

# PRÁCTICA 1. Trabajo final

## Análisis del tiempo de propagación de circuitos combinacionales y síntesis en suma de minterms de dispositivos sencillos (Xor y Half-adder)

### Objetivos que se deben alcanzar al realizar la práctica

Después de realizar esta práctica, además de haber mejorado el nivel de consecución de los objetivos necesarios para preparar la práctica (citados en la tabla anterior) y de los objetivos de la práctica anterior relativos al manejo del programa LogicWorks, el alumno será capaz de:

- 1) Analizar el comportamiento temporal de las puertas lógicas y de circuitos o dispositivos combinacionales contruidos con puertas, usando para ello el simulador LogicWorks.

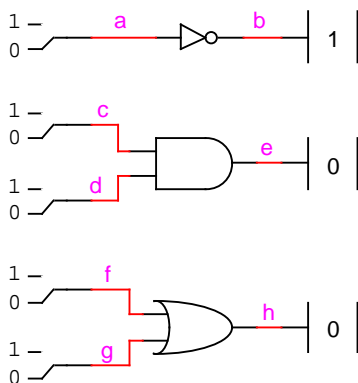
### Directorio de la práctica

Mi PC\\I:\ic\Prac1



## 1.1 Las puertas lógicas básicas. Comportamiento temporal



En el trabajo previo de esta práctica os dimos el resultado del análisis temporal de las puestas Not-1 y Or-2 de nuestra librería. En esta sección vais a obtener vosotros los tiempos de propagación de las puertas Not-1, And-2 y Or-2 de nuestra librería (LibPrac1), para lo que teneos que seguir los siguientes pasos:

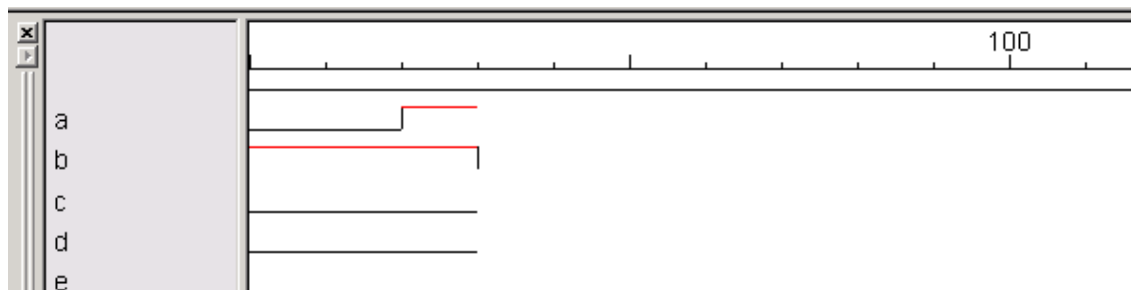
- 1 Como en todas las sesiones de laboratorio lo primero es encender el PC y copiar la carpeta de la práctica (Prac1) del disco donde se encuentra (que es común y está protegido para escritura) a nuestro escritorio. Ahora podemos abrir el LogicWorks y abrir la librería de dispositivos de esta práctica que se encuentra dentro de la carpeta Prac1 y que se denomina LibPrac1. Estos pasos ya los conocemos de la práctica anterior.
- 2 Ahora vamos a crear un circuito muy simple sobre la hoja en blanco de diseño de circuitos que aparece en la pantalla de LogicWorks. Este circuito tiene las puertas Not-1, And-2 y Or-2 sin conectar entre sí. Cada puerta tiene un Binary Switch en cada entrada y un Binary Probe en la salida. Esto ya lo hicisteis en la práctica anterior y debe tener el aspecto de la figura. Fijaos que como hemos usado un mismo circuito para poner las tres puertas hemos dado nombres distintos a todas las señales. En LogicWorks, si en un circuito hay dos señales que se llaman igual es como si esas dos señales estuvieran unidas por un cable (tendrán siempre el mismo valor lógico). Y eso no es lo que queremos aquí.



- 3 Ahora vamos a trabajar con la ventana de tiempos del simulador LogicWorks. Es la ventana horizontal que aparece en la parte inferior de la pantalla y sirve para visualizar la evolución en el tiempo de las señales del circuito (cronogramas). Esta ventana aparece abierta por defecto, pero si no lo está la abrimos yendo a la opción del menú View → Timing Window (o haciendo clic en el

icono Show/hide timing  de la barra de herramientas). En la ventana de tiempos aparece una fila por cada señal (cable) a visualizar, con el nombre de la señal a la izquierda y el cronograma a la derecha. Si no se ven todas las señales, la ventana de tiempo se puede hacer más grande empujando con el cursor hacia arriba una vez seleccionado el marco superior de la ventana de tiempos. Por defecto, está activada la opción de menú Simulation → Add Automatically que hace que cada señal que tiene nombre en el circuito aparezca en la ventana de tiempo. Para borrar señales de la ventana de tiempo hay que hacer clic con el botón derecho sobre el nombre de la señal en la ventana de tiempo para que aparezca un menú desplegable y seleccionar Remove. Por otro lado, si la opción View → Timing Window no está activada y queremos visualizar una señal concreta tenemos que hacer clic sobre la señal para que se ilumine de amarillo y seguidamente hacer clic en el botón Add signals to timing (icono ) o a través del menú Simulation → Add to timing.

- 4 Vamos a inicializar la simulación para borrar los cronogramas referentes a lo que hemos simulado anteriormente y ver con más claridad lo que queremos hacer ahora. Para ello hacemos clic sobre el icono  de la barra de herramientas que inicializa la simulación (*reset*). Ahora ya podemos ir dando valores a las señales de entrada e ir midiendo el tiempo que tardan en propagarse los cambios de las entradas a la salida. Por cada cambio que producimos en una entrada haciendo clic sobre el Binary Switch, el simulador dibuja un trozo de cronograma, justo hasta que el cambio se ha propagado a todas las señales del circuito (en este caso hasta la señal de salida de la puerta). Observad con cuidado el valor final de la salida de la puerta después de cada cambio a la entrada, ya que si, por ejemplo, la salida pasa de 0 a 1 como consecuencia de un cambio en la entrada, sólo vemos, al final del cronograma de la salida, la línea vertical que pasa de 0 a 1, pero no vemos ningún trozo de línea con valor 1. Este trozo de línea lo vemos al siguiente cambio de alguna entrada. Para medir mejor los tiempos de propagación aumentad el zoom de la pantalla del tiempo. Para cambiar el zoom de esta pantalla y poder apreciar mejor los tiempos utilizad los iconos “Zoom in”, “Zoom out” y “Normal size” . Inicialmente, una marca de tiempo (en la regla de encima de los cronogramas) marca 25 unidades de tiempo (u.t.). Esto es cuando está en “Normal size”. Dados los tiempos de propagación de nuestras puertas, es conveniente aumentar la escala para que cada marca de la regla corresponda a 10 u.t. Pasaremos a la escala deseada haciendo clic una vez en el botón “Zoom in”. La siguiente figura muestra el cronograma de LogicWorks al pasar de 0 a 1 el switch de la entrada de la Not-1.



- 5 Si queremos producir un cambio de valor de las dos entradas –o en general sobre varias entradas– a la vez (en el mismo instante de tiempo) tenemos que:
  - parar la simulación (hacer clic en el botón de la persona sentada),
  - cambiar el valor de las entradas haciendo clic en los Binary Switch y
  - volver a poner a correr la simulación (hacer clic en el botón de la persona corriendo).

Ocurren dos cosas curiosas:

- a) al hacer clic sobre un Binary Switch cuando la simulación está parada, el cambio de conexión (de valor) del Binary Switch no se visualiza hasta que ponemos a correr la simulación y
- b) si seguidamente hacemos un segundo clic sobre el mismo Binary Switch o sobre otro, aparece una pantalla que nos advierte de que “los switches no operan cuando el simulador está parado” y nos pregunta si queremos que se ponga en marcha la simulación en este momento. Si queremos seguir cambiando el valor de ese u otros Binary Switch antes de volver a iniciar la simulación, tenemos que hacer clic en el botón de cancelar que aparece en la ventana. ¡Ya iniciaremos la simulación cuando hayamos cambiado todos los Binary Switch, haciendo clic en la persona corriendo!

Vamos a analizar exhaustivamente el comportamiento temporal de la puerta And-2 usando la misma notación y el mismo tipo de tablas que usamos en el trabajo previo para la Not-1 y Or-2. A modo de recordatorio se indica a continuación lo que ya vimos en el trabajo previo.

Denominamos  $t$  al instante de tiempo en que se produce un cambio en el valor de la entrada "a" de la puerta Not-1. La primera y la segunda columna de la siguiente tabla muestran el valor de la entrada un instante antes ( $t-\Delta t$ ) y un instante después ( $t+\Delta t$ ) de producirse el cambio, respectivamente. Llamamos por  $d$  (*delay*) al número de unidades de tiempo tales que la salida se estabiliza al valor correcto en el instante  $t+d$ .

Para la puerta Not-1  $d$  vale 10 u.t. ya que la salida "b" cambia de 1 a 0 pasadas 10 unidades de tiempo desde que la entrada ha pasado de 0 a 1. Para el cambio de la entrada de 1 a 0,  $d$  también vale 10. Esto se muestra en la siguiente tabla.

Valor de la entrada "a" en $t-\Delta t$	Valor de la entrada "a" en $t+\Delta t$	$d$ ( <i>delay</i> )
0	1	10
1	0	10

Para la puerta Or-2 os dimos la siguiente tabla:

Valor de las entradas "f" y "g", (f, g) en $t-\Delta t$	Valor de las entradas "f" y "g", (f, g) en $t+\Delta t$	$d$ ( <i>delay</i> )
(0, 0)	(0, 1)	20
	(1, 0)	20
	(1, 1)	20
(0, 1)	(0, 0)	20
	(1, 0)	0 (n.c.)
	(1, 1)	0 (n.c.)
(1, 0)	(0, 0)	20
	(0, 1)	0 (n.c.)
	(1, 1)	0 (n.c.)
(1, 1)	(0, 0)	20
	(0, 1)	0 (n.c.)
	(1, 0)	0 (n.c.)

## □ Informe final

### Pregunta 1

Analizad, usando los cronogramas de LogicWorks los tiempos de retardo de la puerta And-2. En este caso, algunos cambios en las entradas no producen cambio en la salida. Explicad por qué ocurre esto. En estos casos poned un 0 en la columna  $d$  y al lado, entre paréntesis (n.c.): no cambio. A la vista de los cronogramas rellenad la siguiente tabla.

Valor de las entradas "c" y "d", (c, d), en $t-\Delta t$	Valor de las entradas "c" y "d", (c, d), en $t+\Delta t$	$d$ ( <i>delay</i> )
(0, 0)	(0, 1)	
	(1, 0)	
	(1, 1)	
(0, 1)	(0, 0)	
	(1, 0)	
	(1, 1)	
(1, 0)	(0, 0)	
	(0, 1)	
	(1, 1)	
(1, 1)	(0, 0)	
	(0, 1)	
	(1, 0)	

Las conclusiones del análisis temporal de las puertas Not-1 y Or-2 ya las dimos en el trabajo previo, pero las repetimos aquí considerando las tres puertas básicas Not-1, And-2 y Or-2, dada su importancia:

1. Cuando un cambio en una entrada produce un cambio en la salida, el tiempo que tarda en propagarse el cambio a la salida es el mismo si la salida pasa de 0 a 1 que si pasa de 1 a 0. Por ello, cuando hablemos de **tiempo de propagación (Tp) desde una entrada a la salida** no distinguiremos entre los dos casos posibles, a diferencia de lo que ocurre en el mundo real.
2. El comportamiento temporal (y lógico) que se produce en la salida a consecuencia de un cambio en una entrada depende del valor lógico de la otra entrada (caso de la And-2 y Or-2). Por ejemplo, en la Or-2 cuando la entrada "f" pasa de 0 a 1
  - si la entrada "g" vale 1 no se produce cambio en la salida y podríamos decir que en este caso el tiempo de propagación es 0, mientras que
  - si la entrada "g" vale 0 sí que se produce cambio en la salida y el tiempo de propagación es de 20 u.t.

A pesar de ello, cuando hablemos de **tiempo de propagación desde una entrada a la salida (Tp<sub>f-h</sub>, Tp<sub>g-h</sub>)** no tendremos en cuenta las diferencias según el valor de la otra entrada, sino que consideraremos el caso peor, en el que la otra entrada tiene el valor que hace que se produzca un cambio en la salida.
3. En las puertas And-2 y Or-2 el tiempo de propagación desde una entrada a la salida es el mismo para cualquiera de las dos entradas. Por ello, cuando nos refiramos a una de estas puertas no diferenciaremos entre tiempo de propagación desde la entrada x o desde la entrada y, sino que hablaremos de **tiempo de propagación de la puerta (Tp)**.

#### □ Informe final

##### Pregunta 2

¿Cuál es el tiempo de propagación de cada una de las tres puertas básicas de nuestra librería?


$$Tp(\text{Not-1}) = \quad \quad \quad Tp(\text{And-2}) = \quad \quad \quad Tp(\text{Or-2}) =$$

Recordad los tiempos de propagación de estas tres puertas básicas, ya que usaremos estas puertas en el resto de circuitos que diseñaremos y estos tiempos influyen directamente el tiempo de propagación de los circuitos más complejos.

## 1.2 La puerta Xor de 2 entradas (Xor-2)

Ahora vamos implementar en LogicWorks el circuito combinacional de la puerta Xor de dos entradas usando el diseño en suma de minterms que hicisteis en el trabajo previo. Para ello usad las puertas Not-1, And-2 y Or-2 de nuestra librería.

### 1.2.1 Especificación

Xor-2		<table><tr><th>x</th><th>y</th><th>w</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	w	0	0	0	0	1	1	1	0	1	1	1	0	$w = x \oplus y$
x	y	w																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

### 1.2.2 Síntesis del circuito en suma de minterms

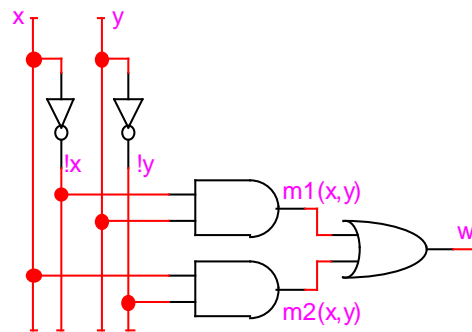


Fig. Esquema lógico interno de la puerta Xor-2 en suma de minterms.

### 1.2.3 Implementación del circuito en LogicWorks

Para implementar el circuito seguid las indicaciones de edición de circuitos que hemos comentado en esta práctica y en la anterior (obtención de dispositivos de la librería, conexión de cables, dar nombre a los cables...). Os será útil recordar lo siguiente:

- 1) Como podéis ver en el circuito anterior, las señales  $x$  e  $y$  vienen dadas por dos cables verticales. Para crear estos cables utilizad el icono de la barra de herramientas. Una vez hayáis hecho clic sobre él, el cursor se convertirá en una cruz. El primer punto donde hagáis clic será el punto de inicio del cable, moviendo el ratón podéis estirar el cable, y para terminarlo haced doble clic sobre el punto donde queráis que acabe.
- 2) La forma de dar nombres a los cables es la misma que ya se ha comentado, y para unir patillas de salida-entrada de dos dispositivos, o para unir una patilla de un dispositivo a un cable, simplemente haced clic en el extremo de la patilla, arrastrad el ratón para estirar el cable y soltad el botón cuando veáis que se unen. Si lo habéis hecho bien, en la unión de dos cables perpendiculares cuando uno de ellos continúa, aparecerá un punto rojo (no aparece punto en un cable que hace un ángulo de 90 grados).
- 3) Seguid el siguiente orden a la hora de implementar el circuito:
  - Primero dibujad los cables verticales para las señales  $x$  e  $y$ .
  - Después colocad las puertas Not conectadas a los cables verticales.
  - Estirad las salidas de las Not para crear los cables verticales de las señales  $\neg x$  e  $\neg y$ .
  - Seguidamente colocad las puertas And y sus conexiones a los cables verticales.
  - Por último, la puerta Or, su conexión y la salida.

### 1.2.4 Comprobación del circuito

Para comprobar el circuito Xor-2 en suma de minterms, debéis añadir Switches a las entradas de los cables  $x$  e  $y$ , y un Binary Probe a la salida  $w$  y también a  $m1$  y  $m2$  para ver los resultados intermedios. Comprobad que el circuito construido se comporta como dice la tabla de verdad de la función Xor.

### 1.2.5 Tiempo de propagación

Ahora vamos a simular el comportamiento temporal del circuito de la puerta Xor-2 en suma de minterms, al igual que hicisteis en el trabajo previo, pero ahora usando el simulador LogicWorks. Para ello haced lo mismo habéis hecho para probar el funcionamiento de las puertas básicas.

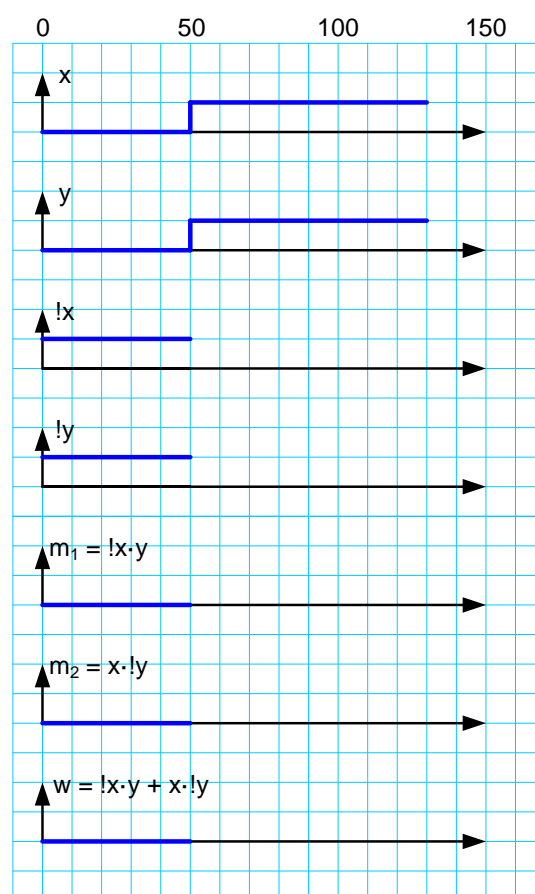
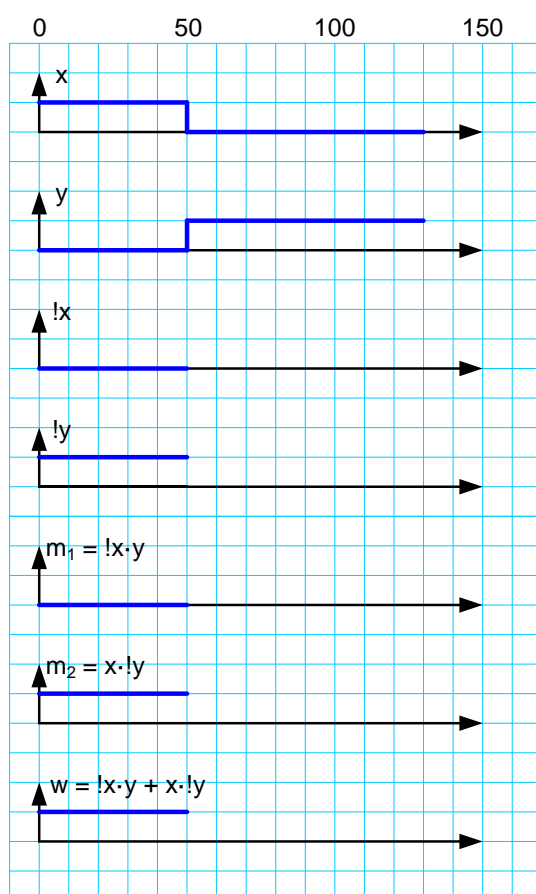
#### □ Informe final

##### Pregunta 3

- a) Simulad, usando los cronogramas de LogicWorks estos dos casos de cambios en las entradas y copiad el resultado sobre estos cronogramas incompletos.

Caso a: (1, 0) -> (0, 1)

Caso b: (0, 0) -> (1, 1)



- b) ¿El cronograma que dibujasteis para los casos a y b de la pregunta 2 del informe previo, coincide con el obtenido ahora con el simulador LogicWorks? Si no coincide explicad el porqué.
- c) Rellenad la columna d de la siguiente tabla, que es equivalente a la que usamos al analizar la puerta And-2 y Or-2, pero ahora para el circuito Xor-2 en suma de minterms. En la columna d indicar los casos en que la salida no cambia con 0 u.t. (n.c.) y los casos de glitch con el tiempo seguido de la palabra glitch entre paréntesis.

Valor de las entradas "x" e "y", (x, y), en t-Δt	Valor de las entradas "x" e "y", (x, y), en t+Δt	d (delay)
(0, 0)	(0, 1)	
	(1, 0)	
	(1, 1)	
(0, 1)	(0, 0)	
	(1, 0)	
	(1, 1)	
(1, 0)	(0, 0)	
	(0, 1)	
	(1, 1)	
(1, 1)	(0, 0)	
	(0, 1)	
	(1, 0)	

- d) Explicad para alguno de los casos en que se ha producido un *glitch* por qué se ha producido e indicad la duración del pulso y por qué dura lo que dura.
- e) ¿Cuál es el tiempo de propagación de cada una de las entradas a la salida?

$$T_{p_{x-w}} = \quad ; \quad T_{p_{y-w}} =$$


## 1.2.6 Encapsulado del circuito

Lo primero que debemos hacer es preparar el circuito para poderlo encapsular. Primero quitamos los Binary Switch y los Binary Probe que hemos introducido en el circuito. También podemos quitar el nombre de las puertas y de las señales, si lo queremos dejar más limpio. Luego ya podemos usar el procedimiento que explicamos en la práctica 0 y que no vamos a repetir aquí.

En la librería de dispositivos LibPrac1 encontraréis un dispositivo denominado Xor-2-P1. Es un símbolo de una puerta Xor, que nosotros hemos creado, para que no tengáis que dibujar vosotros símbolos con el editor de dispositivos. Este símbolo no tiene asociado (*attached*) el circuito correcto de una Xor. Esto es lo que tenéis que hacer vosotros (sólo tiene asociado un circuito con los port connectors de las entradas x e y así como el de la salida w, para que vosotros completéis el circuito).

## 1.2.7 Comprobación del circuito encapsulado

Ahora vamos a comprobar que el dispositivo Xor-2-P1 que acabamos de crear tiene el comportamiento esperado. Para ello, abrid un nuevo circuito y añadid el dispositivo a la hoja de edición de circuitos como si de una puerta básica se tratara.

Si hacéis doble clic sobre la puerta Xor-2-P1 que habéis colocado en la hoja de edición se os abrirá una nueva ventana donde veréis su implementación interna, la que tiene asociada, que es la que vosotros habéis construido en esta práctica. Para volver a la ventana anterior utilizad Ctrl+Shift+U. o haced clic en el botón  de cerrar ventana (ojo no hacedlo en la ventana que contiene toda la aplicación).

Añadid los Binary Switch y Binary Probe necesarios y simulad su comportamiento para cada una de sus posibles entradas, como ya sabemos hacer de secciones anteriores.

### □ Informe final

#### Pregunta 4

Cuando estéis seguros del correcto funcionamiento de vuestro circuito Xor-2-P1, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo.

En la sección 1.3.5 usaremos esta puerta Xor que acabamos de crear para hacer un circuito más complejo, como es el Half-adder, que primero implementaremos en suma de minterms.

## 1.3 El Half-adder (Ha)

### 1.3.1 Especificación

Un Half-adder es un circuito combinacional que tiene dos bits de entrada y dos de salida. El Half-adder **cuenta** el número de unos que hay en sus dos entradas y codifica el resultado en binario en los dos bits de salida (c, s), el bit de mayor peso es el c (que significa *carry*, acarreo, me llevo una) y el de menor peso es el s (que significa suma). Dado que como máximo habrá dos unos en las entradas, con dos bits es suficiente para codificar el resultado de la cuenta, y todavía sobra la combinación  $c=1, s=1$  que nunca se dará. Los posibles usos del Half-adder los comentaremos en la próxima práctica, pero podéis suponer que es un elemento muy útil y que vale la pena crearlo. Veamos su especificación lógica mediante su tabla de verdad:

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

### 1.3.2 Implementación del circuito en LogicWorks

Ahora, abrid un circuito nuevo de LogicWorks y construid el circuito Half-adder en suma de minterms que propusisteis como solución a la pregunta 4 del informe previo, siguiendo el estilo empleado para la puerta Xor-2.

### 1.3.3 Comprobación del circuito

Añadid Switches en las entradas y poned Probes a los minterms y en las salidas c y s. Comprobad que los valores obtenidos para cada entrada se corresponden con los de la tabla de verdad del Ha.

### 1.3.4 Tiempo de propagación

Como podéis comprobar por observación, en este circuito un cambio en una entrada no tarda lo mismo en propagarse a cada una de las dos salidas. En general, como ya hemos dicho, en un circuito (o dispositivo) complejo, con varias entradas y varias salidas, el tiempo de propagación desde la entrada  $i$  a la salida  $j$  no tiene porqué ser el mismo para cada pareja de señales de entrada y salida  $(i, j)$ . Todos estos tiempos caracterizan al circuito (o dispositivo).

Ahora, calculad “a mano” el tiempo de propagación de todas las parejas entrada-salida de vuestro circuito Ha, sabiendo los tiempos de propagación reales de las puertas involucradas que habéis encontrado en esta práctica.

Comprobad midiendo el tiempo en los cronogramas, que los tiempos que habéis calculado a mano coinciden con lo simulado por LogicWorks.

#### ❑ Informe final

##### Pregunta 5

Listad los tiempos de propagación de cada pareja entrada-salida para el Half-adder que hemos creado con las puertas de la librería de LogicWorks es:

Tx-c =      ; Tx-s =      ; Ty-c =      ; Ty-s =

### 1.3.5 Otra implementación del Half-adder

Fijaos en la salida s de tabla de verdad de Half-adder y en la tabla de verdad de la puerta Xor. También podéis fijaros en la implementación que habéis hecho de la salida s del Half-adder y en la implementación de la Xor-2-P1. La conclusión de esta observación es que la salida s es la Xor de x e y,  $s = x \oplus y$ .

Ya que tenemos implementada la puerta Xor, vamos a usarla en el diseño del Half-adder. Eliminad las puertas que implementan la salida s y sustituidlas por el dispositivo Xor-2-P1 que habéis creado y que se encuentra en vuestra librería de dispositivos. Salvad este circuito con el nombre Ha-P1, pero no cerréis el circuito.

### 1.3.6 Encapsulado del circuito

Encapsulad el nuevo circuito. Seguid el método de encapsulado usual. En este caso, utilizad el elemento Ha-P1 de la librería como símbolo, el cual ya tiene asociados los port connectors.

### 1.3.7 Comprobación del circuito encapsulado

Comprobad el correcto funcionamiento del circuito Ha-P1 una vez encapsulado tal y como habéis hecho con la Xor-2. Comprobad que las salidas se ajustan a lo esperado según la tabla de verdad para cada una de las combinaciones de valores de las entradas.

#### ❑ Informe final

##### Pregunta 6

Cuando estéis seguros del correcto funcionamiento de vuestro circuito Ha-P1, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo



# Informe final Práctica-1

Apellidos y nombre: .....Grupo.....

Apellidos y nombre: .....Grupo.....

(Por orden alfabético)

## Pregunta 1 (1 punto)

Valor de las entradas "c" y "d", (c, d), en $t-\Delta t$	Valor de las entradas "c" y "d", (c, d), en $t+\Delta t$	d (delay)
(0, 0)	(0, 1)	
	(1, 0)	
	(1, 1)	
(0, 1)	(0, 0)	
	(1, 0)	
	(1, 1)	
(1, 0)	(0, 0)	
	(0, 1)	
	(1, 1)	
(1, 1)	(0, 0)	
	(0, 1)	
	(1, 0)	

Explicación:

## Pregunta 2 (0.25 puntos)

$T_p(\text{Not-1}) =$

$T_p(\text{And-2}) =$

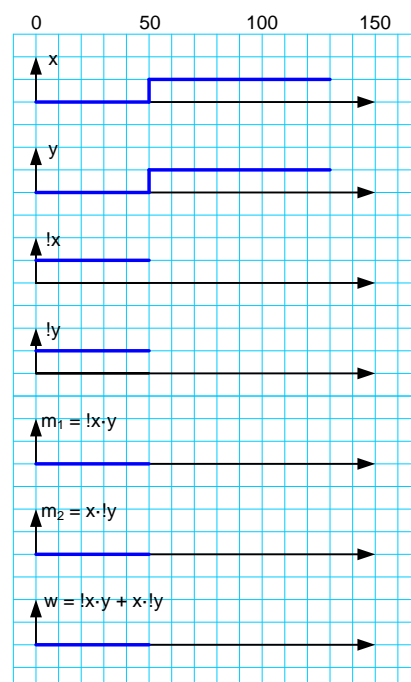
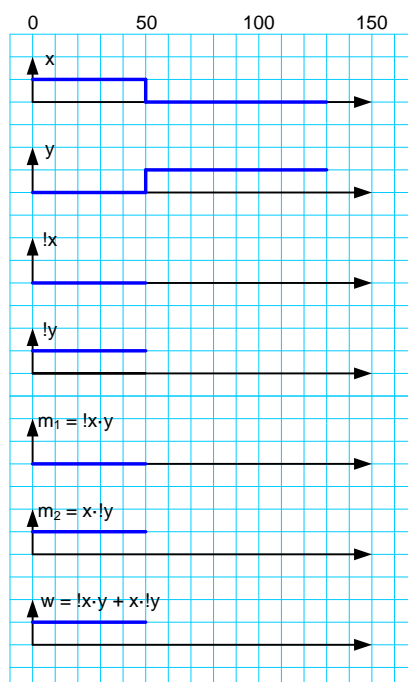
$T_p(\text{Or-2}) =$

## Pregunta 3 (3.75 puntos)

a) (1 punto)

Caso a: (1, 0)  $\rightarrow$  (0, 1)

Caso b: (0, 0)  $\rightarrow$  (1, 1)



b) (0.25 puntos)

c) (1 punto)

Valor de las entradas "x" e "y", (x, y), en $t-\Delta t$	Valor de las entradas "x" e "y", (x, y), en $t+\Delta t$	d ( <i>delay</i> )
(0, 0)	(0, 1)	
	(1, 0)	
	(1, 1)	
(0, 1)	(0, 0)	
	(1, 0)	
	(1, 1)	
(1, 0)	(0, 0)	
	(0, 1)	
	(1, 1)	
(1, 1)	(0, 0)	
	(0, 1)	
	(1, 0)	

d) (0,5 puntos)

e) (1 punto)  $T_{px-w} =$  ;  $T_{py-w} =$

**Pregunta 4** (2 puntos)

Xor-2-P1.

Comentario del profesor:

Firma del profesor:

**Pregunta 5** (1 punto)

$T_{x-c} =$

$T_{x-s} =$

$T_{y-c} =$

$T_{y-s} =$

**Pregunta 6** (2 puntos)

Ha-P1.

Comentario del profesor:

Firma del profesor: