

VERILOG

Lab #1

Sumador binario sincrónico de 4, 8 y 32 bits

Especificaciones

1. Sumador binario sincrónico de 4 bits, con los siguientes modos de funcionamiento:
 - a) Suma
 - b) Resta
 - c) Mantener valor anterior
 - d) Limpiar el sumador
2. Entradas del contador:
 - a) **CLK** – Entrada de reloj del sumador. El flanco activo de la señal *CLK* es el flanco creciente. Entonces, con cada flanco positivo del reloj el sumador cambia de estado dependiendo del estado de las señales de *MODO* si la señal *ENB*=1.
 - b) **ENB** – Entrada de habilitación del sumador. Si *ENB*=1 el sumador funciona normalmente respondiendo a los flancos activos de *CLK* para cambiar de estado de acuerdo a la señal *MODO*. Si *ENB*=0 el contador mantiene su estado actual sin importar los flancos de *CLK*.
 - c) **MODO[1:0]** – Entrada de modo que consta de dos líneas y sirve para definir cuál será el próximo estado del sumador al llegar el flanco activo del reloj en la entrada *CLK*. Si el sumador se encuentra en el estado *Q* antes del flanco activo del reloj, luego del flanco activo, su estado será:
 - *MODO* = 00 → *Q*
 - *MODO* = 01 → *A+B*
 - *MODO* = 10 → *A-B*
 - *MODO* = 11 → 0
 - d) **A[3:0]** – Entrada de datos *A* consta de 4 líneas. El valor que tengan las entradas *A[3:0]* corresponde al primer operando de la suma o resta, según la señal de *MODO* en el flanco activo de *CLK* si *ENB* = 1.
 - e) **B[3:0]** – Entrada de datos *B* consta de 4 líneas. El valor que tengan las entradas *B[3:0]* corresponde al segundo operando de la suma o resta, según la señal de *MODO* en el flanco activo de *CLK* si *ENB* = 1.
3. Salidas del contador:
 - a) **Q[3:0]** – Salida *Q* que consta de cuatro líneas que indican el estado presente del sumador. El estado del sumador cambia con el flanco activo de la señal *CLK* mientras *ENB*=1 y de acuerdo con el modo seleccionado con las líneas *MODO[1:0]*.
 - b) **RCO** – Salida de llevo “*Ripple-Carry Out*” que indica cuando el sumador llega a su cuenta límite para que la siguiente etapa, en sumadores de más de 4 bits, se habilite para que realice su actualización de estado. Note que *RCO* se pone en 1 dependiendo de la señal *MODO* (acarreo o rebase).

Trabajo a realizar sobre el dispositivo a diseñar

1. Escribir una descripción conductual del sumador de 4 bits usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado.
2. Usando varias instancias del sumador de 4 bits, construya un sumador de 8 bits y otro de 32 bits.
3. La descripción en Verilog deberá tener al menos un módulo de banco de pruebas, un módulo probador, un módulo con la descripción del sumador de 4 bits y un módulo parametrizado para construir contadores de 8 bits o 32 bits.
4. Definir un plan de pruebas para garantizar el funcionamiento del diseño. El plan de pruebas debe cubrir todos los modos de operación del sumador. En la sección de abajo **“Propuesta de Plan de Pruebas Mínimo”** se da una idea del conjunto mínimo de pruebas que debería tener. El módulo probador debe suministrar las señales necesarias para que las pruebas se realicen.

Propuesta de Plan de Pruebas Mínimo

El conjunto de pruebas mínimas que se deben hacer para validar el diseño conductual solicitado por lo menos debe incluir:

1. **Prueba #1, suma de 4 bits.** El sumador inicia en cero y se definen valores para las entradas A y B. Se envía un flanco activo de reloj y el sumador realiza la operación correspondiente. Se debe repetir este proceso para varias posibles sumas, incluyendo casos que generen acarreo, para verificar el correcto funcionamiento de la señal RCO.

Un ejemplo de cómo se puede estructurar la prueba se muestra en la siguiente secuencia:

1. Establecer $MODO[1:0]=11$ para limpiar el contador.
 2. Poner $ENB=1$.
 3. Enviar flanco activo en CLK . Con esto se pone el contador a cero.
 4. Establecer $MODO[1:0]=01$. Pone modo de suma.
 5. Enviar flanco activo en CLK . El estado de contador debería pasar a $Q=A+B$.
 6. Variar los valores de A y B para observar distintos resultados, incluyendo casos con acarreo donde $RCO=1$.
-
2. **Prueba #2, resta de 4 bits.** El sumador inicia en cero y se definen valores para las entradas A y B para ejecutar la resta correspondiente. También se debe verificar que la señal RCO se pone en 1 cuando se cumple la condición de rebase, ¿en qué estado debería ponerse RCO cuando A es menor que B?
 3. **Prueba #3, mantener el valor en modo 00.** El sumador realiza una operación inicial y se modifica el modo para ponerlo en 00. Luego se envían varios flancos positivos del reloj y se verifica que la salida se mantiene con el resultado de la última operación realizada.
 4. **Prueba #4, mantener el valor cuando $ENB = 0$.** El sumador realiza una operación inicial y después de completada se pone $ENB=0$. Luego se envían varios flancos positivos del reloj y se verifica que la salida se mantiene con el resultado de la última operación realizada.

5. **Prueba #5, limpiar el contador.** Verificar que si el contador se encuentra en cualquier estado inicial posible, es posible regresar su salida a cero poniendo la señal de $MODO=2'b11$ y sosteniéndola durante un ciclo de reloj.
6. **Prueba #6, sumador de 8 bits.** Construya un sumador de 8 bits utilizando dos contadores de 4 bits de los ya probados. Verifique que el sumador de 8 bits funciona en todos los modos de funcionamiento del sumador de 4 bits. Para esto diseñe una prueba mínima dado que las pruebas de la #1 a la #5 ya cubren una buena porción de la funcionalidad esperada.
7. **Prueba #7, sumador de 32 bits.** Construya un sumador de 32 bits utilizando ocho contadores de 4 bits de los ya probados o bien 4 contadores de 8 bits. Verifique que el sumador de 32 bits funciona en todos los modos de funcionamiento del sumador de 4 bits. Para esto diseñe una prueba mínima dado que las pruebas de la #1 a la #5 ya cubren una buena porción de la funcionalidad esperada.