

# VERILOG

## Lab #2

### Descripción conductual de un contador binario síncronico

#### Especificaciones

1. Contador binario síncronico de 4 bits, con los siguientes modos de funcionamiento:
  - a) Cuenta hacia arriba.
  - b) Cuenta hacia abajo.
  - c) Cuenta de tres en tres hacia abajo.
  - d) Carga en paralelo.
2. Entradas del contador:
  - a) **CLK** – Entrada de reloj del contador. El flanco activo de la señal *CLK* es el flanco creciente. Entonces, con cada flanco positivo del reloj el contador cambia de estado dependiendo del estado de las señales de *MODO* si la señal *ENB*=1.
  - b) **ENB** – Entrada de habilitación del contador. Si *ENB*=1 el contador 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 contador al llegar el flanco activo del reloj en la entrada *CLK*. Si el contador se encuentra en el estado *Q* antes del flanco activo del reloj, luego del flanco activo, su estado será:
    - $MODO = 00 \rightarrow Q + 1$
    - $MODO = 01 \rightarrow Q - 1$
    - $MODO = 10 \rightarrow Q - 3$
    - $MODO = 11 \rightarrow D$
  - d) **D[3:0]** – Entrada de datos *D* consta de 4 líneas. El valor que tengan las entradas *D*[3:0] será almacenado en *Q*[3:0] 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 contador. El estado del contador 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 contador llega a su cuenta límite para que la siguiente etapa, en contadores 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*.

## Trabajo a realizar sobre el dispositivo a diseñar

1. Escribir una descripción conductual del registro desplazable usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado.
2. La descripción en Verilog deberá tener al menos un módulo de banco de pruebas, un módulo probador, y un módulo con la descripción del contador. Use Icarus Verilog para hacer esto.
3. Definir un plan de pruebas para garantizar el funcionamiento del diseño. El plan de pruebas debe cubrir todos los modos de operación del registro desplazable. 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, cuenta ascendente.** El contador inicia en un valor predeterminado y se le mandan suficientes flancos activos de reloj para que haga una secuencia completa de estados. También se debe verificar que la señal RCO se pone en 1 cuando el contador alcanza el final de la cuenta, antes de repetir la secuencia de nuevo. Un ejemplo de cómo se puede estructurar la prueba se muestra en la siguiente secuencia:
  1. Establecer  $MODO[1:0]=11$  y  $D[3:0]=0000$ . Pone modo de cargar estado.
  2. Enviar flanco activo en  $CLK$ . Con esto se pone el contador a cero.
  3. Establecer  $MODO[1:0]=00$ . Pone modo de cuenta ascendente.
  4. Enviar flanco activo en  $CLK$ . El estado de contador debería pasar a  $D[3:0]=0001$ .
  5. Enviar 15 flancos activos en  $CLK$ . Esto debería hacer que el contador por cada uno de los siguientes 15 estados y la cuenta final quede en  $D[3:0]=0000$ . Se espera que cuando  $D[3:0]=1111$  la señal  $RCO$  se haga 1. El resto del tiempo estaría en cero.
2. **Prueba #2, cuenta descendente.** El contador inicia en un valor predeterminado y se le mandan suficientes flancos activos de reloj para que haga una secuencia completa de estados. También se debe verificar que la señal RCO se pone en 1 cuando el contador alcanza el final de la cuenta, antes de repetir la secuencia de nuevo. ¿En qué estado debería ponerse  $RCO=1$  para este modo de cuenta?
3. **Prueba #3, cuenta descendente de tres en tres.** El contador inicia en un valor predeterminado y se le mandan suficientes flancos activos de reloj para que haga una secuencia completa de estados. También se debe verificar que la señal RCO se pone en 1 cuando el contador alcanza el final de la cuenta, antes de repetir la secuencia de nuevo. ¿En qué estado debería ponerse  $RCO=1$  para este modo de cuenta? ¿Esto ocurre para un único estado o varios?
4. **Prueba #4, carga en paralelo.** Verificar que si el contador se encuentra en cualquier estado inicial posible, es posible cargarle cualquier valor que se coloque en las entradas  $D$ . Tome en cuenta la posibilidad de usar un estado indeterminado, como estado inicial  $D[3:0]=XXXX$ , antes de cargar cualquier valor en  $D$ .
5. **Prueba #5, contador de 16 bits.** Construya un contador de 16 bits utilizando cuatro contadores de 4 bits de los ya probados. Verifique que el contador de 16 bits funciona en

todos los modos de funcionamiento del contador de 4 bits. Para esto diseñe una prueba mínima dado que las pruebas de la #1 a la #4 de arriba ya cubren una buena porción de la funcionalidad esperada.