Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0523 Circuitos Digitales II
II ciclo 2022

Tarea 2

Descripción estructural del contador sincrónico de 16 bits: síntesis automática

Javier Solera Bolaños B66963
Grupo 1

Profesor: Enrique Coen Alfaro

20/09/2022

Índice

1. Resumen

En este trabajo se realizó un diseño estructural del contador de 16 bits que se realizó en la tarea 1 mediante la herramienta de yosys. El cual se llevo a cabo una síntesis de alto nivel al contador de 16 bits, es decir, primero se convierte los procesos a un netlist (primer paso para la síntesis) y también se realizo un mapeo tecnológico y este producía una descripción estructural usando componentes comerciales y generara una síntesis mas completa y por ultimo también se modificó los archivos cmos_cells.v para realizar una simulación con retardos y así generar una síntesis con el fin de tratar de simular los retador como si fueran en la vida real.

2. Descripción Arquitectónica

Mediante la herramienta de yosys, se realizaron 3 síntesis del contador de 16 bits realizado en la tarea 1. La primera síntesis se realizo con la librería interna de yosys, la cual genera una descripción de alto nivel RTLIL. La segunda síntesis que se realiza es por medio de mapeo de tecnología y usando la librería de cmos_cells.lib. Y la Tercera síntesis que se realiza, es igual a la anterior, solo que aplicándole retardo en esta librería, de manera que cada vez que haya una NAND, NOR, NOT, BUF o flipflop, la señal se retarda.

Para este diseño, la herramienta de yosys y por medio del comando $abc -liberty\ ./cmos\_cells.lib$, se determinaron que la síntesis utiliza los siguientes componentes:

Lista de Componentes	
NAND cells:	17
NOR cells:	34
NOT cells:	13
internal signals:	90
input signals:	10
output signals:	5

Tabla 1: Lista de componentes, resultados del ABC

3. Plan de Pruebas

Las pruebas que se esperan son muy parecidas a las de la tarea 1, pero únicamente con el contador de 16 bits, y utilizando los archivos que se generaron mediante la herramienta de yosys y esta prueba se realizo sin retardos y se hizo un prueba con retardos.

Entonces para las 2 pruebas que eran sin retardos. Se realizaron 4 pruebas para cada modo, cuenta ascendente, cuenta descendente, cuenta descendente de 3 en 3 y carga en paralelo, de manera que se usa 4 contadores de 4 bits crear un contador de 16 bits:

1. Cuenta ascendente MODO[1 : 0] = 00: Se desea realizar una cuenta ascendente de 1 en 1 y el Ripple carry out (RCO) se va a activar cada vez que hay un llevo y cuando el RCO se activa, el contador 2 entraba en funcionamiento, luego cuando el contador 2 activa el RCO, el contador 3 entra en funcionamiento y asi con el contador 4.

   1.1) El registro inicia en 0000000000000000, para lograr esto se toma el MODO[1 : 0] = 11 y D[15 : 0] = 0000000000000000

1.2) Cada flanco positivo del reloj, me activa la suma, entonces la salida Q se vería de esta manera: Q[15 : 0] =0000000000000000 − > Q[15 : 0] =0000000000000001 − > Q[15 : 0] =0000000000000010 − > ... − > Q[15 : 0] =1111111111111111.

1.3) En esta prueba se trato de analizar que el RCO funcionara correctamente, es decir, que cada vez que el RCO se activara, el contador 2 empezara a funcionar.

2. Cuenta descendente MODO[1 : 0] = 01: Se desea realizar una cuenta descendente de 1 en 1 y el Ripple carry out (RCO) de va a activar cada vez que hay un llevo, al igual que el modo anterior, cuando el RCO se activa, activa el contador 2 y cuando el contador 2 activa el RCO, el contador 3 se activa y asi sucesivamente.

   2.1) El registro inicia en 1111111111111111, para lograr esto se toma el MODO[1 : 0] = 11 y D[15 : 0]= 1111111111111111

   2.2) Cada flanco positivo del reloj, me activa la resta, entonces la salida Q se vería de esta manera: Q[15 : 0] =1111111111111111 − > Q[15 : 0] =1111111111111110 − > Q[15 : 0] =1111111111111110 − > ... − > Q[15 : 0] =0000000000000000.

   2.3) En esta prueba se trato de analizar que el RCO funcionara correctamente, es decir, que cada vez que el RCO se activara, el contador 2 empezara a restar y asi sucesivamente.

3. Cuenta descendente 3 en 3 MODO[1 : 0] = 10: Se desea realizar una cuenta descendente de 3 en 3 y el Ripple carry out (RCO) de va a activar cada vez que hay un llevo

   3.1) El registro inicia en 1111111111111111, para lograr esto se toma el MODO[1 : 0] = 11 y D[15 : 0]= 1111111111111111

   3.2) Cada flanco positivo del reloj, me activa la resta, entonces la salida Q se vería de esta manera: Q[3 : 0] =1111111111111111 − > Q[3 : 0] =1111111111111100 − > Q[3 : 0] =1111111111111001 − > ... − > Q[3 : 0] =0000000000000000.

   3.3) Al igual que en las pruebas anteriores, RCO se va a activar siempre que alguno de los contadores llegue a estos valores: 0000, 0001 o 0010, y asi activara al siguiente contador.

4. Carga en Paralela MODO[1 : 0] = 11:

   4.1) El registro inicia en XXXXXXXXXXXXXXXX, para lograr esto se toma el MODO[1 : 0] = 11 y D[3 : 0] =XXXXXXXXXXXXXXXX

   4.2) Cada flanco positivo del reloj, a la entrada D, se le va a sumar 1 y la salida Q va a copiar el valor que haya en D, por ejemplo si en D[15 : 0]= 1010000011001010, la salida Q en el siguiente flanco positivo del reloj va a ser Q[15 : 0]= 1010000011001010

   4.3) En esta prueba el RCO se va a encender siempre que D[3 : 0]= 1111 de alguno de los 4 contadores de 4 bits.

Para las pruebas con retardos, se realizaron las mismas pruebas pero esta vez en el archivo.v se le agregaron unos retardos en las compuertas NAND, NOT, NOR, BUF y en los flipflops, entonces cada vez que la señal llega a uno de estos elementos, va a sufrir un retardo, lo que produce que la salida no ocurre simultáneamente con el flanco positivo del reloj, si no que ocurre un cierto tiempo despues, en mi caso se realizaron pruebas para que ocurriera 2 ns despues, es decir, para el modo 00, cada flanco positivo del reloj, manda la señal de activar la suma, pero esta se activa hasta 2ns despues de que llegara el flanco positivo del reloj, y las pruebas seria las mismas que en el punto anterior.

4. Instrucciones de utilización de la simulación

Es importante que los archivos flipflop.v, tb_Counter.v, teste.v, Counter _ synth _ rtlil.v, Counter_full_synth.v y el Makefile se encuentren en la misma carpeta.

Entonces puede abrir la terminal y ubicarse en donde están los archivos mencionados anteriormente y correr el comando make, este ejecutará el archivo Makefile y complilará y correrá los archivos.

5. Ejemplos de los resultados

5.1. Pruebas sin Retardos

5.1.1. Archivo Counter_synth_rtlil.v

Figura 1: Cuenta Ascendente, RTLIL

Figura 2: Cuenta Descendente, RTLIL

Figura 3: Cuenta Descendente de 3 en 3, RTLIL

Figura 4: Carga en paralelo, RTLIL

5.1.2. Archivo Counter_full_synth.v

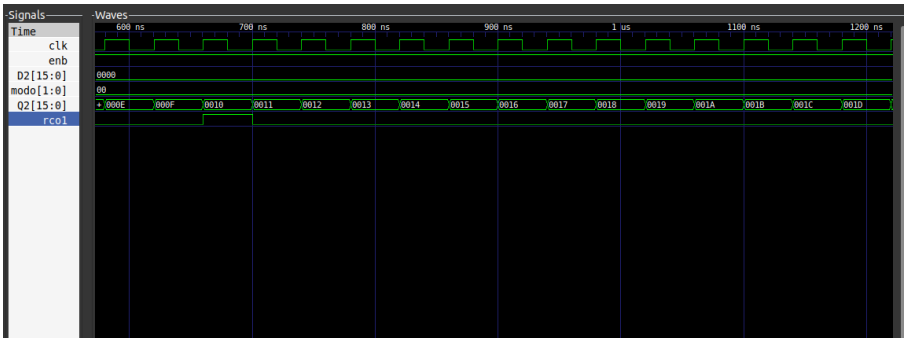

Figura 5: Cuenta Ascendente, sintetizado

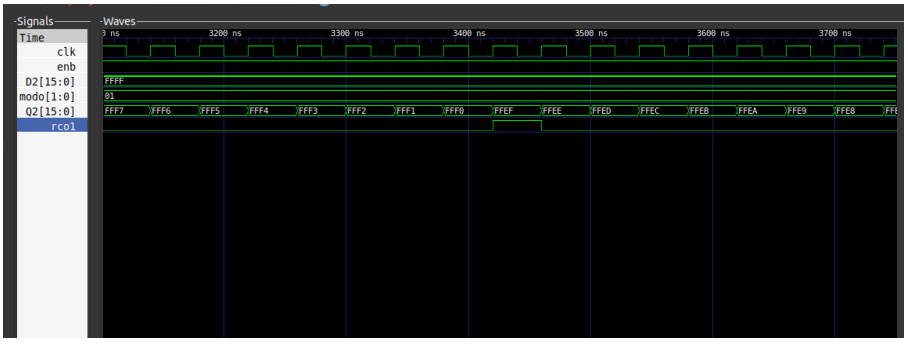

Figura 6: Cuenta Descendente, sintetizado

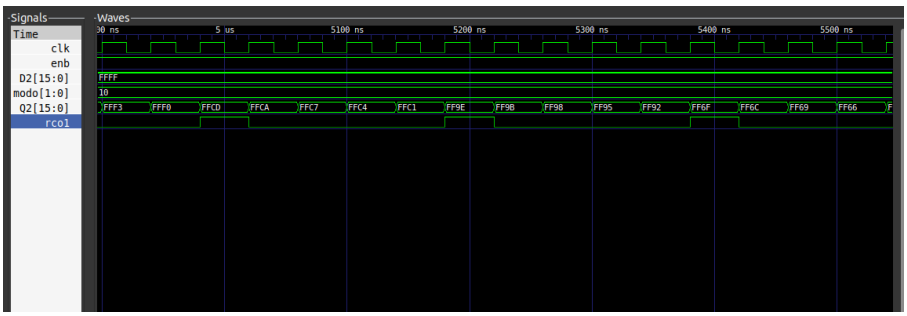

Figura 7: Cuenta Descendente de 3 en 3, sintetizado

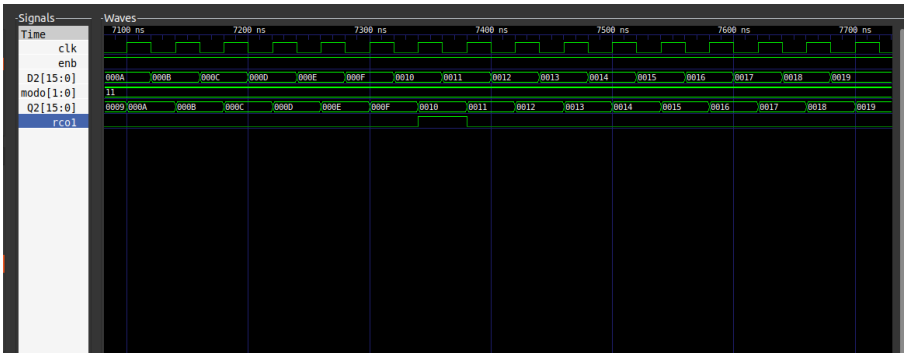

Figura 8: Carga en paralelo, sintetizado

5.2. Pruebas con Retardos

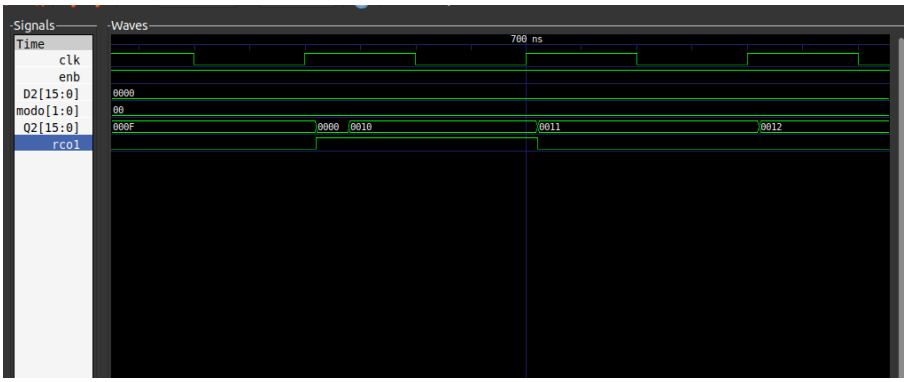

Figura 9: Cuenta ascendente con retardo

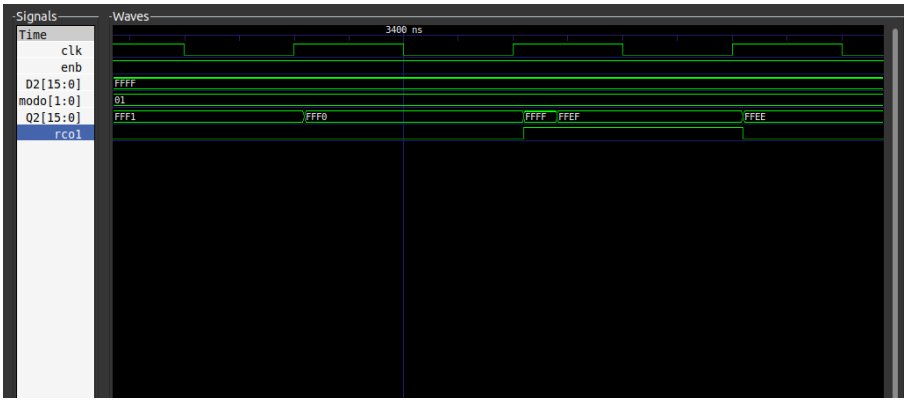

Figura 10: Cuenta descendente con retardo

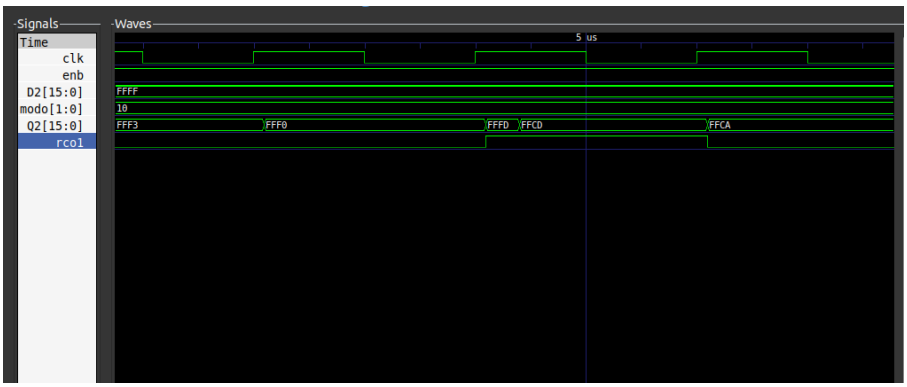

Figura 11: Cuenta 3 en 3 descendente con retardo

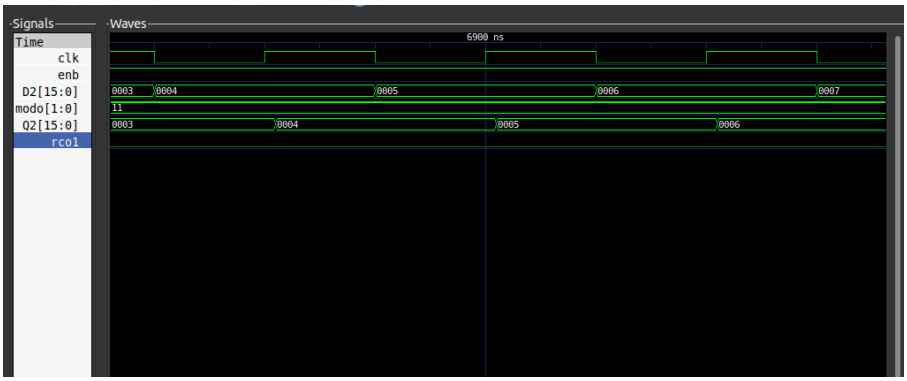

Figura 12: Carga en paralelo con retardo

6. Conclusiones y recomendaciones

Se realizó hacer una descripción estructural mediante una descripción conductual. Se logro sintetizar de alto nivel el contador de 16 bits que se realizo en la tarea 1 por medio de la herramienta yosys, esta síntesis procura sintetizar de una manera de que no dependa de ninguna tecnología , y se comprobó su correcto funcionamiento mediante iverilog. Luego se realizo la selección de una biblioteca para realizar el mapeo tecnológico y así realizar una sintetización completa del código y se comprobó su correcto funcionamiento. Por ultimo se le asignaron retardos para tratar de simularlo como si fuera la vida real, estos retador se asignaron a las NAND, NOR, NOT BUF, y flipflops que el sistema asigno para hacer la sintesis. Se recomienda cambiar las and y or por && y ||||, ya que yosys no acepta esa sintaxis en el código y genera errores.