

## PRÁCTICA 2. Trabajo previo

### Implementación combinacional de un sumador e introducción a los circuitos secuenciales.

#### 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 4: Sección 4.2	4.1.1, 4.2.2, 4.3.2.
Capítulo 6: Secciones 6.1.3, 6.1.4 y 6.1.5. Sección 6.2.2 y 6.3. Secciones 6.5.2 y 6.5.3. Sección 6.1.8	6.3 y 6.4. 6.10. 6.14. 6.15 y 6.16.

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

**Estructura.** Esta práctica tiene dos partes bien diferenciadas, de temática muy diferente: 1) implementación de un sumador combinacional conectando Full-adders en propagación del acarreo y 2) introducción al biestable D activado por flanco (secuencial más simple que usaremos para construir otros más complejos) e implementación de un circuito secuencial contador.

## 2.1 Implementación combinacional de un sumador

Ya sabemos implementar circuitos combinacionales de pocas entradas. En la práctica 1 hemos construido un circuito combinacional con dos entradas ( $x$ ,  $y$ ) y dos salidas ( $c$ ,  $s$ ), denominado Half-adder, que suma dos bits con el mismo peso. En esta parte de la práctica, primero vamos a utilizar el bloque Half-adder para construir un bloque combinacional que suma tres bits con el mismo peso, denominado Full-adder, y por último vamos a usar los Full-adder para diseñar un sumador de números naturales de 16 bits.

### 2.1.1 Diseño del Full-adder con Half-adders

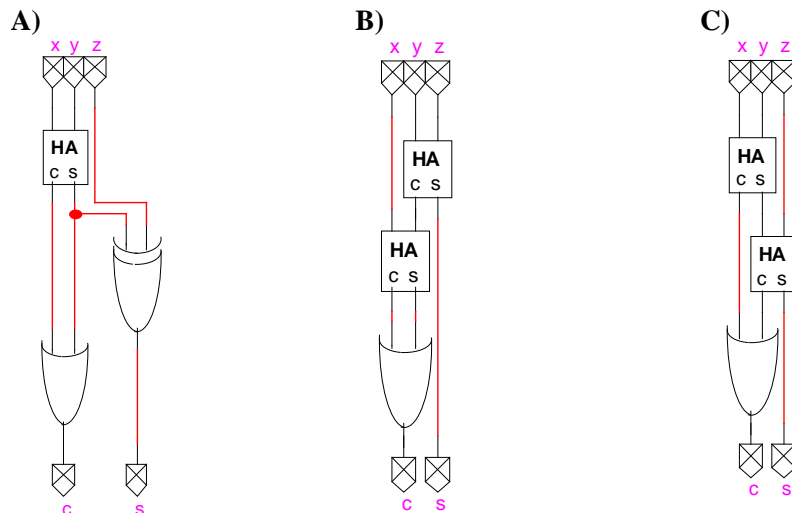
Un Full-adder tiene tres entradas y dos salidas, por lo que se puede sintetizar perfectamente con el método sistemático que hemos estudiado: formulación de la tabla de verdad e implementación en suma de minterms. Sin embargo, ahora vamos a encontrar otra implementación del Full-adder sin seguir ningún método sistemático. Sabemos que un Half-adder suma dos bits del mismo peso y que un Full-adder suma tres bits del mismo peso. ¿Sabríamos sumar tres bits haciendo varias sumas de dos bits cada una? Si sabéis hacerlo sobre el papel y con ejemplos, ¡también podéis implementar un circuito que lo haga!

#### ➤ Informe previo

##### Pregunta 1

Dibuja un circuito que sume tres bits del mismo peso (que haga la funcionalidad de un Full-adder) usando solamente dispositivos Half-adder (puedes usar los que desees, pero sólo Half-adders, ¡ni una puerta más!).

A continuación mostramos tres diseños, uno de ellos es un Full-adder implementado con menos de tres Half-adders y alguna puerta más.



➤ **Informe previo**  
**Pregunta 2**

- a) Construye una tabla de verdad para cada uno de los tres diseños y compárala con la tabla de verdad del Full-adder. ¿Cuál de ellos es un Full-adder? ¿Por qué no se realiza correctamente la suma de los tres bits para los circuitos que no son un Full-adder? (Piensa en los pesos de los bits a sumar y en los pesos de los resultados parciales y finales. Si los dos bits que entran en un Half-adder tienen peso  $2^0$ , la salida  $s$  tiene también peso  $2^0$ , pero la salida  $c$  tiene peso  $2^1$ ). Al diseño correcto del Full-adder obtenido de esta pregunta lo denominaremos **Fa-ha** (las siglas ha indican que está construido con half-adders)
- b) Cada uno de los Half-adders que hemos usado en estos diseños está construido con una And-2 para la salida  $c$  y una Xor-2 para la salida  $s$  (como hicimos en la práctica 1). Si el tiempo de propagación de una And-2 es de 20 u.t., el de una Xor-2 es de 50 u.t. y el de una Or-2 es de 20 u.t., ¿cuál es el tiempo de propagación de cada entrada a cada salida, para cada uno de los tres circuitos anteriores?

Diseño A)	Diseño B)	Diseño C)
$T_{p_{x-c}} =$	$T_{p_{x-c}} =$	$T_{p_{x-c}} =$
$T_{p_{y-c}} =$	$T_{p_{y-c}} =$	$T_{p_{y-c}} =$
$T_{p_{z-c}} =$	$T_{p_{z-c}} =$	$T_{p_{z-c}} =$
$T_{p_{x-s}} =$	$T_{p_{x-s}} =$	$T_{p_{x-s}} =$
$T_{p_{y-s}} =$	$T_{p_{y-s}} =$	$T_{p_{y-s}} =$
$T_{p_{z-s}} =$	$T_{p_{z-s}} =$	$T_{p_{z-s}} =$

- c) ¿Son intercambiables las entradas del Fa-ha en cuanto a tiempo de propagación (¿ $T_{px-c} = T_{py-c} = T_{pz-c}$ ?, ¿ $T_{px-s} = T_{py-s} = T_{pz-s}$ ? )?

Como conclusión podemos decir que hay distintas formas de implementar un Full-adder (como cualquier otro dispositivo). Todas tienen el mismo comportamiento lógico (la misma tabla de verdad) pero pueden tener distinto comportamiento en cuanto a tiempos de propagación, número de puertas y tipos de puertas utilizadas.

## 2.1.2 Algoritmo de suma en binario

El objetivo principal de esta práctica es implementar un sumador de dos números naturales,  $X_u$  e  $Y_u$ , representados en binario con 16 bits cada uno:  $X = x_{15}x_{14} \cdots x_1x_0$  e  $Y = y_{15}y_{14} \cdots y_1y_0$  con  $x_i, y_i \in \{0,1\} \forall i$ . Como  $X_u$  e  $Y_u$  pueden valer cada uno:  $0 \leq x, y \leq 2^{16} - 1$ , la suma  $W_u = X_u + Y_u$  puede valer  $0 \leq W_u \leq 2^{17} - 2$ . Esto nos indica que el resultado  $w$  necesita un vector de 17 bits para representar todos los posibles resultados de la suma:  $W = w_{16}w_{15} \cdots w_1w_0$ . Así pues, nuestro circuito combinacional tendrá 32 entradas y 17 salidas de un bit. Ya no es un circuito con pocas entradas y salidas, ¿verdad?

➤ Informe previo  
Pregunta 3

- ¿Cuántas filas tiene la tabla de verdad que especifica el funcionamiento del sumador binario de dos números de 16 bits?
- No es viable construir el sumador binario de números de 16 bits con la metodología que hemos usado en la práctica 1 (especificación mediante tabla de verdad, obtención de las expresiones en suma de minterms e implementación directa de estas expresiones con puertas Not, And y Or) ¿Por qué? Justifica las razones.

Dada la inviabilidad de construir un sumador a partir de la especificación exhaustiva del valor de las salidas para todos los posibles valores de las entradas, vamos a implementar el sumador usando un algoritmo equivalente al que usamos cuando sumamos en decimal con lápiz y papel, pero ahora en base sumador binario en propagación del acarreo.

➤ Informe previo  
Pregunta 4

Completad las siguientes tablas siguiendo el algoritmo de suma con propagación del acarreo:

- Suma en Hexadecimal ( $b=16$ ) de los vectores de dígitos hexadecimales  $X=A3B2$  e  $Y=8E64$

		Dígito 4	Dígito 3	Dígito 2	Dígito 1	Dígito 0
	$X$		A	3	B	2
	$Y$		8	E	6	4
$k=0$	$w_0$					
	$c_1$					
$k=1$	$w_1$					
	$c_2$					
$k=2$	$w_2$					
	$c_3$					
$k=3$	$w_3$					
	$c_4$					
	$W$					

- Obtened los valores  $X_u$ ,  $Y_u$  y  $W_u$  (en decimal) de los números naturales representados hexadecimal por los vectores de dígitos  $X$ ,  $Y$  y  $W$ . Realizad el cálculo  $X_u + Y_u$  (en decimal) y comprobad que os da igual al valor  $W_u$  obtenido. Si es así, la suma en hexadecimal que habéis hecho en el apartado anterior es correcta.

$X_u =$   
 $Y_u =$   
 $W_u =$   
 ¿ $X_u + Y_u = W_u$ ?

- Suma en binario ( $b=2$ ) de los vectores de bits  $X=0101$  e  $Y=1101$

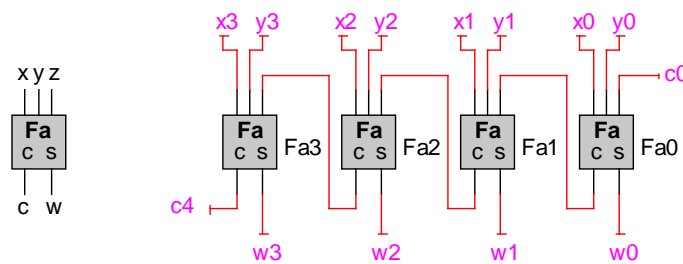
		Dígito 4	Dígito 3	Dígito 2	Dígito 1	Dígito 0
	$X$		0	1	0	1
	$Y$		1	1	0	1
$k=0$	$w_0$					
	$c_1$					
$k=1$	$w_1$					
	$c_2$					
$k=2$	$w_2$					
	$c_3$					
$k=3$	$w_3$					
	$c_4$					
	$W$					

- d) Obtened los valores  $X_u$ ,  $Y_u$  y  $W_u$  (en decimal) de los números naturales representados en binario por los vectores de dígitos  $X$ ,  $Y$  y  $W$ . Realizad el cálculo  $X_u + Y_u$  (en decimal) y comprobad que os da igual al valor  $W_u$  obtenido. Si no es así, la suma en binario que habéis hecho en el apartado anterior no es correcta.

$X_u =$   
 $Y_u =$   
 $W_u =$   
 $\checkmark X_u + Y_u = W_u?$

### 2.1.3 Diseño del sumador binario combinacional con propagación del acarreo

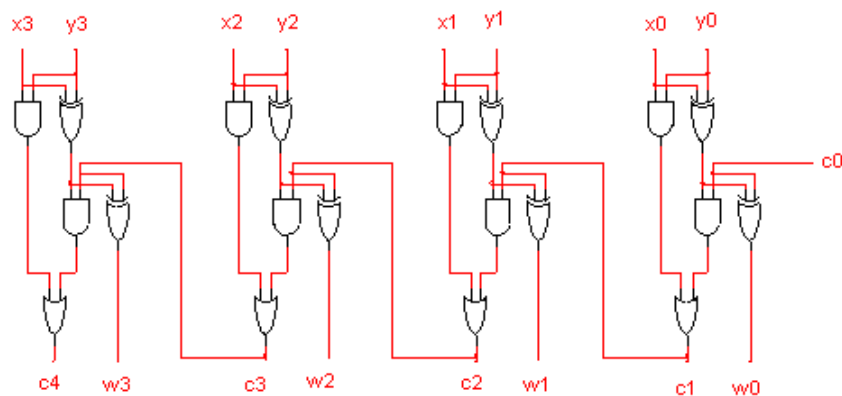
La implementación combinacional del algoritmo de suma en binario se obtiene disponiendo de  $n$  Full-adders (tantos como bits tienen los números a sumar, en nuestro ejemplo  $n = 4$ ) y conectandolos como indica la siguiente figura. Aunque el primer Fa podría ser un Ha porque  $c_0$  vale 0, dejamos un Fa-ha. A pesar de que la entrada  $c_0$  no la usamos en esta práctica, es útil en otras ocasiones.



Fijaos que en el diseño de la figura, la salida  $w_4$  la hemos denominado  $c_4$  (esto es correcto ya que  $w_4 = c_4$ ). No hemos nombrado  $w_4$  porque en los computadores generalmente todos los datos tienen el mismo número de dígitos. Cuando la salida  $c_n$  del sumador vale 1 nos indica que el resultado correcto de la suma no puede representarse en binario con  $n$  bits.

### 2.1.4 Tiempo de propagación del sumador

Vamos a calcular el tiempo de propagación de un sumador de dos números de 4 bits implementado con Fa en propagación del acarreo para el caso de implementar cada Fa con dos Ha y una puerta Or-2. Para ello es de gran ayuda la siguiente figura que muestra el sumador binario con dispositivos Fa-ha pero sin que se visualicen los bloques Fa-ha, sino viendo directamente todas las puertas con que están contruidos los Half-adders que se usan para construir los Full-adders. También os ayudará para estos ejercicios que recordéis los tiempos de propagación de las puertas, que la Xor-2 puede tener un tiempo de propagación de 40 o de 50 dependiendo de los valores concretos de sus entradas, por lo que para calcular caminos críticos hay que considerar que el tiempo es el de la combinación de valores que produce el máximo retardo, 50 u.t.



➤ **Informe previo**  
**Pregunta 5**

Para la implementación basada en los Full-adders **Fa-ha** (suponed que el tiempo de propagación de las puertas And-2, Or-2 y Xor-2 es 20, 20 y 50 u.t. respectivamente):

- Indicad qué pareja entrada-salida es la que tiene mayor tiempo de propagación (o una de las parejas de entrada-salida si hay varias con el mismo tiempo máximo). Listad la secuencia de Full-adders, indicando la entrada y la salida de cada Full-adder, por la que pasa el camino crítico de la entrada a la salida con mayor tiempo de propagación del sumador. Por ejemplo, si el camino crítico pasara por el Full-adder 1 (el que suma los bits de peso 2<sup>1</sup>) entrando por z y saliendo por c, en la lista debe indicarse ..., -> Fa1z-c -> .....
- ¿Cuál es el tiempo de propagación del sumador? ¿Cómo has obtenido este tiempo?

➤ **Informe previo**  
**Pregunta 6**

- Expresa, para  $n = 1$ ,  $n = 2$  y para el caso general de  $n > 2$  bits, el tiempo de propagación del sumador binario construido con Fa-ha.

Nos centramos ahora en el sumador basado en Fa-ha, para reducir el trabajo de la siguiente pregunta. El circuito sumador que hemos diseñado usando Fa-ha, como en general ocurre con cualquier circuito, no tarda siempre lo mismo en dar el resultado correcto y estable en sus salidas. Este tiempo depende de las entradas concretas en las que se produce el cambio, del valor de las entradas antes y después de producirse un cambio y del valor de las entradas que no cambian. (Cuando hablamos de tiempo de propagación estamos considerando el peor caso). El siguiente ejercicio trata de este aspecto, aunque es un tema que no debe obsesionarnos, pues excepto en estos ejercicios siempre hablamos de tiempo de propagación considerando que los valores de las entradas son los que producen el mayor tiempo de propagación y lo obtenemos calculando el camino crítico.

Es útil que recordéis para responder a las siguientes preguntas que cuando una puerta And-2 tiene una entrada fijada a 0, la salida es 0 independientemente del valor de la otra entrada. De forma dual, cuando una puerta Or-2 tiene una entrada fija a 1, la salida vale 1 independientemente del valor de la otra entrada. Además, como ya vimos en la práctica anterior, la puerta Xor-2 tiene un tiempo de propagación que depende de los valores concretos que cambian en sus entradas (realmente, según el cambio a veces el tiempo es de 0 u.t., 40 u.t. o 50 u.t.). Copiamos a continuación la tabla de tiempos de la puerta Xor-2.

Valor de "x" e "y" en t-Δt	Valor de "x" e "y" en t+Δt	d (delay)
0 0	0 1	40
	1 0	40
	1 1	50
0 1	0 0	40
	1 0	50
	1 1	50
1 0	0 0	40
	0 1	50
	1 1	50
1 1	0 0	0
	0 1	50
	1 0	50

➤ **Informe previo**  
**Pregunta 7**

- Suponed que en el sumador de dos números de 4 bits construido con Fa-ha todas las entradas han valido 0 durante mucho tiempo antes del instante t. En el instante t las entradas pasan a valer:  $X = 1010$ ,  $Y = 0101$  y  $C_0 = 1$ . La salida  $w_k$  se estabiliza con su valor correcto en el instante de tiempo  $t+d_k$ , para  $k = 0, \dots, 3$  y la salida  $c_4$  se estabiliza en el tiempo  $t+d_4$ . ¿Cuánto vale  $d_k$  para cada una de las cinco salidas del sumador,  $c_4, w_3, \dots, w_0$ ?

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

b) Repetid el ejercicio anterior para  $X = 1010$ ,  $Y = 0101$  y  $C_0 = 0$ .

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

c) Explicad las diferencias entre el caso a) y b):

- ¿Por qué en el caso a) d<sub>3</sub> es mayor que d<sub>2</sub>, d<sub>2</sub> mayor que d<sub>1</sub>, y d<sub>1</sub> mayor que d<sub>0</sub> mientras que en el caso b) d<sub>3</sub>, d<sub>2</sub>, d<sub>1</sub> y d<sub>0</sub> tienen el mismo tiempo?
- ¿Por qué en el caso a) d<sub>4</sub> es menor que d<sub>3</sub> mientras que en el b) d<sub>4</sub> es 0?

## 2.2 Introducción a los circuitos secuenciales

En esta segunda parte de la práctica primero estudiamos el comportamiento del dispositivo **secuencial** más sencillo que vamos a utilizar para luego construir otros circuitos secuenciales más complejos: el biestable D activado por flanco. Para nosotros, los biestables son a los circuitos secuenciales como las puertas a los combinacionales: los elementos más simples con los que construiremos el computador completo.

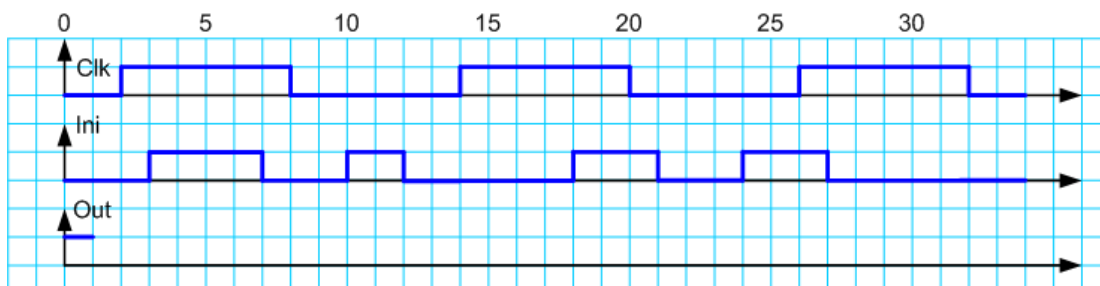
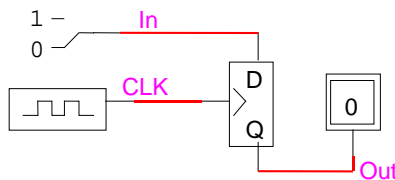
### 2.2.1 El biestable D activado por flanco

Un biestable es un circuito secuencial que consta de una entrada de datos D y otra de reloj, CLK y una salida Q, todas ellas de 1 bit. Su función es muy simple, cuando la señal de reloj pasa de 0 a 1 (se produce un flanco ascendente de reloj), el valor que se encuentra en ese momento en la entrada D del biestable, aparece en la salida Q, y permanece fijo ese valor en la salida aunque cambie el valor de la entrada D, hasta que se produzca otro flanco ascendente de reloj.

En realidad, al igual que ocurre con las puertas lógicas, el biestable no modifica su salida en el mismo instante de tiempo en que llega el flanco ascendente de reloj. Un biestable real tiene un tiempo de propagación distinto de cero. El **tiempo de propagación** es el tiempo desde que se produce el flanco ascendente de la señal de reloj hasta que la salida Q se estabiliza al valor que tenía la entrada D en el instante en que llegó el flanco. (Además hay otros parámetros temporales importantes en un biestable, como el tiempo de *start-up* y el tiempo de *hold*, que para simplificar no consideramos en este curso).

#### ➤ Informe previo Pregunta 8

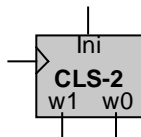
Dado el circuito de la figura, rellenad el siguiente cronograma con el valor que tomará la señal Out en cada instante de tiempo. Suponed que el tiempo de propagación del biestable es 1 u.t. (1 cuadrícula del dibujo).



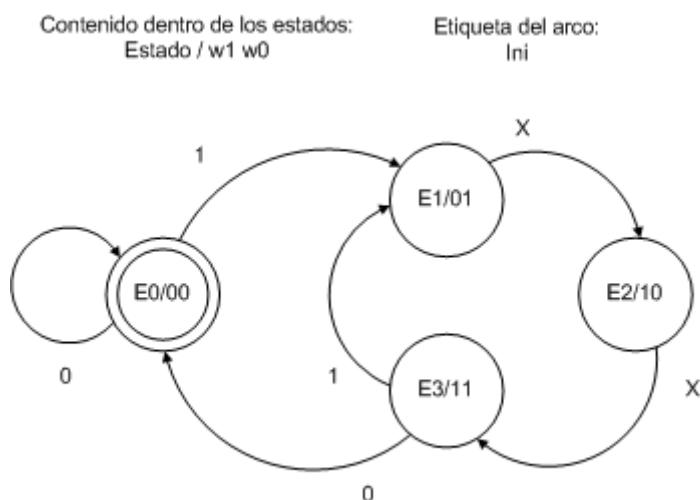
## 2.2.2 Generador de secuencia

Vamos a implementar un circuito lógico secuencial muy simple, pero que se parece mucho al que diseñaremos en la práctica 3 como unidad de control de un multiplicador secuencial. La funcionalidad de este circuito, que ahora nos puede parecer extraña, no debe importarnos por el momento, sólo tenemos que saber cómo se comporta el circuito (dibujar un cronograma) a partir del grafo de Moore que especifica su funcionamiento y también tenemos que saberlo implementar.

El circuito, cuya caja se muestra a continuación tiene una entrada Ini y dos salidas w1 y w0 (además de la entrada de reloj, Clk, como tienen todos los circuitos secuenciales).



El comportamiento del circuito queda especificado en el siguiente grafo de estados.



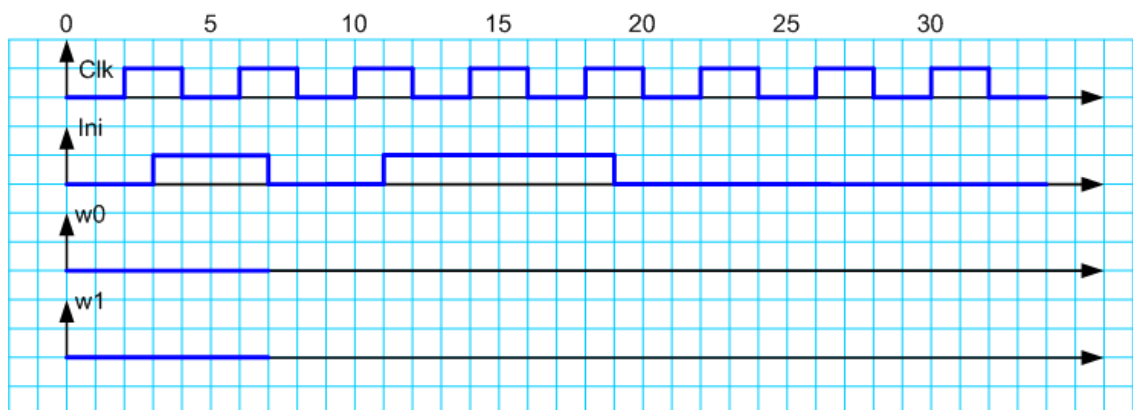
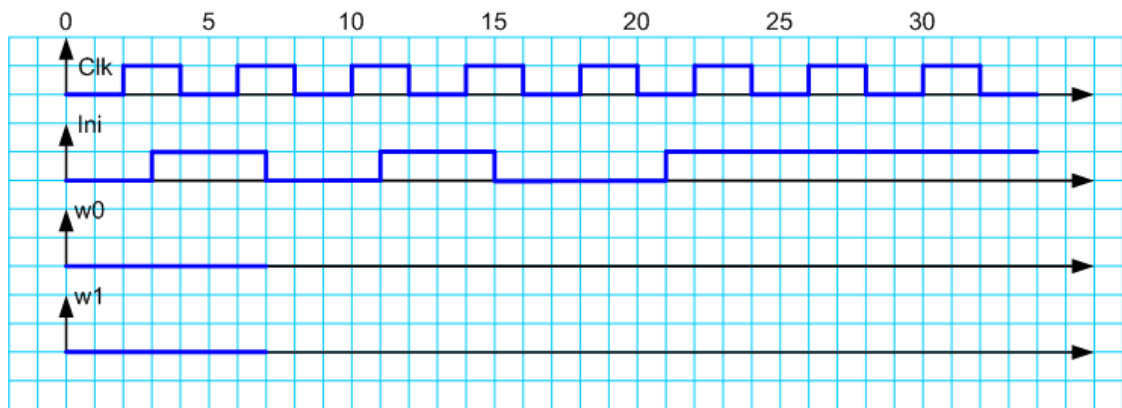
Si consideramos que las salidas w1 y w0 representan un número natural codificado en binario con dos bits, podemos decir que el circuito genera la secuencia 1, 2, 3 y la repite indefinidamente si en el estado E3 la entrada Ini vale 1. Si Ini vale 0 en el estado E3 el sistema se va al estado inicial, E0, generando la salida 0 durante todos los ciclos que permanezca en este estado. Para salir del estado inicial Ini tiene que valer 1 durante un ciclo. El valor de Ini en los estados E1 y E2, en los que la salida vale 1 y 2 respectivamente, no afecta al funcionamiento del circuito.

### ➤ Informe previo Pregunta 9

A partir del grafo anterior, obtened la tabla transiciones (del estado siguiente) y la de las salidas, obtened la expresión mínima en suma de productos (usando mapas de karnaugh) de las señales del estado siguiente y de las salidas y dibujad el esquema lógico interno del circuito CLS-2 usando el mínimo número de biestables D para implementar el estado y puertas Not, And-2 y Or-2 para implementar las expresiones mínimas del estado siguiente y de las salidas (si necesitáis puertas And y Or de más de dos entradas podéis implementarlas con la estructura en árbol de puertas de dos entradas).

### ➤ Informe previo Pregunta 10

Completa los dos cronogramas siguientes, suponiendo que el retardo desde que llega el flanco ascendente de reloj hasta que se estabilizan las entradas del circuito al valor correcto es de 1 unidad de tiempo (una cuadrícula del dibujo) y que el resto de tiempos de propagación de puertas y biestables es 0.



### Tiempo de ciclo

En un circuito secuencial el tiempo de ciclo (periodo) de la señal de reloj tiene que ser suficientemente grande como para que todas las señales que llegan a la entrada D de los biestables estén estables en su valor correcto cuando llegue el flanco ascendente de reloj. Si esto no ocurre, los biestables almacenarán valores incorrectos y el sistema no funcionará como debe. Para ello el periodo de la señal de reloj (el tiempo entre dos flancos ascendentes de reloj) debe ser mayor que el tiempo del camino crítico del circuito. Vamos a calcular el tiempo de ciclo mínimo del CLS-2.

### ➤ Informe previo

#### Pregunta 11

Considerando los tiempos de propagación de las puertas Not, Or-2 y And-2 son 10, 20 y 20 unidades de tiempo respectivamente (si necesitáis puertas And y Or de más de dos entradas podéis implementarlas con la estructura en árbol de puertas de dos entradas) y que el tiempo de propagación del biestable es de 100 unidades de tiempo,

- dibujad sobre el esquema del circuito CLS-2 que has propuesto en la pregunta 9 del informe previo, el camino crítico (dibuja uno de ellos si es que hay varios caminos críticos) del circuito e indica el tiempo de este camino como la suma de tiempos de propagación de los dispositivos por los que pasa (recordad que en general el tiempo de propagación de un combinacional es distinto dependiendo de la entrada y salida que se consideren).
- ¿cuál es el tiempo de ciclo mínimo para que el circuito funcione correctamente?



# Informe previo Práctica-2

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

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

(por orden alfabético)

## Pregunta 1

## Pregunta 2

a)

b)

Diseño A)	Diseño B)	Diseño C)
$T_{p_{x-c}} =$	$T_{p_{x-c}} =$	$T_{p_{x-c}} =$
$T_{p_{y-c}} =$	$T_{p_{y-c}} =$	$T_{p_{y-c}} =$
$T_{p_{z-c}} =$	$T_{p_{z-c}} =$	$T_{p_{z-c}} =$
$T_{p_{x-s}} =$	$T_{p_{x-s}} =$	$T_{p_{x-s}} =$
$T_{p_{y-s}} =$	$T_{p_{y-s}} =$	$T_{p_{y-s}} =$
$T_{p_{z-s}} =$	$T_{p_{z-s}} =$	$T_{p_{z-s}} =$

c)

### Pregunta 3

a)

b)

### Pregunta 4

a)

		Dígito 4	Dígito 3	Dígito 2	Dígito 1	Dígito 0
	$X$		A	3	B	2
	$Y$		8	E	6	4
$k=0$	$w_0$					
	$c_1$					
$k=1$	$w_1$					
	$c_2$					
$k=2$	$w_2$					
	$c_3$					
$k=3$	$w_3$					
	$c_4$					
	$W$					

b)

c)

		Dígito 4	Dígito 3	Dígito 2	Dígito 1	Dígito 0
	$X$		0	1	0	1
	$Y$		1	1	0	1
$k=0$	$w_0$					
	$c_1$					
$k=1$	$w_1$					
	$c_2$					
$k=2$	$w_2$					
	$c_3$					
$k=3$	$w_3$					
	$c_4$					
	$W$					

d)

### Pregunta 5

a)

b)

### Pregunta 6

a)

### Pregunta 7

a)

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

b)

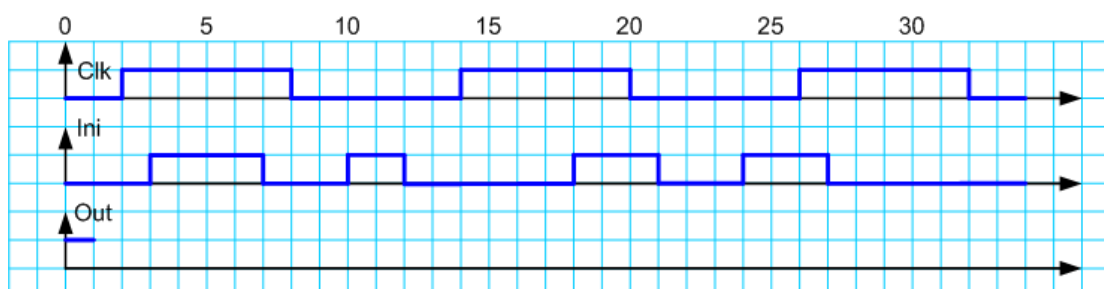
d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>

c) Explicad las diferencias entre el caso a) y b):

i. ¿Por qué en el caso a) d<sub>3</sub> es mayor que d<sub>2</sub>, d<sub>2</sub> mayor que d<sub>1</sub>, y d<sub>1</sub> mayor que d<sub>0</sub> mientras que en el caso b) d<sub>3</sub>, d<sub>2</sub>, d<sub>1</sub> y d<sub>0</sub> tienen el mismo tiempo?

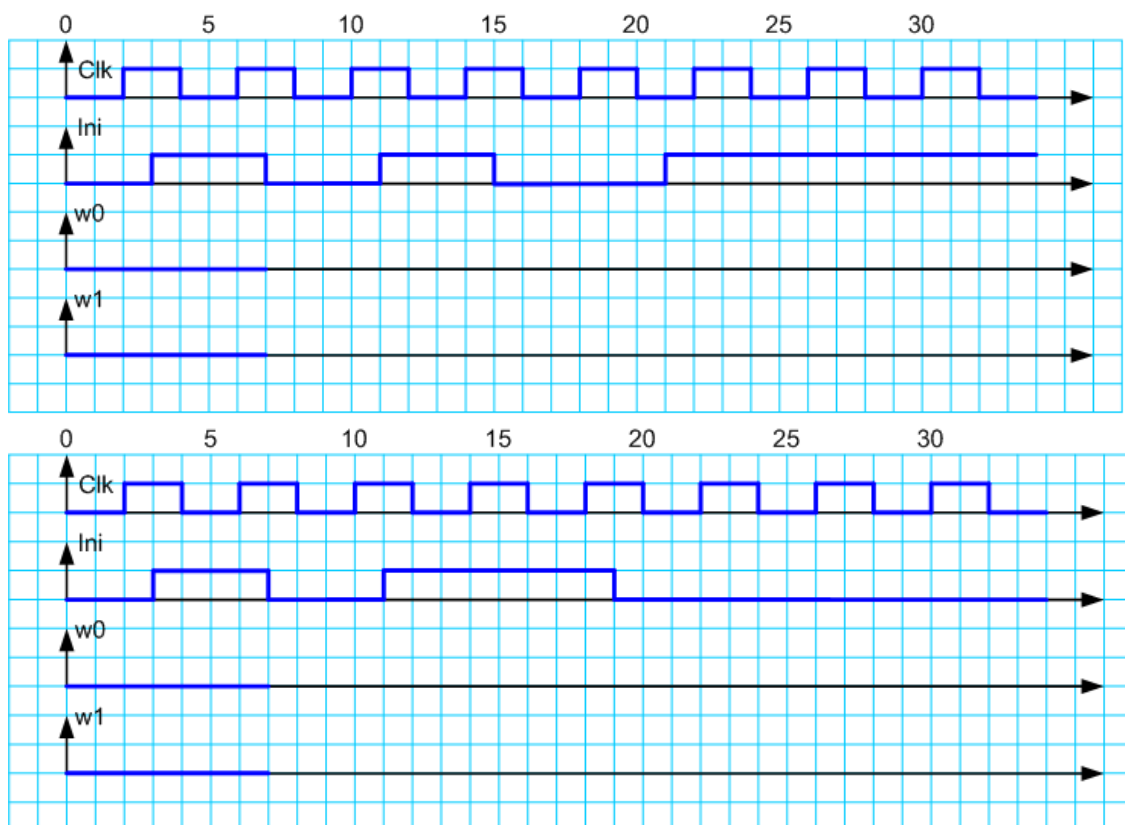
ii. ¿Por qué en el caso a) d<sub>4</sub> es menor que d<sub>3</sub> mientras que en el b) d<sub>4</sub> es 0?

### Pregunta 8



### Pregunta 9

### Pregunta 10



### Pregunta 11

- a) (Dibujado sobre el esquema respuesta de la pregunta 10 del informe previo)
- b)