

### Tarea #1

(Entrega 11 de setiembre del 2022)

#### Descripción conductual de un contador binario sincrónico

#### Especificaciones

1. Contador binario sincrónico 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.

### Rúbrica de Calificación

Tarea #1: Descripción conductual de un contador binario síncrono	Categoría	% Categoría	% Rubro	% Total
Existe una descripción conductual en Verilog del contador de 4 bits. Esta descripción incluye al menos un módulo de banco de pruebas (testbench.v), un módulo probador (tester.v) y un módulo para el dispositivo bajo prueba (DUT).	Código	20%	20%	4%
Se construye un contador de 16 bits a partir del módulo de 4 bits y se incluye los módulos adicionales necesarios para probar su correcto funcionamiento.	Código	20%	20%	4%
Las descripciones de Verilog se entregan en archivos distintos al reporte, listos para ser simulados, e incluyen un archivo de Makefile, de modo que la simulación se corre con una sola línea de comando.	Código	20%	10%	2%
Las descripciones en Verilog están comentadas adecuadamente para que otras personas entiendan la lógica de la descripción.	Código	20%	10%	2%
Las descripciones en Verilog compilan sin producir errores.	Código	20%	20%	4%
Las descripciones en Verilog ejecutan correctamente. Es decir, corren, entregan algunos resultados y finalizan.	Código	20%	20%	4%
El dispositivo completa una operación de cuenta ascendente, de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	10%	6%
El dispositivo completa una operación de cuenta descendente, de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	10%	6%
El dispositivo completa una operación de cuenta de tres en tres, de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	10%	6%
El dispositivo completa una operación de carga en paralelo, de forma correcta de acuerdo con la especificación dada.	Pruebas	60%	10%	6%
El módulo de 16 bits completa correctamente todas las operaciones descritas en la especificación.	Pruebas	60%	60%	36%
El reporte contiene las siguientes secciones: Resumen, descripción arquitectónica, plan de pruebas, instrucciones de utilización de la simulación para quien califica, ejemplos de resultados, conclusiones y recomendaciones.	Reporte	20%	40%	8%
El reporte explica con claridad los detalles relevantes del diseño particular que se hizo, las partes del diseño que dieron más trabajo para completar y por qué, una explicación de los problemas que se presentaron y cómo se solucionaron.	Reporte	20%	40%	8%
La longitud del reporte no excede 10 páginas.	Reporte	20%	20%	4%

## Guía para el reporte

Se debe entregar en forma electrónica un documento, a lo sumo de 10 páginas de longitud, que incluya los siguientes puntos:

1. **Resumen:** Breve (Media página máximo) descripción de todo el proyecto. Esta sección es fundamental pues puede determinar si el lector se interesa o no en leer los detalles del proyecto. Un resumen mal hecho puede esconder un excelente proyecto. El resumen debería incluir:
  - a) Descripción breve del sistema, es decir, qué hace. Incluya alguna característica que considere que distingue este diseño en particular.
  - b) Las pruebas que se realizaron y qué resultados se obtuvieron. Indique problemas que se tuvieron que considere importante resaltar.
  - c) Conclusiones más importantes y recomendaciones para un diseño posterior.
2. **Descripción Arquitectónica:** Incluye un diagrama de bloques con las señales más importantes que sirve como base para describir el funcionamiento del sistema. La descripción va en términos de lo que se espera que el sistema haga. Es decir, se debe detallar la funcionalidad del sistema, el protocolo de las señales que se usan para que funcionen cada una de las partes y las secuencias de eventos que se deben dar. Esta descripción podría ir acompañada de tablas de verdad, tablas de transición de estados, diagramas de estados, diagramas temporales, etc.
3. **Plan de Pruebas:** Aquí se deben enumerar, esto es, se debe presentar una **lista detallada** de las pruebas que se le van a hacer al diseño para verificar que está funcionando de acuerdo a las especificaciones dadas. La lista debe contener por lo menos los siguientes elementos i) Nombre/número de prueba, ii) Descripción de la prueba, y iii) Una indicación de si el diseño la falló o la pasó. Estas pruebas podrían incluir la generación de vectores de entrada para probar en forma exhaustiva todas las líneas de una tabla de verdad o tabla de estados, patrones aleatorios de entradas para tratar de causar errores en la respuesta del diseño, o patrones específicos que ejerciten un cierto modo de funcionamiento. Cada prueba debería ser claramente enumerada en el plan para que también se pueda hacer referencia a ella en el código del banco de pruebas del diseño.
4. **Instrucciones de utilización de la simulación:** Esta sección debe mostrar los comandos necesarios para hacer funcionar la simulación en todos los casos que especifica el plan de pruebas. Hay que suponer que el diseño de un grupo puede ser utilizado por otro grupo o el profesor. Si los resultados no se pueden repetir porque no se conocen los comandos para hacer funcionar la simulación entonces es como si el diseño no funcionara del todo. Se recomienda crear un Makefile de modo que se pueda correr todas las pruebas del caso con un solo comando en Icarus Verilog y GTKwave.
5. **Ejemplos de los resultados:** Una descripción de los resultados más importantes acompañados de los diagramas temporales de la simulación (GTKWave) o cualquier otra salida que demuestre claramente el comportamiento descrito. No es necesario incluir una muestra exhaustiva de resultados, sino que los más representativos del diseño. El punto es mostrarle al lector los comportamientos más sobresalientes para formarle una idea clara

del funcionamiento del diseño. Ya verá el lector si desea más detalles, entonces podrá correr una simulación.

6. **Conclusiones y recomendaciones:** Basado en los resultados obtenidos se indica aquí qué se logró con el proyecto. Puede ser que se concluya que con el diseño propuesto se tiene una limitación en la velocidad de respuesta de... etc. O que con ciertas combinaciones de entradas el diseño se vuelve inestable o los resultados no son los esperados. También se puede concluir qué ventajas o problemas encontraron al seguir el plan de trabajo. A raíz de las conclusiones se puede también recomendar cómo se podría mejorar el diseño o qué otras pruebas se le podrían hacer para garantizar su funcionamiento en otras condiciones que al principio no se consideraron, o también cómo se debería planear el siguiente proyecto para poder cumplirlo a tiempo.