DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC2343 - Arquitectura de Computadores (II/2025)

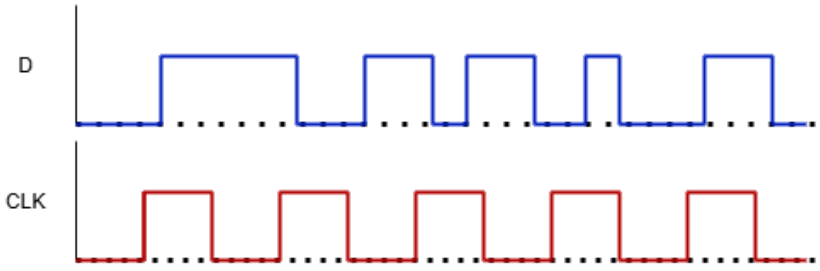
Guía de Ejercicios: Almacenamiento de datos

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Nicolás Romo Aubele (nroma@uc.cl)

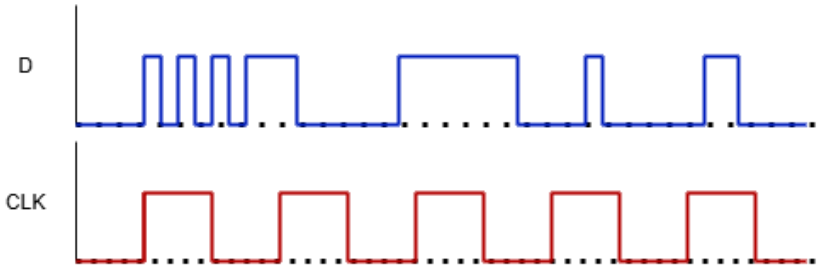
Nota: La gran mayoría de ejercicios de esta guía pueden tener más de una solución, por lo que se debe considerar esto al momento de realizarlos, dado que las soluciones mostradas corresponder a una manera de abordarlos.

Pregunta 1: Latches y flip-flops

A continuación se presentan dos diagramas en los cuales se muestran dos señales: D y CLK, estas representan una señal de entrada en un instante dado y una señal de reloj respectivamente. Para cada uno de ellos dibuje las señales de salida de un Latch D y un Flip-flop D. Puede asumir que ambas salidas parten en 0, es decir, nada ha ocurrido antes.



Pregunta a)



Pregunta b)

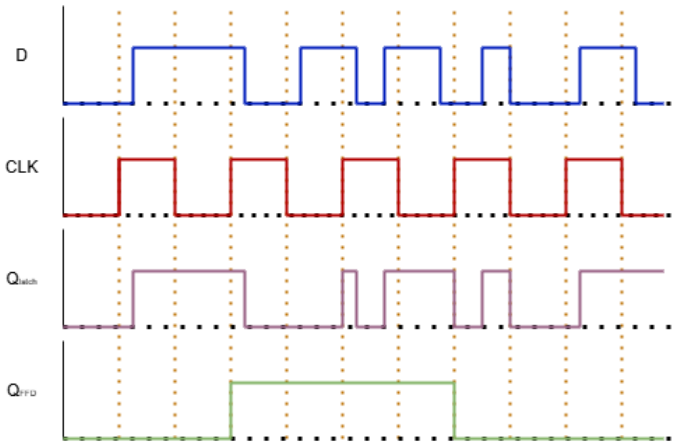
Solución:

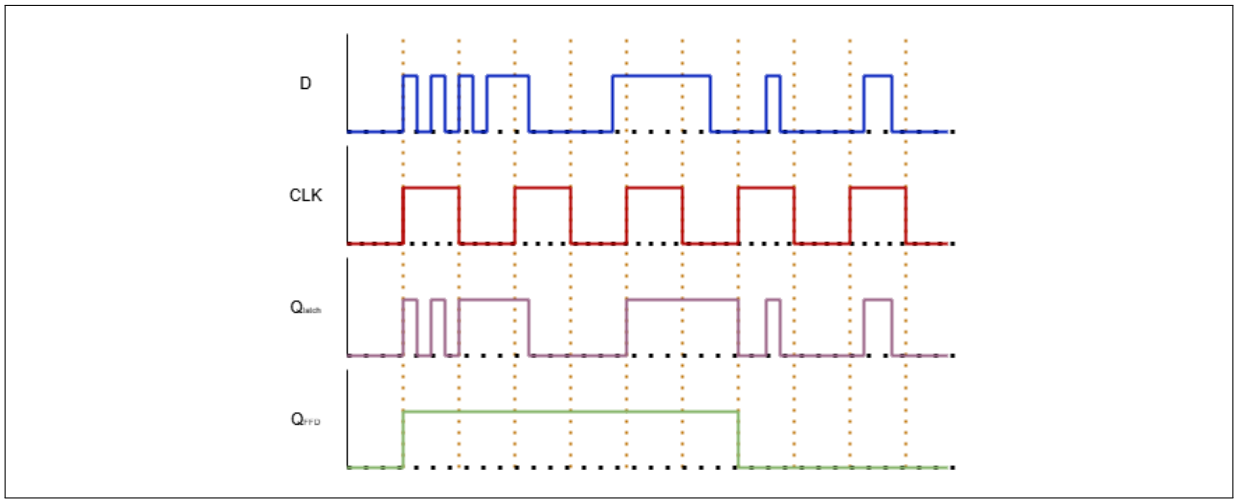
Nota: Este tipo de ejercicios no es típico de pruebas, pero es útil para entender el comportamiento y la relación que tienen ambos componentes con el CLK, y cómo guardan su información, por lo que, de todas maneras es útil comprenderlo.

Para comprender la solución de estos ejercicios es útil conocer qué son, y cómo se comportan estos dos componentes. En primer lugar se puede entender un latch como un dispositivo con memoria que tiene la capacidad de guardar el estado anterior de una señal. Para el caso particular del latch D, si la señal de control es 0, entonces su salida se mantiene igual, sin embargo, si señal de control es 1, entonces la señal de salida tomará automáticamente el valor de la señal de entrada D.

Sin embargo, es posible observar que el latch D tiene una dependencia muy directa de la señal de control, puesto que guardará la información de la entrada cada vez que la señal de control esté activada lo cual genera que su valor almacenado pueda tener una gran variación durante el tiempo. Por lo tanto, es natural que pueda surgir la duda de ¿cómo almacenar de forma más controlada un dato? Para ello, serán de utilidad los flancos de subida, que aparecen en la señal CLK.

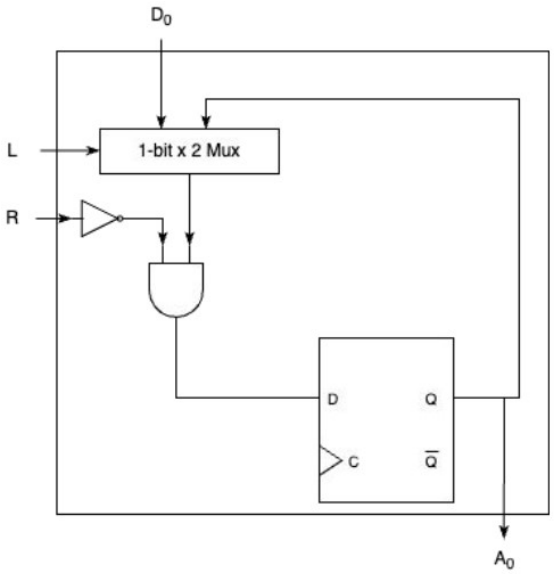
En particular, se puede observar que si la señal de entrada D es muy variable entonces esta podría, eventualmente, cambiar mientras la señal CLK=1, para evitar esto se construyen los flip-flops, en el caso del curso el más estudiado es el flip-flop D, este se construye a partir de dos latch D. La particularidad que tienen estos componentes es que permiten que este cambio de valor ocurra **solo** en el instante en que la señal CLK pasa `de 0 a 1`. Con ello se se puede lograr un mayor control en el contexto de almacenamiento de memoria, dado que el cambio en el dato almacenado sólo se dará en el flanco de subida, y no en todo el tiempo en que la señal CLK sea igual a 1. Una vez mencionado este marco teórico, a continuación se muestran las salidas de los componentes para ambos casos:





Pregunta 2: Modificaciones de un registro

Considere la siguiente estructura de un registro, la cual corresponde a la vista en clases:



A partir de ella, realice las siguientes modificaciones a la **estructura interna** de este para que:

1. El valor por defecto al momento de hacer *Reset* sea 1 y en lugar de 0.
2. Al momento de tener las señales *Load* y *Reset* en 1, se conserve el estado.
3. Agregando una nueva señal N, al momento de que esta tome el valor 1 el registro invierta su valor almacenado. Es decir, si tenía guardado 1 este pasa a ser un 0, y viceversa.

Haga cada modificación de forma independiente.

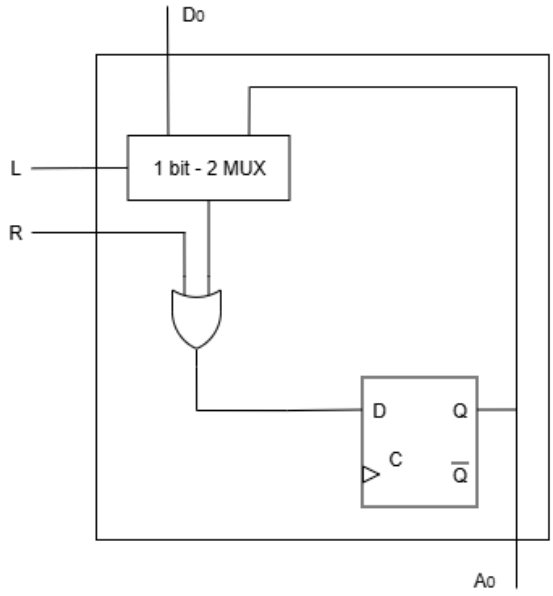
Solución:

Modificación 1

En este caso, se puede notar que, en el registro original, dado que existen un `NOT` y un `AND`, siempre que se le entregue la señal $R = 0$ la salida del `AND` será el valor seleccionado por el MUX entre el valor de entrada y el valor almacenado, denotemos a esta salida como M. Sin embargo, es posible notar que cambiando estas compuertas por una única puerta `OR` que conecte la señal directamente la señal R con M, entonces se puede obtener el comportamiento requerido, para comprender esto puede ser de utilidad la siguiente tabla:

R	M	$L \vee M$
0	0	0
0	1	1
1	0	1
1	1	1

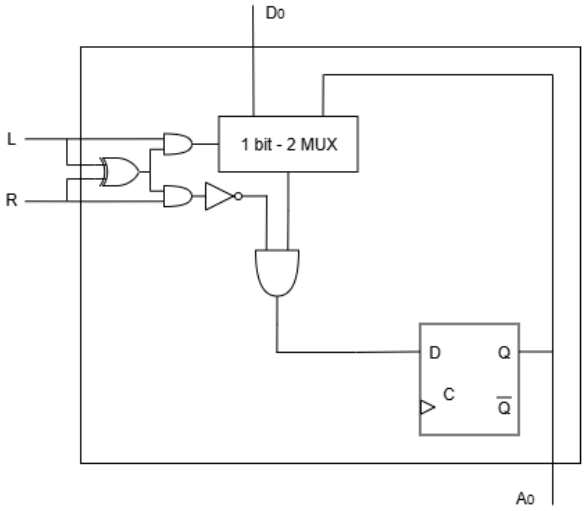
Note que en ella, si la señal R es 0, entonces el valor almacenado no sufre cambios, sin embargo, si la señal es 1 en ambos casos se guarda un 1, es decir, el valor por defecto cuando se hace *Reset* es 1, que justamente es el comportamiento deseado. El diagrama de esta modificación se muestra a continuación:



Modificación 2

Para esta pregunta es importante recordar que el registro visto en clases selecciona su valor almacenado Q si la señal de selección $L = 0$, mientras que si $L = 1$ se elige el valor de la entrada D.

Considerando lo anterior, se puede notar que el si las señales L y R son conectadas a una compuerta `XOR` y, a su vez, la salida de esta compuerta se conecta a cada una de las señales de entrada, entonces se logra el comportamiento deseado, esta idea se puede observar en el siguiente diagrama:



En el diagrama se puede notar que si ambas señales son 1, entonces la salida de la compuerta `XOR` será 0, lo cual inhabilitará ambas señales con los `AND`, es decir, de ambas sale un 0, por lo que el registro seleccionará el valor que tenía guardado y la señal de R hará que este vuelva a guardarse en el.

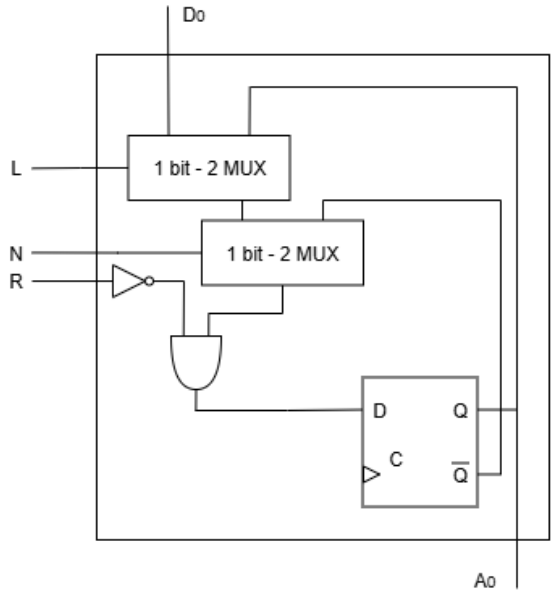
Modificación 3

Note que en este caso, a diferencia de los anteriores, se pide agregar una nueva señal N, la cual nos indica que queremos guardar el valor inverso del registro, por lo tanto, esto nos puede idea de como abordar el problema.

En particular, una forma en la que se puede se puede conseguir el funcionamiento pedido es conectando la salida del multiplexor seleccionado por L a un nuevo multiplexor, el cual tiene como señal de selección N, y como otra entrada el valor $\overline{Q}$ (Recuerde que los flip-flop D también tienen su salida negada, que generalmente no se usa, pero en este caso nos es de utilidad). Este multiplexor selecciona con el $N = 0$ el valor de Q y con $N = 1$ el valor de $\overline{Q}$.

Luego, se conecta la salida del multiplexor N a la compuerta `AND`, y con ello se logra lo pedido

en el enunciado. Esto se puede ver más directamente en el siguiente diagrama:



Note que si la señal $N = 0$, entonces se guardará el valor de Q, mientras que si $N = 1$ el nuevo MUX seleccionará el valor de opuesto, por lo que ese será el valor que entra en la compuerta `AND`.

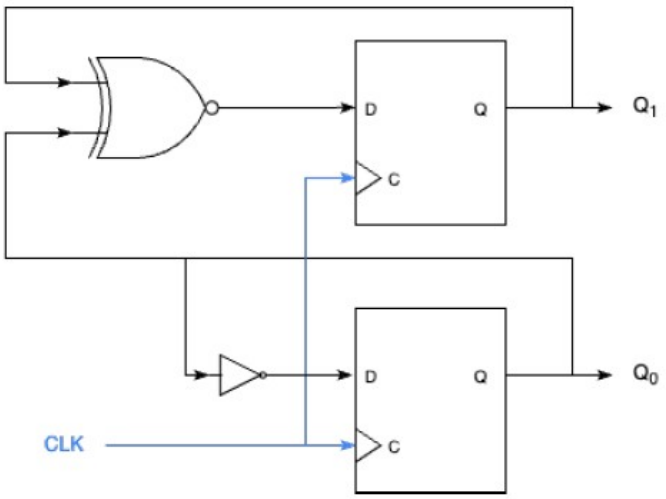
Pregunta 3: I1 2024-1 Pregunta 2

Diseñe un contador secuencial de 2 bits que se decrementa con cada flanco de subida de la señal de control. Este contador, además, debe recibir una señal de entrada B de 1 bit, correspondiente al valor del botón del *timer*. Si el valor de B es igual a 1 durante el flanco de subida de la señal de control (i.e., el botón está presionado), entonces el contador debe actualizar su valor a 3 en vez de decrementarse en una unidad.

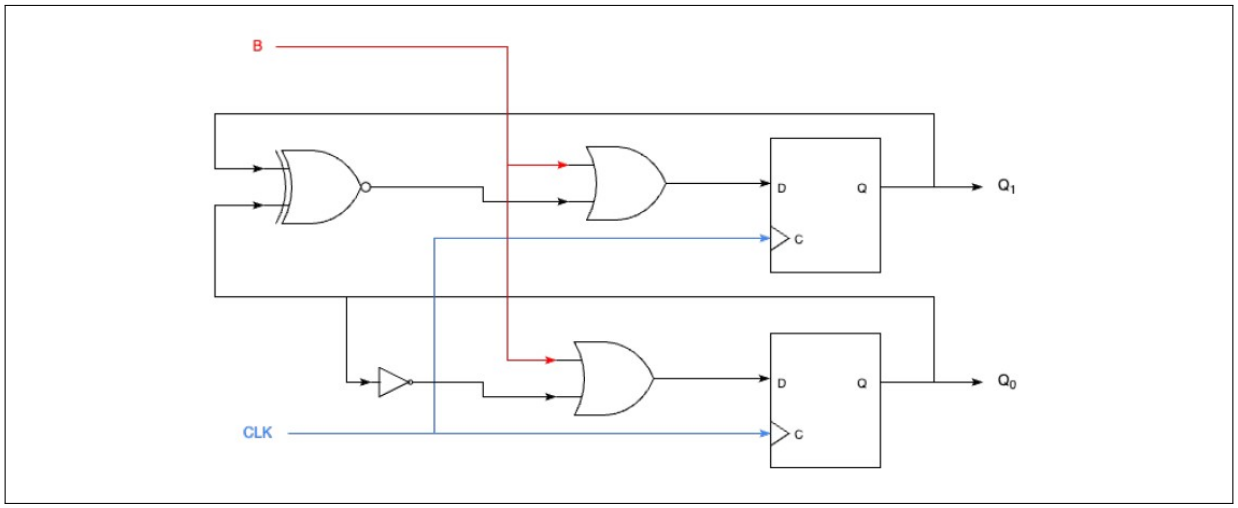
Solución: Si se denota el estado del contador en un ciclo t de la señal de control CLK como $Q_1^t Q_0^t$, y su estado en el próximo ciclo, es decir en $t+1$ como $Q_1^{t+1} Q_0^{t+1}$, entonces es posible elaborar la siguiente tabla de verdad en la cual se denota la relación que debe existir entre ambos estados:

Q_1^t	Q_0^t	Q_1^{t+1}	Q_0^{t+1}
1	1	1	0
1	0	0	1
0	1	0	0
0	0	1	1

En ella se puede observar que para cada estado el siguiente es su valor decrementado en uno. Además, de ella es posible deducir que $Q_0^{t+1} = \neg Q_0^t$, mientras que $Q_1^{t+1} = (Q_0^t \oplus Q_1^t)$. Es decir, Q_0 será igual a $\neg Q_0$, mientras que Q_1 será igual a Q_0 XNOR Q_1. Haciendo uso de flip-flops y las compuertas antes señaladas, se obtiene al siguiente diagrama:

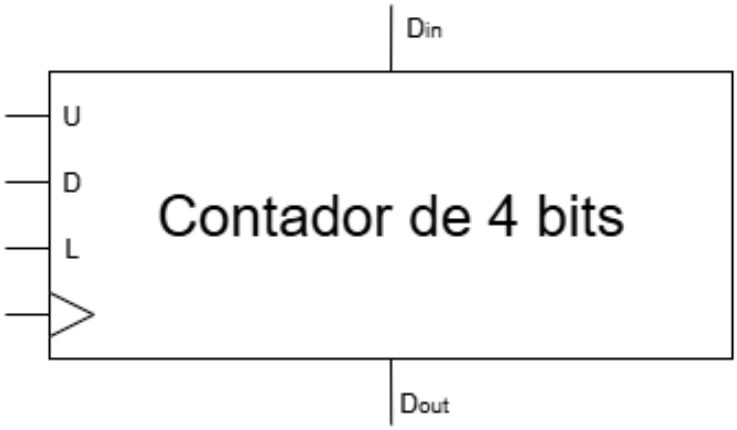


Note que el circuito anterior funciona, es decir, en cada ciclo de clock decrementa en una unidad su valor guardado, sin embargo, todavía falta considerar la señal B. Del enunciado se sabe que si $B_t = 1$, entonces $Q_1^{t+1} Q_0^{t+1} = 11$. Por lo tanto, haciendo uso de la ley de dominación, se pueden ajustar las expresiones de los estados del contador de la siguiente forma: $Q_0^{t+1} = B_t \vee Q_0^t$; y $Q_1^{t+1} = B_t \vee (Q_0^t \oplus Q_1^t)$. Por lo que, finalmente, el siguiente diagrama representa el contador modificado conectado a la señal B:



Pregunta 4: Tarea 1 2023-1

Considere que tiene el siguiente contador de 4 bits, el cual tiene señales de *Up* y *Down* las cuales permiten que este aumente o disminuya en uno su valor almacenado.



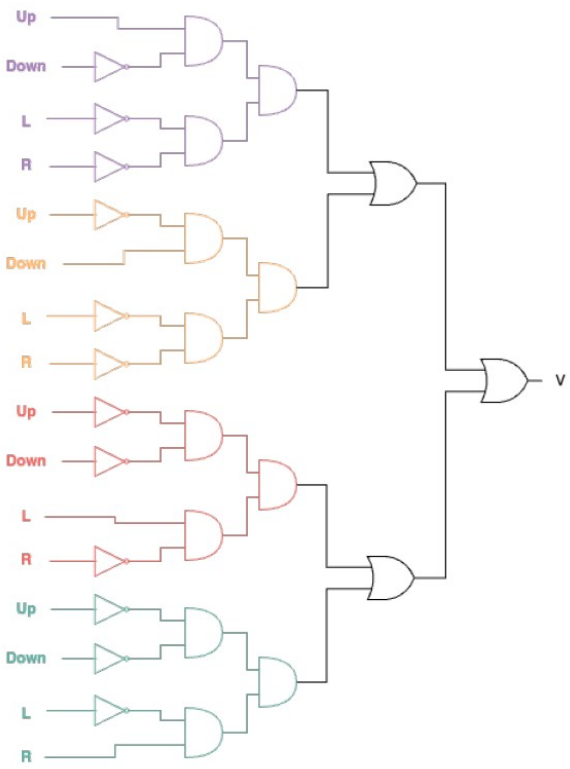
Extiéndalo para que cualquier entrada de señales inválida preserve el estado del componente. Considere que una entrada inválida es cualquier combinación de señales que implique más de una modificación sobre el valor registrado. Por ejemplo: $Up = 1$, $Down = 1$ es una combinación inválida ya que el registro no puede incrementar y disminuir su valor al mismo tiempo.

Solución: Para abordar esta pregunta es importante considerar en qué casos se quiere que el contador funcione correctamente, en particular, estos casos son:

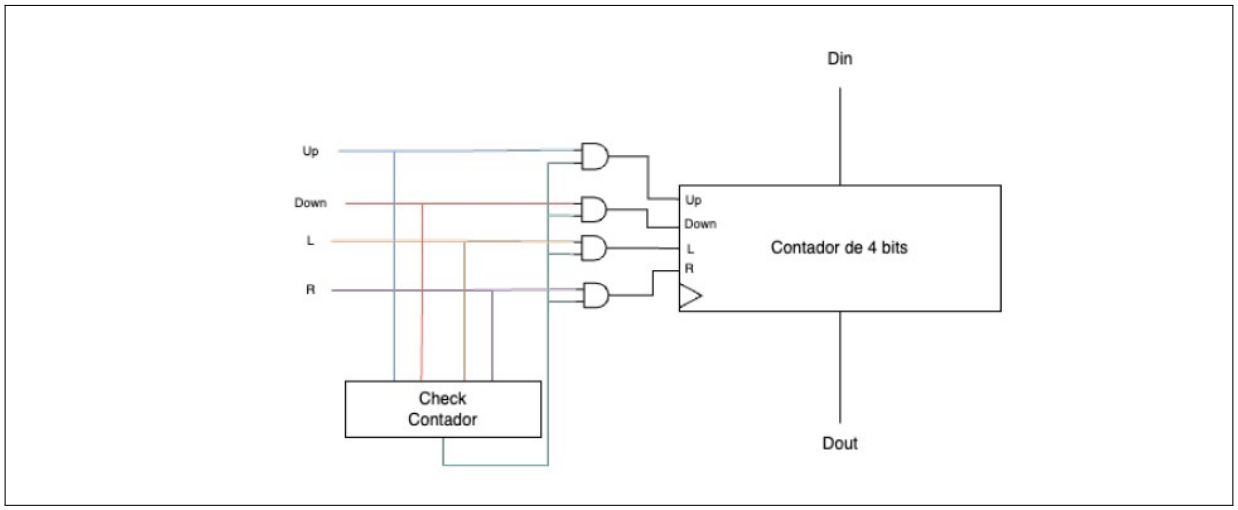
- La señal $Up = 1$, $Down = 0$ y el resto 0.

- La señal $Up = 0$, $Down = 1$ y el resto 0.
- La señal $Load = 1$ y el resto 0.
- La señal $Reset = 1$, y el resto 0.

En todos estos casos necesitamos que el contador funcione de manera normal, mientras que en cualquier otra combinación de señales, el contador deberá mantener su valor. A continuación se puede observar el siguiente circuito, el cual entrega en su salida V un bit, el cual tiene valor 1 si se cumple una de las condiciones mencionadas anteriormente, mientras que si ninguna de estas se cumple, entonces el valor de este bit será 0:

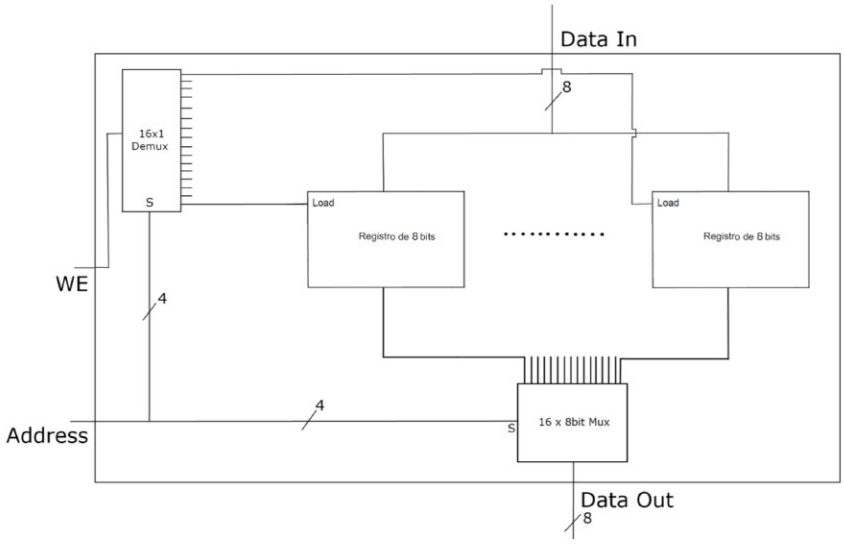


Note que, con la inclusión de las puertas `AND`, se logra el comportamiento esperado, debido a que si ninguna de las condiciones mencionadas se cumple, entonces todas tendrán valor 0, mientras que si hay alguna que se cumple, entonces la salida final tendrá valor 1. Finalmente, es posible observar que el componente que recién construido puede ser usado como un mecanismo de protección del contador, pues si la salida V de este se conecta a cada una de las entradas por medio de una compuerta `AND`, entonces las entradas mantendrán su valor solamente si se cumplen las condiciones. La integración de este componente para generar la protección del registro se puede observar en la siguiente figura:



Pregunta 5: I1 2023-2 Pregunta 2

A continuación, se adjunta un posible diagrama interno de una RAM de 16 direcciones:



Modifique este diagrama para que reciba entradas $\texttt{DataIn}_1$; $\texttt{DataIn}_2$; $\texttt{Address}_1$; $\texttt{Address}_2$; $\texttt{WE}_1$; $\texttt{WE}_2$, y salidas $\texttt{DataOut}_1$ y $\texttt{DataOut}_2$ para leer y escribir datos en dos direcciones de memoria distintas de forma simultánea.

Solución: Lo que se pide, en resumen, es crear una RAM que permita el acceso simultáneo a dos direcciones de memoria distintas. Para ello, se deben realizar los siguientes cambios:

- Se agrega un segundo MUX cuya señal de selección es igual a $\texttt{Address}_2$ y recibe como entradas los valores de todos los registros internos. Esto permite escoger una segunda salida $\texttt{DataOut}_2$.
- Se agrega un segundo DEMUX cuya señal de selección es igual a $\texttt{Address}_2$ y recibe como entrada la señal $\texttt{WE}_2$. Esto permite transmitir solo a uno de los 16 registros la segunda señal de escritura.
- Como cada registro recibe dos señales de escritura, se conectan ambas a una compuerta `OR`. Se habilitará la escritura sobre un registro si, y solo si $\texttt{WE}_1$ está activo para su dirección $\texttt{Address}_1$ o si $\texttt{WE}_2$ está activo para su dirección $\texttt{Address}_2$.
- Para manejar las dos entradas $\texttt{DataIn}_1$ y $\texttt{DataIn}_2$, se conectan cada una de ellas a un DEMUX para transmitir su valor solo a uno de los 16 registros.
- Como cada registro recibe dos valores de entrada, se conectan ambos a una compuerta `OR` de 8 bits. Como las direcciones de memoria son **distintas**, nunca se tendrán dos valores de entrada distintos de 0. Por lo tanto, la compuerta cumple con seleccionar solo el valor de escritura transmitido si corresponde.

Las modificaciones anteriores se ven reflejadas en el siguiente diagrama: