

IE0411 Microelectrónica -G01-

Tarea 1: verificación

Jorge Muñoz Taylor (A53863) - jorge.munoztaylor@ucr.ac.cr

II-2020

1. Pruebas realizadas sobre los contadores

Los tres contadores tienen cinco entradas y tres salidas, para probar el funcionamiento de cada contador se siguen las pautas a continuación:

- Primero se resetea el contador colocando *enable* en 0 y *reset* en 1, para que inicialice en $Q = 0$, luego de 10 ciclos se pone *enable* en 1 y *reset* en 0 para comenzar el conteo. También en 1410 se cambia los valores a *enable* = 0-*reset* = 0 y en 1760 se cambian a *enable* = 1-*reset* = 0 para verificar el comportamiento de Q .
- Se toman los, uno a uno, los cuatro modos de operación del contador para verificar que cada uno cuenta como debe hacerlo: 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 al igual que las entradas *reset* y *enable*, 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.

El diseño del *testbench* se realizó siguiendo la tabla de verdad que se muestra en el cuadro 1, note que la entrada *modo no importa* (X) en tres casos, esto es así a pesar de que no se menciona en las especificaciones 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* el valor que tenga, de igual forma en los dos casos siguientes solo que con el *reset*, como es 1 todas las salidas deben valer 0.

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 sigue el circuito contador descrito en las especificaciones.

Para los tres contadores se utilizó el mismo escenario de prueba. Además los resultados de cada simulación se guardan en 3 archivos diferentes: *log_A.txt*, *log_B.txt* y *log_C.txt*, donde cada letra en mayúscula corresponde al contador que se utilizó para generar el *log*.

2. Composición de la prueba

Se compone de 7 archivos:

- **test_1.v:** Contiene las señales que se inyectarán en el contador incluido el reloj.
- **tasks.v:** Contiene 4 *tasks* que se encargan de verificar las salidas y las señales *reset-enable*:

- **verificar_enable_reset**: Determina si las señales *reset-enable* generan una salida *Q* según la especificación.
 - **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.
 - **testbenchA.v**: Contiene las instancias de los módulos *test_1*, *counter* y *checker*. Note que el módulo *counter* que se instancia corresponde al contador A.
 - **testbenchB.v**: Contiene las instancias de los módulos *test_1*, *counter* y *checker*. Note que el módulo *counter* que se instancia corresponde al contador B.
 - **testbenchC.v**: Contiene las instancias de los módulos *test_1*, *counter* y *checker*. Note que el módulo *counter* que se instancia corresponde al contador C.

3. Resultados

Para analizar la información obtenida de las simulaciones se crearon 4 señales orientadas a ello, las llamaré señales de **fallo**:

- enable_reset_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.

3.1. Contador A

En las figuras 1 y 2 se puede ver que el contador funciona correctamente en los 4 modos de conteo, es decir, cuenta de tres en tres, de menos uno en menos uno, de uno mas uno y en modo de carga. El *RCO* presenta fallos, por ejemplo, cuando $Q = 15$ el rco se activa siendo este resultado incorrecto. La salida *LOAD* tampoco funciona adecuadamente ya que debería ponerse en *ALTO* únicamente cuando se está en modo de carga, sin embargo puede verse como se activa durante el modo 10.

Enable-reset también presenta un error, cuando $enable = reset = 0$ la salida Q debe estar en alta impedancia, sin embargo en este caso es 0.

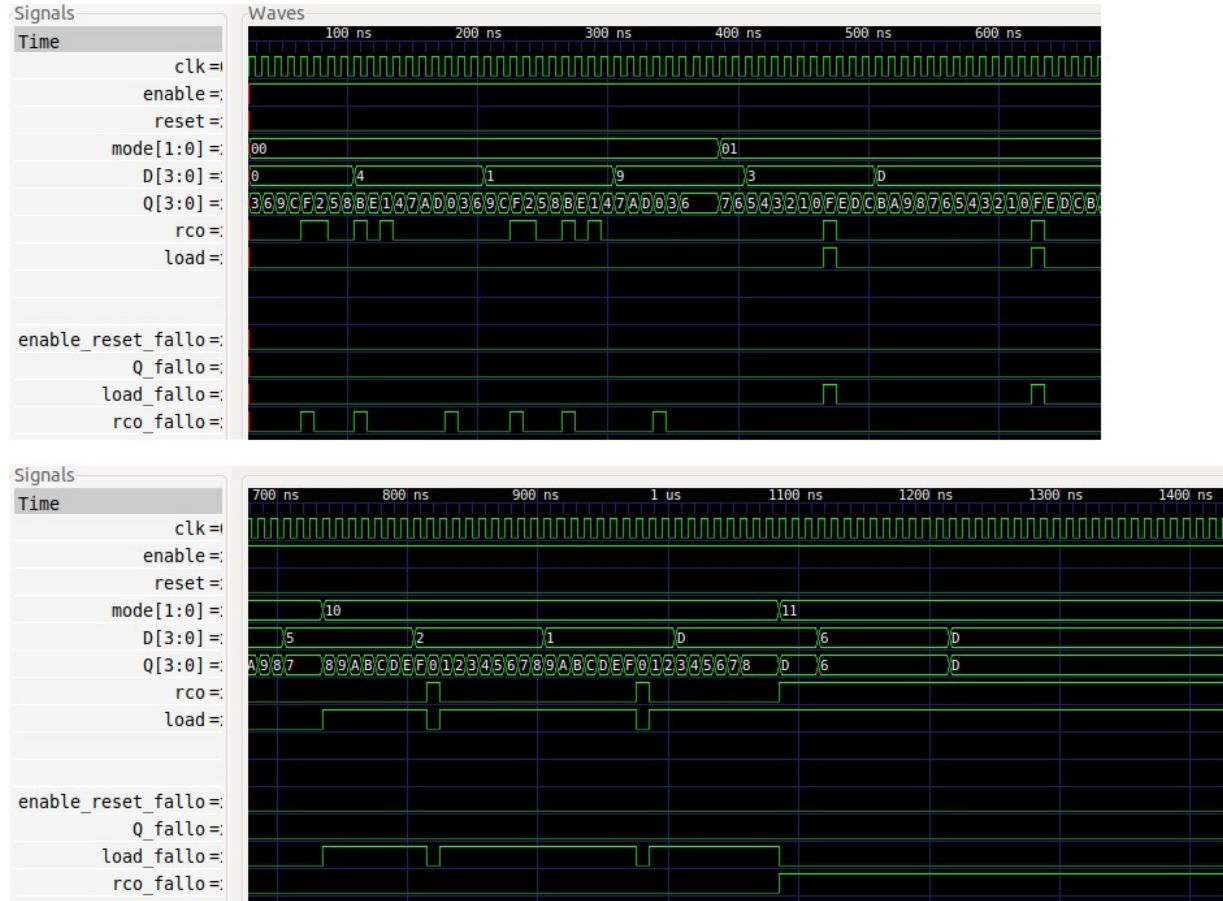


Figura 1: Resultado de la parte no aleatoria de la simulación del contador A.

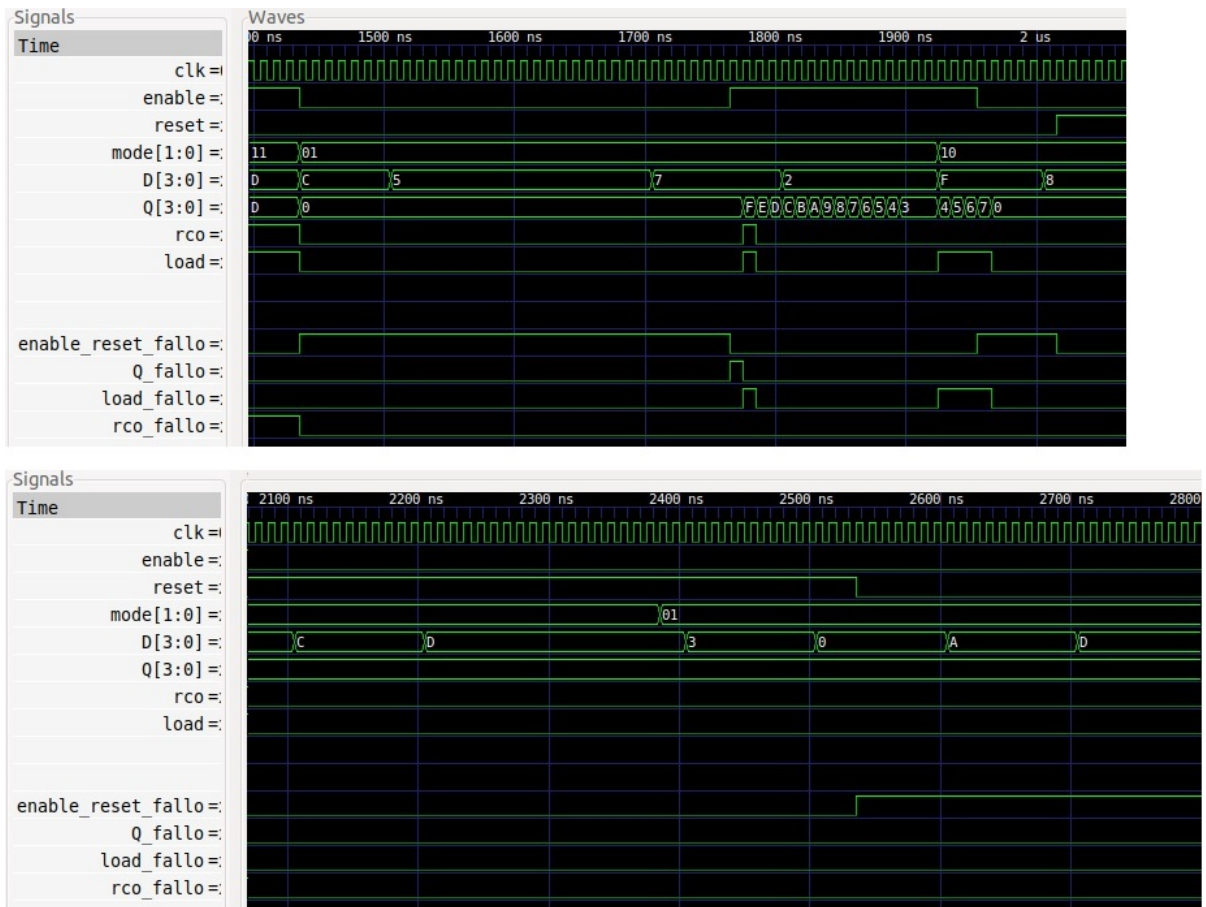


Figura 2: Resultado de la parte aleatoria de la simulación del contador A.

3.2. Contador B

De las figuras 3 4 se desprende que el contador no funciona correctamente en el modo 01 ($Q - 1$), en la salida LOAD ocurren errores durante el modo 00 para $Q = 8$ también cuando $enable = reset = 0$ (en este caso load se pone en alta impedancia). El rco presenta una anomalía cuando $Q = A$ y cuando Q está en alta impedancia.



Figura 3: Resultado de la parte no aleatoria de la simulación del contador B.



Figura 4: Resultado de la parte aleatoria de la simulación del contador B.

3.3. Contador C

De la figura 6 se puede notar que cuando $\text{reset}=\text{enable}=0$ la salida Q no se pone en alta impedancia, además el rco tampoco trabaja como debería.

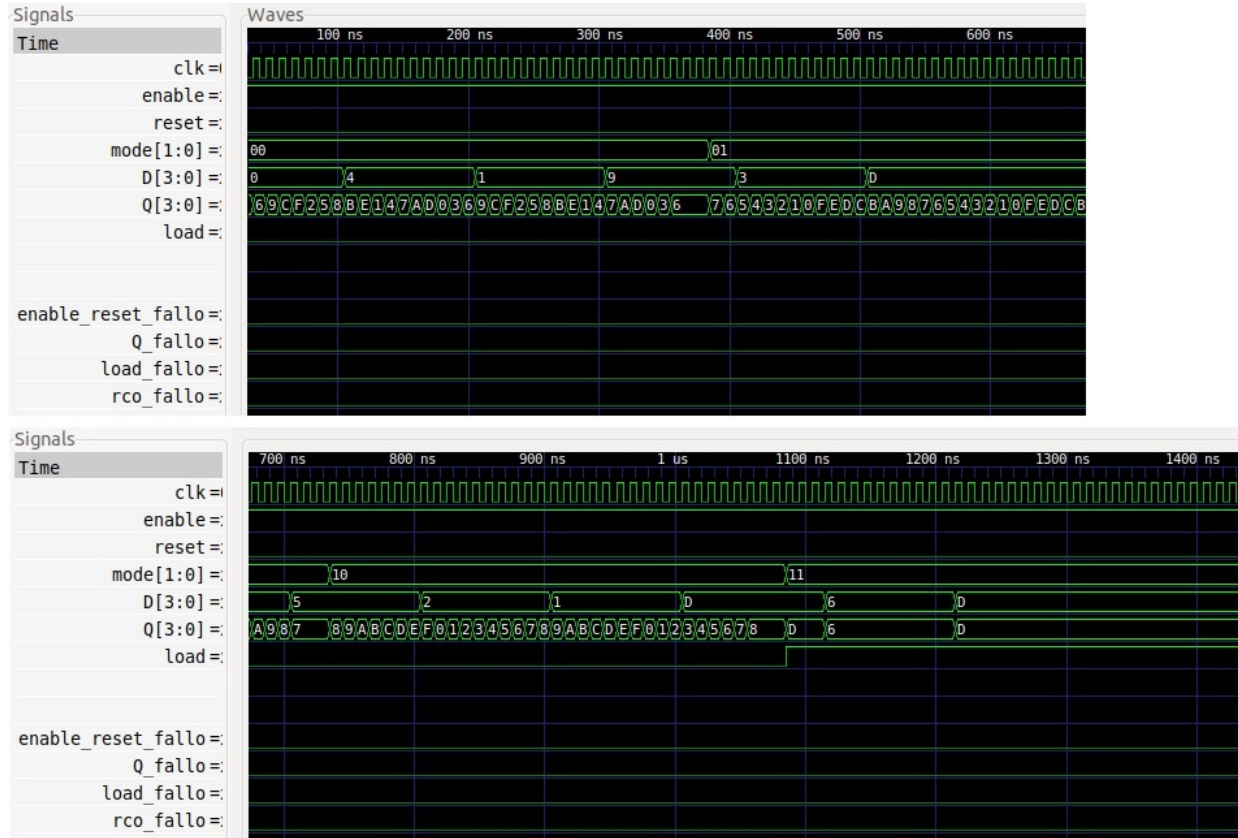


Figura 5: Resultado de la parte no aleatoria de la simulación del contador C.

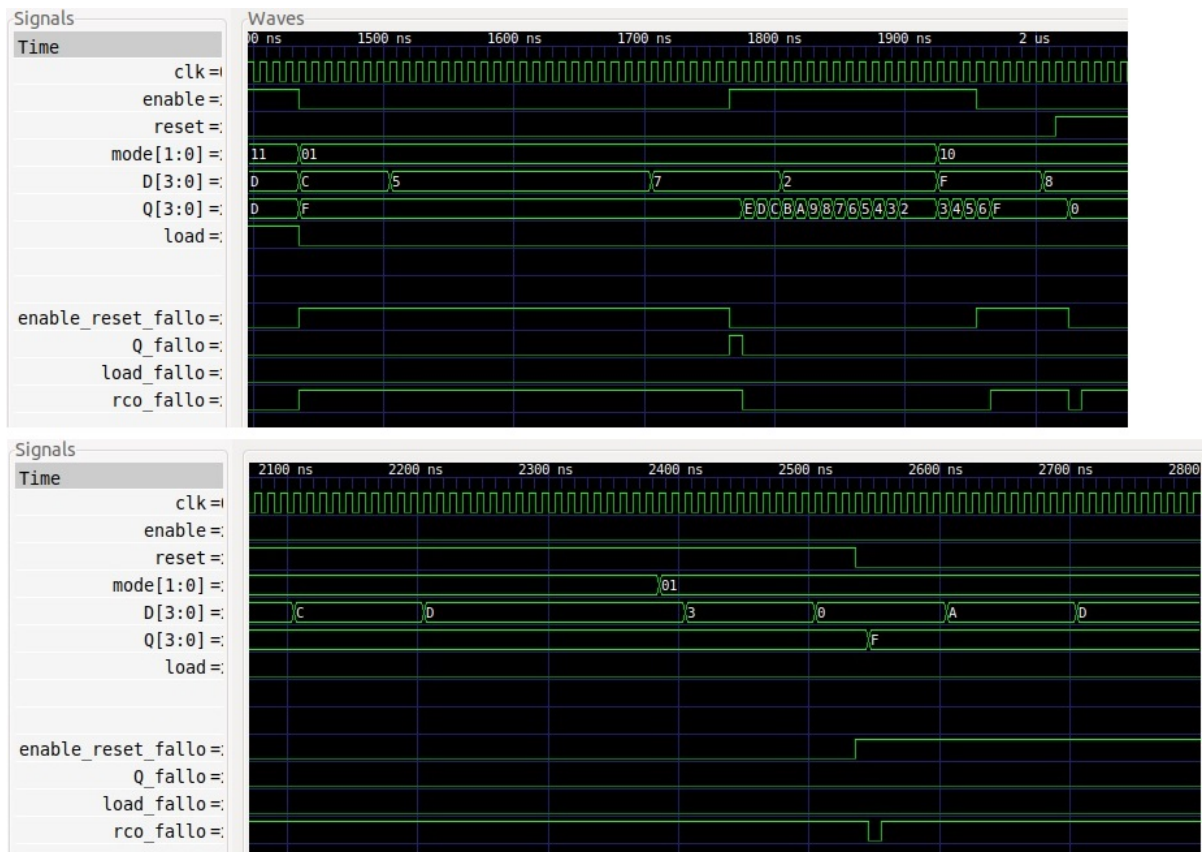


Figura 6: Resultado de la parte aleatoria de la simulación del contador C.

4. Conclusiones

- El contador A, en base a las pruebas, presenta los siguientes fallos:
 - Q: En modo 11 no se pone en alta impedancia.
 - rco
 - load: La señal es 1 en otros modos diferentes a 11.

Sin embargo los modos de conteo funcionan según las especificaciones.

- El contador B, presenta las siguientes anomalías:
 - Q: No funciona en el modo 01.
 - rco: Hay errores cuando Q está en alta impedancia y cuando Q es igual a A.
 - load: Presenta errores en el modo 00 para $Q = 8$ y para $enable = reset = 0$.

La operación para los modos 00, 10 y 11 es la que se espera.

- El contador C, tiene las siguientes malfunciones:
 - Q: No se pone en alta impedancia cuando $reset = enable = 0$.
 - rco: Falla cuando $reset = enable = 0$.

Los valores de Q son los adecuados para los 4 modos de operación.