

Apellidos:.....**SOLUCIÓN**.....

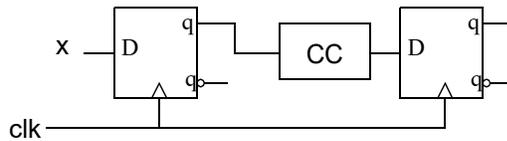
1	2	3	4

Nombre:.....

**DURACIÓN 2:00**

**1.- [2 Puntos]**

- a) Defina breve y claramente el concepto de tiempo de *setup* y de *hold* y a qué tipo de circuitos se aplica.
- b) Para el circuito de la figura el biestable D tiene un comportamiento real con los tiempos que se muestran en la tabla. Si la frecuencia del reloj es de 100 Mhz, ¿cual es el retraso máximo que puede tener el circuito combinacional CC para que el comportamiento de todo el circuito sea correcto? Justifique la respuesta.
- c) Si el tiempo de retraso de CC fuese mayor, ¿qué habría que hacer para el todo el circuito funcionara de forma correcta?



$$t_{pd(D \rightarrow q)} = 2ns$$

$$t_{su} = 1ns$$

$$t_h = 0,5ns$$

**SOLUCIÓN**

a) Tiempo de *setup*: tiempo que tienen que permanecer estables las entradas síncronas del biestable antes del flanco activo de reloj.

Tiempo de *hold*: tiempo que tienen que permanecer estables las entradas síncronas del biestable después del flanco activo de reloj.

b) Dado que el reloj es de 100MHz, su periodo es de 10ns. Es decir, hay 10ns entre flancos activos de reloj. La salida q del primer biestable cambia 2 ns después del flanco activo de reloj. Este cambio debe verse en la entrada del segundo biestable al menos 1 ns antes del siguiente flanco activo de reloj para que se satisfaga el tiempo de *setup*. Esto implica que el retraso máximo del circuito CC debe ser de 7 ns.

c) Si el retraso de CC es mayor de 7 ns la solución pasa por reducir la frecuencia del reloj (ampliar su periodo).

**CRITERIO DE CORRECCIÓN**

- 2.- [1 Punto] Explique de forma clara y concisa el funcionamiento de los codificadores. Comente las situaciones conflictivas que puede haber en su uso y cuales son las soluciones.

**SOLUCIÓN**

Además del funcionamiento hay que explicar las siguientes situaciones conflictivas que puede haber:

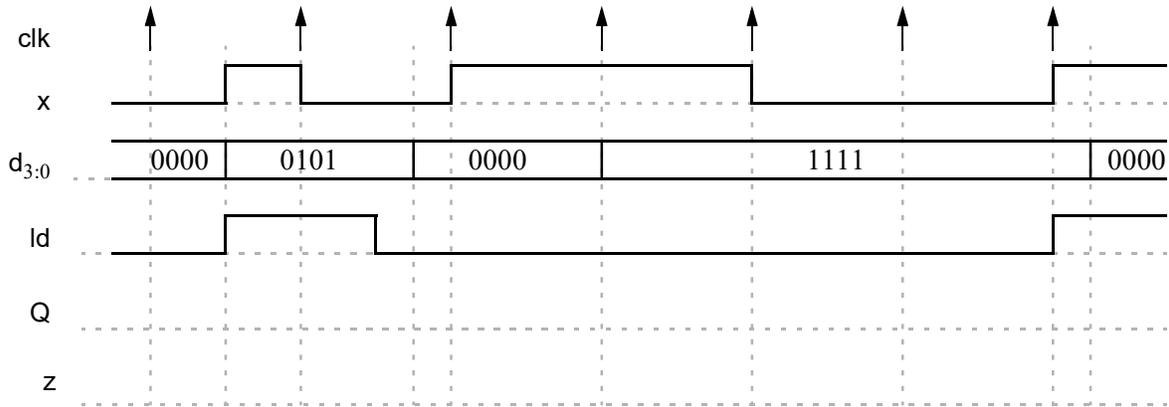
- Activación de varias entradas a la vez. La solución es un codificador con prioridad.
- El no diferenciar si el codificador está activado/desactivado/sin entradas activas. La solución es poner dos salidas adicionales, GS y EO.

Para una descripción más detallada, consultar los apuntes de clase.

**CRITERIO DE CORRECCIÓN**

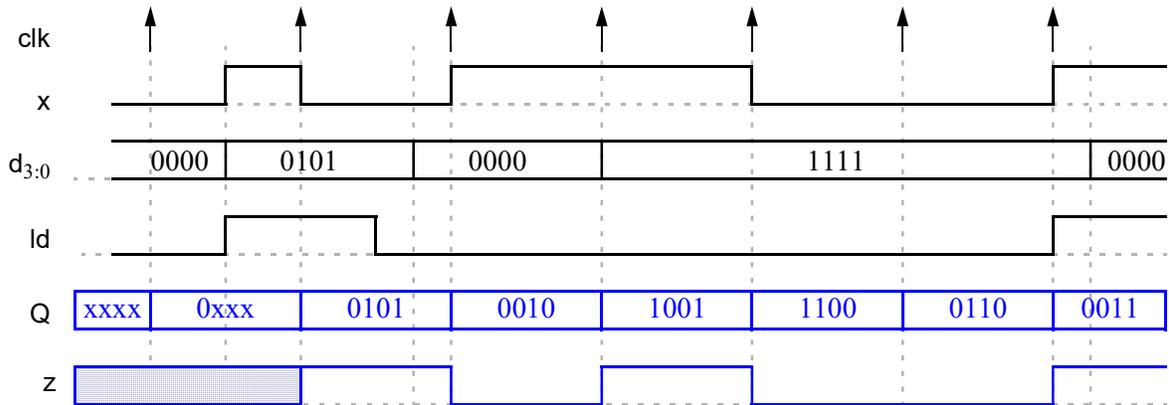
- 3.- [2 Puntos] Rellene el siguiente cronograma, sabiendo que se tiene un registro paralelo-serie de 4 bits con entrada serie x por la izquierda, una entrada paralelo d3:0, una entrada de control ld y una salida serie z. Cuando la entrada ld está a 1 realiza una carga en paralelo de los datos d3:0 y cuando está a 0 realiza un desplazamiento a la derecha cargando el valor de la entrada x en el bit más significativo. La salida z es el bit menos significativo del registro. Q representa lo que está almacenado

en el registro. Inicialmente se desconoce el estado del registro.



### SOLUCIÓN

Al ser un registro paralelo-serie realiza operaciones de carga y de desplazamiento. Cuando la entrada cambia en el flanco activo de reloj, al ser el comportamiento ideal, se toma el valor anterior. Los cambios del registro sólo se pueden hacer cuando hay un flanco activo de reloj. Con todo esto el contenido es:



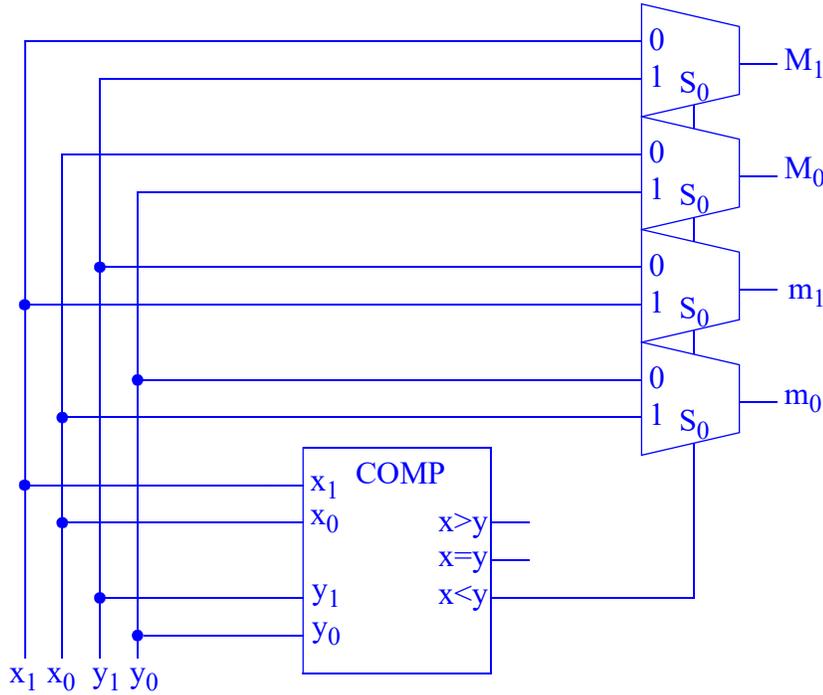
### CRITERIO DE CORRECCIÓN

- 4.- [3 Puntos] Dispone de 4 números de 2 bits sin signo A, B, C y D. Usando subsistemas combinacionales no programables y puertas, calcule la diferencia entre el mayor y el menor de los 4.

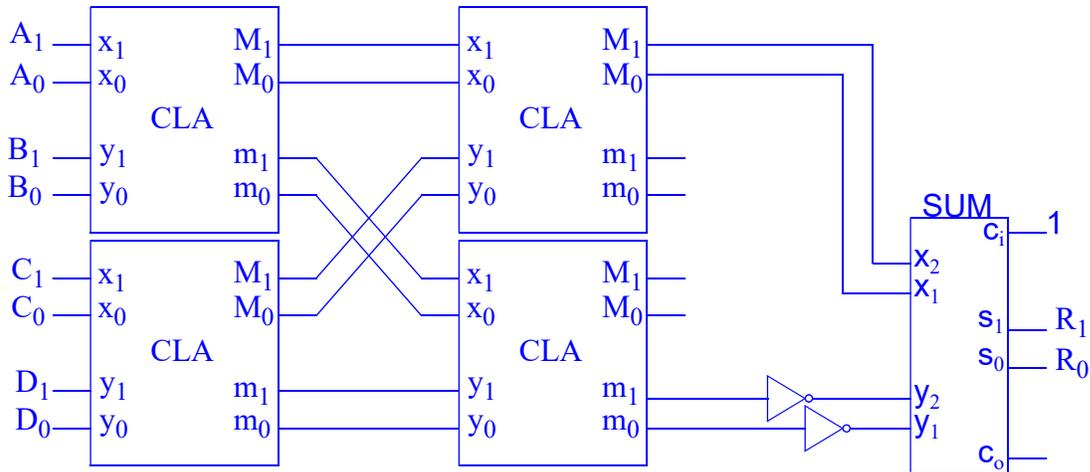
### SOLUCIÓN

El diseño del circuito va a llevarse a cabo construyendo otro más simple (CLA) que compare sólo dos

números ( $x_{1:0}$  e  $y_{1:0}$ ) y proporcione a su salida, tanto el mayor ( $M_{1:0}$ ) como el menor ( $m_{1:0}$ ):



El circuito pedido se obtendrá interconectando varios de ellos del siguiente modo:



En la segunda etapa se usa el mismo módulo CLA para seleccionar el mayor de los mayores y el menor de los menores. En estos módulos CLA no se usan ni el menor de los mayores ni el mayor de los menores, por lo que se podrían simplificar sus circuitos para ahorrar 2 MUX2:1 en cada uno de ellos. No obstante, se han dejado en aras de un diseño modular más sencillo. Una vez obtenidos ambos, sólo queda calcular la diferencia usando un sumador en configuración restadora.

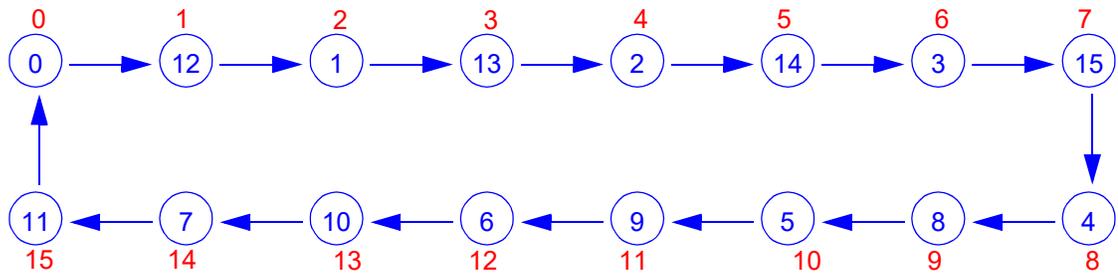
### CRITERIO DE CORRECCIÓN

- 5.- [2 Puntos] El último contador diseñado por el jefe de I+D de una conocida empresa antes de ser ingresado en un psiquiátrico, tenía una forma curiosa de contar. Disponía de 4 salidas y empezaba en el código 0. Aparentemente el código saltaba hacia adelante y hacia atrás de forma alternativa: 0, 12, 1, 13, 2, 14, 3, 15, 4, 8, 5, 9, 6, 10, 7, 11 y vuelta a empezar. Afortunadamente, el becario se dio cuenta que podía realizar dicho contador usando un simple contador ascendente y alguna puerta. ¿Podría indicar el diseño del becario?

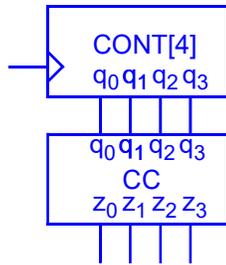
### SOLUCIÓN

Como puede observarse mirando la secuencia de salida, se trata de un contador módulo 16, ya que pasa por 16 estados antes de repetir la secuencia. Se podría diseñar el contador usando la técnica estándar de diseño de CSS, pero resulta más sencillo (y además así se pide en el enunciado) basar el diseño en un contador ascendente y

modificarlo para obtener el nuevo comportamiento. A continuación se muestra el diagrama de estados del contador sobre el que se ha hecho la asignación de estados de un contador ascendente en rojo:



A la salida del contador ascendente módulo 16 se le añadirá un circuito combinacional que generará la secuencia solicitada:



	$q_3q_2$	00	01	11	10
$q_1q_0$	00	0000 <sup>0</sup>	0010 <sup>4</sup>	0110 <sup>12</sup>	0100 <sup>8</sup>
	01	1100 <sup>1</sup>	1110 <sup>5</sup>	1010 <sup>13</sup>	1000 <sup>9</sup>
	11	1101 <sup>3</sup>	1111 <sup>7</sup>	1011 <sup>15</sup>	1001 <sup>11</sup>
	10	0001 <sup>2</sup>	0011 <sup>6</sup>	0111 <sup>14</sup>	0101 <sup>10</sup>
		$z_3z_2z_1z_0$			

Separando cada salida en su mapa-K, obtenemos las expresiones mínimas de las mismas:

	$q_3q_2$	00	01	11	10
$q_1q_0$	00	0 <sup>0</sup>	0 <sup>4</sup>	0 <sup>12</sup>	0 <sup>8</sup>
	01	1 <sup>1</sup>	1 <sup>5</sup>	1 <sup>13</sup>	1 <sup>9</sup>
	11	1 <sup>3</sup>	1 <sup>7</sup>	1 <sup>15</sup>	1 <sup>11</sup>
	10	0 <sup>2</sup>	0 <sup>6</sup>	0 <sup>14</sup>	0 <sup>10</sup>
		$z_3$			

	$q_3q_2$	00	01	11	10
$q_1q_0$	00	0 <sup>0</sup>	0 <sup>4</sup>	1 <sup>12</sup>	1 <sup>8</sup>
	01	1 <sup>1</sup>	1 <sup>5</sup>	0 <sup>13</sup>	0 <sup>9</sup>
	11	1 <sup>3</sup>	1 <sup>7</sup>	0 <sup>15</sup>	0 <sup>11</sup>
	10	0 <sup>2</sup>	0 <sup>6</sup>	1 <sup>14</sup>	1 <sup>10</sup>
		$z_2$			

$$z_3 = q_0$$

$$z_2 = \overline{q_3}q_0 + q_3\overline{q_0} = q_3 \oplus q_0$$

$$z_1 = q_2$$

$$z_0 = q_1$$

	$q_3q_2$	00	01	11	10
$q_1q_0$	00	0 <sup>0</sup>	1 <sup>4</sup>	1 <sup>12</sup>	0 <sup>8</sup>
	01	0 <sup>1</sup>	1 <sup>5</sup>	1 <sup>13</sup>	0 <sup>9</sup>
	11	0 <sup>3</sup>	1 <sup>7</sup>	1 <sup>15</sup>	0 <sup>11</sup>
	10	0 <sup>2</sup>	1 <sup>6</sup>	1 <sup>14</sup>	0 <sup>10</sup>
		$z_1$			

	$q_3q_2$	00	01	11	10
$q_1q_0$	00	0 <sup>0</sup>	0 <sup>4</sup>	0 <sup>12</sup>	0 <sup>8</sup>
	01	0 <sup>1</sup>	0 <sup>5</sup>	0 <sup>13</sup>	0 <sup>9</sup>
	11	1 <sup>3</sup>	1 <sup>7</sup>	1 <sup>15</sup>	1 <sup>11</sup>
	10	1 <sup>2</sup>	1 <sup>6</sup>	1 <sup>14</sup>	1 <sup>10</sup>
		$z_0$			

Efectivamente, el contador “desquiciado” se puede implementar con un contador “cuerdo” y una puerta XOR.

### CRITERIO DE CORRECCIÓN