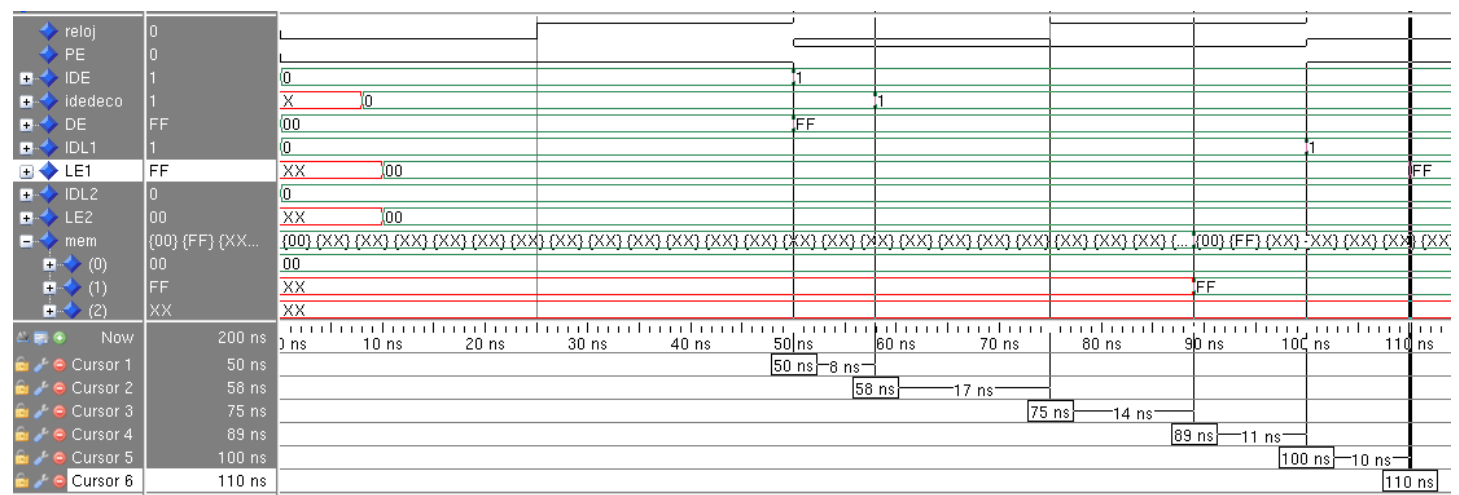


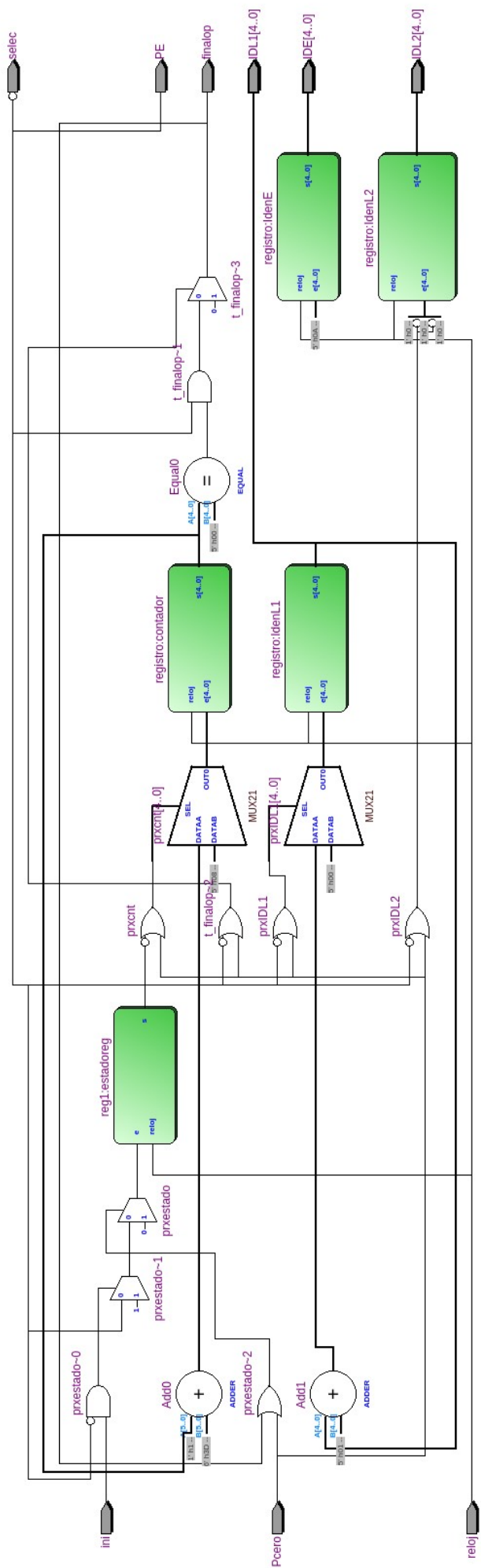
**Pregunta 1:** Considere el proyecto del diseño del banco de registros (página 185). Entregue una copia de la ventana temporal de la simulación que muestre el retardo de los componentes del Banco de registros involucrados en una operación de lectura por el puerto 1. Muestre en el mismo diagrama los retardos de los componentes involucrados en una operación de escritura. Para ello, modifique el programa de prueba si es necesario. Utilice los cursores para mostrar los retardos.



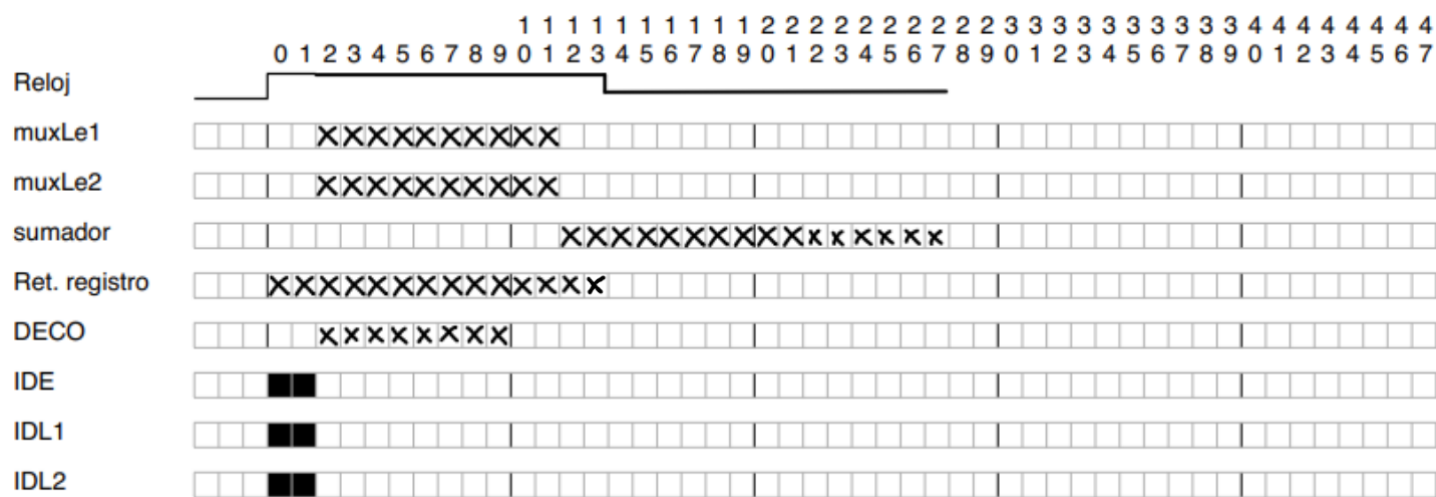
En el **Cursor1** es donde se prepara la operación de escritura y donde el IDE ya tiene el valor del puerto.  
 En el **Cursor2** es el instante donde varía el valor del decodificador que es  $58\text{ns} - 50\text{ ns} = \mathbf{8\text{ ns}}$ .  
 En el **Cursor3** es donde se empieza la escritura en sí.  
 En el **Cursor4** es donde acaba la escritura y podemos ver  $89\text{ ns} - 75\text{ ns} = \mathbf{14\text{ ns}}$ .  
 En el **Cursor5** es donde empieza la lectura del registro.  
 En el **Cursor6** es cuando ya sale el dato del registro por el multiplexor de salida  $110\text{ ns} - 100\text{ ns} = \mathbf{10\text{ ns}}$ .

**Pregunta 2:** Considere el módulo de control que genera la secuencia de operaciones con dependencias (página 196). Entregue una copia del esquema RTL de la unidad de control efectuada por Quartus. Identifique en el esquema la lógica de próximo estado del autómata principal y de los cuatro autómatas subordinados (página 192).

El autómata principal es el “**estadoreg**” en el esquema RTL i los autómatas subordinados son “**contador**”, “**idenL1**”, “**idenL2**” y “**idenE**”.



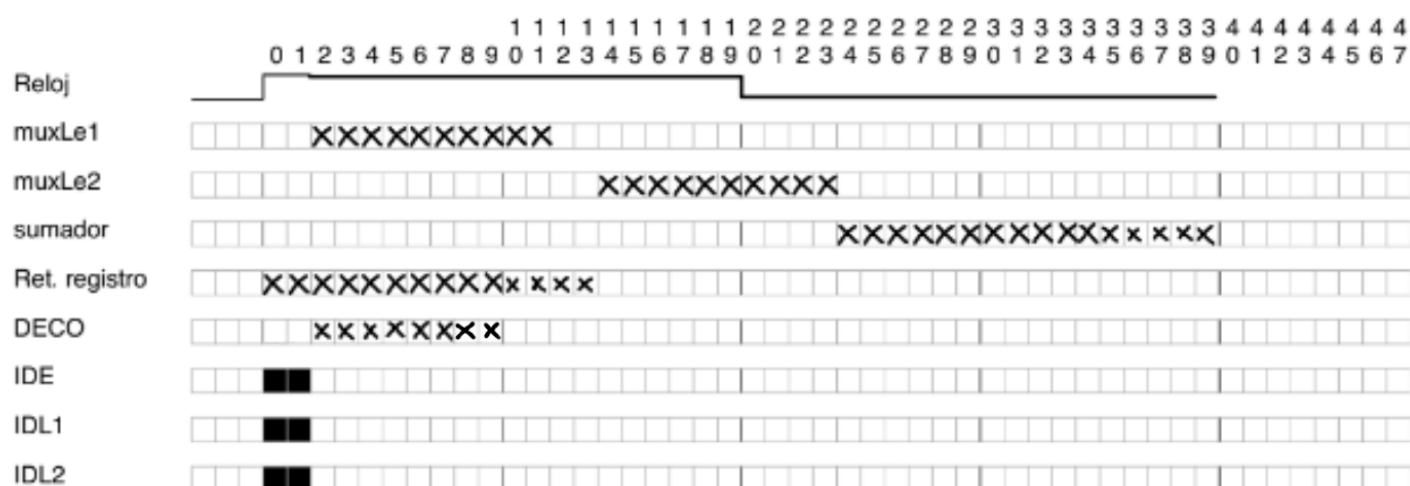
**Pregunta 3:** Suponga la secuencia de operaciones independientes (página 186). Marque, oscureciendo las casillas, los retardos de cada componente para el segundo ciclo de la secuencia de operaciones (IDL1 = 1, IDL2 = 10, IDE = 19). El retardo de un componente se empieza a marcar a partir del instante de tiempo en el cual todas las señales de entrada, que determinan el valor de su salida, son válidas. En un ciclo de la señal Reloj, entendemos que una señal es válida a partir del instante de tiempo en que no se modifica.



Tiempo de ciclo mínimo: **28 ns**

Dado que el primer paso es escribir el resultado en el registro anterior, pero en registros independientes de lectura el sumador puede comenzar con la suma antes de que se complete la escritura. Por tanto, el tiempo de ciclo depende de los retardos del decodificador, del multiplexor de lectura y del sumador.

**Pregunta 4:** Suponga la secuencia de operaciones dependientes (página 186). Marque, oscureciendo las casillas, los retardos de cada componente para el segundo ciclo de la secuencia de operaciones (IDL1 = 1, IDL2 = 10, IDE = 10).

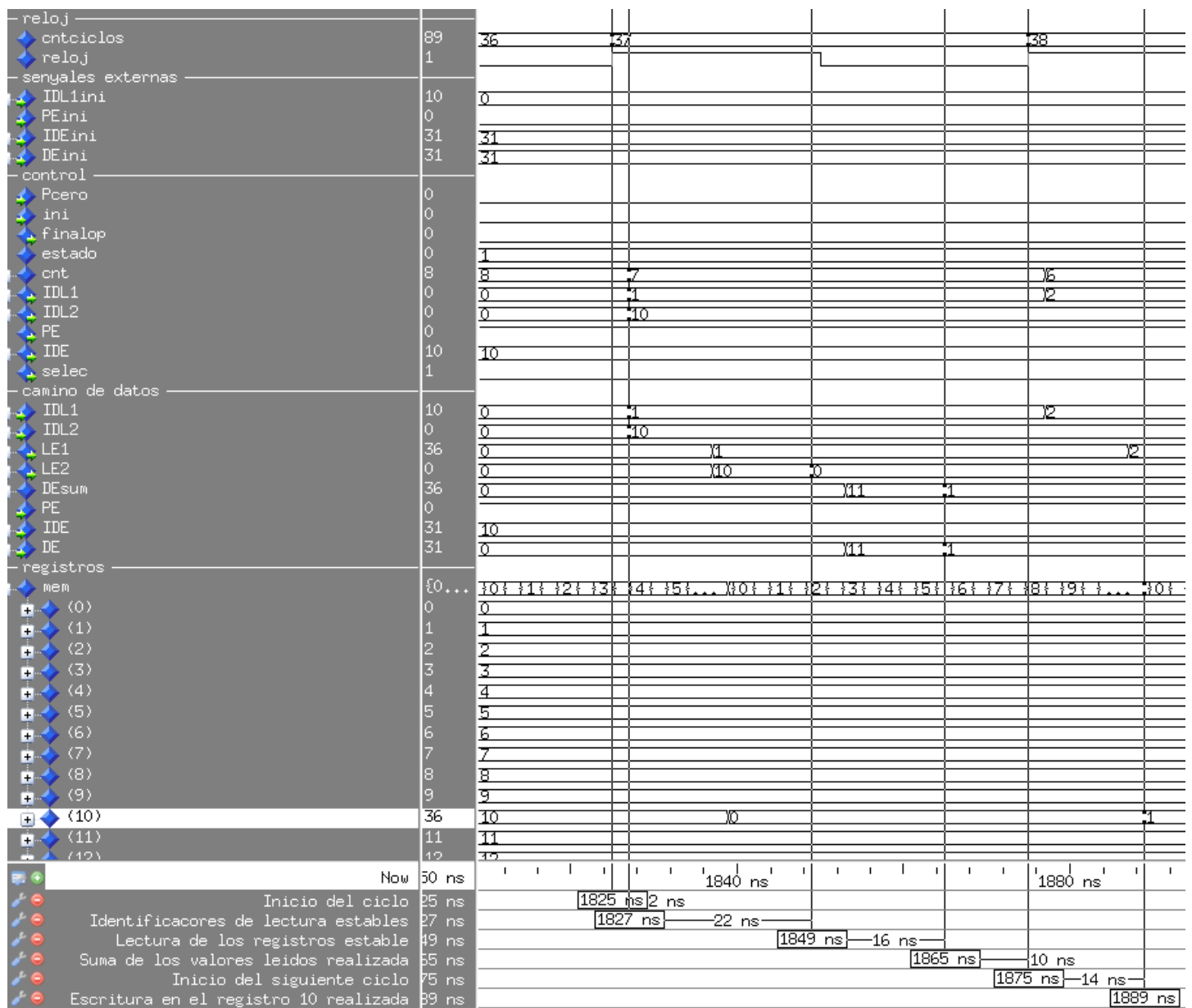


Tiempo de ciclo mínimo: **40 ns**

Dado que el primer paso es escribir el resultado en el registro anterior, debemos esperar a que escriba el resultado antes de leer el segundo el valor de el muxLe2 (el registro donde escribe el resultado anterior es el mismo de IDL2). Cuando tengamos los dos registros ya podremos hacer la suma en el sumador.

Por tanto, el tiempo de ciclo depende de los retardos de la escritura anterior, del multiplexor de lectura 2 y del sumador.

**Pregunta 5:** Entregue una copia del contenido de la ventana temporal que se obtiene con Modelsim cuando el camino de datos procesa la secuencia de operaciones dependientes (página 186). El objetivo es mostrar que el periodo del reloj es correcto. Por tanto, en la ventana de tiempo se debe poder identificar claramente el periodo de la señal de reloj. Identifique en el cronograma el ciclo correspondiente a la segunda operación y muestre los retardos de los componentes mediante cursores.



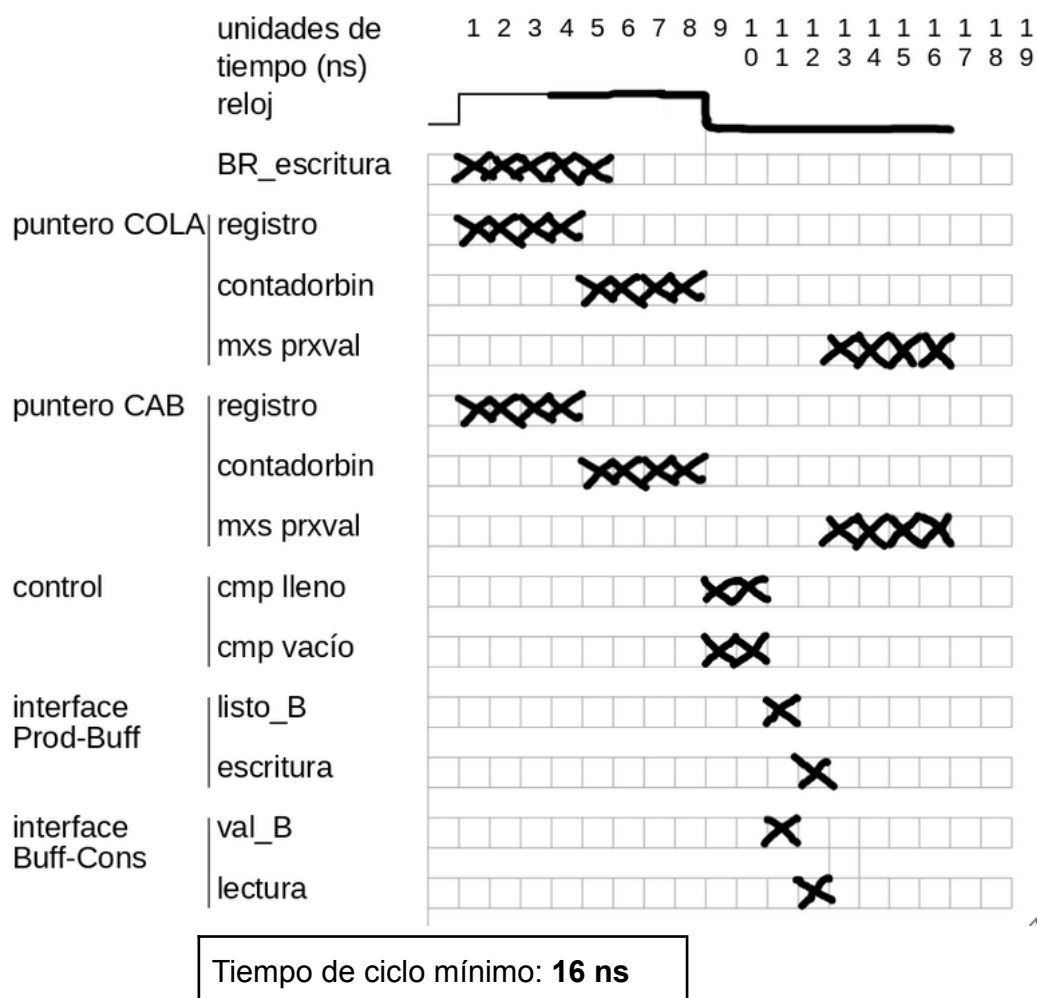
El primer cursor indica el inicio del ciclo. Dos nanosegundos después (segundo cursor) los identificadores de registro de lectura (IDL1 e IDL2) ya son estables en la entrada de los multiplexores de lectura. La lectura del banco de registros se estabiliza 22 ns después (tercer cursor) del paso anterior dado que el registro 10 (leído por LE2) aún está siendo modificado por la escritura provocada en el ciclo anterior. Podemos apreciarlo en el instante en que la salida en LE1 se estabiliza: hay un valor incorrecto (10) en la salida de LE2, que no es correcto hasta los 1849 ns. Después de 16 ns para obtener el cálculo correcto del sumador (cuarto cursor), vemos que el tiempo de ciclo mínimo es  $2\text{ ns} + 22\text{ ns} + 16\text{ ns} = 40\text{ ns}$ , tal y como habíamos calculado antes.

En el inicio del siguiente ciclo (penúltimo cursor) empieza la escritura en el registro 10, que no se completa hasta 14 ns después (último cursor).

**Pregunta 6:** En el programa de prueba asociado al diseño del elemento de acceso al banco de registros del buffer circular (página 205) se planifican 4 acciones de escritura. Sin embargo, al final de la simulación se observa que el valor del puntero cola es 5, por tanto, se efectúan 5 escrituras. Justifique este comportamiento. Analice los procesos estímulos y gen\_reloj del programa de prueba y los procedimientos involucrados.

Esto se debe a que el proceso gen\_reloj comprueba el valor de la variable *final* justo antes de generar un flanco descendente en la señal *reloj*, y como se da el valor a la variable *final* justo después de esperar a un flanco descendente, las señales de *lectura* y *escritura* se mantienen a 1 un ciclo más de lo esperado y por tanto los punteros aumentan su valor.

**Pregunta 7:** Indique el retardo necesario en una acción de actualización del buffer circular. Considere el diseño que no utiliza todas las entradas.



En el flanco ascendente de reloj se efectúa la escritura anterior en el BR y se actualizan los registros de los punteros. Una vez tienen el nuevo valor de cola y cabeza, el sumador binario calcula el próximo valor de éstos. Después del cálculo, el módulo de control ya tiene las tres señales con las que opera estables, y por tanto puede hacer la comparación correctamente.

Después de que se hayan estabilizado las señales *lleno* y *vacío*, se propagan por las puertas de las interfaces del búfer circular para actualizar *listo\_B* y *val\_B*, seguido de *escritura* y *lectura*,

David Cañadas López y Roger Ortega Castilló

respectivamente. Una vez estas últimas tengan su valor estable en la entrada de los multiplexores de los punteros, cuatro nanosegundos después (cada uno se compone de dos multiplexores seguidos) habrán calculado la salida correspondiente.