

IE0411 Microelectrónica -G01-

## Tarea 2: Temporización

Jorge Muñoz Taylor (A53863) - [jorge.munoztaylor@ucr.ac.cr](mailto:jorge.munoztaylor@ucr.ac.cr)

II-2020

# Índice

<b>1. Desarrollo</b>	<b>3</b>
1.1. Diseño del contador . . . . .	3
1.2. Componentes usados para el diseño . . . . .	3
1.3. Sintetización . . . . .	4
1.4. Diseño de la prueba para el contador . . . . .	5
1.5. Composición de la prueba . . . . .	6
<b>2. Resultados</b>	<b>7</b>
2.1. Sin tiempos de propagación . . . . .	7
2.2. Con tiempos de propagación . . . . .	10
2.3. Diferencia de tiempos . . . . .	11
<b>3. Conclusiones</b>	<b>12</b>
<b>4. Recomendaciones</b>	<b>13</b>
<b>Referencias</b>	<b>14</b>

# 1. Desarrollo

## 1.1. Diseño del contador

El contador está compuesto por 5 entradas y 3 salidas, su funcionamiento es sincrónico con transiciones en el flanco positivo de la entrada clk. En el cuadro 1 se muestra la tabla de verdad que se siguió para diseñar el contador.

clk	enable	reset	modo	load	Q	rco
posedge	0	0	x	0	Hi-z	0
posedge	0	1	x	0	0	0
posedge	1	0	00	0	Q+3	activo
posedge	1	0	01	0	Q-1	activo
posedge	1	0	10	0	Q+1	activo
posedge	1	0	11	1	D	0
posedge	1	1	x	0	0	0

Cuadro 1: Tabla de verdad que describe el funcionamiento del contador en base a cada entrada.

Algunos estados se tuvieron que asumir puesto que no fueron mencionados explícitamente en las especificaciones:

1. Sin importar su enable está en bajo o en alto, cuando la entrada reset está en alto las salidas Q y rco se ponen en bajo.
2. La salida load se pone en alto sólo cuando enable está en alto, reset en bajo y modo en 11.
3. Luego de reset = enable = bajo las salidas Q y rco se ponen en bajo, es decir, se reinicia el contador.
4. Cuando Q está en alta impedancia las salidas rco y load están en bajo.
5. La entrada modo tiene un valor X (*no importa*) en tres casos, esto es así ya que, en el primer caso al tener enable y reset en 0 la salida Q debe estar en el tercer estado por lo que el valor que tenga la entrada modo no importa, de igual forma en los dos casos siguientes solo que con el reset.

## 1.2. Componentes usados para el diseño

El criterio de selección que se usó fué: capacitancia de salida de 50pF, tensión de  $5 \pm 0,5$ , temperatura entre  $-40^\circ \sim 85^\circ$  y se tomaron los valores máximos de los retardos.

	modelo	Costo	t pdh	t pdl	t setup	t hold
NOT[1]	74AC11004	\$2.085	6.3ns	5.5ns	-	-
NAND de 2 entradas[2]	MC74VHCT00A	\$0.1467	9ns	9ns	-	-
NAND de 3 entradas[3]	SN74ACT10-Q1	\$0.456	10ns	9.5ns	-	-
NOR de 2 entradas[4]	74AC02	\$0.660	8ns	7ns	-	-
NOR de 3 entradas[5]	74VHC27	\$0.48	9ns	9ns	-	-
FFD[6]	SN74LVC1G80	\$0.086	3.8ns	3.8ns	1.1ns	0.4ns

Cuadro 2: Modelos de los componentes usados junto su costo **unitario** y tiempos de retardo.

La cantidad de componentes que se obtuvieron al sintetizar el contador con *Yosys* se muestran en el cuadro 3, en dicho cuadro se puede ver que Yosys utilizó todas las compuertas cuando sintetizó el circuito.

Componente	Cantidad
NOT	13
NAND de 2 entradas	4
NAND de 3 entradas	6
NOR de 2 entradas	27
NOR de 3 entradas	16
FFD	7

Cuadro 3: Conteo de componentes obtenido con *Yosys*.

### 1.3. Sintetización

Para obtener la netlist del circuito se utilizó el siguiente código para Yosys:

```
read_verilog src/counter.v
synth -top counter
dfflibmap -liberty cells/cmos_cells.lib
abc -liberty cells/cmos_cells.lib
clean
write_verilog src/contador_synt.v
```

Un detalle **muy importante** es que Yosys **no puede generar salidas de tercer estado (alta impedancia)** ya que no está preparado para ello, en cambio cada vez que se requiera tener una salida de tercer estado Yosys colocará un cero, por esa razón tome en cuenta que para *reset=enable=0* en lugar de tener la salida *Q* en tercer estado se tendrá *Q=0*. El creador del Yosys confirmó ese detalle en su *Reddit*:

*“Clifford Vienna:*

*Yosys does not really support tristate logic yet and it is not on the short term todo list.”*

Por lo tanto, el checker y los task considerarán este detalle a la hora de analizar las salidas del contador. Note como en la figura 1, que corresponde a la versión sintetizada, cuando enable y reset están en bajo la salida *Q* tiene un valor de 0, mientras que en la figura 2 (diseño conductual) se tiene un valor de tercer estado para la salida *Q*.

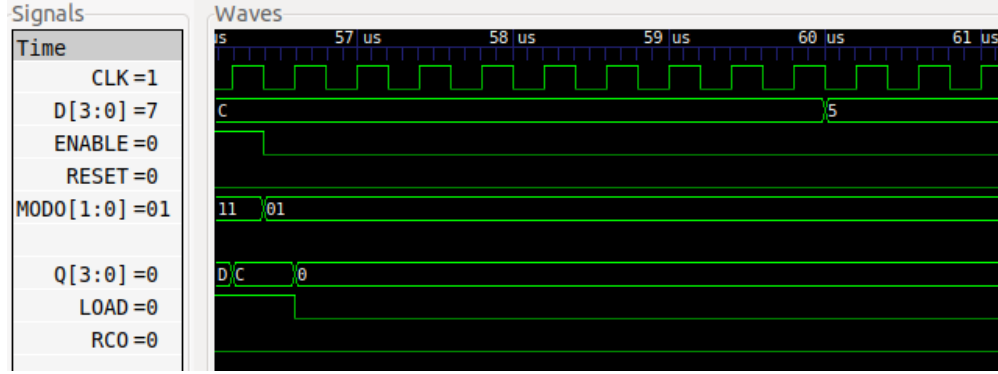


Figura 1: Simulación del circuito sintetizado del contador. Note que la salida Q vale 0 a pesar de que reset y enable están en bajo.

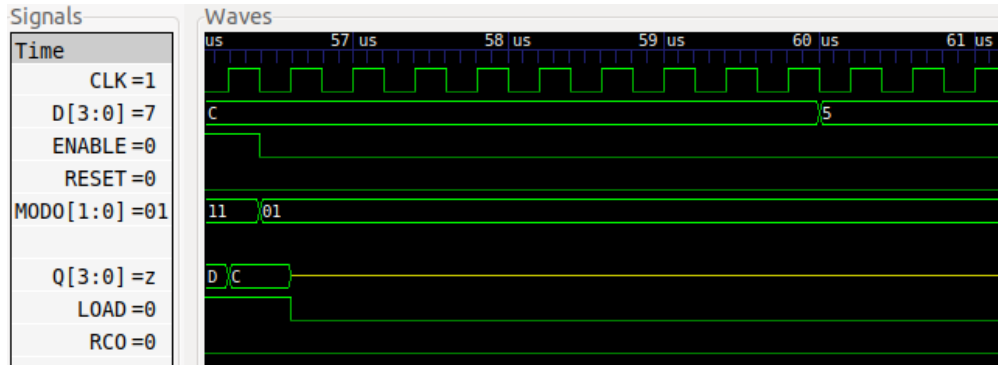


Figura 2: Simulación conductual del contador. Note que la salida Q está en tercer estado cuando reset y enable están en bajo.

#### 1.4. Diseño de la prueba para el contador

El diseño del *test* se realizó siguiendo la tabla de verdad que se muestra en el cuadro 1. El test es un módulo que se encarga de generar los estímulos que se utilizarán en el DUV y en el checker para verificar si el DUV funciona bien.

- Primero se resetea el contador colocando *enable* en 0 y *reset* en 1 para que se inicialicen las salidas en  $Q = 0$ ,  $rco = 0$  y  $load = 0$ . Luego se pone *enable* en 1 y *reset* en 0 en el tiempo ' $CLK * 2^1$ ' para comenzar el conteo. Después se cambian  $enable = 0$ - $reset = 0$  en  $CLK * 282$  y  $enable = 1$ - $reset = 0$  en  $CLK * 352$  para verificar el comportamiento de  $Q$ .
- Se toman los cuatro modos de operación en orden y se verifica que las salidas son las adecuadas: en 00 cuenta de 3 en 3, en 01 resta 1, en 10 suma 1 y el 11 cada el valor de la entrada D en Q.
- A partir de cierto punto de la simulación y por medio de un loop *forever*, la entrada *mode* cambia aleatoriamente cada  $CLK * 70$  tiempos al igual que las entradas *reset* y *enable* (estas

<sup>1</sup>La constante ' $CLK$ ' (no confundir con  $clk$ ) está definida en el archivo *defines.v* e indica el tiempo en el que el reloj cambia de estado, las pruebas se construyeron en función a la variable mencionada para mantener la similitud en las pruebas cuando se cambie el reloj.

cada  $CLK * 10 CLK * 20$  tiempos), de esta forma no se tiene control sobre las posibles salidas y así comprobar los resultados obtenidos en la parte no aleatoria.

- La entrada D es aleatoria durante toda la simulación y cambia cada  $CLK * 20$ .

## 1.5. Composición de la prueba

Se compone de 6 archivos:

- **test.v:** Contiene las señales que se inyectarán en el contador incluido el reloj.
- **tasks.v:** Contiene 3 *tasks* que se encargan de verificar las salidas del contador:
  - **verificar\_Q:** Determina si la salida Q tiene el comportamiento esperado.
  - **verificar\_LOAD:** Determina si la salida *LOAD* es 1 cuando la señal *mode* sea 11.
  - **verificar\_rco:** Verifica si la señal de salida *rco* es alta cuando ocurre un *overflow*.
- **checker.v:** Despliega en la terminal e imprime en el archivo de log cuando ocurre un error durante la simulación. Las señales de error provienen de los módulos definidos en *task.v*.
- **defines.v:** Aquí están definidos algunos parámetros utilizados, así es más fácil cambiar el comportamiento de la simulación en cualquier momento sin necesidad de modificar todo el código.
- **testbench.v:** Contiene las instancias de los módulos *test\_1*, *counter* y *checker*.
- **testbench\_synt.v:** Contiene las instancias de los módulos *test\_1*, *counter* y *checker*. Note que el módulo *counter* que se instancia corresponde al módulo sintetizado en Yosys.

## 2. Resultados

Para analizar la información obtenida de las simulaciones se crearon 3 señales orientadas a ello, las llamaré señales de **fallo**:

- Q\_FALLO
- LOAD\_FALLO
- RCO\_FALLO

Estas señales se ven reflejadas en las simulaciones en *GTKwave* en la parte inferior de la imagen, cuando algún error ocurre en la señal de salida o *reset/enable* las señales de fallo pasarán al estado ALTO de lo contrario permanecerán en BAJO.

Cada simulación está subdividida en 4 imágenes, la primera figura muestra la parte **no** aleatoria de la simulación mientras que la segunda figura muestra la parte aleatoria, por razones de espacio cada parte de la simulación se dividió en dos y se colocaron una sobre otra, pero recuerde que forma parte de la misma simulación.

Bastará con un fallo para considerar que esa característica no funciona adecuadamente.

### 2.1. Sin tiempos de propagación

Se tomaron cuatro trozos de la simulación para mostrar el comportamiento del circuito sintetizado, en el primer trozo (figura 3) está el comportamiento del modo 00 donde se puede notar que la salida Q inicia el conteo en  $t = 6ns$ , luego en cada flanco positivo de CLK se da la suma del valor anterior más 3. Cabe resaltar como en  $t = 26ns$  la salida RCO es alta, esto ocurre porque al pasar de  $f$  a 2 se produce un rebase.

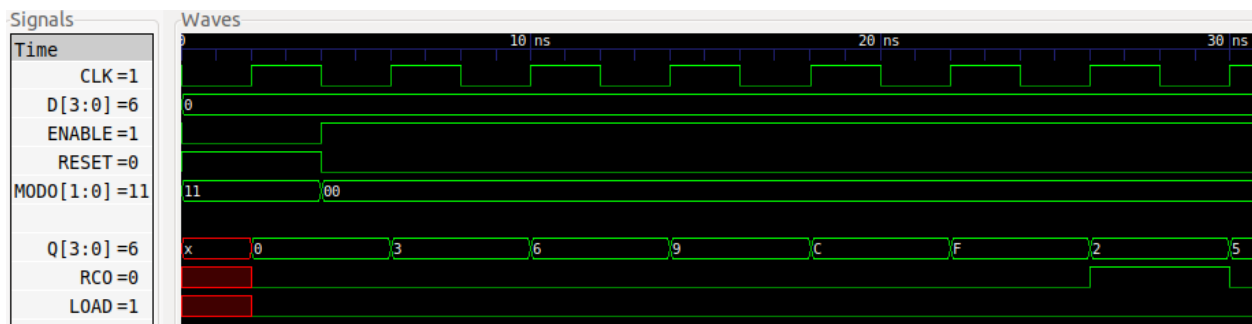


Figura 3: Resultado de la simulación SIN tiempos de propagación.

En el segundo trozo (figura 4) está representado el modo 01, note por ejemplo, para  $t = 166ns$  se da la transición de  $Q = 4$  a  $Q = 3$  lo que corresponde a una resta de 1, ocurre lo mismo para los tiempos  $t = 170ns$  y  $t = 174ns$ . Note como en  $t = 182ns$  la salida RCO es alta, esto ocurre porque al pasar de 0 a  $F$  se produce un rebase.

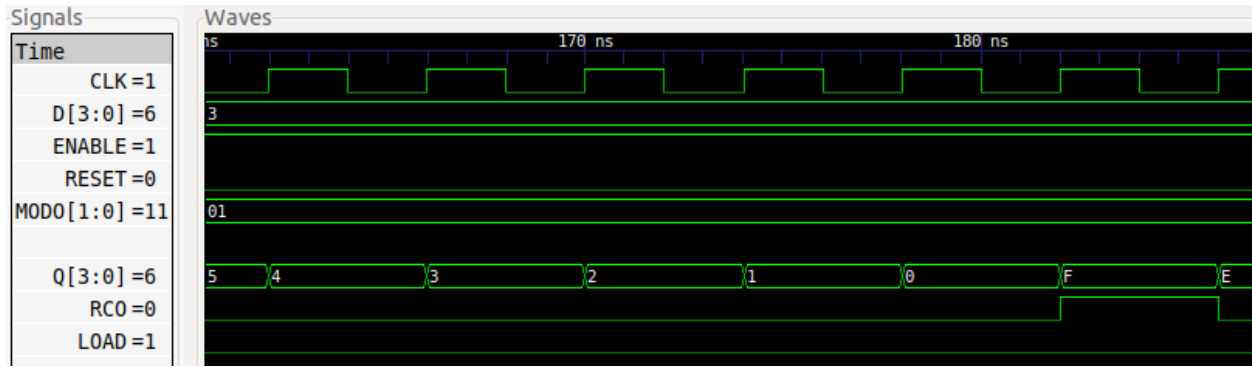


Figura 4: Resultado de la simulación SIN tiempos de propagación

En la figura 5 se muestra el modo 10, en este modo el conteo es ascendente y unitario, puede verse como en  $t = 310ns$  Q pasa de C a D, también en  $t = 314ns$  Q pasa de D a E. Cabe resaltar como en  $t = 322ns$  la salida RCO es alta, esto ocurre porque al pasar de F a 0 se produce un rebase.

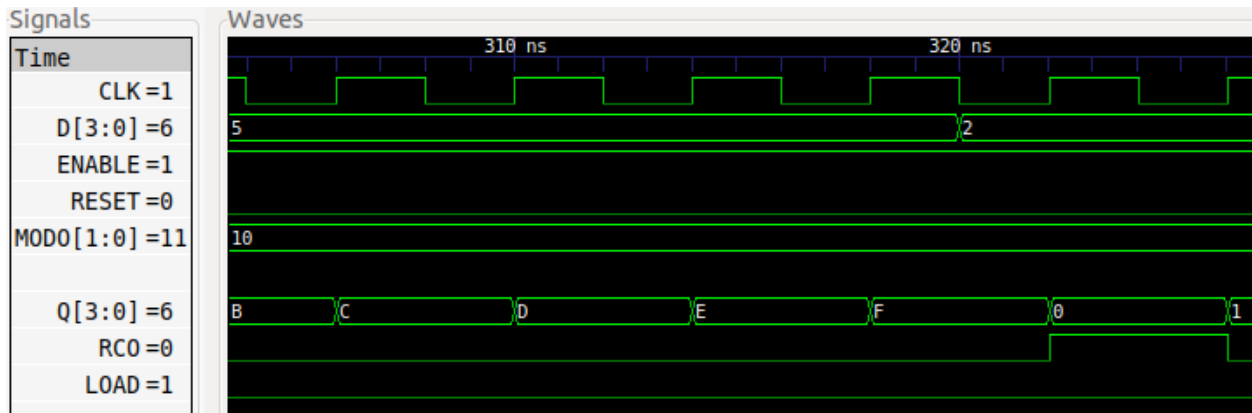


Figura 5: Resultado de la simulación SIN tiempos de propagación

La última figura (la 6) muestra el modo 11, en  $t = 426ns$  se puede ver como el valor en la entrada D se refleja en la salida Q, lo mismo en  $t = 442ns$  que es justamente el comportamiento especificado. En este caso la salida RCO siempre está en bajo sin importar el valor y la salida LOAD siempre está en alto, excepto cuando reset está en alto y cuando reset-enable están en bajo.



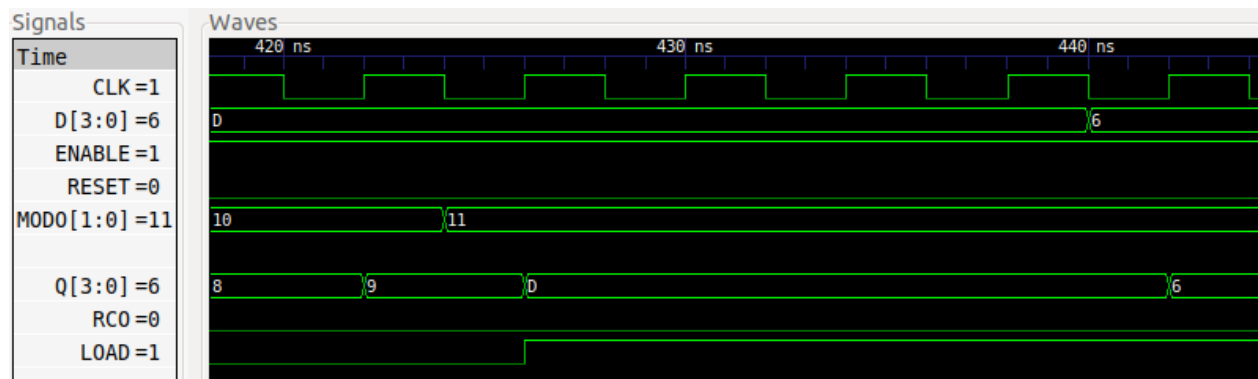


Figura 6: Resultado de la simulación SIN tiempos de propagación

## 2.2. Con tiempos de propagación

Se solicita probar el circuito sintetizado activando los tiempos de propagación de los componentes seleccionados (ver cuadro 2) para CLK igual a  $2ns$ ,  $4ns$ ,  $20ns$  y  $200ns$  en ese orden.

Para determinar si el CLK es el correcto se tomaron las señales del módulo *verificar\_Q* en GTKWave, de estas salidas la más importante es *Q\_fallo* que se pone en 1 cuando la salida del DUV no es la correcta, **basta con que se de un fallo en la salida para considerar que el tiempo no cumple con la temporización del circuito**. Note que la salida *Qi* corresponde a la *Q* del DUV y *Q\_ant* corresponde a la salida que genera el checker.

En las figuras 7, 8, 9 y 10 se muestran las verificaciones para los cuatro tiempos mencionados, para  $2ns$ ,  $4ns$  y  $20ns$  la señal *Q\_fallo* se pone en alto, por lo tanto dichos tiempos no cumplen con la temporización del circuito. Para  $200ns$  (figura 10) las señales *Qi* y *Q\_ant* son iguales en todas las transiciones de igual forma *Q\_fallo* está en bajo en todo momento, por lo tanto, se puede concluir que el valor de tiempo indicado para el reloj del circuito es de  $200ns$ .

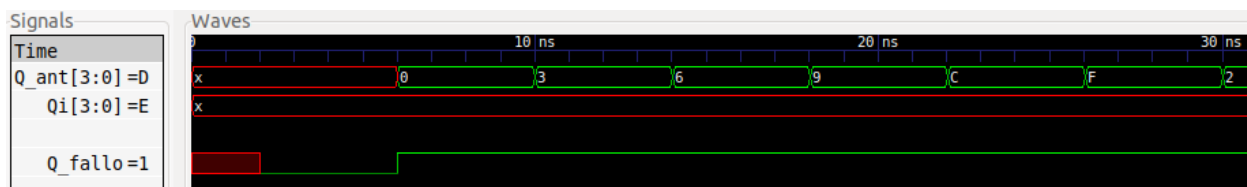


Figura 7: Salidas del módulo *verificar\_Q* para  $2ns$  de transición de la señal clk.

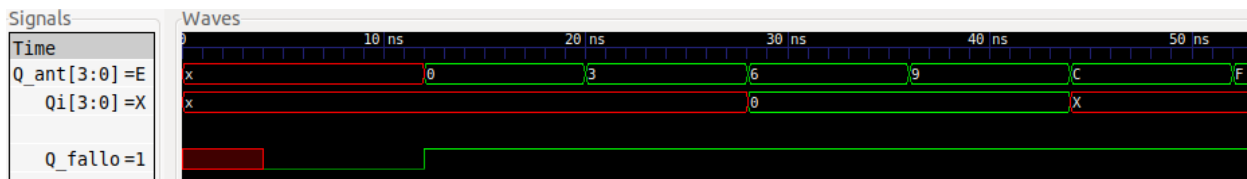


Figura 8: Salidas del módulo *verificar\_Q* para  $4ns$  de transición de la señal clk.

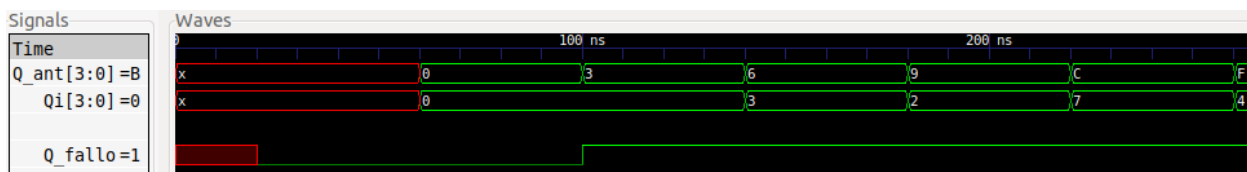


Figura 9: Salidas del módulo *verificar\_Q* para  $20ns$  de transición de la señal clk.

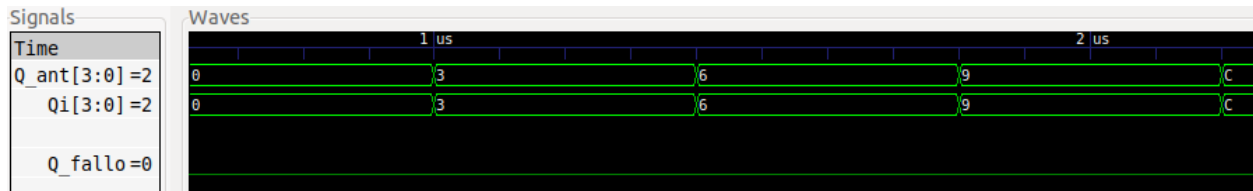


Figura 10: Salidas del módulo *verificar\_Q* para 200ns de transición de la señal *clk*. Note que la salida *Q\_fallo* nunca está en alto.

### 2.3. Diferencia de tiempos

Los resultados para la simulación con retardos se ajustan a lo esperado, hay que tener en cuenta que los retardos de cada una de las compuertas instanciadas durante la sintetización se van sumando, por lo tanto los paths ven sus retardos incrementados, para corregir este detalle resulta necesario disminuir la frecuencia del reloj y hacer el contador más lento, de esta forma se da tiempo a las señales para que puedan atravesar el path.

En este caso, el retardo más grande lo tiene la compuerta NAND de 3 entradas con  $10ns$ , ya sólo con este detalle se pueden descartar los tiempos de 2ns y 4ns para el reloj puesto que son tan pequeños que no permiten a las señales de entrada de la NAND desplegar una salida, de igual forma para 20ns.

### 3. Conclusiones

- El contador se pudo diseñar en base a las especificaciones dadas en el enunciado del proyecto.
- El contador pasó las pruebas para Q, RCO y LOAD tanto para su versión conductual como la para versión sintetizada. La versión sintetizada sólo funcionó para un período de reloj de 200ns.
- El contador sintetizado con Yosys **no puede generar salidas de tercer estado** debido a que el programa no está preparado para hacerlo, en su lugar se verán ceros en la salida (cuando reset=enable=0).
- Es necesario tener en cuenta que el cambio de valor en una señal se de únicamente en un bloque always, es decir, que la misma señal no se manipule en varios bloques always.

## 4. Recomendaciones

- Evitar los latch inferidos, esto se consigue evitando los bloques always @(\*) y colocar siempre la condición else al usar if.
- Es mejor utilizar la salida no bloqueante en lugar de la bloqueante en los bloques always, esto se ajusta más al comportamiento real del hardware.
- Cuando se estén creando las bibliotecas de componentes hay que recordar definir la función del componente luego del bloque specify, de lo contrario no funcionará, ya que el archivo .LIB es usado durante la sintetización NO durante las simulaciones.

## Referencias

- [1] T. instruments. 74ac11004 hex inverter datasheet. [Online]. Available: [https://www.ti.com/lit/ds/symlink/74ac11004.pdf?ts=1602514791834&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252F74AC11004%253FkeyMatch%253DINVERTER%2526tsearch%253DSearch-EN-everything%2526usecase%253DGPN](https://www.ti.com/lit/ds/symlink/74ac11004.pdf?ts=1602514791834&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252F74AC11004%253FkeyMatch%253DINVERTER%2526tsearch%253DSearch-EN-everything%2526usecase%253DGPN)
- [2] Onsemi. Quad 2-input nand gate: Mc74vhct00a. [Online]. Available: <https://www.onsemi.com/pub/Collateral/MC74VHCT00A-D.PDF>
- [3] T. instruments. Sn74act10-q1 triple 3-input positive-nand gate. [Online]. Available: [https://www.ti.com/lit/ds/symlink/sn74act10-q1.pdf?ts=1602517413545&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/sn74act10-q1.pdf?ts=1602517413545&ref_url=https%253A%252F%252Fwww.google.com%252F)
- [4] STMicroelectronics. 74ac02 quad 2-input nor gate. [Online]. Available: <https://datasheetspdf.com/pdf-file/246165/STMicroelectronics/74AC02/1>
- [5] Mouser. 74vhc27 triple 3-input nor gate. [Online]. Available: <https://www.mouser.com/datasheet/2/149/74VHC27-101406.pdf>
- [6] T. instruments. Sn74lvc1g80 single positive-edge-triggered d-type flip-flop. [Online]. Available: [https://www.ti.com/lit/ds/symlink/sn74lvc1g80.pdf?ts=1602512485400&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/sn74lvc1g80.pdf?ts=1602512485400&ref_url=https%253A%252F%252Fwww.google.com%252F)