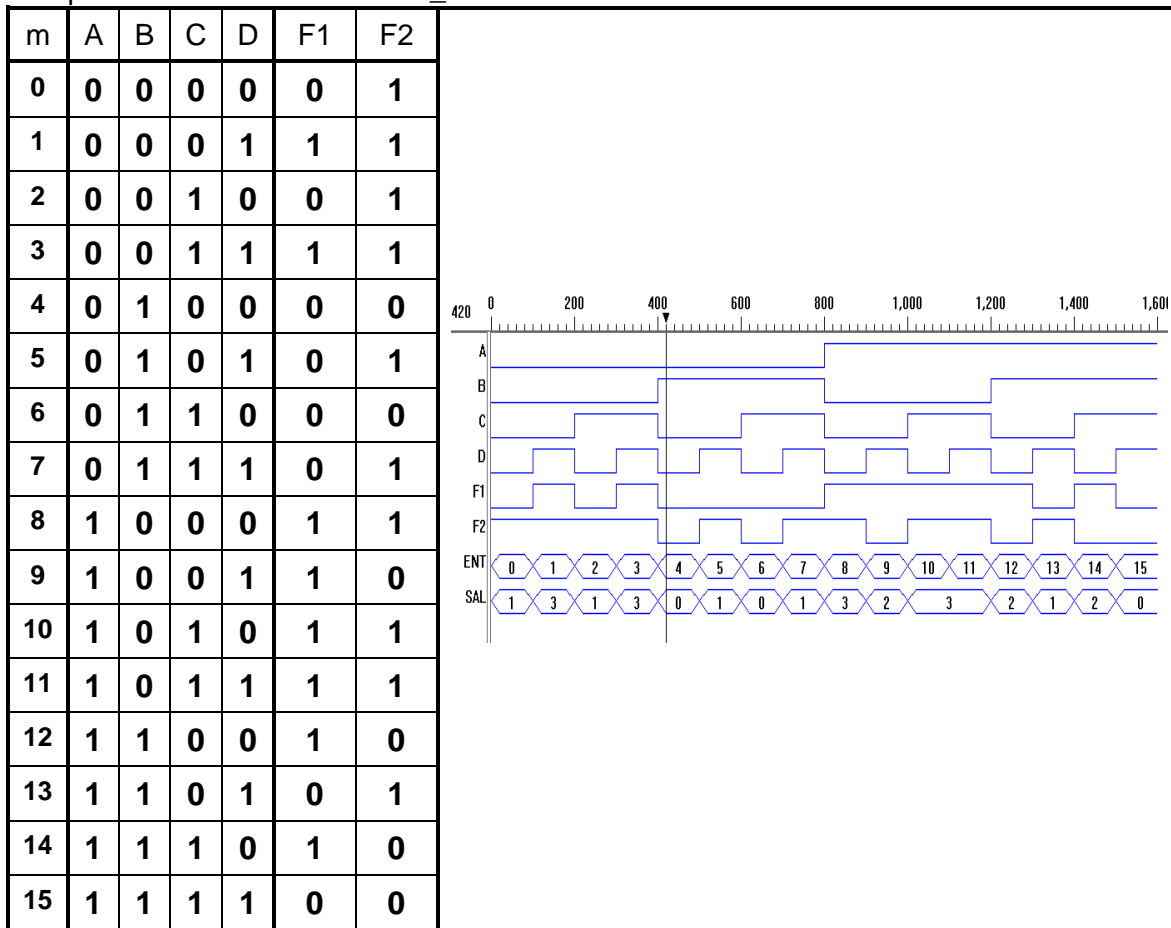

En el código en ABEL-HDL Module se pueden asignar convenientemente las terminales de entradas y salidas siempre y cuando estén disponibles para ese propósito, en la siguiente figura se muestran las terminales disponibles para entrada y salida para GAL22V10.

Distribución de terminales (pin out)

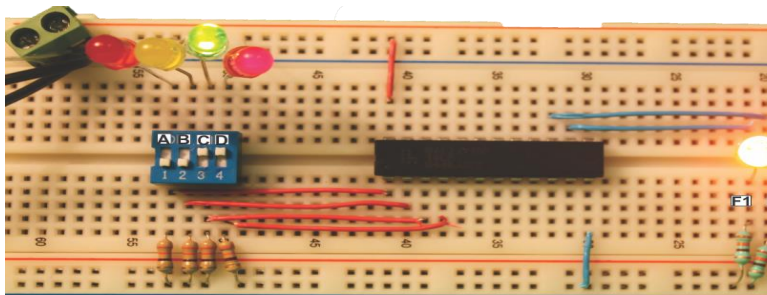


Para cualquier aclaración sobre la distribución de terminales de su dispositivo consulte el manual del fabricante.

Compare los resultados del Abel_test vector con la tabla de verdad:



Considerando la distribución de terminales (Pin Out) del archivo reporte, implementar físicamente el circuito como se muestra en la siguiente figura y comprobar todas las combinaciones coincide con las obtenidas en la tabla de verdad.



A	1	24	Vc
B	2	23	
C	3	22	
D	4	21	
	5	20	
	6	19	
	7	18	
	8	17	
	9	16	
	10	15	VE
	11	14	F1
GND	12	13	

Simulación en Proteus

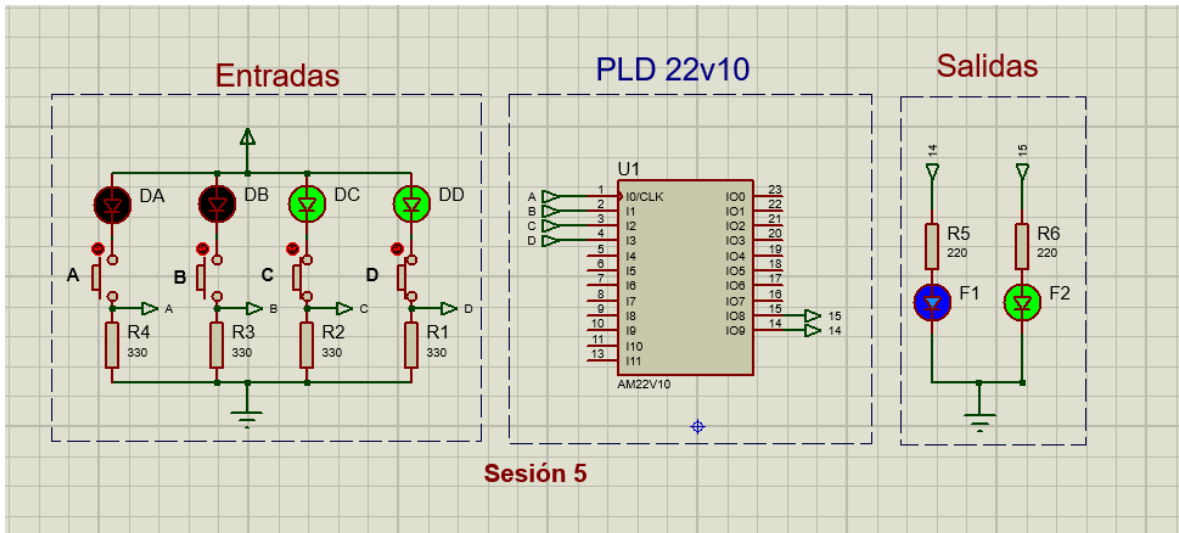


Diagrama elaborado en Proteus para efectuar la simulación y comprobar los resultados obtenidos

Comprobación de las ecuaciones mínimas

En el archivo reporte generado en la compilación, en la página 2 se muestran los resultados de las ecuaciones implementadas por el IspLEVER.

$$F1 = (!B \& D \# A \& !D);$$

$$F2 = !(B \& !D \# A \& B \& C \# A \& !B \& D \& !C);$$

Muy probablemente estos resultados son diferentes a las propuestas en el archivo ABEL-HDL MODULE, ya que el programa realiza un proceso de minimización para optimizar el uso del dispositivo.

Comprobaremos estos resultados con el uso del mapa de Karnaugh.

Reglas para el uso del mapa de Karnaugh (Kmap).

- 1.- Formar el menor número de grupos.
- 2.- Cada grupo lo más grande posible.
- 3.- Todos los unos o ceros deberán de ser agrupados.
- 4.- Un solo uno o cero puede formar un grupo.
- 5.- Casillas de un grupo pueden formar parte de otro grupo.

Grupo = Unos adyacentes enlazados (paralelogramos) en una cantidad igual a una potencia entera de dos, ejemplo (1, 2, 4, 8,16, etc.).

Con los valores obtenidos en la tabla de verdad, llenaremos con los valores de salida F1 y F2 en la localidad correspondiente para cada uno de los mapas.

Mapas de Karnaugh (Kmap) para F1 y F2

		AB			
		00	01	11	10
CD	00	0 0	4 0	12 1	8 1
	01	1 1	5 0	13 0	9 1
	11	3 1	7 0	15 0	11 1
	10	2 0	6 0	14 1	10 1

		AB			
		00	01	11	10
CD	00	0 1	4 0	12 0	8 1
	01	1 1	5 1	13 1	9 0
	11	3 1	7 1	15 0	11 1
	10	2 1	6 0	14 0	10 1

Solución:

Ecuaciones mínimas para F1 y F2

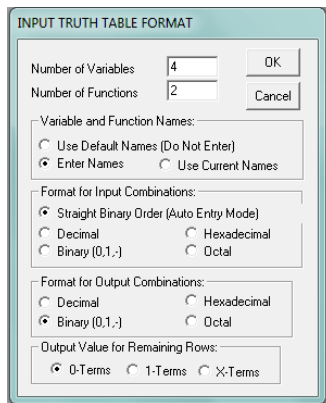
Agrupando unos		Agrupando ceros	
$F1 = A D' + B' D$	$F1 = (B' + D') (A + D)$		

Agrupando unos		Agrupando ceros	
$A'D + B'D' + B'C + B C'D$	$(B' + D) (A' + B + C + D') (A' + B' + C')$		

Ecuaciones mínimas en LogicAid

Utilizando tabla de verdad obtenga las ecuaciones mínimas en el software LogicAid y compárelos con los resultados obtenidos en el Mapa de karnaugh.

1. Con el software LogicAid seleccione en el menú **Input** la opción de **Truth Table** (o presione CTRL+T).
2. Para este ejercicio seleccione 4 variables, 2 funciones y proporcione los nombres como lo muestra la siguiente figura.

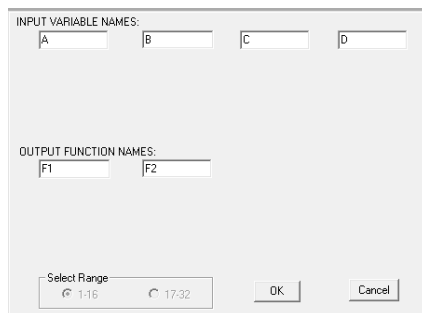


Number of Variables **4**

Number of Functions **2**

Seleccionar la opción de **Enter Names**

3. Asigne los nombres de las variables de entrada y salida correspondientes a su actividad.



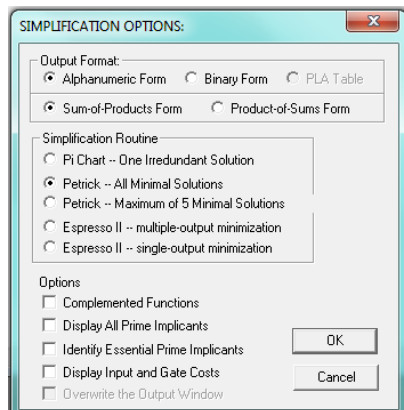
Aquí se asignaran los nombres de las variables de su ejercicio tomando en cuenta que una variable en mayúscula es diferente a minúscula así como las salidas F1 y F2

4. Capture los valores de la tabla de verdad

Input Variable Name	Output Function Name
0000	01
0001	11
0010	01
0011	11
0100	00
0101	01
0110	00
0111	01
1000	11
1001	10
1010	11
1011	11
1100	10
1101	01
1110	10
1111	00

En esta parte proporcionará solo los valores de salida F1 y F2 ya que los de entrada se proporcionan en forma automática.

5. En el menú de ROUTINE seleccione la opción de SIMPLIFICATION, obtenga todas las posibles soluciones mínimas por el método de Petrick en la forma suma de productos



En esta parte se selecciona el formato de salida de los resultados.

En forma alfabética

Suma de Productos

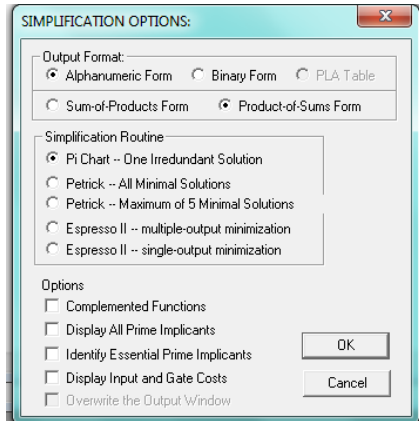
Todas las posibles soluciones por el método de Petrick.

Una vez proporcionada la información de salida presione OK, y aparecerá una nueva ventana indicando los resultados en Suma de Productos.

$$F1 = B'D + A D'$$

$$F2 = A'D + B'D' + B'C + B C'D$$

Regrese a la ventana de la tabla de verdad y de nuevo seleccione la opción de SIMPLIFICATION, obtenga todos los posibles soluciones mínimas por el método de Petrick en la forma pero ahora la opción productos de suma.



En esta parte proporcionara el formato de salida de los resultados.

En forma alfabética

Productos de Suma

Todas las posibles soluciones por el método de Petrick.

Resultados en Productos de Suma.

$$F1 = (A + D) (B' + D')$$

$$F2 = (B' + D) (A' + B + C + D') (A' + B' + C')$$

Comparación de resultados de las ecuaciones mínimas:

	Archivo Reporte	Mapas de Karnaugh	LogicAid
F1	(!B & D # A & !D)	A D' + B' D	B' D + A D'

Para el caso de F1 los tres resultados coinciden.

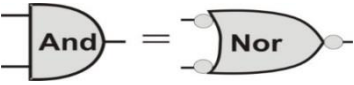
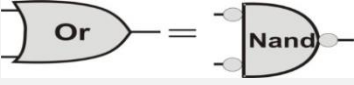
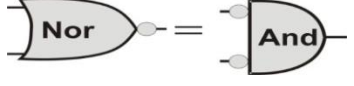
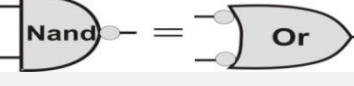
La F2 del Archivo Reporte está en la forma And/Nor

$F2 = !(B \& !D \# A \& B \& C \# A \& !B \& D \& !C)$; And/Nor

Los resultados del mapa de Karnaugh y del LogicAid está en forma Or/and

$F2 = (B' + D) (A' + B + C + D') (A' + B' + C')$ Or/And

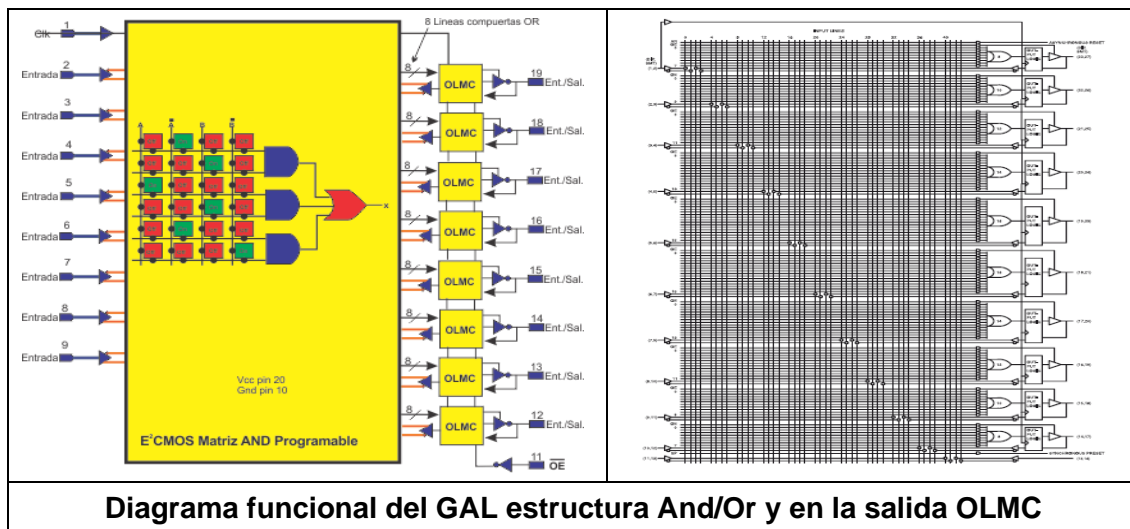
Para hacer coincidir estos resultados es necesario aplicar el teorema de D'Morgan

Teorema de D Morgan	
$AB = (A' + B')'$	
Reemplazar los operadores And por Nor negando sus entradas	
$A+B = (A' B')'$	
Reemplazar los operadores Or por Nand negando sus entradas	
$(A+B)' = A'B'$	
Reemplazar los operadores Nor por And negando sus entradas	
$(A B)' = A' + B'$	
Reemplazar los operadores Nand por Or negando sus entradas	

Por medio de este teorema podemos obtener las ocho formas estándar en la que se expresan las funciones booleanas.

Minitérminos (SOP)	Maxitérminos (POS)
And/Or	Or/And
Nand/Nand	Nor/Nor
Or/Nand	And/Nor
Nor/Or	Nand/And

En el GAL por su estructura interna que es fija, solo soporta las formas And/Or y And/Nor.



Para nuestro propósito de obtener la forma **And/Nor** debemos de partir de la forma **Or/And**

$$F2 = (B' + D) (A' + B + C + D') (A' + B' + C') \text{ And/Or}$$

Como primer paso aplicaremos el Teorema de D' Morgan solo a la And la reemplazaremos por Nor negando las entradas

$$F2 = [(B' + D)' + (A' + B + C + D')' + (A' + B' + C')'] \text{ Nor/Nor}$$

Ahora reemplazaremos las compuertas Nor de los paréntesis redondos por And con entradas negadas.

$$F2 = [B D' + A B' C' D + A B C] \text{ forma And/Nor}$$

$$F2 = !(B \& !D \# A \& B \& C \# A \& !B \& D \& !C); \text{ forma And/Nor}$$

Comparando los resultados concluimos que el programa IspStarter utiliza la forma And/Nor en este caso para expresar la función F2.

Reporte

Los reportes del Laboratorio deberán de contener la siguiente información.

Portada:

Reporte (lista de Cotejo, Check List)

1	Portada.
2	Ecuaciones asignadas.
3	Código en formato ABEL-HDL Module (incluyendo el Test_vectors).
4	Imagen de la simulación.
5	Tabla de verdad
6	Imagen de la Distribución de terminales (Pin Out)
7	Foto del circuito implementado
8	Ecuaciones mínimas del archivo reporte
9	Imagen del Circuito en Protesus
10	Mapas de Karnaugh y sus resultados.
11	Archivos de LogicAid y sus resultados.
12	Comparación de los resultados (Reporte, KMap y LogicAid)
13	Obtención de la forma And/Nor por el teorema de D' Morgan.
14	Conclusiones
15	Recomendaciones
16	Subir a Google Classroom los archivos entregables solicitados

En la página <http://jagarza.fime.uanl.mx/> podrás encontrar el video de esta sesión, así como manuales de uso del lenguaje ABEL.HDL, así como el software LogicAid

Archivos entregables todos incluidos en un **Zip o RAR** de nombre, sesión, hora y numero de lista ejemplo S5MM1NL3.zip martes M1 numero de lista 3

Reporte	Captura Esquemática	Abel HDL Module	JEDEC	LogicAid		Animación	PROTEUS
PDF	SCH	ABL	JED	Aid	Out	GIF	PDSRJR

Una vez cumplido lo anterior es necesario agendar y efectuar la entrevista presencial para presentar el prototipo funcionando correctamente, así como explicar los procedimientos y resultados obtenidos en forma oral y escrita.