

Tarea Temporización

27 de septiembre de 2020

Consideraciones generales:

- La entrega de esta tarea consiste en un informe de resultados, los cuales deben ser debidamente justificados.
- Toda imagen con resultados debe tener una buena calidad y debidamente justificada y referenciada para ser calificada.
- Se deben entregar todos los ejecutables que comprueben el análisis de resultados.
- Como parte de la solución debe incluir un Makefile con su respectivo README que contenga la forma de ejecutar las pruebas.
- Se castigará severamente cualquier intento o asomo de copia durante el examen o en las soluciones.
- El trabajo debe de ser realizado en LATEX.
- Todo informe debe contener las siguientes secciones: portada, índice, desarrollo, resultados, conclusiones y recomendaciones, bibliografía y anexos.

Especificaciones

Esta tarea consiste en escribir una descripción conductual del contador usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado. Como requisito de diseño deberá utilizar al menos un *case* en la descripción conductual del contador. El contador deberá tener los siguientes modos de operación:

- Cuenta hacia arriba.
- Cuenta hacia abajo.
- Cuenta de tres en tres hacia arriba.
- Carga en paralelo.

Además del contador se deberá crear una librería de compuertas lógicas dentro del archivo *cmos_cells.v* que deben incluir los siguientes módulos:

- Inversor
- NAND 2 entradas
- NOR 2 entradas
- NAND 3 entradas
- NOR 3 entradas
- DFF Flip Flop

Como requisito en la descripción de la compuertas debe incluir los detalles de temporización utilizando el bloque *specify* en Verilog. Los tiempos a incluir son:

- Tiempo propagación en alto t_{pdh}
- Tiempo propagación en bajo t_{pdl}
- Tiempo de preparación o setup t_{setup}
- Tiempo de sostenimiento o hold t_{hold}

Los valores de los tiempos deben de obtenerse de la hoja de datos del fabricante de su elección. Las mismas deben incluirse como bibliografía. El informe debe incluir el valor unitario o del lote por cada componente. Cada estudiante tiene libertad de seleccionar los parámetros como la capacitancia de carga, temperatura, valores mínimos o máximos para escoger sus tiempos de propagación, solamente deben cumplir con que la selección sea uniforme para todas las compuertas. Deberán utilizar una escala de referencia en *ns* con una precisión en *ps* en la implementación del código.

Utilice yosys para sintetizar su modelo utilizando la librería *cmos_cells.lib*. Luego, guarde la descripción estructural con el nombre *contador_synt.v* e incluya este contador en su *top.v* para ser verificado. Como parte del procedimiento debe incluir la librería *cmos_cells.v* en su archivo *contador_synt.v* antes de utilizar icarus verilog.

Utilice la estructura de Testbench diseñado en la tarea 3 para verificar el diseño y el netlist generado con yosys.

Realice dos verificaciones:

- Sin tiempos de propagación.
- Con tiempos de propagación.

Para el segundo caso deberá ejecutar la siguiente línea de comando *iverilog -o algo -gspecify algo.v* para que se consideren los valores de temporización. Todas las pruebas deben iniciar con un periodo de 2 ns para el reloj. Puede aumentar el periodo al siguiente valor permitido, pero no puede saltar entre periodos no consecutivos si encuentra problemas en su contador, el aumento del periodo deberá justificarse técnicamente. Los valores permitidos para el periodo del reloj son:

- 2 ns
- 4 ns
- 20 ns
- 200 ns

Especificación de las entradas del contador

1. **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 MODE y si la señal ENABLE = 1.
2. **ENABLE**: Entrada de habilitación del contador. Si ENABLE = 1, el contador funciona normalmente respondiendo a los flancos activos de CLK para cambiar de estado de acuerdo a la señal MODO. Si ENABLE=0 y RESET = 0, el contador tendrá una salida de alta impedancia.
3. **RESET**: Entrada que pone todas las salidas del contador en cero. Depende de la señal de reloj.
4. **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 ENABLE = 1 y MODO = 11.
5. **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 → Q + 3
 - MODO = 01 → Q - 1
 - MODO = 10 → Q + 1
 - MODO = 11 → D

Especificación de las salidas del contador

1. **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].
2. **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 dependiendo de la señal MODO, además, debe permanecer en bajo

durante el MODO = 11.

3. **LOAD** : Salida que indica cuando el contador esta en modo de carga debe permanecer en bajo para el resto de los modos.

Consideraciones

1. Calcule la diferencia de tiempos entre el modelo sin retardo y el modelo con retardo para las señales de: LOAD, RCO y Q.
2. Explique los problemas que encuentra entre su scoreboard y su contador utilizando los tiempos de propagación.
3. Cree un resumen de las compuertas utilizadas para la síntesis de su modelo. ¿Se usaron todas las librerías?.