PRÁCTICA 0. Trabajo previo

Introducción al LogicWorks, puertas lógicas, análisis lógico de circuitos combinacionales y encapsulado de un dispositivo sencillo

Objetivos que se deben haber alcanzado antes de realizar la práctica

Antes de preparar esta práctica se deben haber alcanzado los objetivos específicos que se indican en la siguiente tabla. Para ello es recomendable haber estudiado las secciones de la documentación que se indican en la primera columna de la tabla.

Secciones de la documentación a estudiar	Objetivos específicos a alcanzar
Capítulo 2: Todo el capítulo (secciones 2.1 a 2.3).	2.4
Capítulo 3: Secciones 3.1 y 3.2.	3.1, 3.3 y 3.5

Estos objetivos serán evaluados en el informe previo (última página de este documento) que debéis entregar al inicio de la sesión de laboratorio y en la prueba previa individual que se hará al inicio de la sesión.

0.1 Introducción al sistema de numeración binario

De momento nos interesa que sepáis expresar en binario un número natural que os demos en decimal (Obtener el vector de bits X a partir de Xu) y al revés, expresar en decimal un número que os demos en binario (Obtener Xu a partir de X). Saber esto os ayudará en esta práctica para ordenar las filas de una tabla de verdad y en la siguiente para la síntesis de circuitos combinacionales en suma de minterms. En las siguientes prácticas construimos un sumador y un multiplicador de números naturales codificados en binario con 16 bits.

- **Informe previo**

  Pregunta 1

  Escribid el valor en decimal que representan, según el sistema binario, cada uno de los siguientes vectores de bits:

  a) 10011101
  b) 1100000011100001
  c) 0000000000101010

- **Informe previo**

  Pregunta 2

  Rellena la siguiente tabla, indicando cada vector de 4 bits que representa a cada uno de los números naturales del 0 al 15.

Vector de bits X= $x_3\ x_2\ x_1\ x_0$	Valor X_u
	0
	1
	2
	3
	4
	5
	6
	7

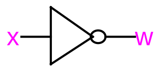
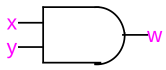
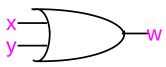
Copyright © 2017, Juan J. Navarro, Universitat Politècnica de Catalunya.

	8
	9
	10
	11
	12
	13
	14
	15

0.2 Las puertas lógicas básicas

En esta sección nos familiarizamos con el funcionamiento de tres dispositivos que denominamos puertas lógicas básicas, las únicas puertas que usaremos para construir el nuestro computador: *Not*, *And* y *Or*.

En la siguiente tabla se muestra, para cada puerta: el nombre que le damos en nuestra librería de dispositivos LogicWorks, su símbolo, y la notación usada para la función (operación) lógica que implementa (entre paréntesis se indica alguna notación alternativa para cada operación).

Puerta	Nombre del dispositivo en nuestra librería	Símbolo	Expresión lógica
Not	Not-1		w = !x (w = $\overline{x}$) (w = x´)
And	And-2		w = x y (w = x · y) (w = x && y)
Or	Or-2		w = x **+** y (w = x || y)

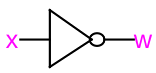
0.2.1 Comportamiento lógico. Tabla de verdad

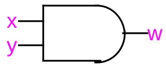
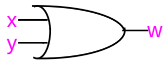
Desde un punto de vista lógico, podemos decir que en una puerta lógica, como en cualquier circuito combinacional, el valor de la salida en cada momento es función de lo que valen las entradas en ese preciso momento. Por ello, el funcionamiento de una puerta se puede especificar, al igual que la función lógica que implementa, mediante una tabla, que se denomina **tabla de verdad**.

- **Informe previo**

  Pregunta 3

  Completad las tablas de verdad de las tres puertas básicas. Esta información la tenéis que saber de memoria. Buscadla en la documentación de la asignatura.

Puerta	Símbolo	Tabla de verdad
Not		*x* | *w* 0 | 1 |

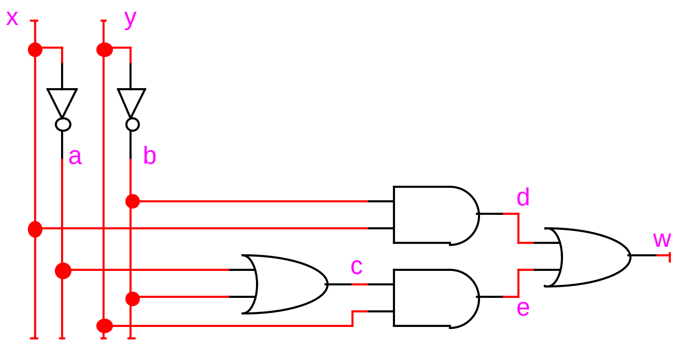
<table>
<tr><td>And</td><td></td><td><table><tr><th>x</th><th>y</th><th>w</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table></td></tr>
<tr><td>Or</td><td></td><td><table><tr><th>x</th><th>y</th><th>w</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table></td></tr>
</table>

0.3 Análisis de un circuito formado por varias puertas

Ahora vamos a analizar con lápiz y papel un circuito combinacional formado por puertas Not, And y Or. Este circuito, que denominamos C-P0 (**C**ircuito de la **P**ráctica **0**), es muy sencillo. Se podría decir que es una puerta lógica ya que tiene solamente 1 salida (implementa una sola función lógica) y 2 entradas. Después, en el laboratorio, este circuito lo vamos a implementar con el simulador LogicWorks y lo vamos a asociar a un símbolo (una caja etiquetada con C-P0 con dos patillas de entrada y una de salida), para formar un dispositivo de la librería y poderlo usar en la siguiente práctica para formar circuitos o dispositivos más complejos.

0.3.1 Esquema lógico

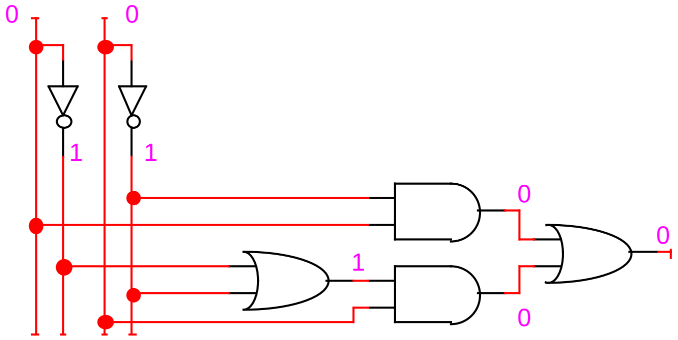
La siguiente figura muestra un esquema lógico interno del circuito C-P0. Por convenio, dos líneas que se cruzan no están conectadas excepto que se dibuje un punto gordo en la intersección. Para ayudarnos en el análisis del circuito hemos dado nombre a las líneas de salida de cada puerta que forma el circuito: a, b, c, d y e.



0.3.2 Análisis del circuito

El análisis lógico de un circuito combinacional consiste en saber qué hace el circuito. Esto es, saber el valor lógico (0 o 1) que toma cada una de sus salidas para cada una de las posibles combinaciones de valores lógicos de sus entradas. Esto se indica mediante la tabla de verdad del circuito.

Así pues, a partir del esquema lógico del circuito vamos a encontrar su tabla de verdad. Para ello, usando las tablas de verdad de las puertas Not-1, And-2 y Or-2 (que debéis saber de memoria) podemos ver, para cada combinación de las variables de entrada x e y, el valor lógico que toman cada una de las variables intermedias a, b, c, d y e, y finalmente la salida w. La siguiente figura muestra el valor de los cables del circuito (las variables lógicas) para la combinación de valores de entrada x=0 e y=0.



➢ **Informe previo**
Pregunta 4

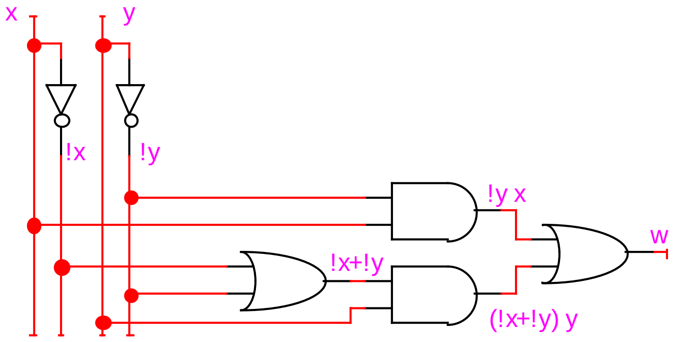
a) Completa cada una de las filas de la tabla de verdad siguiente, con el objetivo de encontrar la tabla de verdad del circuito combinacional C-P0, procediendo como se ha hecho en el párrafo anterior para la primera fila de la tabla de verdad.

x	***y***	***a***	***b***	***c***	***d***	***e***	***w***
0	**0**	1	1	1	0	0	0
0	**1**						
1	**0**						
1	**1**						

b) Especifica la tabla de verdad del circuito C-P0, rellenando la tabla siguiente.

Veamos otra forma un poco más rápida de encontrar la tabla de verdad del circuito. En la siguiente figura indicamos la expresión lógica de la salida de cada puerta en función de las entradas del circuito, x e y. Así, este circuito implementa la función cuya expresión lógica obtenida directamente del esquema del circuito es:

w = !y x + (!x + !y) y.



Ésta es una de las posibles expresiones de la función lógica w (pues aplicando propiedades algebraicas de las funciones lógicas Not, And y Or podemos encontrar otras expresiones equivalentes, aunque esto no lo vamos a hacer, de momento). La expresión w = !y x + (!x + !y) y podría ser ya el resultado del análisis lógico del circuito, pues nos especifica claramente qué hace el circuito. No obstante, si queremos la tabla de verdad del circuito, que es una especificación exhaustiva del valor de la salida para cada posible combinación de los valores de las entradas, podemos proceder como antes, pero ahora completamos la tabla por columnas. La columna !x la obtenemos negando la columna x, lo equivalente para la columna !y, para la columna !y+!x hacemos la or lógica de las columnas !x e !y y así para el resto de columnas.

- **Informe previo**

  Pregunta 5

  Completa cada una de las columnas de la tabla de verdad siguiente, con el objetivo de encontrar la tabla de verdad del circuito combinacional C-P0. Comprueba que el resultado es el mismo que el anterior.

x	*y*	*!x*	*!y*	*!x+!y*	*!yx*	*(!x+!y)y*	*w*
0	**0**	1	1	1			
0	**1**	1	0	1			
1	**0**	0	1	1			
1	**1**	0	0	0			

Si no os habéis equivocado, la tabla de verdad resultante de la pregunta 4a y 5 del informe previo debe ser la misma.

Informe previo Práctica-0

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

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

(Por orden alfabético)

Pregunta 1

a) Xu =

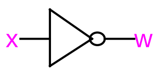
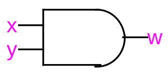
b) Xu =

c) Xu =

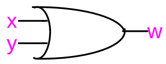
Pregunta 2

Vector de bits *X*= $x_3\ x_2\ x_1\ x_0$	Valor X_u
	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15

Pregunta 3

<table>
<tr><th>Puerta</th><th>Símbolo</th><th>Tabla de verdad</th></tr>
<tr><td>Not</td><td>
</td><td><table><tr><th>x</th><th>w</th></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr></table></td></tr>
<tr><td>And</td><td>
</td><td><table><tr><th>x</th><th>y</th><th>w</th></tr><tr><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td></td></tr></table></td></tr>
</table>

Copyright © 2017, Juan J. Navarro, Universitat Politècnica de Catalunya.

<table>
<tr>
<td>Or</td>
<td>
</td>
<td>

x	*y*	*w*
0	0	
0	1	
1	0	
1	1	

</td>
</tr>
</table>

Pregunta 4

a)

x	***y***	***a***	***b***	***c***	***d***	***e***	***w***
0	**0**	1	1	1	0	0	0
0	**1**						
1	**0**						
1	**1**						

b)

Pregunta 5

x	***y***	***!x***	***!y***	***!x+!y***	***!yx***	***(!x+!y)y***	***w***
0	**0**	1	1	1			
0	**1**	1	0	1			
1	**0**	0	1	1			
1	**1**	0	0	0			

Copyright © 2017, Juan J. Navarro, Universitat Politècnica de Catalunya.