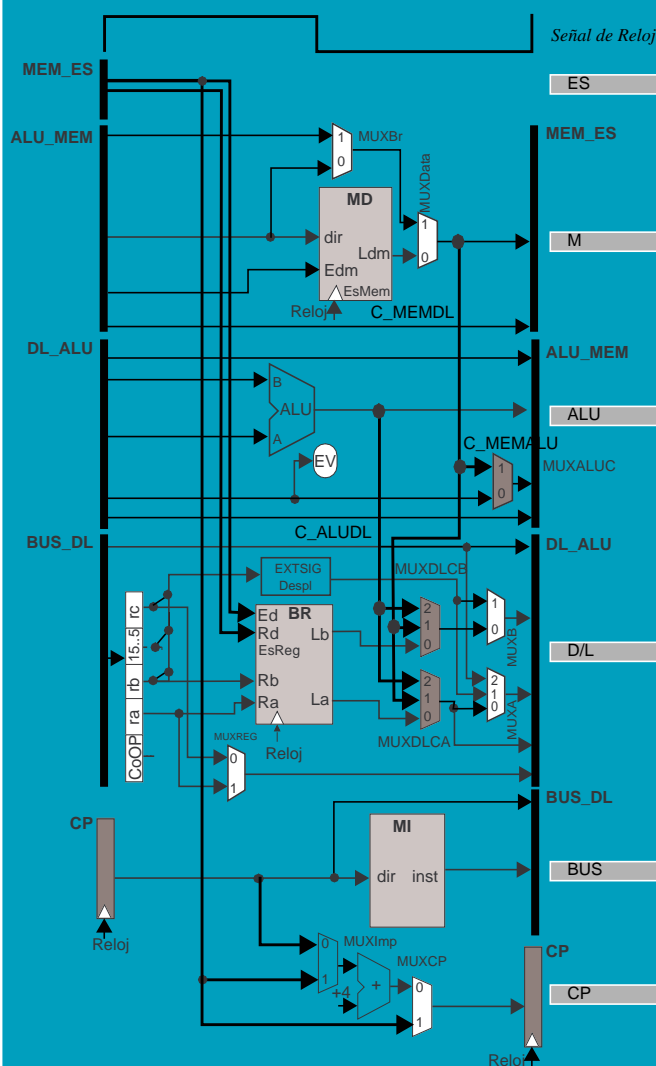


# Documentación de prácticas

## Arquitectura de Computadores II

### (Grau - AC2)



```

process (D, clock)
begin
  -- clock rising edge
  if ( clock='1' and clock'event ) then
    Q <= D;
  end if;
end process;

```



E. Herrada  
J.M. Llabería  
A. Olivé

© Copyright 2011, 2015 los autores, Universidad  
Politécnica de Cataluña

● ● ● ● ● ●

## Práctica 1 Sumador de 1 bit

1

Sumador de un bit	1
Estructura básica de un modelo en el lenguaje VHDL	2
Interfaz	2
Arquitectura	3
Paquetes y librerías	5
Elementos léxicos de VHDL	6
Operadores	7
Utilización de LogicWorks	8
Entorno	8
Creación de una descripción VHDL	9
Compilación	14
Simulación	15
Comprobación del funcionamiento lógico	16
Modelo VHDL con retardo	19
Objetos de datos en VHDL: Señales y constantes	21
Especificación de retardos en VHDL	22
Descripción en VHDL del sumador de 1bit con retardos	22
Creación de un elemento para utilizarlo en un circuito	23
Herramienta de dibujo (captura) de esquemas	26
Elementos para crear un circuito	26
Selección de un elemento de la librería	28
Patillas	28
Selección de un elemento o un icono	29
Conexión de elementos	29
Borrado un elemento	30
Otras posibilidades y utilidades	30
Señales de entrada y de salida y simulación lógica	31
Asignación y visualización de valores lógicos	31
Simulación lógica	32
Utilización de vectores de comprobación	33
Medida del tiempo de retardo	34
Ventana de Tiempo	34
Etiquetas	34
Funcionalidades en la ventana de Tiempo	36
Simulación temporal del circuito	37
Observación precisa de los retardos	40
Utilización de ficheros de excitación o estimulación	43
Apéndice 1.1: Entorno de simulación en LogicWorks	47

Directorio de trabajo .....	47
Fichero de configuración .....	47

# Práctica 1

## Sumador de 1 bit

.....

Los objetivos de esta sesión son, por un lado, familiarizarse con el entorno de trabajo de LogicWorks y un subconjunto del lenguaje VHDL (VHSIC, “Very High Speed Integrated Circuits”, **H**ardware **D**escription **L**anguage) que permite especificar circuitos digitales. Por otro lado, aprender una metodología de diseño, de simulación de circuitos y comprobación de un diseño de forma que se verifique el correcto funcionamiento del mismo. En este sentido se utilizará como ejemplo conductor un **sumador de 1 bit** o “**full adder (FA)**”.

### Sumador de un bit

La expresión algebraica correspondiente a la suma de 3 bits con la misma ponderación es la siguiente:

$$x + y + c = 2 \times c_1 + s$$

El resultado son dos bits, el bit de menor ponderación se denomina suma (s) y el otro bit se denomina acarreo de salida ( $c_1$ ).

**Expresiones lógicas.** A partir de las entradas (x, y) y de un *acarreo* de entrada ( $c_{en}$ ) en la Figura 1.1 se muestran unas expresiones lógicas, que se obtienen de la tabla de verdad, para la suma (s) y el *acarreo* de salida ( $c_{sal}$ ).

$c_{en}$	x	y	$c_{sal}$	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$s = (x \oplus y) \oplus c_{en} \quad \oplus \text{ puerta XOR}$$

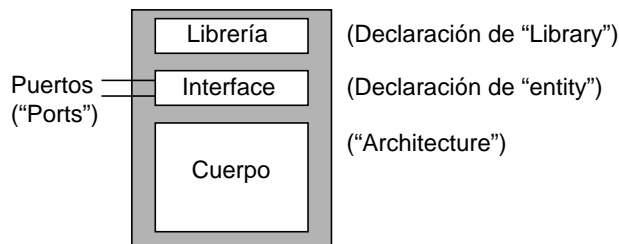
$$c_{sal} = x \cdot y + c_{en} \cdot y + c_{en} \cdot x \quad + \text{ puerta OR}$$

$$\cdot \text{ puerta AND}$$

**Figura 1.1** Tabla de verdad y expresiones lógicas para un sumador de 1 bit.

## Estructura básica de un modelo en el lenguaje VHDL

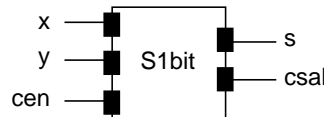
Cuando se diseña un sistema digital es una buena práctica distinguir la interfaz de la implementación (Figura 1.2). Por interfaz entendemos cómo se observa el sistema digital desde el exterior. Por implementación entendemos cómo se obtiene el comportamiento. Esto es, qué elementos se utilizan para obtener el comportamiento deseado. Además de los elementos anteriores se distingue la especificación de librerías.



**Figura 1.2** Descripción de un modelo en VHDL: consta de una interfaz y un cuerpo de la “architecture” e indicación de las librerías utilizadas.

## Interfaz

En VHDL una declaración “entity” da nombre al sistema o módulo que estamos diseñando y define su interfaz con el exterior en términos de una lista de puertos. Por ejemplo, una declaración “entity” es la equivalencia textual del símbolo de un sumador de 1 bit (Figura 1.3). El símbolo muestra la interfaz: señales de entrada (x, y, cen) y señales de salida (S, csal).



**Figura 1.3** Símbolo de un sumador de 1 bit.

En VHDL la interfaz se especifica de la siguiente forma.

Ejemplo	entity S1bit is			Sintaxis	entity NAME_OF_ENTITY is		
	port (	x: in	std_logic;		port (signal_names: mode		type;
		y: in	std_logic;		signal_names: mode		type;
		cen: in	std_logic;		...		
		s: out	std_logic;		signal_names: mode		type);
		csal: out	std_logic);		end [NAME_OF_ENTITY] ;		
		end S1bit;					

Se empieza con la palabra clave “entity”, seguido por el nombre del módulo y la palabra clave “is”. Seguidamente se declaran los puertos utilizando la palabra clave “port”. Una declaración “entity” siempre finaliza con la palabra clave “end”, opcionalmente ([ ]) seguida del nombre de la “entity”.

El nombre de la “entity” es un identificador seleccionado por el usuario. Los “signal\_names” constan de una lista separada por punto y coma de uno o más identificadores que especifican las señales de la interfaz externa. Las señales (“signal\_names”) tienen un “mode” que es una palabra reservada para indicar la dirección de la señal: a) señal de entrada (“in”) y b) señal de salida (“out”). Los puertos se pueden enumerar en cualquier orden.

Un “signal\_name” es un objeto denominado “signal”. La implementación de un objeto “signal” es una secuencia de pares tiempo-valor. Con ello se describe el comportamiento de una señal transportada por un cable en un circuito digital. A la secuencia de pares tiempo-valor se la denomina frente de onda.

**Tipos (“type”).** Una señal tiene un tipo predefinido o definido por el usuario. Los tipos básicos en VHDL son “bit” y “bit\_vector”. Una señal de tipo “bit” puede tomar los valores 0 y 1. El tipo “bit\_vector” es un vector cuyos elementos son de tipo “bit”. Los tipos “bit” y “bit\_vector” no son suficientemente versátiles para especificar todas las posibilidades en una señal digital. Por ello se han definido otros tipos con un mayor rango de valores. En total se utilizan 9 posibilidades estando entre ellas el valor no definido (“U”), alta impedancia (“Z”), no importa o no especificado (“don’t care”, “-”), el valor desconocido (“X”) y codificaciones que indican también la fortaleza de la señal. Como ejemplo, el valor alta impedancia permite que una señal pueda ser activada desde múltiples fuentes.

La definición de tipo usualmente utilizada para señales de un bit es “std\_logic” que codifica los 9 valores descritos. Esta definición de tipo está incluida en el “package” std\_logic\_1164 de la librería IEEE<sup>1</sup>.

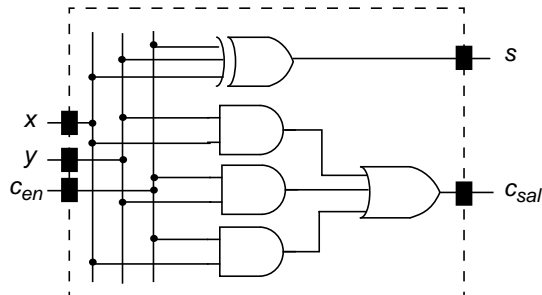
La definición de valores en el tipo std\_logic se ha efectuado mediante el tipo carácter. Puesto que el tipo carácter se utiliza en la definición, la forma de especificar los valores es parte de la definición y deben especificarse de la misma forma. Por ello, los valores deben especificarse con mayúsculas.

## Arquitectura

La operación o función de una “entity” se describe en lo que se denomina cuerpo de la “architecture”, el cual está constituido por una declaración y una descripción de la operación mediante sentencias de asignación de señales concurrentes. En la Figura 1.4 se muestra la descripción de un sumador de 1 bit con puertas lógicas que mimetiza las expresiones lógicas descritas previamente.

---

1. Posteriormente detallaremos otras definiciones de tipo usuales.



**Figura 1.4** Esquema de puertas de la operación suma de 1 bit.

En VHDL la función se especifica en el cuerpo de la arquitectura ("architecture") de la siguiente forma.

**Ejemplo**

```
architecture compor of S1bit is
begin
  s <= (x xor y) xor cen;
  csal <= (x and y) or (x and cen) or (y and cen);
end compor;
```

**Sintaxis**

```
architecture architecture_name of NAME_OF_ENTITY is
begin
  -- Concurrent Statements
end architecture_name;
```

La línea cabecera del cuerpo de la arquitectura define el nombre de la arquitectura, por ejemplo "compor", y lo asocia con la declaración "entity", S1bit. El nombre de "architecture" puede ser cualquier identificador que sea legal. El cuerpo principal de la arquitectura empieza con la palabra clave "begin" y finaliza con la palabra clave "end" seguida del nombre dado a la "architecture". Pueden existir varios cuerpos de la arquitectura que se corresponden con una única definición de la interfaz ("entity"). Cada uno de los cuerpos describe alternativas para implementar la operación del circuito.

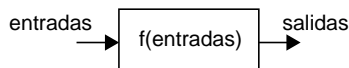
**Comentarios.** Una utilización apropiada de comentarios mejora la lectura y comprensión de cualquier código en VHDL. Un comentario empieza con dos guiones (--). El compilador VHDL ignora cualquier texto posterior a "--" hasta el final de la línea en la que aparecen los dos guiones. Un comentario que ocupa varias líneas necesita empezar cada línea con dos guiones. En LogicWorks no deben utilizarse caracteres con acentos o la "ñ". El compilador indica error.

Los puertos especificados en la interfaz son visibles en el cuerpo de la arquitectura. Los puertos se observan como terminales (pins) externos de un circuito. Ahora bien, desde el punto de vista interno del circuito diseñado, un terminal es un cable con una conexión externa que transporta una señal.

Para modelar el sumador de 1 bit utilizaremos un modelo que describe el comportamiento o la función.



**Modelo de comportamiento (“Behavioral Model”).** Se especifican funciones que relacionan las salidas con las entradas (Figura 1.5).



**Figura 1.5** Modelo de comportamiento.

**Sentencia de asignación de señal.** En el cuerpo de la arquitectura se utilizan sentencias concurrentes las cuales se denominan de asignación de señales. Estas sentencias especifican funciones que hacen uso de operadores lógicos. Los operadores lógicos permitidos son: and, or, nand, nor, xor, xnor, y not. Posteriormente se detallan otros tipos de operadores para otros tipos de señales. Como operador de asignación se utiliza el símbolo `<=` que asigna el valor de la expresión que está a la derecha del símbolo a la señal que hay en la izquierda.

Ejemplo	Sintaxis
<code>s &lt;= (x xor y) xor cen;</code>	<code>signal-name &lt;= expresion;</code>

Las sentencias de asignación de señales se dice que son concurrentes porque pueden activarse y ser evaluadas a la vez. En este sentido, el orden con que se escriban no es importante, ya que no se evalúan secuencialmente. En el ejemplo del sumador las dos sentencias que se han explicitado se evalúan concurrentemente ( `s <= . . .` y `csal <= . . .` ).

Una sentencia de asignación de señal se ejecuta, esto es se evalúa la parte derecha y se asigna a la parte izquierda, cuando una o más señales en la parte derecha cambia su valor. Es decir, se produce un evento en una de las señales que se especifican en la parte derecha de la asignación. Adicionalmente se puede especificar un retardo asociado con el cambio, el cual detallaremos posteriormente (“Especificación de retardos en VHDL” en la página 22).

Recopilando, un sistema digital está básicamente conducido por datos y eventos, los cuales se producen en una señal y ello da lugar a que se produzcan eventos en otra señal. La ejecución de las sentencias está determinada por el flujo de valores en las señales. Por tanto, el orden en el cual se especifican las sentencias no es significativo.

## Paquetes y librerías

En VHDL, un “package” (paquete o almacén) es un fichero que contiene declaraciones de objetos, tipos de datos, entre otros, utilizados de forma usual.

Una librería es un lugar donde se almacena información (directorio) que se referencia por un nombre lógico (camino en el árbol de directorios). En la librería se almacenan “packages”. Antes de utilizarse deben declararse utilizando la cláusula o palabra clave “library”.

Una vez se ha declarado una librería, las declaraciones contenidas en un “package” se hacen accesibles a un modelo VHDL mediante la palabra clave “use”.

En resumen, al principio de un fichero VHDL se utilizan las palabras clave “library” y “use” de la siguiente forma. La extensión “.all” indica que se utiliza todo el “package” y si sólo es un ítem se especifica el ítem (“my\_func”).

Ejemplo	Sintaxis
<b>library</b> ieee; <b>use</b> ieee.std_logic_1164.all;	<b>library</b> logical-library-name-1, logical-library-name-2; <b>use</b> logical-library-name-1.package1.all ; <b>use</b> logical-library-name2.package2.my_func;

Previamente hemos comentado que el tipo “std\_logic” está definido en el “package” std\_logic\_1164 perteneciente a la librería ieee. Por tanto, para utilizar el tipo “std\_logic” debemos de especificar la librería y el “package”. Esto es ieee.std\_logic\_1164.all.

Las librerías STD y WORK están implícitamente declaradas. La primera contiene “packages” estándar suministrados por el distribuidor. La segunda se refiere al directorio de trabajo, el cual se puede establecer dentro del entorno de simulación que se usa (Apéndice 1.1).

## Elementos léxicos de VHDL

Un identificador es una palabra definida por el usuario para nombrar objetos en modelos VHDL. Cuando se elige un identificador se deben tener en cuenta las siguientes reglas.

- Puede contener sólo caracteres alfanuméricos (A hasta Z, a hasta z, 0 - 9) y el carácter (\_) guion bajo.
- El primer carácter debe ser una letra y el último carácter no puede ser un guion bajo.
- Un identificador no puede incluir dos guiones bajos consecutivos.
- En un identificador no se distingue entre mayúsculas o minúsculas.
- Un identificador puede ser de cualquier longitud.

Ciertos identificadores son utilizados por VHDL como palabras clave para constructores específicos y por ello no pueden utilizarse para identificar objetos o señales. Usualmente las palabras clave se muestran en negrita o en otro color.

**Caracteres.** Un carácter se especifica entre comillas: Por ejemplo, los valores de tipo std\_logic se especifican como: ‘0’, ‘1’, ‘U’, ‘X’, ‘-’.

**Espacios en blanco.** VHDL no diferencia entre uno o varios espacios en blanco o tabulaciones.

**Sentencias en VHDL.** De forma similar a otros lenguajes de programación, cada sentencia VHDL finaliza con punto y coma.

**Paréntesis.** VHDL es laxo en los requisitos de uso de paréntesis. En cualquier caso, para mejorar la lectura y comprensión de un código, es buena idea utilizar paréntesis siempre que se crea necesario con el objetivo de ayudar a entender el propósito de un código.

## Operadores

Los operadores lógicos están definidos para tipos “bit” y “std\_logic” y sus vectores. Estos operadores se utilizan para definir expresiones lógicas o para efectuar operaciones bit a bit en vectores de bits. El resultado que se obtiene es del mismo tipo que los datos.

**Operadores lógicos:** and, or, nand, nor, xor, xnor, y not.


Notemos que algunos operadores, como nand o nor, no son asociativos. En consecuencia deben utilizarse paréntesis para indicar el orden de evaluación. En la Figura 1.6 se muestran funciones a nivel de puerta lógica disponibles en VHDL.

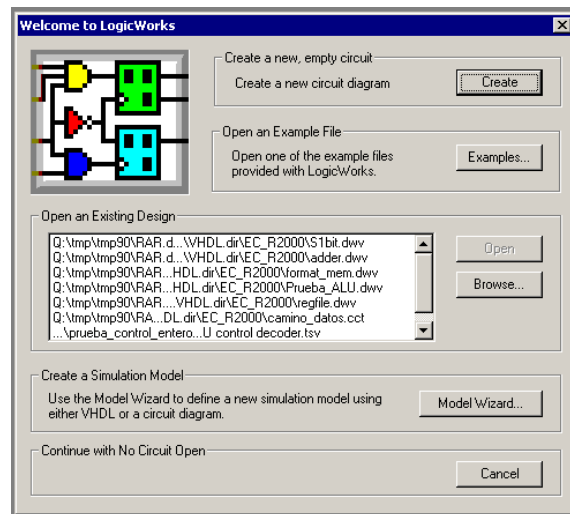
Expresión lógica	Codificación VHDL	Operación
$Z1 = A1 \cdot B1$	<code>z1 &lt;= a1 AND b1;</code>	AND
$Z2 = A2 + B2$	<code>z2 &lt;= a2 OR b2;</code>	OR
$Z3 = A3 \oplus B3$	<code>z3 &lt;= a3 XOR b3;</code>	OR exclusiva
$Z4 = \overline{A4}$	<code>z4 &lt;= NOT a4;</code>	negación
$Z5 = \overline{(A5 \cdot B5)}$	<code>z5 &lt;= a5 NAND b5;</code> <code>z5 &lt;= NOT(a5 AND b5);</code>	NAND
$Z6 = \overline{(A6 + B6)}$	<code>z6 &lt;= a6 NOR b6;</code> <code>z6 &lt;= NOT(a6 OR b6);</code>	NOR

**Figura 1.6** Funciones lógicas disponibles en VHDL.

## Utilización de LogicWorks

### Entorno

Cuando se inicializa LogicWorks emerge una pantalla donde se muestran distintas opciones que no utilizaremos por ahora. Para seguir se posiciona el cursor  en el rótulo “Cancel” y se pulsa el botón izquierdo del ratón (Figura 1.7).



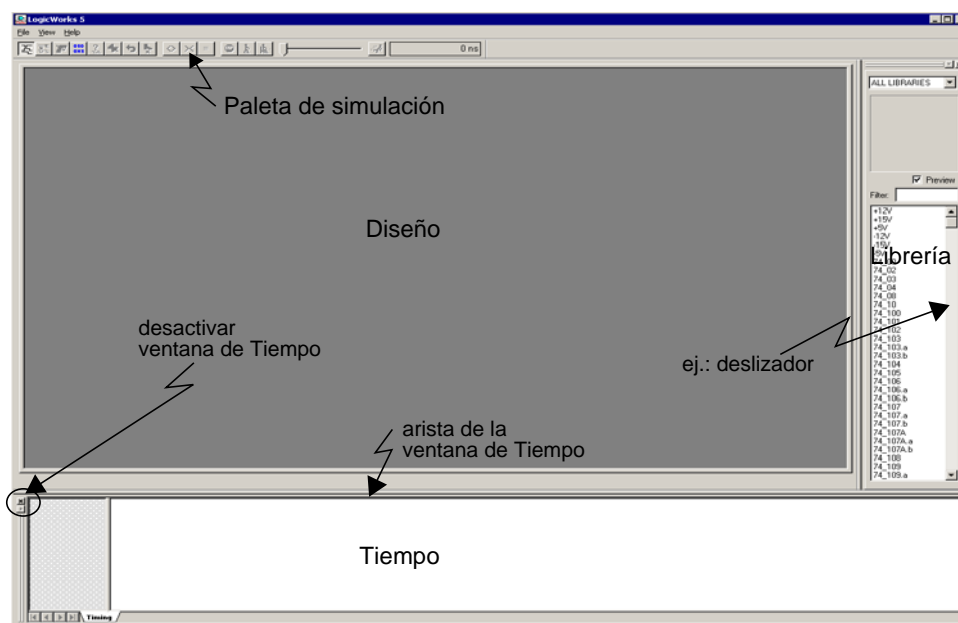
**Figura 1.7** Ventana de bienvenida.

Una vez inicializado LogicWorks, se observan varias ventanas en la pantalla (Figura 1.8).


- La ventana de Diseño se utiliza para editar ficheros y para crear los esquemas de circuitos y manipularlos con las herramientas disponibles.
- La ventana de Tiempo se utiliza por el programa de simulación de retardos para visualizar la evolución temporal de las señales etiquetadas en el esquema del circuito.
- La paleta de simulación se utiliza para controlar acciones relacionadas con la simulación. Por ejemplo, inicio, pausa, velocidad, desactivar el simulador.
- La ventana de Librería muestra los elementos que pueden utilizarse para diseñar el circuito cuando se dibuja el esquema de un circuito.

Las ventanas visualizan sólo parte de la información. Para visualizar el resto se utiliza el deslizador (horizontal, vertical) de la ventana.

El tamaño (horizontal/vertical) de algunas ventanas se puede modificar posicionando el cursor en una arista de la ventana. El símbolo del cursor se modifica y en ese instante se puede redimensionar la ventana en la dirección deseada pulsando el botón izquierdo y moviendo el cursor.



**Figura 1.8** Ventanas de trabajo en LogicWorks.

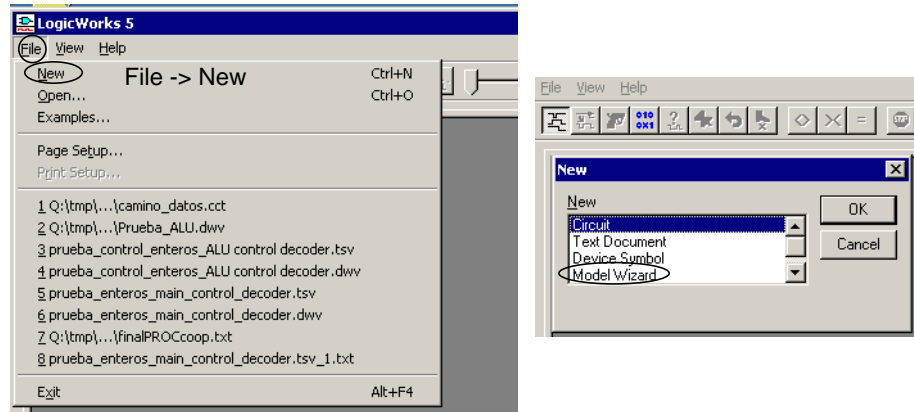
Las ventanas de Librería y Tiempo se pueden desactivar posicionando el cursor sobre el símbolo  que aparece en una esquina de la ventana y pulsando el botón izquierdo del ratón. Posteriormente se explicará cómo activarlas de nuevo.

## Creación de una descripción VHDL

En este apartado se muestra cómo utilizar una herramienta denominada "Model Wizard" para que nos ayude a efectuar una descripción VHDL.

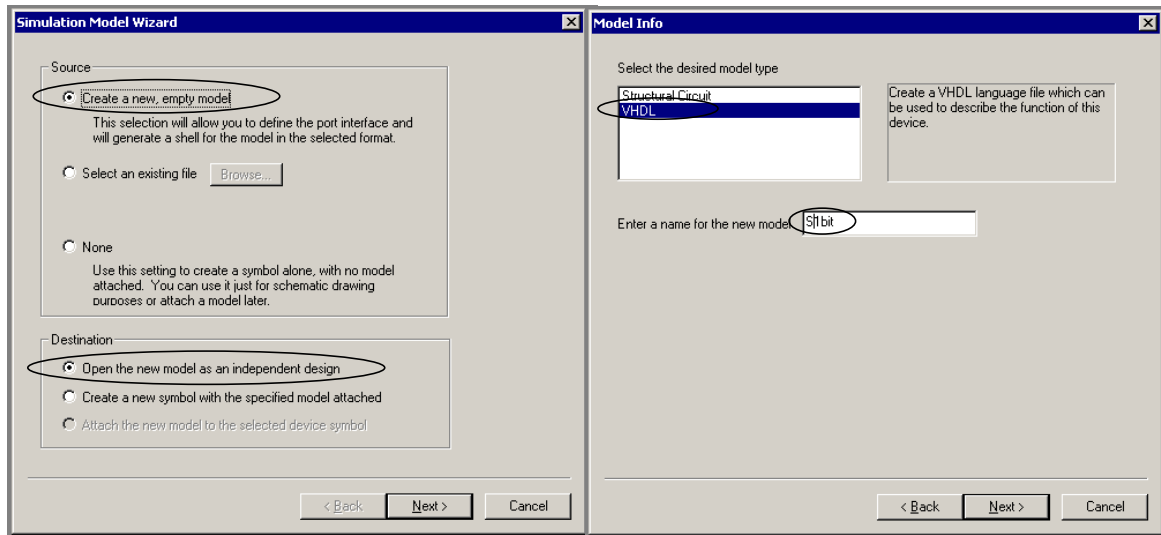
Para utilizar la herramienta debemos de efectuar los siguientes pasos: a) posicionar el cursor en el rótulo "File" ubicado en la parte izquierda superior de la ventana de LogicWorks, b) pulsar el botón izquierdo del ratón y emerge una ventana con rótulos, c) liberar el botón, mover el cursor hasta el rótulo que especifica la orden deseada (en este caso "New") y d) pulsar el botón izquierdo del ratón (selección de la orden). En la parte izquierda de la Figura 1.9 se muestra una simbología para explicitar los pasos previos. Una flecha (->) indica que el rótulo está en otra ventana que se visualiza al pulsar el botón

izquierdo del ratón en el rótulo (leyenda) previo. Al conjunto de pasos lo denominaremos dar la orden “último rótulo en el que se ha posicionando el cursor”. Finalmente, en la ventana emergente se selecciona “Model Wizard” (parte derecha de la Figura 1.9).



**Figura 1.9** Ventana para iniciar o proseguir un diseño y selección de la herramienta de ayuda (“Model Wizard”).

En la ventana emergente “Simulation Model Wizard” deben efectuarse las dos selecciones que se muestran en la parte izquierda de la Figura 1.10. Una es relativa a la fuente (“source”) y otra es relativa al destino (“destination”). Para ello hay que posicionar el ratón en el botón que se quiera seleccionar de la pantalla emergente y pulsar el botón izquierdo del ratón.



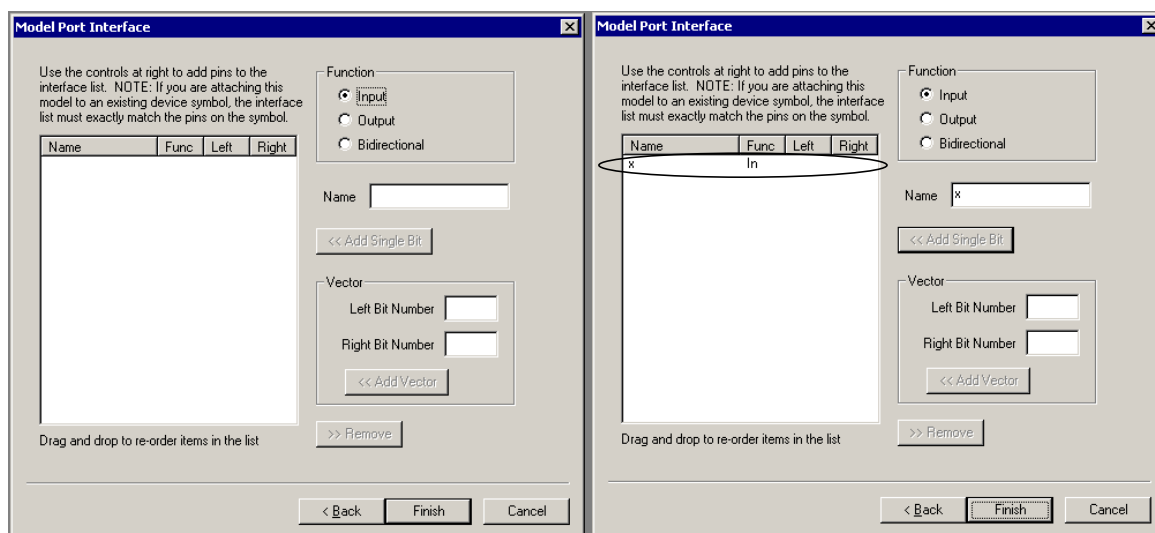
**Figura 1.10** Creación de un módulo como diseño independiente.

Seguidamente se solicita que la herramienta muestre la siguiente ventana pulsando el rótulo “Next”. En la nueva ventana “Model Info” se selecciona el rótulo VHDL y se indica un nombre para el módulo (parte derecha de la Figura 1.10). Por ejemplo S1bit.

**Trabajo:** Inicie la creación de un fichero VHDL para diseñar un sumador de 1 bit.

Una vez se ha especificado el nombre, la herramienta muestra la siguiente ventana pulsando el rótulo “Next”. La ventana emergente “Model Port Interface” se utiliza para especificar las señales de entrada y salida del dispositivo que queremos diseñar (Figura 1.11).

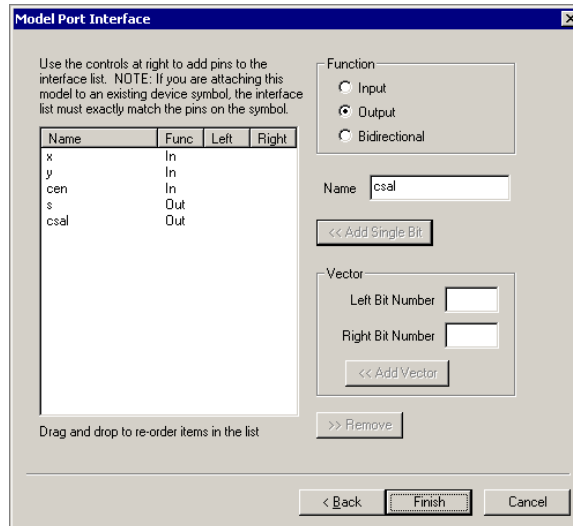
Para especificar una señal de entrada se marca en “Function” el botón “Input” y en “Name” se especifica el nombre de la señal. Una vez se ha introducido el nombre se añade la señal a la especificación pulsando en el rótulo “<<AddSingle Bit”, el cual se activa después de introducir el nombre de la señal. Una vez se ha pulsado en el rótulo “<<AddSingle Bit” la señal se muestra en la parte izquierda de la ventana (parte derecha de la Figura 1.11). Para indicar una señal de salida se marca en el botón Output.



**Figura 1.11** Ventana para especificar las entradas y salidas del módulo o dispositivo. Detalle de la ventana utilizada mostrando la especificación de una señal de entrada.

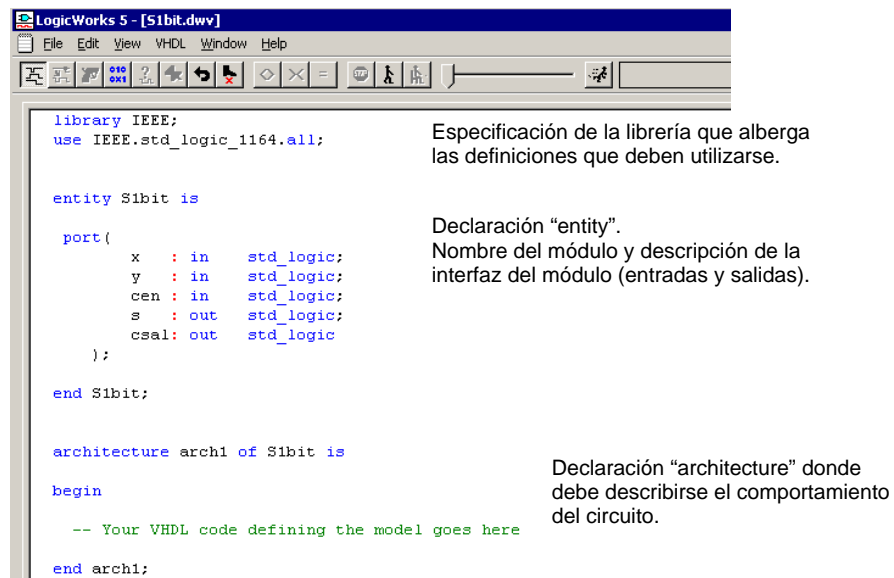
**Trabajo:** Especifique todas las entradas y salidas del sumador de 1 bit descrito previamente.

Una vez especificadas todas las señales de entrada y salida se observa la siguiente información en la ventana (Figura 1.12).



**Figura 1.12** Detalle de todas las señales de entrada y salida especificadas en el módulo S1bit.

Para finalizar se pulsa en el rótulo “Finish” y aparece una ventana que muestra el fichero cuyo nombre es S1bit.dwv (Figura 1.13).



**Figura 1.13** Documento generado con la herramienta que permite especificar la interfaz de un modelo VHDL.



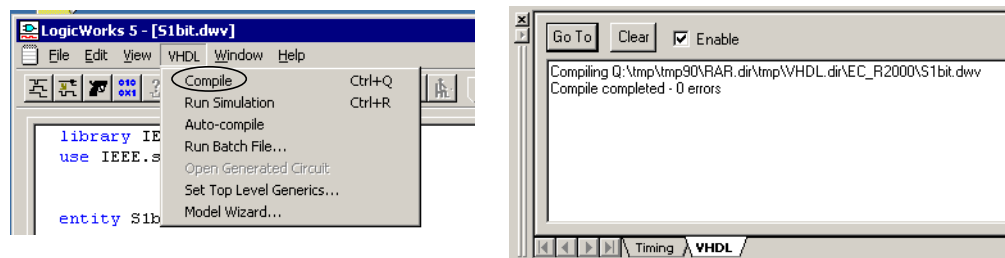


Para almacenar el fichero en disco efectúe los pasos que se muestran en la parte derecha de la Figura 1.14. Hay dos opciones en función de si ya existe o no existe el fichero:

- “Save”: si ya existe el fichero.
- “SaveAs ...”: si es la primera vez o se quiere modificar el nombre del fichero. Utilizando la ventana que emerge nos ubicamos en el directorio donde queremos que se almacene el fichero y establecemos el nombre de fichero que queremos. Notemos que la extensión del fichero es “.dwv”.

## Compilación

Seguidamente debe compilarse el fichero. Para ello posicione el cursor en el rótulo VHDL y pulse el botón izquierdo del ratón para que emerja una ventana desplegable (parte izquierda de la Figura 1.15). En esta ventana se selecciona “Compile”.



**Figura 1.15** Ventana para indicar la acción de compilación del fichero VHDL que se está editando. Resultado de la compilación.

En la parte inferior de la ventana de LogicWorks aparece un nuevo panel, denominado VHDL, donde se muestra el resultado de la compilación (parte derecha de la Figura 1.15).

Si la compilación da algún error, en el listado se indica el número de línea del fichero VHDL donde se detecta el error y una indicación del tipo de error. Para posicionarse en la línea seleccione la línea en el listado y pulse el botón “Go to” que se muestra en la parte izquierda superior del panel VHDL (parte derecha de la Figura 1.15).

---

**Trabajo:** Compile la especificación VHDL del sumador de 1 bit.

---

El compilador crea un fichero con el mismo nombre que el fichero compilado siendo la extensión “.dwa”.

## Simulación

Después de una compilación sin errores se puede efectuar una simulación. Para ello se pulsa el icono de la paleta de simulación que se muestra en la Figura 1.16.



**Figura 1.16** *Icono para activar la simulación.*

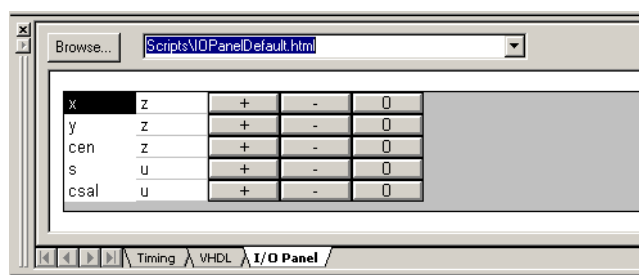
El fondo del documento VHDL toma un tono grisáceo para indicar que no se puede editar mientras se efectúa la simulación.

Para establecer valores en las entradas y comprobar las salidas se pulsa el icono de la paleta de simulación que se muestra en la Figura 1.17.



**Figura 1.17** Icono para activar la ventana de entrada de datos y observación de las salidas.

En la parte inferior de la ventana de LogicWorks aparece un nuevo panel denominado "I/O Panel" (Figura 1.18). En este panel se muestran las entradas y salidas especificadas en la definición del módulo y unos botones para establecer valores en las señales de entrada. Si no se visualizan todas las señales redimensione el tamaño del panel. Para ello, posicione en la arista superior de panel. Cuando observe que el símbolo del cursor se modifica mueva el cursor verticalmente hacia arriba.



**Figura 1.18** *Ventana de entrada de datos y observación de las salidas.*


*Nota: Si el "I/O"Panel ya ha sido usado previamente debemos pulsar en la lista de selec-*



*ción y seleccionar "Scripts\IOPanelDefault.html" para que volvamos a observar las señales y botones que permiten establecer valores en las señales de entrada.*

Pulse el botón que muestra un cero para inicializar con este valor cada una de las señales de entrada. Posteriormente se utilizan los botones + y - para establecer valores en las señales de entrada. Concurrentemente se observa el resultado en las señales de salida.

*Trabajo: Efectúe las acciones descritas para activar la simulación.*

Del modo de simulación se sale pulsando el icono  (Figura 1.19) de la paleta de simulación. Esto es, se para el simulador.



**Figura 1.19** Icono que para el simulador.

## Comprobación del funcionamiento lógico

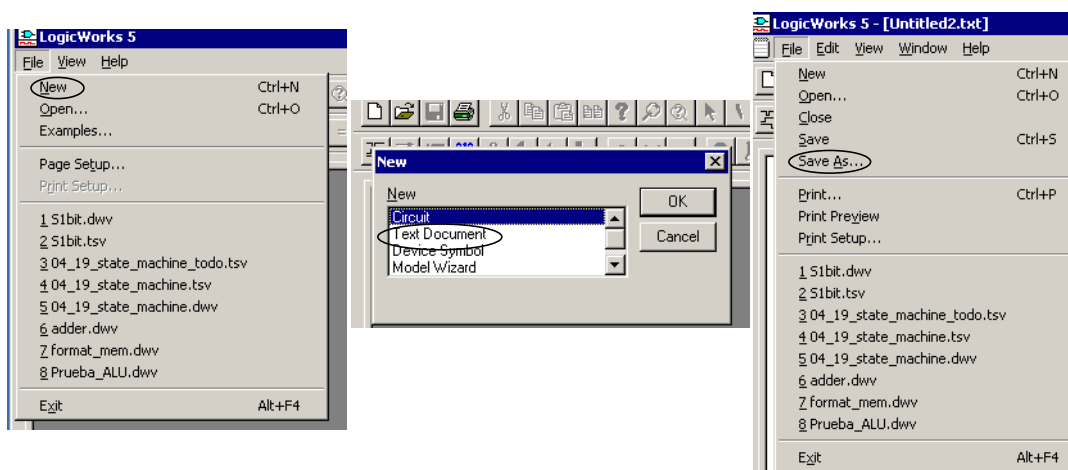
Para comprobar el funcionamiento del dispositivo utilizaremos la tabla de verdad del sumador de 1bit (Figura 1.20). Uno a uno establecemos los valores de las señales de entrada correspondiente a cada fila de la tabla de verdad. Seguidamente comprobamos si las señales de salida del módulo VHDL se corresponde con los valores asociados a las señales de salida especificadas en la misma fila en la tabla de verdad.

entradas			salidas	
cen	x	y	csal	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

**Figura 1.20** Tabla de verdad de un sumador de 1 bit.

**Trabajo:** Compruebe la tabla de verdad utilizando el “I/OPanel”.

Para comprobar el funcionamiento de una forma metódica pero menos tediosa se utiliza un fichero de comprobación (“test”). Este fichero se edita utilizando LogicWorks. Para editar el fichero se selecciona en la ventana de LogicWorks “File -> New” y en la ventana emergente se selecciona “Text Document” (centro de la Figura 1.21).



**Figura 1.21** Edición de un fichero de texto. Nombrar un fichero y almacenar en disco.

Seguidamente debemos dar un nombre al fichero y almacenarlo. Para ello se efectúa la selección que se muestra en la parte derecha de la Figura 1.21. Emerge una ventana que se utiliza para ubicarse en el directorio de trabajo y especificar el nombre del fichero. El nombre del fichero de test o comprobación debe ser “nombre del fichero VHDL” que se quiere comprobar con la extensión “.tsv”. En el caso que nos ocupa el nombre es S1bit.tsv.

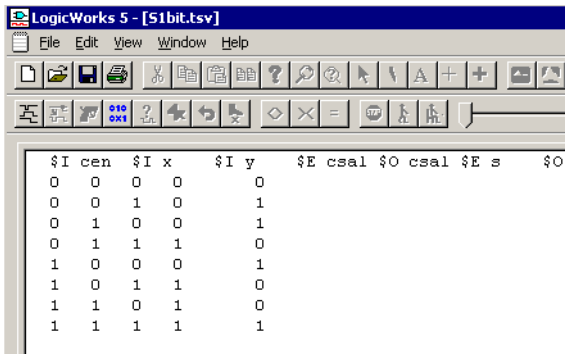
El formato del fichero es el siguiente:

- Primera fila: nombre de las variables de entrada y salida precedidas respectivamente por \$I y \$E. La separación entre los nombres de las variables se efectúa mediante tabulación.
- En las siguientes filas se escriben los valores de las señales de entrada que queremos comprobar y los valores de las señales de salida que esperamos obtener con estos valores de entrada. Los valores en una fila se separan mediante tabulación. No es necesario que en el fichero se observen las columnas perfectamente alineadas. Si los nombres de las variables junto con los prefijos ocupan más espacio que el número de caracteres de un tabulador, no es posible que las columnas queden alineadas.

En resumen, en una columna del fichero identificamos el nombre de la señal en la primera fila y en las siguientes filas el valor de la señal de entrada o el valor de la señal de salida que esperamos obtener.

Además de las señales de entrada y salida, en el fichero de comprobación se puede especificar que se nos muestren los valores de las señales de salida que evalúa el módulo diseñado. Para ello, después del nombre de una variable de salida (\$E variable de salida) insertamos un tabulador y volvemos a especificar el nombre de la variable de salida con el prefijo \$O. En las filas donde se han especificado los valores de las señales también debe insertarse en la posición correspondiente el tabulador que se ha insertado en la primera línea.

En la parte izquierda de la Figura 1.22 se muestra el fichero editado en LogicWorks.




\$I cen	\$I x	\$I y	\$E csal	\$O csal	\$E s	\$O s
0	0	0	0		0	
0	0	1	0		1	
0	1	0	0		1	
0	1	1	1		0	
1	0	0	0		1	
1	0	1	1		0	
1	1	0	1		0	
1	1	1	1		1	

**Figura 1.22** Fichero de entrada para la comprobación del circuito.

Observe que la distancia entre la cuarta columna y la quinta columna es mayor que la distancia entre las otras columnas. Esta diferencia se corresponde con la tabulación requerida debido a que se demanda que la simulación muestre la señal de salida csal (\$O csal). En el fichero se solicitan las señales \$O csal y \$O s. En la parte derecha de la Figura 1.22 se muestra el formato con las columnas alineadas.

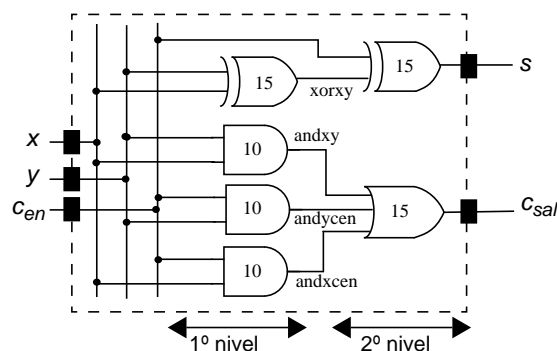
Una vez editado el fichero de comprobación se debe almacenar en disco. Para ello se selecciona en la ventana de LogicWorks "File -> Save" o "File-> Save As ..." en función de si el fichero ya ha sido o no almacenado previamente.

Una vez se ha almacenado el fichero en disco se abre, si no se tiene abierto, el fichero con la especificación VHDL del modelo que se quiere comprobar. Seguidamente se posiciona el cursor en el rótulo VHDL y se pulsa el botón izquierdo del ratón para que emerja una ventana desplegable. En esta ventana se selecciona "Run Simulation" (Figura 1.15). También se inicia la simulación pulsando en el icono  ("Run Simulator") de la paleta de simulación. En la parte inferior de la ventana de LogicWorks se muestra un panel denominado "TestPanel" en el cual se muestra el resultado de la comprobación del funcionamiento (Figura 1.23).



Estamos interesados en una simulación detallada del circuito donde se modelen las transiciones de todas la señales en una implementación física. Entonces, además de las señales de la interfaz queremos modelar las señales internas del circuito.

En la Figura 1.24 se muestra un esquema con puertas lógicas de un sumador de 1bit y el retardo asociado a cada puerta. El número de niveles de puertas del sumador depende del tipo de puertas disponible y del número de entradas de las puertas. Las puertas utilizadas tienen 2 entradas (AND y XOR) y 3 entradas (OR). En esta sección se describe cómo especificar el cuerpo de la arquitectura para describir con detalle el esquema de la Figura 1.24. Notemos que la descripción de la interfaz ("entity") no se modifica.



Los números en el interior de las puertas indican el retardo en ns

**Figura 1.24** Esquema de puertas lógicas de un sumador de 1 bit.

En el esquema de la Figura 1.24 se observan señales internas, las cuales deben describirse en el modelo. Así mismo debe de especificarse el retardo de cada puerta lógica.

Las sentencias de asignación de señales concurrentes, en la especificación de la arquitectura, se comunican mediante señales ("signals"), las cuales representan las entradas y salidas y cables del circuito digital. En la declaración de la interfaz se declaran los puertos de entrada y salida del módulo. Si son necesarios cables adicionales se declaran señales ("signals") en la parte declarativa del cuerpo de la arquitectura. Esto es, en la parte donde se describe el funcionamiento.

Teniendo en cuenta la declaración de señales y la especificación de constantes para especificar retardos, el aspecto del cuerpo de la arquitectura es el siguiente.

Ubicación	<b>architecture</b> architecture_name <b>of</b> NAME_OF_ENTITY <b>is</b>	parte declarativa
	<b>constant</b> . . .	
	<b>signal</b> . . .	
	<b>begin</b>	
	-- Concurrent Statements	
	<b>end</b> architecture_name;	



## Objetos de datos en VHDL: Señales y constantes

Un objeto de datos (“Data Object”) es un ítem en VHDL que se crea mediante una declaración, tiene un nombre (identificador asociado), un valor y tipo asociado. Un objeto puede ser una constante (“Constant”) o una señal (“Signal”). Posteriormente explicaremos otros objetos. Hasta ahora hemos visto que los puertos de entrada y salida son señales. Las señales pueden considerarse cables en un esquema de circuito, las cuales pueden tener un valor actual y valores futuros (historia de valores), y estos valores son función de las sentencias de asignación de señales utilizadas. Una constante asocia un valor a un símbolo de un tipo de datos. Seguidamente observaremos que la declaración de los distintos objetos de datos es muy similar.

**Constante (“Constant”).** Una constante se inicializa con un valor al declararla y este valor no se puede modificar durante la simulación. La declaración se efectúa de la siguiente forma.

Ejemplo	Sintaxis
<b>constant</b> retardoand: <b>time</b> := 2ns;	<b>constant</b> constant name: <b>type</b> := initial value;

En la declaración de una constante se especifica el nombre del objeto, el tipo y su valor. Las constantes se declaran al principio del cuerpo de la arquitectura y entonces pueden ser utilizadas en cualquier sentencia dentro de la arquitectura.

**Tipo “time”.** Es un tipo predefinido en el estándar de VHDL. Se especifica un valor numérico, un espacio y las unidades (fs, ps, ns, us, ms, sec, min, hr).

La disponibilidad del tipo “time” es una consecuencia natural del modelado que se quiere obtener al efectuar la simulación.

**Señales (“Signals”).** Las señales se declaran al principio del cuerpo de la arquitectura utilizando la siguiente sentencia.

Ejemplo	Sintaxis
<b>signal</b> andxy, andxcen: std_logic;	<b>signal</b> signal_names: type [= initial value] ;

En la declaración de una señal se especifica el nombre del objeto, el tipo y un posible valor inicial<sup>1</sup>. Los objetos “signal” por defecto están indefinidos.

El objeto “signal” representa señales lógicas transportadas por un cable en un circuito. Una señal no tiene memoria. Por tanto, si la fuente de la señal se elimina la señal no tiene ningún valor. Esto es, está indefinida.

1. En LogicWorks no funciona la inicialización de una señal de tipo std\_logic.

## Especificación de retardos en VHDL

Un retardo en una sentencia de asignación de señal se especifica de la siguiente forma.

Ejemplo	Sintaxis
s <= a and b after 2 ns;	signal_name <= value expression [ after time expresion] ;

La especificación anterior refleja la realidad del hardware, donde la lógica tiene retardos de propagación. El retardo incorporado en una sentencia de asignación de señal tiene en cuenta el retardo de la lógica (evaluar “value expresion”) y cualquier otro retardo de propagación necesario para que el nuevo valor se observe al final del cable.

En resumen, una señal actualiza su valor cuando se ejecuta su sentencia de asignación de señal y ha transcurrido el retardo especificado. Si no se especifica un retardo, la señal se actualiza después de un retardo denominado delta utilizado por el simulador para ordenar los eventos.

## Descripción en VHDL del sumador de 1bit con retardos

Al modelar un circuito el programador especifica eventos, retardos y concurrencia. Los eventos se especifican con sentencias de asignación de señales. Los retardos se especifican en las sentencias de asignación de señales. La concurrencia se especifica utilizando una sentencia de asignación de señal para cada señal que se quiere modelar. El orden de ejecución de las sentencias depende del flujo de datos y es independiente del orden en el que se han especificado las sentencias de asignación de señales.

En la Figura 1.25 se muestra el cuerpo de la arquitectura para el sumador de 1 bit de la Figura 1.24 cuando se utiliza un modelo de flujo de datos y se especifican los retardos.

Se especifica un retardo para cada tipo de puerta mediante objetos “Constant”. También se especifican en la parte declarativa las cuatro señales internas del circuito (Figura 1.24). La parte correspondiente a la declaración “entity” no se modifica (Figura 1.14).

---

*Trabajo: Edite el fichero S1bit.dvw (“File -> Open”). La ventana emergente permite movernos por el árbol de directorios para seleccionar el fichero. Una vez efectuadas las modificaciones que especifican el retardo almacene el fichero en disco con el nombre S1bitDF (“File -> Save As ...”). Compruebe el funcionamiento lógico del circuito. En primer lugar compile el fichero y posteriormente utilice el fichero de comprobación que ha creado previamente. Para ello copie el contenido del fichero en un fichero denominado S1bitDF.tsv.*

---

## Ejemplo

```

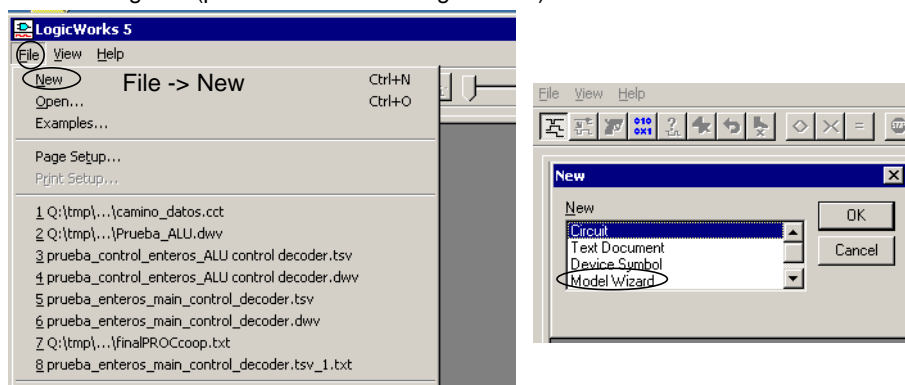
architecture flujodatos of S1bit is
  constant retardoxor: time := 15 ns;
  constant retardoand: time := 10 ns;
  constant retardoor: time := 15 ns;
  signal xorxy, andxy, andxcen, andycen : std_logic;
begin
  xorxy <= x xor y after retardoxor;
  s <= xorxy xor cen after retardoxor;
  andxy <= x and y after retardoand;
  andxcen <= x and cen after retadoand;
  andycen <= y and cen after retadoand;
  csal <= andxy or andxcen or andycen after retardoor;
end flujodatos;

```

**Figura 1.25** Cuerpo de la arquitectura para un sumador de 1 bit especificando retardos y utilizando un modelo de flujo de datos.

## Creación de un elemento para utilizarlo en un circuito

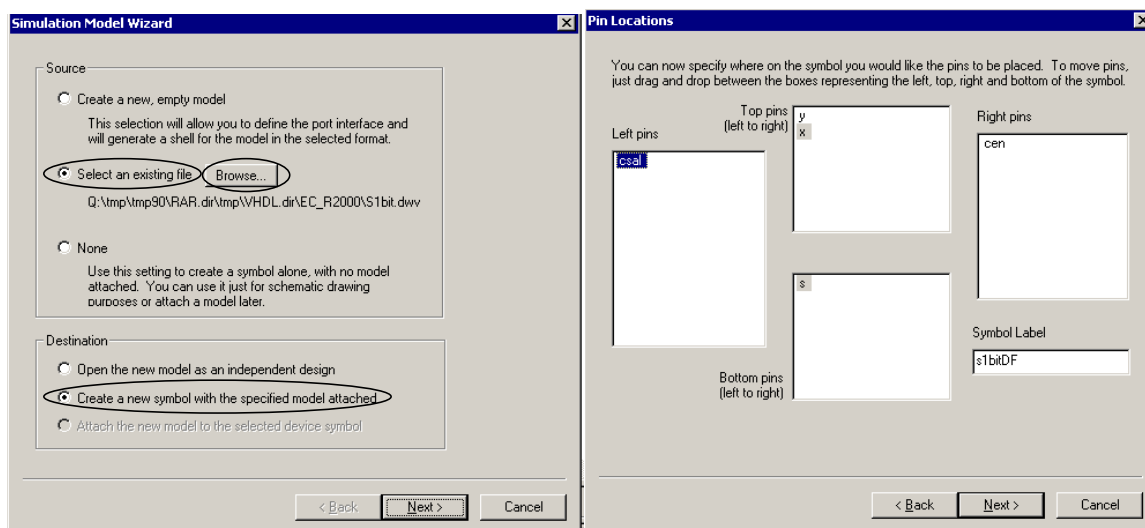
En esta sección se describe cómo crear un elemento que se almacena en una librería para utilizarlo posteriormente en un esquema de un circuito. Para crear el elemento debemos efectuar la acción que se muestra en la parte izquierda de la Figura 1.26. En la ventana emergente (parte derecha de la Figura 1.26) debe seleccionarse “Model Wizard”.



**Figura 1.26** Pasos para crear un elemento de librería.

En la ventana emergente se efectúan las selecciones que se muestran en la parte izquierda de la Figura 1.27. En la sección denominada fuente (“Source”) se selecciona “Select an existing file”. En la sección denominada destino (“Destination”) se selecciona “Create a new symbol ...”. Posteriormente se selecciona el fichero donde se almacena el circuito que se quiere encapsular y almacenar en la librería. Para ello se pulsa en el rótulo

“Browse ..”. La ventana emergente permite desplazarse por el árbol de directorios y seleccionar el fichero VHDL deseado (extensión .dvw) que almacena el circuito que se quiere almacenar en la librería.

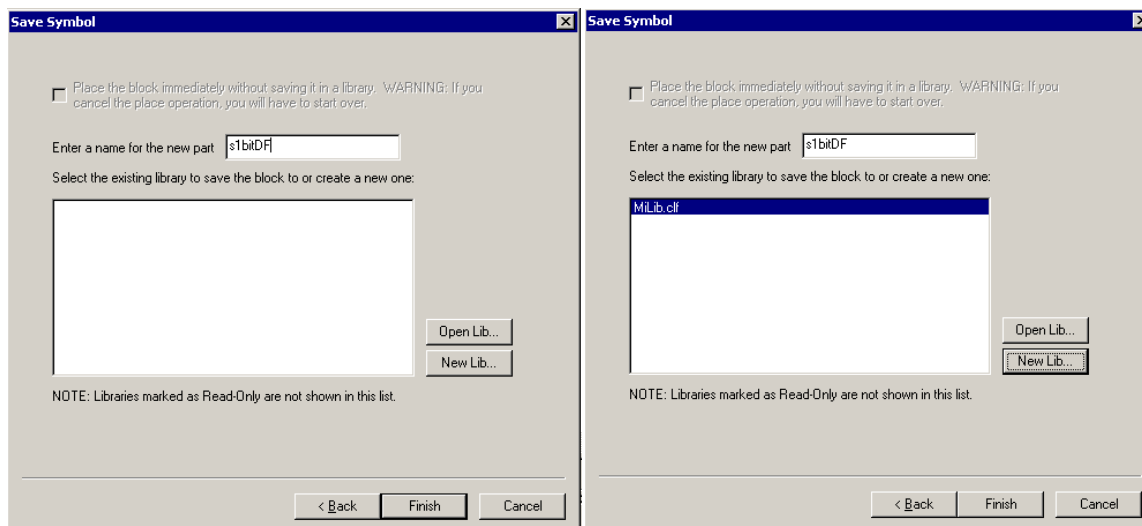


**Figura 1.27** Pasos para crear un elemento en una librería y especificación de la ubicación de los terminales.

El siguiente paso es pulsar el rótulo “Next” y la ventana emergente se utiliza para especificar la ubicación espacial de las patillas en el encapsulado. En esta ventana se identifican los campos (parte derecha de la Figura 1.27): Patillas encima, Patillas debajo, Patillas izquierda, Patillas derecha y Nombre del módulo. El último campo es la etiqueta con que nombraremos al módulo (“Symbol Label”). En esta ventana LogicWorks efectúa una propuesta de ubicación de las patillas en el símbolo que identificará el módulo. Para cambiar la ubicación de una patilla, se selecciona con el botón izquierdo del ratón y se arrastra hasta el lugar deseado (en la parte derecha de la Figura 1.27 se muestra el resultado de editar la propuesta). Notemos que la propuesta de “Symbol Label” es el identificador de la “entity”. Edite el campo y denomine al elemento de librería S1bitDF.

El orden de ubicación de las patillas en el símbolo que identifica el módulo es de arriba a abajo en cada campo de edición.

Una vez editados los distintos campos se posiciona el cursor en el rótulo “Next” y se pulsa el botón izquierdo del ratón. Seguidamente se visualiza una ventana donde se solicita la librería en la cual se quiere almacenar el módulo diseñado (parte derecha de la Figura 1.28). Observe que el nombre propuesto es S1bit, el cual es la etiqueta utilizada después de la palabra “entity” en el fichero relacionado. Edítelo para que sea S1bitDF.



**Figura 1.28** Creación o selección de una librería de componentes para almacenar un circuito encapsulado.

Si no se dispone de librería se ubica el cursor en el rótulo “New Lib..” y se pulsa el botón izquierdo del ratón. En el caso de que ya se disponga de una librería, pero no esté abierta, el rótulo que debe seleccionarse es “Open Lib”. Si la librería está abierta se observa en el recuadro interior.

---

*Nota: Las librerías públicas no deben modificarse. Los módulos que se creen deben almacenarse en librerías propias.*

---

En los dos primeros casos descritos previamente aparece una ventana que permite movernos por el árbol de directorios. Mediante esta ventana nos posicionamos en nuestro directorio y establecemos un nombre para la librería o seleccionamos una librería existente. En la parte derecha de la Figura 1.28 se muestra que la librería creada, seleccionada o abierta es MiLib.clf.

Una vez seleccionada una librería se posiciona el cursor en el rótulo Finish” y se pulsa el botón izquierdo del ratón.

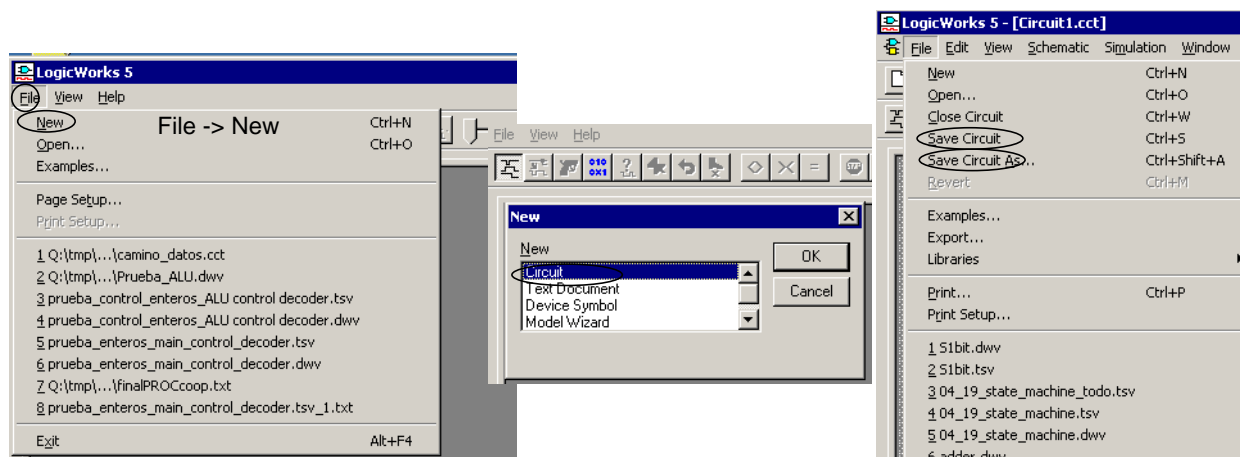
---

*Trabajo: Cree un elemento encapsulado para el sumador de 1 bit con retardo y almacénalo en una librería. Denomine al elemento S1bitDF.*

---

## Herramienta de dibujo (captura) de esquemas

En primer lugar creamos un fichero para ubicar un circuito. Los pasos a seguir se muestran en la parte izquierda de la Figura 1.29. En la ventana que emerge (parte central de la Figura 1.29) se selecciona el rótulo “Circuit”.



**Figura 1.29** Creación de un fichero para construir un circuito mediante captura del esquema. Pasos para almacenar un diseño en un fichero.

Una vez creado el documento damos un nombre al fichero. Para ello se efectúan los pasos mostrados en la parte derecha de la Figura 1.29.

- “Save Circuit”: si ya existe el fichero.
- “Save Circuit As...”: si es la primera vez o se quiere modificar el nombre del fichero. Utilizando la ventana que emerge nos ubicamos en el directorio donde queremos que se almacene el fichero y establecemos el nombre de fichero que queremos. Notemos que la extensión del fichero es “.cct”.

---

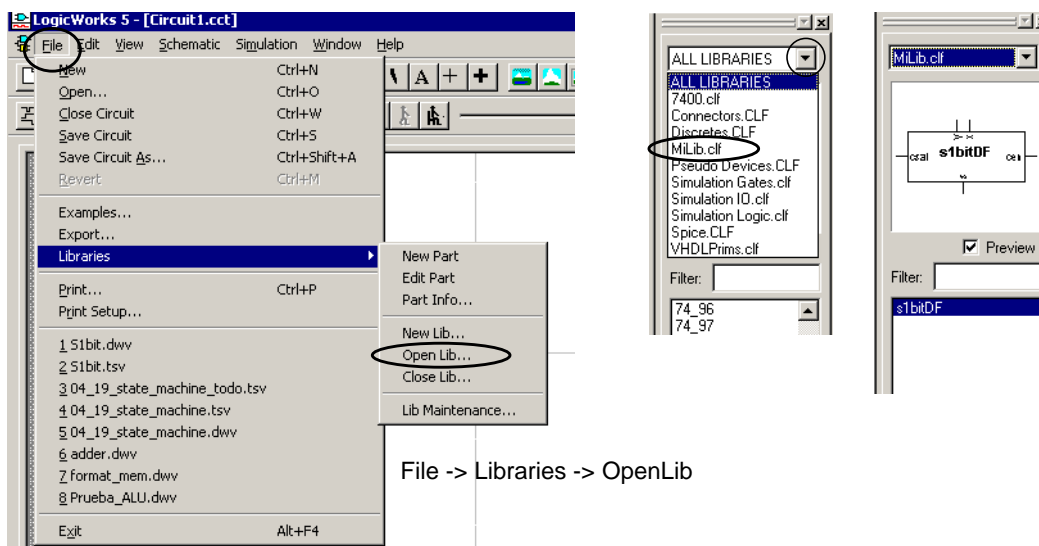
*Trabajo: Establezca un nombre para el fichero. Por ejemplo CircuitoS1bit.cct.*

---

## Elementos para crear un circuito

El sumador se encuentra en la librería *MiLib.clf*. Para utilizar el contenido de la librería es necesario abrir la librería mediante la orden que se muestra en la parte izquierda de la Figura 1.30.

En este caso tenemos una jerarquía de ventanas para efectuar la selección. Cuando se posiciona el cursor en el rótulo que tiene a su derecha el símbolo de la cabeza de una flecha aparece una nueva ventana. Entonces, se mueve el cursor hasta la nueva ventana y se selecciona el rótulo deseado. En la simbología utilizada se observan dos niveles de flechas.



**Figura 1.30** Pasos para abrir una librería de componentes.

Después de dar la orden, se visualiza una ventana que permite movernos en el árbol de directorios hasta el fichero que contiene la librería. Una vez abierta la librería, se visualizan los elementos en la ventana Librería.

*Trabajo: Abra la librería MiLib.clf.*

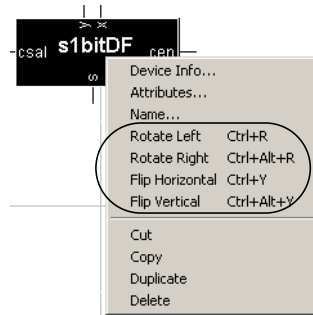
En ocasiones la librería ya está abierta pero se están visualizando todos los elementos de todas las librerías o se están visualizando los elementos de otra librería. En estas condiciones es suficiente posicionar el cursor en la pestaña que hay en la parte superior derecha de la ventana Librería (parte central de la Figura 1.30). Posteriormente se pulsa el botón izquierdo del ratón y emerge una ventana con el nombre de todas las librerías abiertas. Sólo queda seleccionar la librería deseada. Para ello posicione el cursor en ella y libere el botón del ratón. En la parte derecha de la Figura 1.30 se muestra la ventana Librería mostrando únicamente los elementos de la librería seleccionada.

## Selección de un elemento de la librería

Para seleccionar un elemento de la librería se posiciona el cursor sobre el elemento y posteriormente se pulsa el botón izquierdo del ratón.

**Posicionamiento en la ventana de Diseño.** El elemento seleccionado se ubica en la posición deseada de la ventana de diseño moviendo el cursor sin dejar de pulsar el botón izquierdo del ratón. Cuando estemos en la ubicación que nos interesa se libera la pulsación del botón izquierdo del ratón.

**Orientación de un elemento.** Una vez posicionado un elemento su orientación se puede modificar. Para ello se selecciona el elemento posicionando el cursor sobre él y pulsando el botón izquierdo del ratón. Posteriormente se pulsa el botón derecho del ratón y emerge una ventana que se muestra en la Figura 1.31. En esta ventana se muestran 4 opciones para orientar el elemento.



**Figura 1.31** Opciones para modificar la orientación de un elemento.

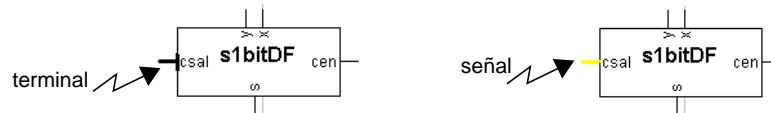
---

**Trabajo:** Posicione el sumador de 1bit en la ventana de diseño.

---

## Patillas

Las entradas y salidas de los elementos se denominan patillas y en una patilla se diferencian dos componentes (Figura 1.32): a) terminal (pin) y b) señal (signal). En la Figura 1.32 se muestra la simbología utilizada por LogicWorks para distinguir los dos componentes. Para seleccionar el terminal de la patilla hay que acercarse lo más posible al elemento y para seleccionar la señal de la patilla hay que alejarse lo más posible.



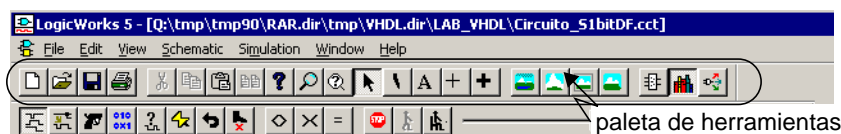
**Figura 1.32** Componentes de una patilla: a) terminal y b) señal.



El símbolo del terminal es una T y para indicar la señal se cambia el color o el tono de la patilla.

## Selección de un elemento o un icono

Los elementos están en la ventana de diseño y los iconos en las paletas de herramientas y simulación (Figura 1.33). Para efectuar una selección hay que posicionar el cursor sobre el elemento o icono y pulsar el botón izquierdo del ratón.



**Figura 1.33** *Paleta de herramientas.*


Para seleccionar varios elementos en la ventana de diseño se pulsa la tecla “*shift*” a la vez que se selecciona un nuevo elemento.

Utilizando el cursor se pueden seleccionar varios elementos en la ventana de diseño delimitando una zona. Para ello se pulsa el botón izquierdo del ratón y se crea un recuadro moviendo el ratón de forma que abarque a los elementos que se quiere seleccionar.

Para liberar la selección se pulsa el botón derecho del ratón.

## Conexión de elementos

Las conexiones entre los elementos de un circuito se pueden efectuar de dos formas:

- Situando el cursor sobre el componente señal de la patilla de un elemento, seleccionándolo y manteniendo pulsado el botón izquierdo del ratón se mueve el cursor en la dirección deseada.
- Utilizando el icono  (*Draw Sig*) que permite dibujar cables (Figura 1.34). Después de seleccionar el icono se posiciona el cursor en la ventana de diseño.

Un cable se puede dibujar aislado de cualquier elemento. Para iniciar el trazado se pulsa el botón izquierdo del ratón y cuando se quiere finalizar el trazado se pulsa dos veces consecutivas el botón izquierdo del ratón.

El trazado de un cable puede efectuarse en tramos que forman ángulo de  $90^\circ$ . Se dibuja un tramo, se deja de pulsar el botón izquierdo del ratón, se mueve el cursor para iniciar el trazado del siguiente tramo y se vuelve a pulsar el botón izquierdo.





**Figura 1.34** Paleta de herramientas. Icono cable.

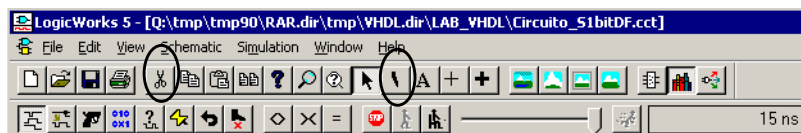
También pueden conectarse patillas de elementos directamente. Se posiciona el cursor en el componente señal de una patilla, se pulsa el botón izquierdo del ratón y se efectúa el trazado hasta la otra patilla que se quiere conectar.

Una forma de conocer los elementos conectados a un cable es seleccionar el cable. Entonces se intensifica la visualización de todas las conexiones o se produce un cambio de color.

## Borrado un elemento

La eliminación de elementos o zonas del circuito se puede efectuar de dos formas:

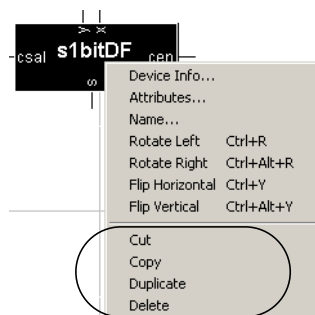
- Cuando se quiere borrar sólo una puerta, un módulo o un trozo de cable se puede utilizar el icono  (Zap, Figura 1.35). Después de seleccionar el icono, el símbolo cursor se convierte en el símbolo seleccionado, se posiciona el símbolo sobre el elemento y se pulsa el botón izquierdo del ratón.
- También puede seleccionarse uno o varios elementos del circuito y posteriormente posicionar el cursor sobre el icono  (Cut, Figura 1.35) y pulsar el botón izquierdo del ratón.



**Figura 1.35** Paleta de herramientas. Iconos cortar y borrar.

## Otras posibilidades y utilidades

Seleccionando un elemento y pulsando el botón derecho del ratón se pueden efectuar algunas de las operaciones comentadas previamente y otras como copia, duplicar, cortar, eliminar, etc (Figura 1.36).



**Figura 1.36** Operaciones accesibles después de seleccionar un elemento y pulsar el botón derecho del ratón.

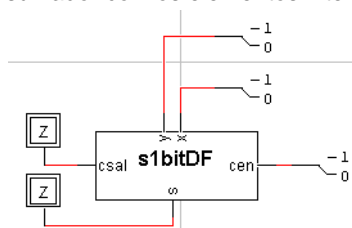
## Señales de entrada y de salida y simulación lógica

### Asignación y visualización de valores lógicos

Una vez diseñado el circuito se comprueba su funcionamiento asignando valores a sus entradas y observando los valores de las salidas. La librería *ac.clf* dispone de dos elementos para este cometido:

- **switch** (interruptor): hace las funciones de interruptor y conectado a una entrada permite establecer el valor 0 o 1 lógico, mediante la pulsación del botón izquierdo del ratón, después de posicionar el cursor sobre el interruptor.
- **probe** (visor): efectúa las funciones de visualizador y conectado a un cable o patilla muestra el valor lógico.

La Figura 1.37 muestra el sumador con los elementos interruptor y visor conectados.



**Figura 1.37** Sumador de 1 bit con elementos de entrada y salida para observar el funcionamiento.


Los valores que se observan en el visor pueden ser:

- 0 o 1 (valores lógicos).

- Z alta impedancia.
- X valor indefinido.
- C indica conflicto y se observa cuando se fuerza un valor por una salida que no se corresponde con el que se genera.




*Trabajo: Abra la librería ac.clf que se encuentra en el directorio de distribución de ficheros comunes de las practicas. Posicione los elementos interruptor y visor en el diseño.*


## Simulación lógica

Para poder obtener y observar los valores de salida es necesario que las señales de entrada se propaguen por las puertas lógicas. Para ello es necesario indicar que se inicie la simulación del circuito mediante el icono  (Run, Figura 1.38). Posicione el cursor en el icono y pulse el botón izquierdo del ratón.

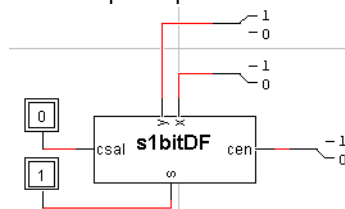


**Figura 1.38** Paleta de simulación. Inicio de la simulación lógica.

La velocidad de la simulación se controla mediante el dial . También, la modificación de la posición del dial, cuando la simulación está parada, inicia directamente la simulación. El icono  se utiliza para pausar la simulación y el icono  se utiliza para salir del modo simulación, o lo que es lo mismo, parar el simulador.

La simulación se suspende automáticamente cuando los valores de todas las señales del circuito son estables. Esto es, no se modifica ninguno de los valores, una vez las entradas del circuito son estables y ha transcurrido el tiempo de propagación de las señales por el circuito. El icono  permite simular paso a paso.

En la Figura 1.39 se muestran los valores de salida después de activar la simulación con los valores de entrada mostrados por la posición de los interruptores.




**Figura 1.39** Valores de salida después de establecer los valores de entrada mostrados y activar la simulación.

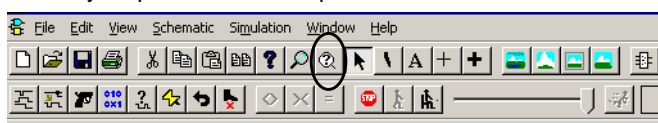
Los valores de las señales de entrada se pueden modificar mediante el elemento interruptor y como resultado se obtienen los nuevos valores en las señales de salida del circuito.

---

*Nota: No modifique los valores de las señales de entrada hasta que se hayan estabilizado las señales de salida con los valores de las señales de entradas anteriores.*

---

**Consulta del valor de un cable o componente señal de una patilla.** La paleta de herramientas dispone del icono  (lupa, Figura 1.40). Una vez seleccionado el icono se posiciona en el cable y se pulsa el botón izquierdo del ratón.



**Figura 1.40** Paleta de herramientas. Icono lupa.

Los valores que se observan son los mismos que con el elemento visor.



---

*Trabajo: Establezca valores en las señales de entrada y salida mediante los interruptores y active la simulación para comprobar el funcionamiento del diseño.*


---

## Utilización de vectores de comprobación

Para verificar el funcionamiento de un esquema de circuito con o sin módulos encapsulados también se puede utilizar un fichero con vectores de comprobación. En principio el esquema del circuito no debería tener ni interruptores ni visores, ya que no se utilizan. Ahora bien, su existencia no es un problema.

- Pare el simulador  .
- Etiquete las señales de entrada y de salida del circuito (se explica en la siguiente sección).
- En el caso que nos ocupa, copie el fichero de vectores de comprobación S1bit.tsv, utilizado previamente, en un fichero cuyo nombre sea idéntico al nombre del fichero que almacena el esquema del circuito, siendo la extensión “.tsv”. Almacene el fichero en el directorio.
- Active la simulación lógica  y en el “TestPanel” se observa el resultado.

Adicionalmente puede observarse la evolución temporal de las salidas con las señales de comprobación en la ventana de Tiempo.

Una vez comprobado el funcionamiento es recomendable modificar el nombre del fichero de comprobación para que no se utilice cada vez que se abra el fichero del esquema de circuito y se active la simulación. El simulador lee el fichero y lo utiliza cada vez que se para el simulador  y posteriormente se activa la simulación.

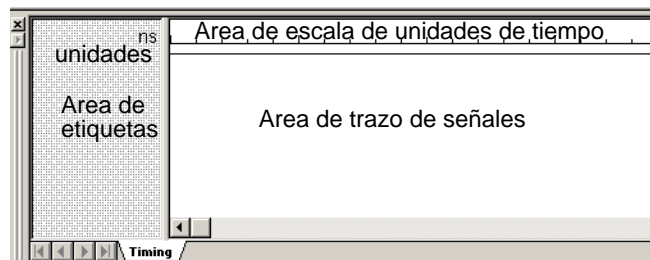
## Medida del tiempo de retardo

LogicWorks puede mostrar al simular la evolución temporal de las señales de un circuito en la ventana de Tiempo. Ello permite observar el retardo de propagación de las señales por el circuito.

El tiempo de retardo de los cables es cero. El tiempo de retardo de las puertas es el establecido en el modelo VHDL.

## Ventana de Tiempo

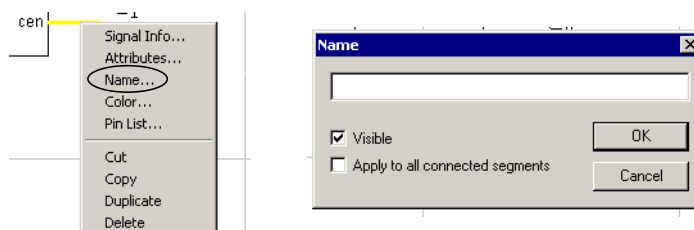
En la ventana de Tiempo se distinguen las áreas que se muestran en la Figura 1.41.



**Figura 1.41** Ventana de tiempo.

## Etiquetas

En primer lugar es necesario etiquetar las señales que se quieren visualizar en la ventana de Tiempo. Para etiquetar un cable o la señal de una patilla se posiciona el cursor y se selecciona el elemento que se quiere etiquetar. Posteriormente se pulsa el botón derecho del ratón. Entonces se visualiza una ventana (parte izquierda de la Figura 1.42) donde debe seleccionarse el rótulo "Name". Seguidamente emerge la ventana que se muestra en la parte derecha de la Figura 1.42 donde se especifica el nombre deseado. Para que la etiqueta se observe en el circuito el rótulo "Visible" debe estar marcado.

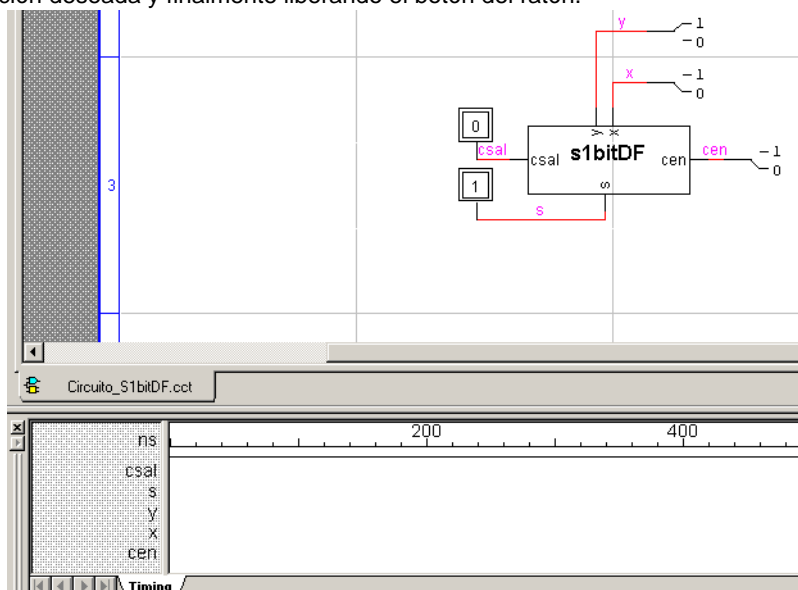


**Figura 1.42** Ventana que permite etiquetar un elemento.

**Modificación de una etiqueta.** Nos posicionamos sobre ella y pulsando dos veces el botón izquierdo del ratón se visualiza la ventana mostrada en la parte derecha de la Figura 1.42. Es suficiente con editar el nombre

En la Figura 1.43 se muestra el sumador con las etiquetas en las señales de entrada y en las señales de salida. Estas etiquetas también se observan en la ventana temporal ubicada en la parte inferior del marco LogicWorks.

El orden vertical de las etiquetas en la ventana de tiempos puede modificarse seleccionando la etiqueta, moviendo el cursor con el botón izquierdo del ratón presionado hasta la posición deseada y finalmente liberando el botón del ratón.



**Figura 1.43** Sumador de 1 bit con las señales de entrada y salida etiquetadas. Observación de las etiquetas en la ventana de tiempos.

**Trabajo:** Etiquete las señales de entrada y salida del circuito.




## Funcionalidades en la ventana de Tiempo

En la paleta de simulación existen algunas opciones de utilidad para observar el detalle de la evolución temporal de las señales (Figura 1.44).

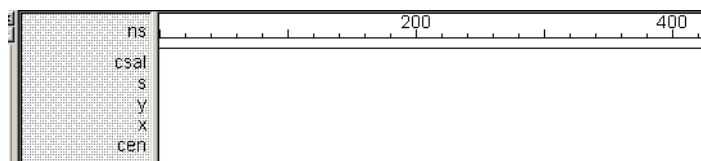


**Figura 1.44** Paleta de simulación.


### Modificación de la escala de unidades de tiempos.

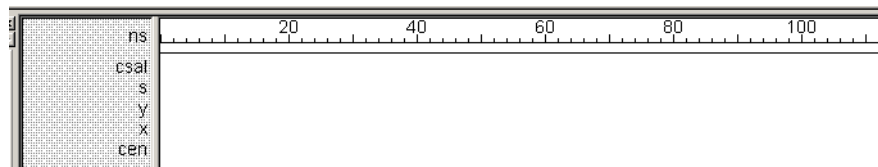
- Pulsando el botón izquierdo del ratón sobre el icono  disminuye la resolución. Se modifica la escala de representación y en menos espacio se representa el mismo tiempo.
- Pulsando el botón izquierdo del ratón sobre el icono  aumenta la resolución. En más espacio se representa el mismo tiempo.
- Pulsando el botón izquierdo del ratón sobre el icono  se reestablece la escala de representación por defecto.

La ventana de Tiempo muestra, por defecto, la escala de tiempo que se observa en la Figura 1.45.




**Figura 1.45** Escala de tiempo por defecto en la ventana de Tiempo.

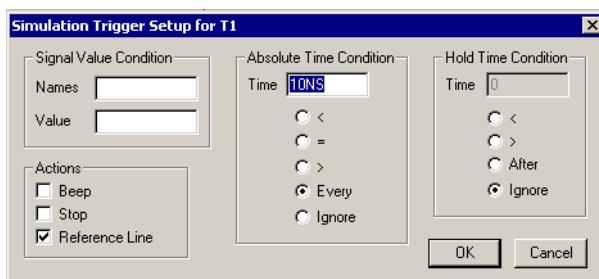
Como conocemos que el retardo de una puerta son 10 ns y tenemos 2 niveles de puerta, interesa aumentar la resolución de la escala de tiempo para observar mejor la evolución de las señales. Entonces aumentamos la resolución de la escala de tiempo posicionando el ratón en el icono  y pulsando 2 veces sucesivas con el botón izquierdo. Ahora el aspecto de la ventana de Tiempo es el que se observa en la Figura 1.46.



**Figura 1.46** Escala de tiempos expandida.

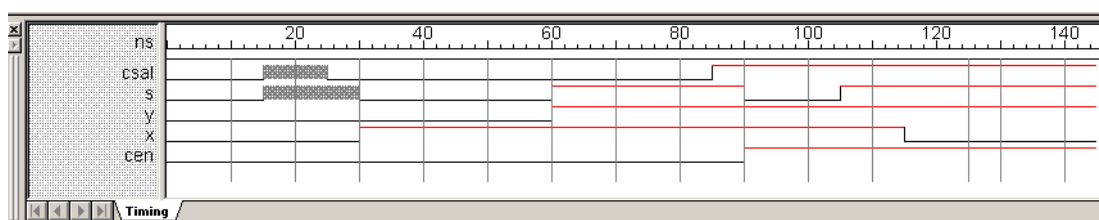


**Marcas en la ventana de Tiempo.** Su función es facilitar la asociación de cambios lógicos en las señales con la escala de unidades de tiempo. Para ello se pulsa con el botón izquierdo del ratón sobre el icono  en la paleta de simulación. Entonces aparece la ventana que se muestra en la Figura 1.47, donde se ha marcado el campo "Every", se han indicado 10 ns y se indica que se quieren líneas verticales de referencia ("Reference Line").



**Figura 1.47** *Ventana para establecer marcas en la ventana de tiempos*

Quando se active la simulación se visualizan en la ventana de Tiempo las líneas de referencia que se observan en la Figura 1.48.



**Figura 1.48** Ventana de Tiempo donde se observan las referencias establecidas.

*Trabajo: Establezca la escala apropiada y las líneas de referencia.*





## Simulación temporal del circuito


Para determinar experimentalmente el retardo del circuito es necesario seguir el proceso de simulación que se describe a continuación.

**Preparación del circuito.** Una forma de garantizar, que al modificar el valor de una señal de entrada, se observe el retardo de propagación de los elementos del circuito, es establecer previamente el valor indefinido en los cables del circuito.

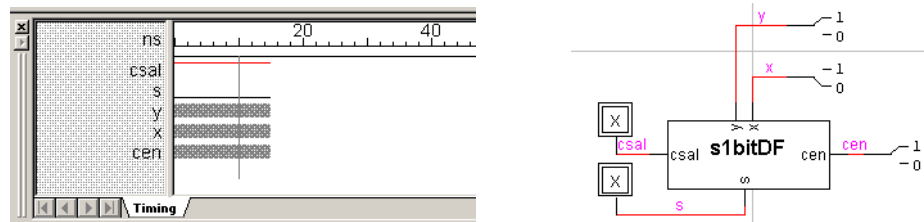
*Nota: Una puerta AND-2 que tiene una entrada con el valor lógico cero no muestra un retardo de propagación si la otra entrada modifica su valor lógico. La situación es idéntica con una puerta OR-2 que tiene una de sus entradas con el valor lógico 1.*

Los pasos que deben efectuarse son los siguientes:

- 1 Selección del icono . Se pausa la simulación  
El icono  puede utilizarse para establecer valores en un cable; por ejemplo 0, 1, x. Una vez se ha posicionando el icono en el cable y manteniendo pulsado el botón izquierdo del ratón se pulsa la tecla correspondiente del teclado. El valor pulsado (en este caso x) no se visualiza inmediatamente. Debe esperarse a que se inicie la simulación.
- 2 Asignación del valor indefinido (x) a todas las entradas.
- 3 Activación de la simulación mediante el icono  o el dial que permite establecer una velocidad de simulación.
- 4 Cuando se estabilizan las señales de salida, la simulación se suspende ya que no se modifican los valores de salida y las entradas permanecen estables. Los valores de las señales se observan en los visores. Entonces se debe pausar el proceso de simulación mediante el icono .

En este instante de tiempo, todas las señales de entrada y salida como cualquier punto intermedio del circuito (señales internas) han tomado el valor lógico indefinido (x). Puede comprobarse mediante el icono .


La ventana de Tiempo muestra la siguiente evolución temporal de las señales cuando en las señales de entrada del sumador se establece el valor x y se activa la simulación (parte izquierda de la Figura 1.49).



**Figura 1.49** Evolución temporal de las señales cuando las entradas del sumador toman el valor indefinido (x) y se activa la simulación.

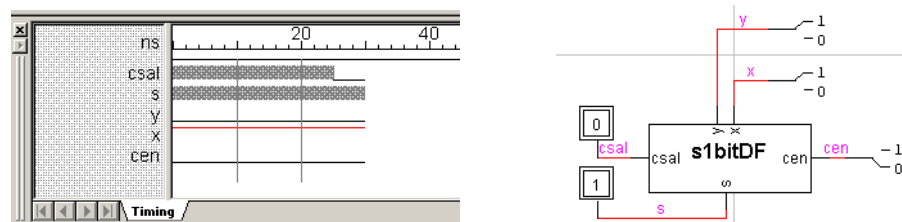
Cuando se establece el valor indefinido (x) en las señales x, y, cen y se activa la simulación, las señales s y csal mantienen los valores de la simulación anterior. Transcurrido un tiempo después de activar la simulación, las señales s y csal toman el valor indefinido




- 7 Activación de la simulación mediante el icono  o mediante el dial que permite controlar la velocidad.

La simulación se suspende cuando las señales de entrada y salida permanecen estables. Ahora todas las señales del circuito tienen un valor lógico que depende de los valores lógicos de las entradas.

En la parte izquierda de la Figura 1.51 se muestra el resultado que se obtiene en la ventana de tiempo cuando se activa la simulación y el circuito simulado es el sumador de 1 bit.



**Figura 1.51** Evolución temporal de las señales cuando se activa la simulación y se han establecido los valores lógicos  $x=1$ ,  $y=0$  y  $cen=0$ .

Las entradas tienen los valores  $x=1$ ,  $y=0$ ,  $cen=0$ . Las señales  $s$  y  $csal$  mantienen el valor indefinido hasta el instante 30 ns y 25 ns respectivamente, a partir del cual toman los valores  $s=1$  y  $csal=0$ . Entonces, el retardo son 30 ns. Para visualizar el valor de la señal  $s$  debe utilizarse el elemento visor (parte derecha de la Figura 1.51) o el icono .

---

**Trabajo:** Establezca valores en las señales de entrada y active la simulación.

---

---

**Nota:** Una vez los valores de las señales de salida son estables, LogicWorks suspende la simulación si no se modifica algún valor de una de las señales de entrada. Entonces, si se simula dos veces consecutivas con la misma entrada, la segunda vez se obtiene un retardo nulo. Si se modifica algún valor de las señales de entrada el retardo observado puede depender de los valores previos de las señales de entrada.

---

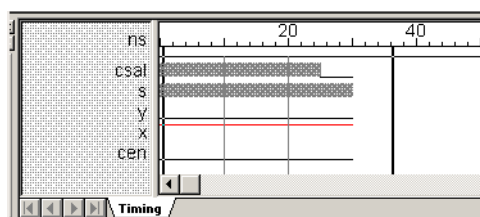
## Observación precisa de los retardos

Discriminar retardos en la ventana de Tiempo es engorroso y puede dar lugar a errores de medición. Por ello utilizaremos la capacidad de LogicWorks para obtener una descripción ASCII de la información que se observa en la ventana de Tiempo. Cuando es una señal, los valores mostrados son 0 y 1. Cuando es un grupo de señales o un bus los valores se muestran en hexadecimal.

Seguidamente se indica cómo delimitar un intervalo de tiempo en la ventana de tiempo y obtener valores numéricos.

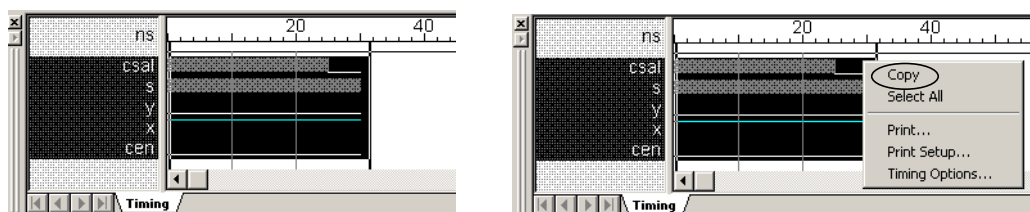
- Se posiciona el cursor en la ventana de Tiempo en el inicio del intervalo de interés y se pulsa el botón izquierdo del ratón.
- Se desplaza el cursor en la dirección horizontal hasta el final del intervalo de interés.

Al finalizar el proceso se observan las marcas mostradas en la Figura 1.52.



**Figura 1.52** Marcas que se observan en la ventana de tiempo al establecer un intervalo mediante el cursor.

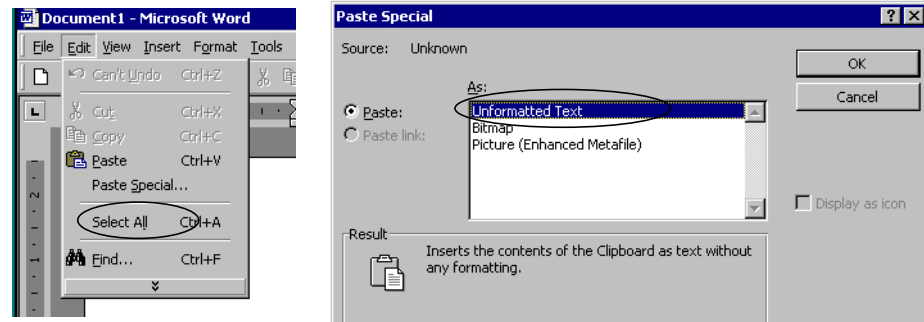
- **Selección de etiquetas.** Una vez seleccionado el intervalo, hay que posicionar el cursor en el área de etiquetas de la ventana de tiempo. Entonces se posiciona el cursor en el símbolo deseado y se pulsa el botón izquierdo del ratón para seleccionar o anular la selección de una etiqueta (parte izquierda de la Figura 1.53). Si se quiere seleccionar más de una etiqueta se pulsa la tecla “shift” después seleccionar la primera etiqueta y antes de seleccionar las siguientes.



**Figura 1.53** Selección de las señales en la ventana de Tiempo. Copia de las señales e intervalo seleccionado.

- Posicionamos el cursor en la parte de la ventana de Tiempo donde se visualiza la evolución de las señales. Se pulsa el botón derecho del ratón y aparece la ventana que se muestra en la parte derecha de la Figura 1.53, donde debe seleccionarse el rótulo “Copy”.
- En el caso de querer seleccionar todas las señales y todo el intervalo de simulación, seleccione “Select All” en la ventana que se muestra en la Figura 1.53.

- Se abre un fichero Word y se utiliza la opción “Paste Special” (parte izquierda de la Figura 1.54). En la pantalla que se visualiza posteriormente se selecciona el formato “Unformatted Text” (parte derecha de la Figura 1.54).



**Figura 1.54** Acciones que hay que efectuar en un fichero Word para copiar la información de las señales seleccionadas de la ventana de Tiempo.

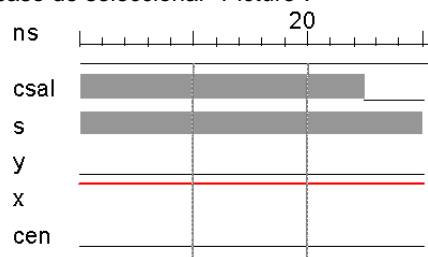
La Figura 1.55 muestra el resultado de la acción.

\$T	\$D	\$O	<u>csal</u>	\$O	s	\$O	y	\$O	x	\$O	<u>cen</u>
0	25NS	X	X	0	1	0					
25NS	5NS	0	X	0	1	0					
30NS	0	0	1	0	1	0					

**Figura 1.55** Valores que se observan en un fichero Word después de copiar la información seleccionada en la ventana de Tiempo.

Las dos primeras columnas de la Figura 1.55 identifican el inicio de un intervalo de tiempo (\$T) y su duración (\$D). Cada etiqueta de la ventana de tiempos aparece precedida de los símbolos \$O. Notemos la deficiencia en el encolumnado.



Para ayudar en la interpretación de los valores numéricos puede visualizarse también en el fichero Word la traza de las señales. Para ello seleccione la opción “Paste Special” (Figura 1.54) y en la ventana que se visualiza posteriormente se selecciona el formato “Bitmap” o “Picture” (parte derecha de la Figura 1.54). La Figura 1.56 muestra el resultado de la acción en el caso de seleccionar “Picture”.



**Figura 1.56** Selección de la opción “Picture”. Trazo de las señales seleccionadas en la ventana de tiempo.



que almacena el esquema del circuito, siendo la extensión “.tsv”. Almacene el fichero en el directorio. En la parte izquierda de la Figura 1.58 se muestra el contenido del fichero de estimulación. Se comprueban todas las posibles combinaciones de entrada.

- Abra el fichero que contiene el esquema de circuito del sumador de 1 bit.
- Pare el simulador .
- Active la simulación lógica  y en la ventana de tiempos se observa el resultado.

Por defecto, el simulador estimula las entradas con un nuevo vector de comprobación cuando todas las señales del circuito son estables.

En la parte derecha Figura 1.58 se muestra parte de la salida ASCII cuando se utiliza el fichero de estimulación indicado previamente. En la Figura 1.59 se muestra la traza de las señales para todas las posibles combinaciones de entrada.

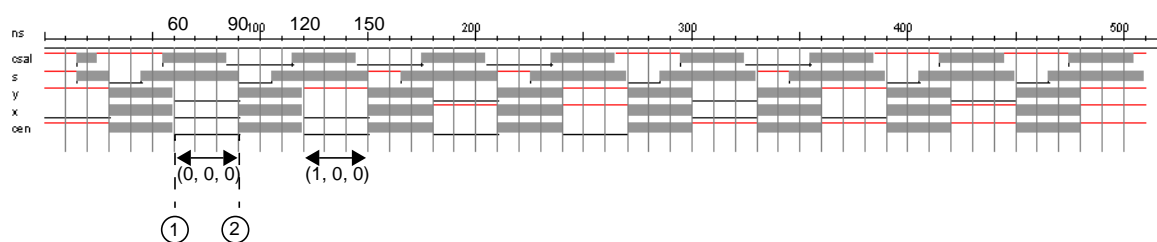
En la tabla de la Figura 1.58 y en la Figura 1.59 se observan en primer lugar los valores correspondientes a un vector de comprobación (y =1, x =0 y cen =1). Estos son los valores que estaban establecidos en las señales de entrada al activar la simulación. Las salidas correspondientes a estos valores son estables al cabo de 30 ns. Posteriormente se observan los resultados correspondientes al fichero de estímulos.

\$I cen	\$I x	\$I y	\$T	\$D	\$O csal	\$O s	\$O y	\$O x	\$O cen
.X	.X	.X	0	15NS	1	1	1	0	1
0	0	0	15NS	10NS	X	X	1	0	1
.X	.X	.X	25NS	5NS	1	X	1	0	1
0	0	1	30NS	15NS	1	0	X	X	X
.X	.X	.X	45NS	10NS	1	X	X	X	X
0	1	0	55NS	5NS	X	X	X	X	X
.X	.X	.X	60NS	25NS	X	X	0	0	0
0	1	1	85NS	5NS	0	X	0	0	0
.X	.X	.X	90NS	15NS	0	0	X	X	X
1	0	0	105NS	10NS	0	X	X	X	X
.X	.X	.X	115NS	5NS	X	X	X	X	X
1	0	1	120NS	25NS	X	X	1	0	0
.X	.X	.X	145NS	5NS	0	X	1	0	0
1	1	0	150NS	15NS	0	1	X	X	X
.X	.X	.X							
1	1	1							

**Figura 1.58** Contenido del fichero de estimulación y Parte del resultado de la simulación en ASCII.



El primer vector de comprobación de interés (0, 0, 0) estimula las entradas en  $t = 60$  ns (marca 1 en la Figura 1.59). Las salidas se estabilizan en  $t = 90$  ns (marca 2 en la Figura 1.59). Entonces el retardo son 30 ns. El vector de comprobación (1, 0, 0) estimula las entradas en  $t = 120$  ns y el retardo para estabilizar las salidas son también 30 ns. Observemos cómo entre vectores de comprobación de interés se inyectan estímulos para indefinir las señales en el circuito. Por ejemplo, en  $t = 30$  ns el vector de estímulos es (x, x, x) y las salidas quedan indefinidas después de 25 ns (Figura 1.58).



**Figura 1.59** Traza observada en la ventana temporal al utilizar un fichero de estimulación con todas las combinaciones posibles de las señales de entrada.



## Apéndice 1.1: Entorno de simulación en LogicWorks

### Directorio de trabajo

Para establecer el directorio de trabajo siga los siguientes pasos: a) Posicione el cursor en el rótulo “Help” ubicado en la parte izquierda superior de la ventana de LogicWorks; b) al pulsar el botón izquierdo emerge una ventana con rótulos; c) pulse con el botón izquierdo la orden “Root Folder...”; d) seleccione el directorio en la ventana emergente (Figura 1.60).Tenga en cuenta que la modificación surge efecto cuando abra de nuevo la aplicación.

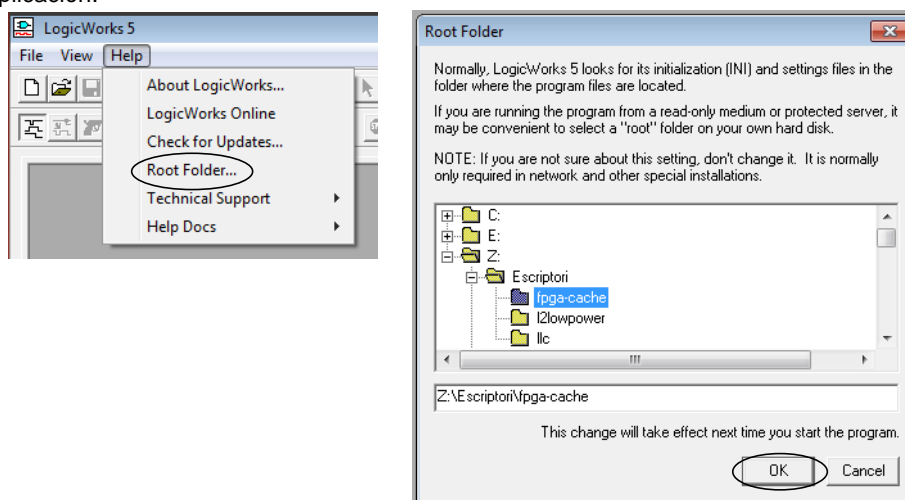


Figura 1.60 Establecimiento del directorio de trabajo

### Fichero de configuración

LogicWorks permite establecer distintas opciones de configuración mediante un fichero de texto denominado lw.ini ubicado en el directorio de trabajo.

- Las opciones se agrupan en secciones. Una sección se identifica mediante una cabecera contenida entre corchetes, por ejemplo: [System].
- Dentro de una sección, cada línea no vacía es una sentencia o un comentario.
- Un comentario es una línea de texto que empieza con los caracteres “//”.
- Las sentencias se escriben de la siguiente forma:

Sintaxis	Ejemplo
palabra_clave = valor	Font = “Courier New” 10 Bold

**Opciones de la sección [Libraries].** Para abrir automáticamente todas las librerías ubicadas en un directorio, la sección [Libraries] debe contener la sentencia:

LIBRARYFOLDER = directory\_path

Los caminos de directorio o fichero son relativos al directorio de trabajo. Tenga en cuenta que no se exploran más niveles de directorios por debajo del especificado.

Para abrir automáticamente una librería en particular se utiliza la sentencia:

LIBRARY = library\_path

En el fichero de configuración también se puede especificar el directorio donde se encuentran las librerías mediante la sentencia:

FOLDER = directory\_path

Ejemplos	
LIBRARYFOLDER = .	abre todas las librerías ubicadas en el directorio de trabajo
LIBRARYFOLDER = milibs	abre todas las librerías ubicadas en el directorio milibs
LIBRARY = .\milibs\lib\lib1.clf	abre la librería lib1 ubicada en el directorio milibs
FOLDER = .	
LIBRARY = lib2.clf	abre la librería lib2 del directorio de trabajo