

**UNIVERSIDAD DE GRANADA.**

**ESCUELA TECNICA SUPERIOR DE  
INGENIERIAS INFORMATICA Y DE  
TELECOMUNICACIÓN.**



**Departamento de Arquitectura y  
Tecnología de Computadores.**

**TECNOLOGÍA Y ORGANIZACIÓN DE  
COMPUTADORES DE COMPUTADORES.**

**SEMINARIO 5.  
INTODUCCIÓN AL MANEJO DE UN SIMULADOR  
LÓGICO Y DE UN ENTRENADOR LÓGICO**

**1º GRADO EN INGENIERÍA INFORMÁTICA.**



## SEMINARIO 5.

### PARTE 1: INTRODUCCIÓN AL MANEJO DE UN SIMULADOR LÓGICO.

### GUÍA DE TRABAJO AUTÓNOMO.

#### 1. Funcionamiento de Logic Works.

Logic Works es un entorno de trabajo que permite diseñar y simular tanto a nivel funcional como temporal circuitos lógicos digitales. El profesor le indicará dónde se encuentra instalado dicho programa. Para poder utilizar este entorno, se debe ejecutar el programa *LogicWorks.exe* dentro del directorio *Program*. Al ejecutar dicho programa, el interfaz de trabajo que se encuentra el usuario se muestra en la Figura S3.1. Dicho interfaz de trabajo se encuentra dividido en varias partes:

- La ventana principal es la **“ventana de diseño”**. En ella se introduce el circuito digital que se quiere diseñar y simular. Junto a ésta, pueden aparecer otras dos pequeñas ventanas que contienen diferentes informaciones. Con la opción de menú “View” se puede controlar que aparezcan o no dichas ventanas. Con la opción de menú "Schematic → Design Preferences" se puede controlar el aspecto de esta ventana principal de diseño.
- La ventana de la derecha es la **biblioteca de componentes (Parts Palette)**. En ella se pueden seleccionar los distintos componentes que se van a utilizar en el diseño. Estos componentes están ordenados según su funcionalidad en distintas bibliotecas. Las bibliotecas de Logic Works más importantes para esta asignatura, entre otras, son:

Biblioteca	Contenido
<i>Simulation Gates</i>	Puertas lógicas: AND, OR, NOT, XOR, etc. con diverso número de entradas
<i>Simulation Logic</i>	Circuitos digitales sencillos: Latches, biestables, registros, contadores, sumadores, etc.
<i>Connect</i>	Puertos de entrada, salida y entrada/salida necesarios para la definición de subcircuitos en Logic Works.
<i>Simulation IO</i>	Dispositivos de entrada/salida para la simulación lógica del circuito (conmutadores de entrada tanto binarios como hexadecimales y displays).

- La ventana inferior es la **ventana de visualización de formas de onda (Timing Window)**, que se utiliza para comprobar el funcionamiento temporal del circuito que se esté diseñando. La descripción funcional de esta ventana se verá en el apartado 4 de este Seminario.

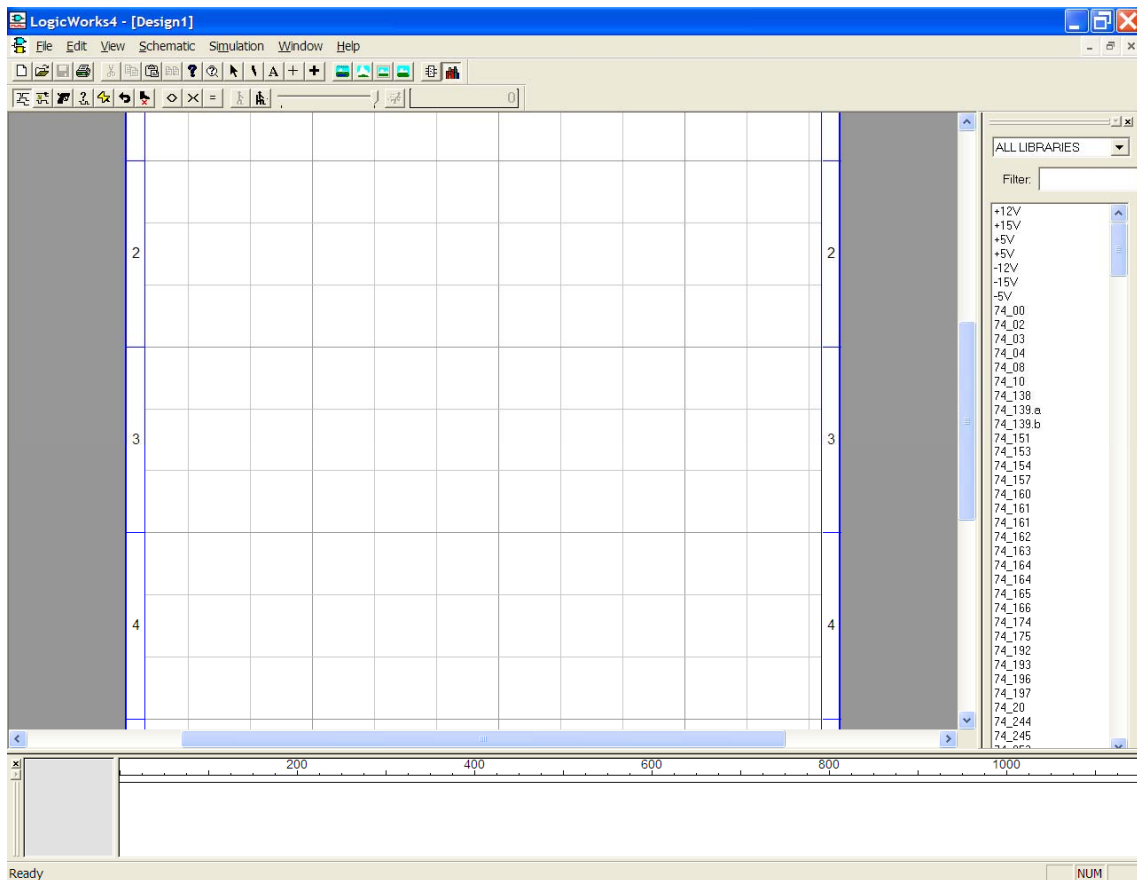


Figura 1: Aspecto inicial de Logic Works.

## 2. Selección de componentes.

Suponga que se desea simular un circuito digital combinacional muy sencillo con 2 entradas  $A_0$  y  $B_0$  que produce 2 salidas  $S_0$  y  $C_1$  cuyas tablas de verdad está indicadas en la Tabla 1.

Tabla 1

$A_0$	$B_0$	$S_0$	$C_1$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Funciones de Conmutación asociadas

$$S_0 = A_0 \oplus B_0$$

$$C_1 = A_0 \cdot B_0$$

Para implementar este circuito en Logic Works se deben seguir los siguientes pasos:

- Seleccione la biblioteca de componentes la biblioteca “Simulation Gates” y haga doble

clic con el ratón sobre “AND-2”. Cuando se desplaza el cursor del ratón hacia la ventana de diseño aparecerá una puerta AND de dos entradas. Para posicionar la puerta sobre el esquemático basta hacer un clic simple con el botón izquierdo del ratón (si se equivoca de puerta bastará con pulsar el botón derecho del ratón o la tecla ESC). Realice la misma tarea con una puerta XOR de dos entradas (ver Figura 2). Cuando se selecciona un componente de una biblioteca se puede cambiar la orientación del componente usando los cursores del teclado mientras que dicho componente no haya sido posicionado definitivamente en la ventana de diseño.

- A continuación, seleccione de la biblioteca de componentes la biblioteca “*Simulation IO*” y haga doble clic sobre “BINARY SWITCH” (conmutador binario). Inserte dos elementos de este componente en el esquemático en posiciones similares a las que se muestran en la Figura 2.

- Igualmente, inserte dos elementos “BINARY PROBE” (sonda ó visualizador binario) de la biblioteca “*Simulation IO*”. Para modificar la posición de cada uno de estos componentes basta seleccionarlo con el ratón y arrastrarlo a la posición deseada. En el caso de los conmutadores binarios, un clic del ratón cambia el estado lógico de la salida del conmutador por lo que para desplazar estos componentes es necesario tener pulsada la tecla “Shift” (mayúsculas) cuando se pulsa el botón del ratón.

### 3. Conexionado entre componentes.

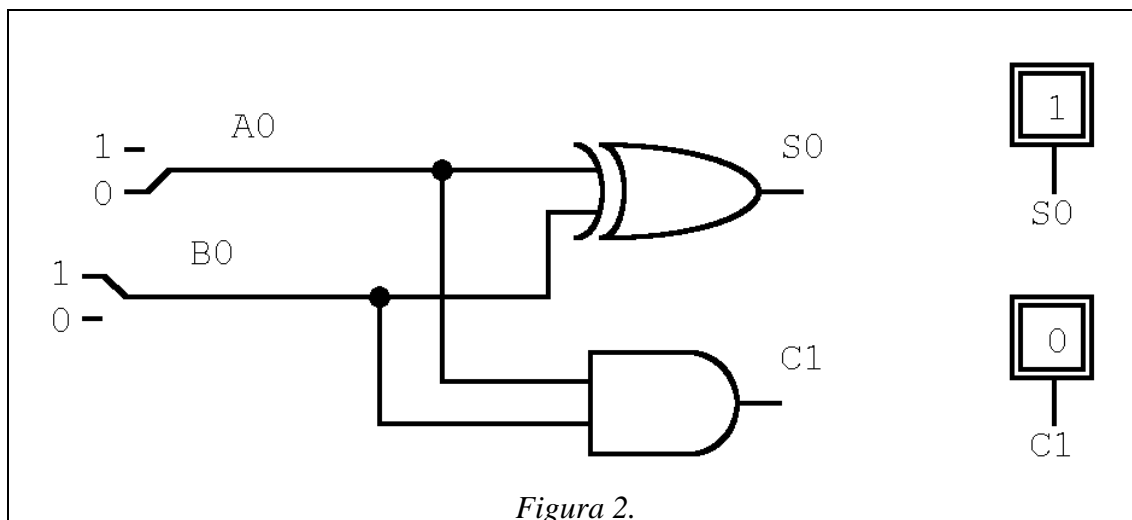
Existen dos formas de realizar el conexionado entre los distintos componentes existentes en la ventana de diseño:

a) La manera más sencilla de realizar el conexionado entre componentes consiste en situar el cursor del ratón cerca del punto final del punto desde donde se quiera que salga la conexión, hacer un clic y arrastrar el ratón hasta la localización final. De esta forma se simula una conexión eléctrica entre el punto de origen y el de final.

b) Una segunda manera de establecer conexiones entre componentes consiste en etiquetar los pines de cada uno de ellos, de tal forma que **dos pines con la misma etiqueta se consideran eléctricamente conectados**. Para insertar una etiqueta en un pin se selecciona la opción “Edit → Text” en el menú (el puntero cambia de forma). Ahora basta seleccionar la parte final (exterior) del pin correspondiente y aparecerá un cuadro de texto donde se puede introducir el nombre que se desea poner al pin. Si todo se ha realizado correctamente, se debe observar que la nueva etiqueta aparece en color rosado y que se añade su nombre en la ventana inferior de visualización de formas de onda, siempre y cuando la opción del menú “Simulation → Add Automatically” esté seleccionada.

Se realizará de la primera forma la conexión entre los conmutadores binarios y las entradas de las puertas lógicas del circuito (ver Figura 2) y de la segunda forma la conexión entre las salidas de las puertas lógicas y los visualizadores binarios. Para ello, se procede a etiquetar con “S0” la salida de la puerta XOR y la entrada de uno de los visualizadores binarios. Igualmente, se usará la etiqueta “C1” para la salida de la puerta AND y la entrada del otro visualizador, tal y como se muestra en la Figura S3.2. Es importante notar que en el simulador, cada vez que se hace doble clic sobre una señal

(cable) se colorean de amarillo todos los conectores que estén eléctricamente conectados a esa señal. De esta forma, se puede asegurar que la salida de cada puerta lógica está realmente conectada con la entrada de su visualizador binario correspondiente. Utilizando la opción *File-Save (Ctrl-S)* guarde el diseño con el nombre *circuito1.cct*.



## 4. Simulación del diseño.

En Logic Works se puede simular un diseño de dos formas distintas.

- La primera de ellas se basa en la utilización de conmutadores binarios o hexadecimales para especificar las entradas lógicas (“BINARY SWITCH” y “HEX KEYBOARD wo/STB<sup>1</sup>” de la biblioteca “*Simulation IO*”) y de visualizadores binarios o hexadecimales para medir las salidas (“BINARY PROBE” y “HEX DISPLAY”). Una vez que se ha definido el circuito digital en la ventana de diseño, se puede realizar una simulación lógica del mismo simplemente haciendo un clic en los conmutadores de entrada y observar los valores de salida en los visualizadores. Compruebe ahora el funcionamiento del circuito verificando que se cumple su tabla de verdad según la Tabla 1.

- La segunda forma se realiza mediante una herramienta de simulación en Logic Works, más potente, que permite la representación gráfica de las formas de onda (cronogramas) de las señales especificadas en el circuito. Para realizar esta segunda forma de simulación hay que seguir una serie de pasos:

### a) Definición temporal de estímulos:

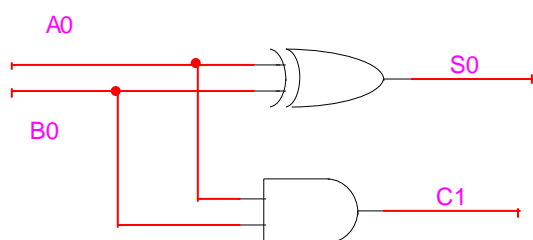
En el apartado 2 de este texto se vio cómo se podían introducir las entradas al circuito a través de conmutadores binarios y hexadecimales. Logic Works también proporciona una herramienta para realizar esta función a través de un fichero de estímulos (con extensión .tim). El formato de dicho fichero consta de una serie de columnas limitadas

<sup>1</sup> HEXADECIMAL KEYBOARD WITHOUT STROBE = Teclado hexadecimal sin señal de habilitación de salida.

por tabuladores (y no por espacios) donde se especifica el instante inicial de tiempo (columna etiquetada con \$T), la duración del periodo de tiempo (\$D) y el valor de cada una de las señales de entrada en ese periodo de tiempo (\$I Nombre\_señal\_entrada).

Por ejemplo, cree el fichero “Seminario5.tim” que se usará para comprobar el comportamiento del circuito de este seminario con los valores que se presentan en la Tabla 2, donde se especifican dos señales de entrada A0 y B0, cuyos valores se fijan en intervalos espaciados cada 20 unidades de tiempo. En forma gráfica, estos estímulos son los que se representan en la Figura 3. Para crear el fichero “Seminario5.tim” puede utilizar, bien un editor de textos convencional proporcionado por el sistema operativo, o bien puede utilizar el propio editor de textos de Logic Works a través del menú “File → New → Text Document”. En cualquier caso, siempre hay que tener precaución de archivar el fichero con la extensión .tim, no con la extensión .txt.

Para introducir el fichero de estímulos en la ventana de visualización de formas de onda, se debe asegurar que al menos las variables de entrada **estén ya presentes** en dicha ventana y de que sus nombres coincidan con los del fichero “.tim”. Se deben eliminar del diseño las entradas lógicas del tipo “BINARY SWITCH” para evitar errores en el simulador (ver Figura 3).



\$T	\$D	\$I A0	\$I B0
0	20	1	1
20	20	0	1
40	20	0	0
60	20	1	0
80	20	0	0
100	20	1	1
120	20	0	0

Tabla 2: Fichero de estímulos.

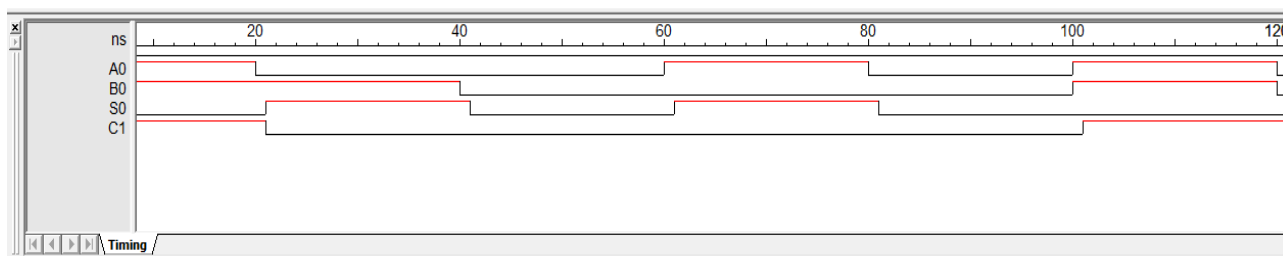


Figura 3: Formas de onda especificadas por el fichero de estímulos de la tabla 2.

### **b) Simulación temporal del circuito:**

Para empezar la simulación temporal del circuito, una vez insertados los estímulos de entrada, se debe acceder a la **barra de botones de control de la simulación** mediante la opción “View → Simulator Toolbar” del menú (ver Figura 1 y Figura 4). Los principales botones existentes en dicha ventana tienen el siguiente significado:



Figura 4: Barra de botones de control de la simulación.

- **“Reset”**: Borra todas las formas de onda de la ventana de análisis temporal para empezar una nueva simulación.
- **“Run”**: Ejecuta la simulación de forma continua.
- **“Step”**: Ejecuta la simulación paso a paso.
- : Modifican la escala de tiempos en la visualización.
- **“Clear X”**: Hace que el simulador intente dar unos valores iniciales a las distintas señales. En Logic Works, cada señal puede tener uno de los 5 valores de la Tabla 3. Inicialmente, los valores de las señales de salida es “X”, es decir, desconocido (0 ó 1).

Valor	Significado
0	Valor lógico 0
1	Valor lógico 1
X	Desconocido (0 ó 1)
Z	Alta impedancia (desconexión eléctrica)
C	Conflicto (hay dos valores de salida que provocan en la señal valores contradictorios)

Tabla 3: Valores para las señales que se utilizan en Logic Works.

Para realizar la simulación, pulse el botón de “Reset” e introduzca el fichero de estímulos “Seminario5.tim” seleccionando del menú la opción “Simulation → Import Timing”. Vaya pulsando sucesivamente el botón de “Step” para ejecutar paso a paso la simulación o bien el botón de “Run” para ejecutar la simulación toda entera. Se visualizarán en la ventana de tiempos los valores de las señales de los estímulos de entrada. Verifique ahora el funcionamiento del circuito contrastándolo con su tabla de verdad (Tabla 1). Ampliando convenientemente la escala de la visualización de la ventana de tiempos, podrá comprobar los retardos en las señales de salida frente a las de entrada. Estos retardos simulan los tiempos de retardo de transmisión de los cambios en las señales de salida frente a los de las entradas. Para variar dicho retardo basta con seleccionar un componente y, con el botón derecho del ratón, elegir la opción de “Attributes → Delay Dev.” y cambiar el valor por defecto que tiene el componente.

## 5. Generación de símbolos para subcircuitos.

En este apartado se muestra cómo se puede generar un símbolo para un circuito diseñado, de modo que se pueda utilizar como un bloque funcional, facilitándose así la implementación de circuitos más complejos. Para generar un símbolo con una funcionalidad determinada, hay que realizar los siguientes pasos:

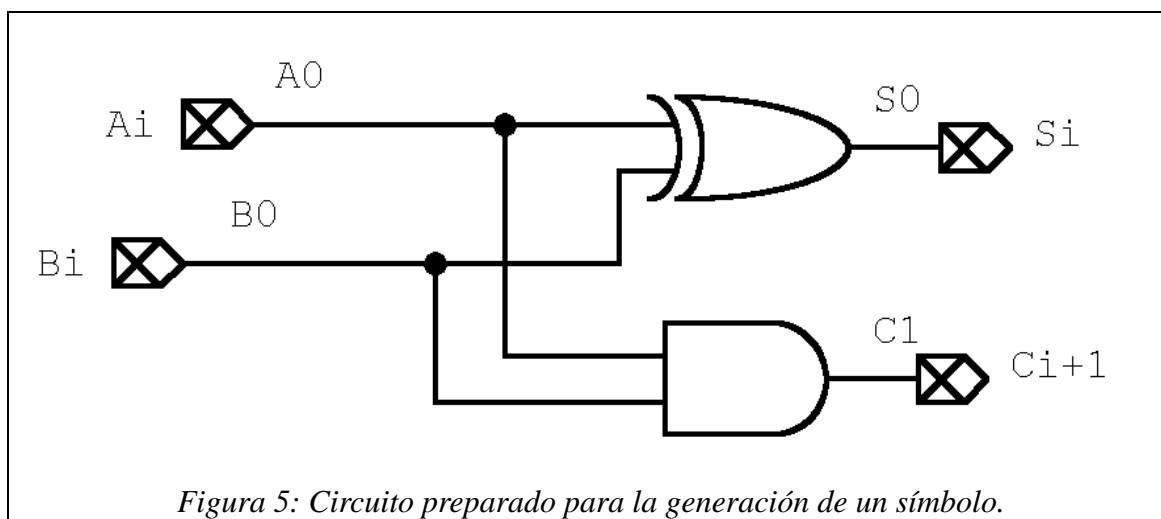
### Paso 1.- Definición de los puertos de E/S del subcircuito:

Cuando se crea un subcircuito, Logic Works debe saber quiénes deben ser las entradas y salidas de éste y qué nombre genérico se les debe dar. Para esto, Logic Works proporciona dos tipos de conectores: “PORT IN” y “PORT OUT” existentes en la biblioteca “Connect”.

Reemplace ahora los dos conmutadores de entrada binarios (BINARY SWITCH) del circuito de la Figura 2 por puertos de entrada (PORT IN) y los dos visualizadores



binarios (BINARY PROBE) por puertos de salida (PORT OUT) de la forma que se muestra en la Figura 5.



Finalmente, Logic Works debe saber qué nombres se les debe dar a las variables de entrada y de salida del subcircuito. Para ello, seleccione “Edit → Text” en el menú y pique con el ratón en el **centro** de cada uno de los puertos de E/S para etiquetarlos. Use las mismas etiquetas que en la Figura 5.

### **Paso 2.- Creación del símbolo para el subcircuito:**

Una vez definido completamente el subcircuito se debe generar un símbolo para él siguiendo los siguientes pasos:

- 1.- Guarde el circuito en un archivo (por ejemplo *circuito2.cct* en el directorio de trabajo) con nombre diferente al del diseño original.
- 2.- Seleccione en el menú desplegable la opción “File → New → Device Symbol”. En la nueva ventana que aparece (que es la del editor de dispositivos) seleccione “Options → Subcircuit / Part Type”. Haga clic ahora en “Create a subcircuit symbol and select an open circuit to attach to it” y elija para el subcircuito el fichero *circuito2.cct* creado. Haga clic en “Done”. En ese momento, Logic Works analiza el circuito y aparece en una de las ventanas del editor las señales de E/S que se han definido en el semisumador previamente.
- 3.- Para generar el símbolo de la forma más rápida posible, seleccione “Options → Autocreate symbol” y dentro de la nueva ventana, a su vez, seleccione “Extract Pin List”. Logic Works asigna por defecto las variables de entrada a la parte izquierda del nuevo componente y las de la salida a la parte derecha. Finalmente, en el campo “Part Name” coloque el nombre “CIRCUITO-2” al símbolo que albergará el *circuito2.cct*. Finalmente, haga clic en “Generate Symbol”.
- 4.- Para guardar el nuevo símbolo en una biblioteca, seleccione “File → Save” y aparecerá una ventana como la de la Figura 6.

Todas las bibliotecas en Logic Works son ficheros con extensión *.clf* que se encuentran en el directorio *\LogicWorks\Libs*. En este seminario se creará una nueva biblioteca llamada “*milib.clf*” que se usará para guardar todos los dispositivos que se vayan diseñando. Para crear dicha biblioteca seleccione en “New Lib” y se crea “MILIB.CLF” (biblioteca de componentes no existente anteriormente), almacenándola el estudiante en algún dispositivo donde tenga derecho de escritura. Posteriormente seleccione la biblioteca recién creada y pulse “Save” para guardar el nuevo componente “” en “*milib.clf*”.

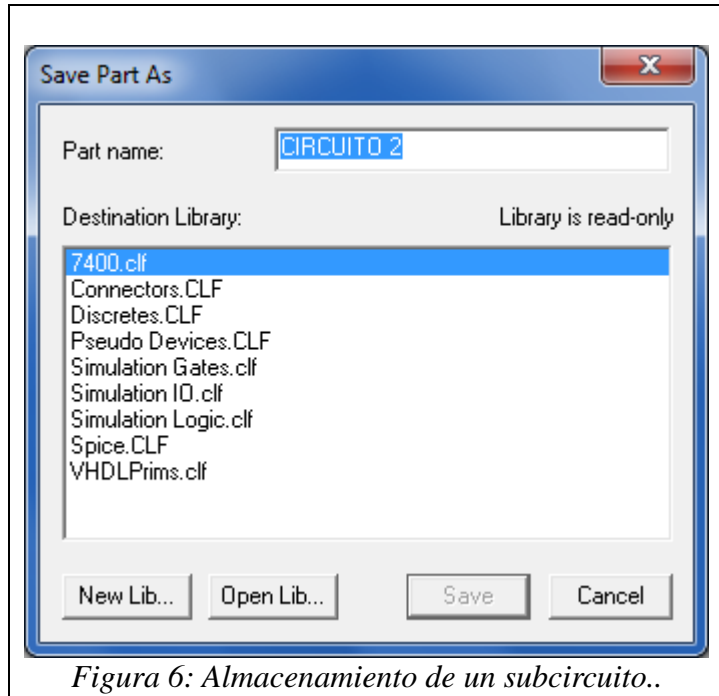


Figura 6: Almacenamiento de un subcircuito..

Para poder usar la biblioteca recién creada en próximas sesiones, basta con hacer clic con el botón derecho del ratón sobre la ventana de componentes y seleccionar la opción “Open Lib” para añadir esta biblioteca a la sesión actual.

### **Paso 3.- Comprobación del subcircuito generado:**

Finalmente, como comprobación, se va a diseñar un nuevo esquema que contendrá el símbolo correspondiente al circuito creado.

Para ello, cierre la ventana de diseño del CIRCUITO-2 y cree un nuevo diseño (“File → New Design”). Inserte ahora el nuevo componente en el esquemático (seleccionando la biblioteca MILIB) y compruebe que el funcionamiento del circuito usando conmutadores binarios en la entrada y visualizadores binarios en la salida (ver Figura 7) coincide con el expuesto en la tabla de verdad de la Tabla 1.

Finalmente, guarde el nuevo diseño con el nombre *circuito3.cct*.

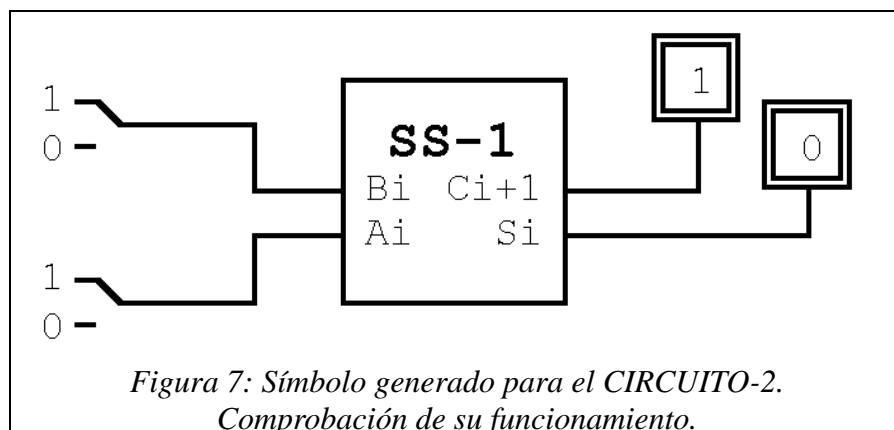


Figura 7: Símbolo generado para el CIRCUITO-2.  
Comprobación de su funcionamiento.

## SEMINARIO 5.

### PARTE 2: INTRODUCCIÓN AL MANEJO DE UN ENTRENADOR LÓGICO.

#### GUÍA DE TRABAJO AUTÓNOMO.

### 1. Descripción del entrenador de prácticas SIDAC DET 2020.

#### *1.1.- Generalidades:*

El entrenador lógico DET 2020 (Figura 8) permite la realización de prácticas sobre circuitos lógicos sin necesidad de utilizar instrumentación exterior. Dicho entrenador contiene las principales funciones lógicas empleadas en los circuitos electrónicos digitales. Estos circuitos están ya alimentados internamente y sus entradas y salidas son accesibles exteriormente para realizar las interconexiones necesarias para el desarrollo de las distintas prácticas.

Mediante un conjunto de cables dotados de conectores de 2 mm. de diámetro se realizan las conexiones entre los diferentes puntos del circuito que se desean interconectar así como con los indicadores de estado lógico, la señal de reloj, los generadores de pulsos o los conmutadores programables de nivel lógico (programadores) que se hallan en la parte inferior del entrenador (ver Figura 8).

En la parte superior del entrenador se han colocado cuatro zócalos de 16 patillas con el fin de poder utilizar, si se desea, cualquier circuito comercial que no esté contenido en el entrenador y de esta forma aumentar la gama de posibilidades del entrenador.

#### *1.2.- Fuente de alimentación:*

El entrenador lógico DET 2020 está alimentado mediante una fuente estabilizada protegida contra cortocircuitos. Mediante el interruptor (1) de la Figura 8 se pone en funcionamiento dicha fuente, se enciende el piloto (2) y se alimentan todos los circuitos del entrenador.

En la parte trasera del bastidor se hallan el fusible de entrada y el cambio de tensión de alimentación.

#### *1.3. - Indicadores de Estado Lógico:*

A través de los bornes de entrada (3) se accede a los indicadores estado lógico (4). Estos indicadores consisten en unos LEDs que se encienden cuando el nivel lógico del punto al que se conectan es alto (Hi) y permanecen apagados si el nivel es bajo (Lo) o están desconectados (Figura 8).

#### *1.4.- Reloj:*

Los pulsos de reloj (Clock) se obtienen a través de los bornes (5) pudiéndose utilizar simultáneamente tanto el pulso positivo como el complementario. La frecuencia de la señal de reloj se puede controlar mediante el selector (6) (para determinar el fondo de escala de la frecuencia) y el potenciómetro (7), con el que se podrá hacer un ajuste fino

de la frecuencia que se desea obtener.

*1.5.- Generadores de pulsos:*

A través de los bornes de entrada (8) se pueden obtener pulsos **sin rebote** cuando se accionan los pulsadores (9). Estos pulsadores generan simultáneamente un pulso positivo y su complementario. Existen dos circuitos generadores de pulsos independientes.

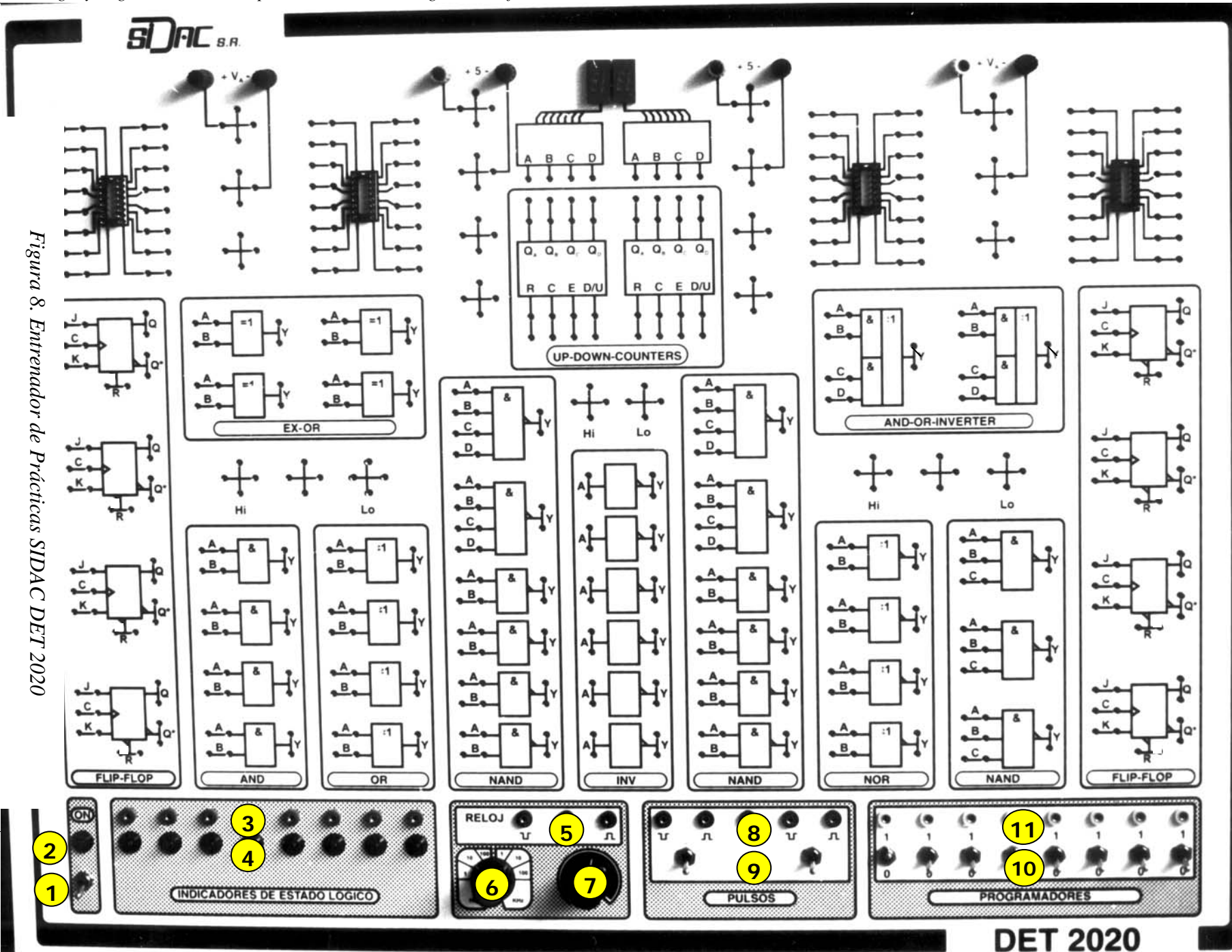
*1.6.- Conmutadores programables de nivel lógico:*

El entrenador dispone de conmutadores (10) con los que se pueden obtener hasta ocho salidas independientes a través de los respectivos bornes de salida (11) al nivel alto (Hi) o bajo (Lo) según su posición.

Resumen (ver Figura 8):

1. Interruptor general
2. LED de encendido
3. Bornes de entrada de los indicadores de estado lógico
4. Indicadores de estado lógico
5. Bornes de salida del reloj
6. Selector del rango de frecuencias de la señal de reloj
7. Potenciómetro para el ajuste fino de la frecuencia de reloj
8. Bornes de salida del generador de pulsos.
9. Pulsadores de accionamiento del generador de pulsos
10. Bornes de salida de los niveles programables
11. Conmutadores de programación de niveles lógicos

Figura 8. Entrenador de Prácticas SIDAC DET 2020



### 1.7.- Zócalos auxiliares:

En la Figura 9 se muestra uno de los cuatro zócalos para circuitos integrados de 16 patillas. No tiene ninguna patilla conectada a la alimentación ni a tierra, por lo que es preciso efectuar todas las conexiones exteriores para su correcta puesta en funcionamiento. Para ello, se deben utilizar las islas de alimentación que se hallan en el interior y que van marcadas como “+5-” (Figura 10) si se trata de TTL o bien usar los bornes auxiliares marcados como “+V<sub>A</sub>-” (Figura 11) si precisan de otra tensión distinta exterior.

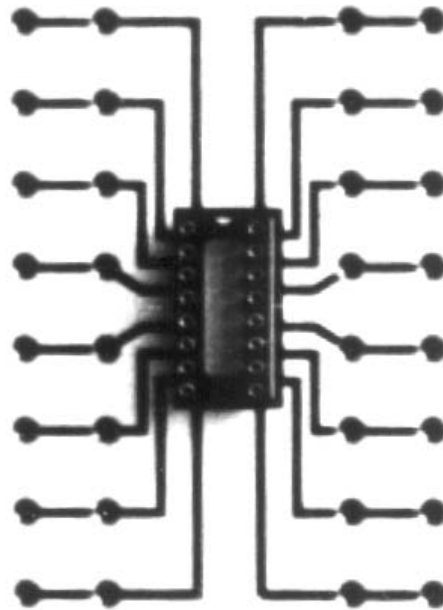


Figura 9

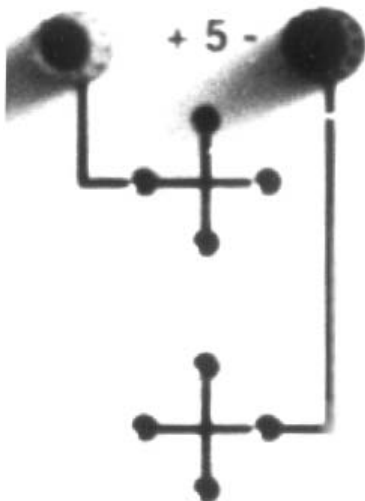


Figura 10

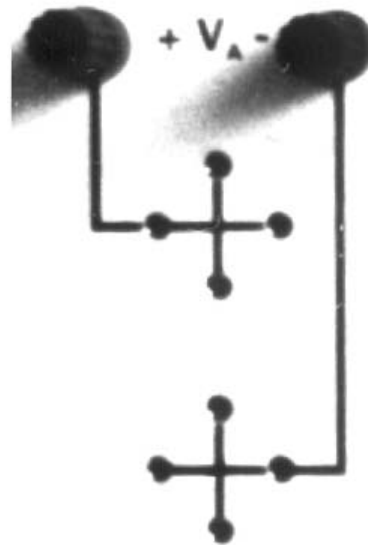


Figura 11

### 1.8.- Islas de nivel fijo y libres:

Cuando se precisa colocar una entrada a nivel fijo alto se puede recurrir a una isla marcada con Hi (Figura 12). Igualmente, si se desea fijar a un nivel bajo se puede recurrir a una marcada con Lo (Figura 13).



Figura 12



Figura 13

Si se desea distribuir una señal a varios puntos se puede utilizar una isla que no está conectada a ningún potencial fijo (Figura 14).



Figura 14

#### 1.9.- Especificaciones:

##### Circuitos:

- 6 inversores
- Ocho puertas NAND de 2 entradas
- 3 puertas NAND de 3 entradas
- 4 puertas NAND de 4 entradas
- 2 circuitos AND-OR-INVERT (AOI)
- 4 puertas EXOR de 2 entradas
- 4 puertas AND de 2 entradas
- 4 puertas OR de 2 entradas
- 4 puertas NOR de 2 entradas
- 8 flip-flops JK con señal de puesta a cero (R) asíncrona activa en alta
- 2 visualizadores de 7 segmentos
- 2 decodificadores BCD-7 segmentos
- 2 contadores bidireccionales

##### Controles:

- Interruptor de red
- Ocho indicadores de estado lógico
- Conmutador selector de frecuencia de reloj (6 márgenes de 0,1 a 100 Khz.)
- Ajuste fijo de la frecuencia de reloj
- Dos generadores de pulsos sin rebote (salida normal y complementaria)
- Ocho interruptores para salidas 1 y 0

*Varios:*

- Cuatro zócalos para circuitos integrados opcionales
- Cuatro entradas para alimentaciones externas

*Características generales:*

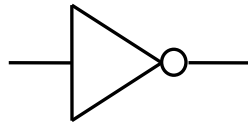
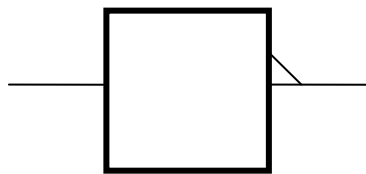
- Circuitos TTL
- Fuente estabilizada y protegida +5V
- Alimentación: 127/230V - 50 hz.
- Dimensiones: 434 x 417 x 90 mm.
- Peso: 5Kg (neto)

*Accesorios:*

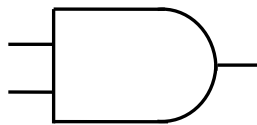
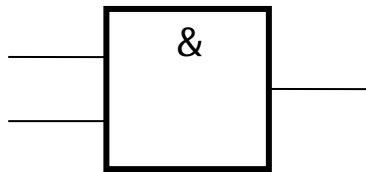
- Juego de circuitos integrados lógicos
- Juego de cables de interconexión con conectores de 2 mm.



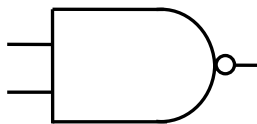
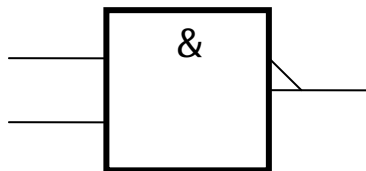
***Símbolos Estándar IEEE para puertas lógicas:***



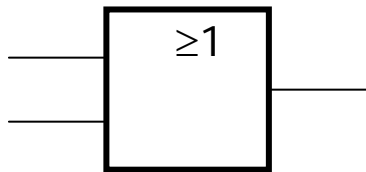
INVERSOR



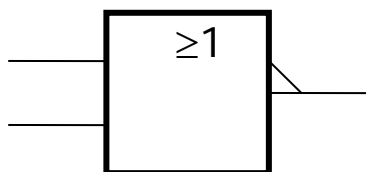
AND-2



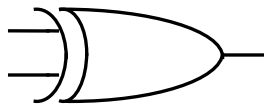
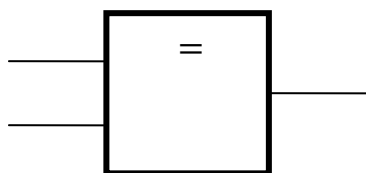
NAND-2



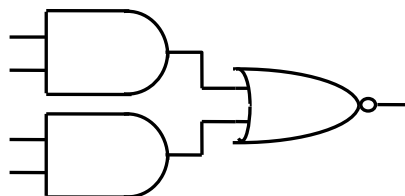
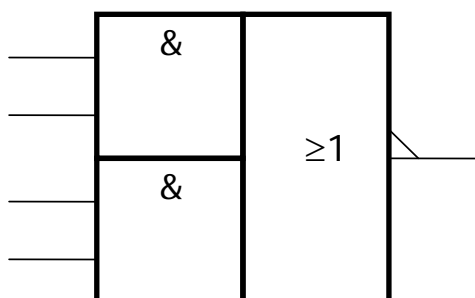
OR-2



NOR-2



XOR-2



AOI

## 2. Realizaciones prácticas.

### *Comprobación de las leyes y teoremas de Álgebra de Boole.*

Compruebe la veracidad de las siguientes ecuaciones lógicas, implementando por separado cada término de la igualdad y comprobando en todos los casos posibles que se cumple dicha igualdad. Ayúdese para la comprobación de Tablas de Verdad. Realice esquemas de los circuitos implementados y rellene convenientemente las Tablas de Verdad correspondientes a las expresiones lógicas implementadas, cotejándolas con las Tablas de Verdad teóricas.

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$x + (y + z) = (x + y) + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

$$x + (\bar{x} \cdot y) = x + y$$

$$x \cdot (\bar{x} + y) = x \cdot y$$

$$x \oplus y = (x \cdot \bar{y}) + (\bar{x} \cdot y)$$

$$\overline{x \cdot y \cdot z \cdot u} = \bar{x} + \bar{y} + \bar{z} + \bar{u}$$

### *Funciones lógicas más comunes.*

Las funciones NAND y NOR son cada una de ellas un conjunto completo de funciones, lo que significa que cualquier otra función se puede expresar sólo utilizando funciones NAND o NOR. Basándose en esto, realice las siguientes tareas:

- a) Dibuje en la siguiente tabla los circuitos que permiten implementar las funciones requeridas, utilizando para ello puertas NAND y NOR de dos entradas (NAND-2 y NOR-2):

<b>Realización Función</b>	<b>Usando solo puertas NAND-2</b>	<b>Usando solo puertas NOR-2</b>
<b>NOT (Inversor)</b>		
<b>AND-2</b>		
<b>OR-2</b>		

b) Basándose en los circuitos obtenidos en la tabla anterior, realice las siguientes funciones para dos variables (Nota: exprese primero las funciones utilizando puertas AND-2, OR-2 y NOT, y después use la tabla anterior para convertir el circuito)

- Función OR y NOR a partir de sólo puertas NAND-2.
- Función AND y NAND a partir de sólo puertas NOR-2.
- Función XOR a partir de sólo puertas NAND-2.
- Función XOR a partir de sólo puertas NOR-2.

### ***Tablas de Verdad.***

Obtenga la tabla de verdad de la siguiente función de conmutación:

$$f(x, y, z, u) = \sum m(3, 4, 5, 7, 11, 15)$$

Minimícela e implemente la expresión mínima AND/OR con dos niveles de puertas lógicas, comprobando que el valor de  $f(x, y, z, u)$  coincide con el de su tabla de verdad teórica. Adapte, si es necesario, la expresión mínima obtenida a los circuitos disponibles en el entrenador de prácticas.

