

**EIE**

Escuela de  
**Ingeniería Eléctrica**

**Universidad de Costa Rica**  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica



**UNIVERSIDAD DE  
COSTA RICA**

**IE-0523 Circuitos Digitales II**

## Tarea 3 y 4

Victor Manuel Yeom Song - victor.yeom@ucr.ac.cr - B78494

I-2020

---

### 1. Tiempo invertido en cada sección

- 40 minutos para leer lo solicitado en la tarea y entender lo que se necesita para cada sección
- 40 minutos de investigación para la búsqueda de componentes y sus parámetros
- 30 minutos para la descripción estructural de cada componente
- 30 minutos para hacer el módulo probador con todas las pruebas
- 10 minutos para hacer el banco de pruebas
- 15 minutos para elaborar pruebas
- 3 minutos para el Makefile
- 40 minutos para la elaboración del reporte

## 2. Descripción arquitectónica/Diagrama del circuito

Se utilizaron como modelo los componentes de Texas Instruments descritos en el Cuadro 1.

Cuadro 1: Información de componentes utilizados en condiciones típicas (temperatura ambiente, tensión de salida en *high* de 3.3 V), obtenidos de Texas Instruments

Componente	Número serial	Precio unitario (\$)	Características temporales típicas (ns)	Consumo de potencia ( $\mu\text{A}$ )
NOT	SN74LV1T04	0.22	$t_{pd} = 4$	$I_{cc} = 1$
AND	SN74AUP2G08	0.35	$t_{pd} = 3,3$	$I_{cc} = 0,9$
OR	SN74LVC2G32-Q1	0.46	$t_{pd} = 5,8$	$I_{cc} = 10$
FFD	SN74LVC1G80-Q1	0.2	$t_{su} = 1,3$ $t_h = 0,9$ $t_{pd} = 4,2$	$I_{cc} = 10$
MUX 2:1 1 bit	SN74LVC2G157	1.11	$t_{pd} = 6$	$I_{cc} = 10$

En el Cuadro 1  $t_{pd}$  es tiempo de propagación desde una entrada a una salida y en todos los componentes es igual para cambios de bajo a alto o de alto a bajo.  $I_{cc}$  es la corriente máxima de alimentación para cada circuito. Específicamente para el FFD se pueden ver las características  $t_{su}$  y  $t_h$ , las cuales son los tiempos de *setup* y de *hold* respectivamente. El tiempo de *setup* es la cantidad de tiempo necesaria que se mantenga estable la señal de entrada del FFD antes del flanco positivo del reloj para tener seguridad de que las pudo registrar o muestrear bien, mientras que el de *hold* es la cantidad de tiempo necesaria después del flanco del reloj para que se pueda guardar la señal a la memoria del FFD. Por esto, estos últimos dos tiempos no son características internas del circuito que conforma el FFD y se deben manejar externo a él. Así, para implementar su control se debería manejar en el probador, pero como el enunciado de la tarea no solicita esto, no se hará, solo se tendrá en cuenta que en diseños más realistas se deben tomar en consideración.

Para construir el multiplexor 2:1 se tomó como base el diagrama que se encuentra en la hoja del fabricante del SN74LVC2G157, el cual se muestra en la Figura 1. Este presenta una entrada de habilitación, por lo que para la presente tarea se adaptó el diagrama para eliminarla. Esto se hace eliminando la entrada denotada por G y eliminando las compuertas AND que la reciben como entrada. Además, se elimina la salida correspondiente a  $\bar{Y}$ .

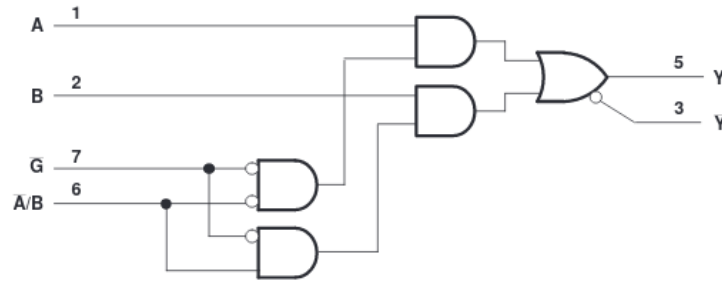


Figura 1: Circuito modelo para implementar el multiplexor 2:1 de 1 bit

A partir de este diagrama, además de las modificaciones mencionadas y las características temporales descritas en el Cuadro 1, se puede determinar el camino crítico del multiplexor. Este sería el que sale desde la señal selectora  $\bar{A}/B$  hasta Y, el cual consiste de una compuerta NOT, una AND y una OR. Así, el tiempo de propagación desde la señal selectora hasta la salida sería la suma del tiempo de propagación de cada uno de estos componentes, equivalente a  $t_{pd,sel} = 13.1$  ns. Nótese que se tomará este como el tiempo de propagación del multiplexor y no el obtenido de la hoja del fabricante, dado que en esta tarea se implementa el multiplexor a partir de las compuertas. También se debe considerar que este es el camino crítico para variaciones en la señal selectora. Para cualquiera de las otras señales, solo se tiene que pasar por un AND y un OR, por lo que el tiempo de propagación desde estas señales es de  $t_{pd,in} = 9.1$  ns

Luego, un multiplexor 2:1 de 2 bits es prácticamente igual a un multiplexor 4:2, como el que se muestra en la Figura 2, donde la señal selectora es igual para la selección de ambas señales. Es decir, para implementar este circuito lo único que se hace es tomar dos multiplexores 2:1 de 2 bits y se conectan en paralelo (la misma señal selectora para ambos y las señales de 2 bits como se muestran en la Figura 2), por lo que tendrá el mismo camino crítico.

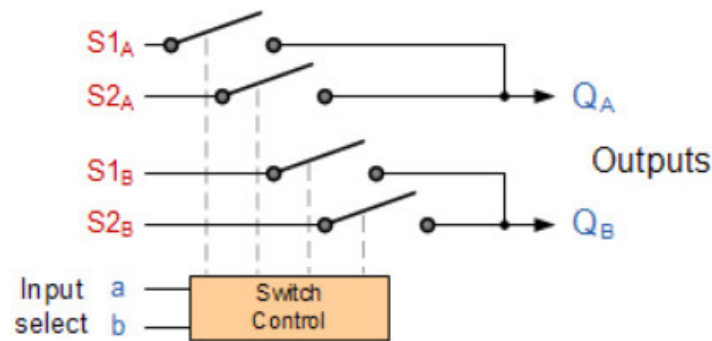


Figura 2: Multiplexor 4:2 general

Finalmente, con el multiplexor 2:1 de 2 bits se puede implementar el circuito deseado de la tarea anterior, cuyo esquemático se muestra en la Figura 3. Nótese que este circuito utiliza un FFD de 2 bits, por lo que simplemente se tomará el componente descrito en el Cuadro 1 y “colocando dos en paralelo” se emula un FFD de 2 bits.

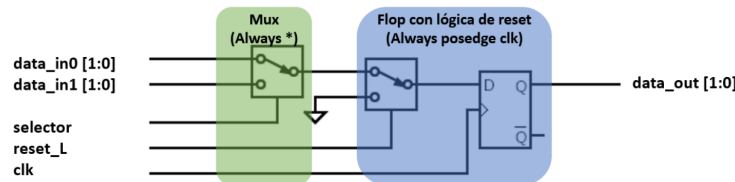


Figura 3: Esquemático del circuito a implementar, con entradas y salidas

## 2.1. Camino crítico

Para este trabajo se considerará el camino crítico como el camino desde las señales de entrada (data\_in) en lugar del camino más largo que corresponde a la señal selectora. Esto debido a que, dependiendo de la aplicación, la señal selectora se varía muy poco y por tanto tendría pocos

problemas en cuanto a propagación de las señales (lo cual se nota con el probador desarrollado), mientras que las señales de entrada pueden variar constantemente.

Además, no se considera el tiempo de lógica secuencial dado que este se encuentra únicamente al final. El efecto que tiene este tiempo sería en un desfase en la señal de salida física con respecto a la señal de salida ideal, siempre y cuando no se tenga una señal de reloj más pequeña que esta. Sin embargo, el tiempo invertido en lógica combinacional tiene un peso considerablemente mayor a éste, por lo que usualmente no tendría un efecto.

Así, se determina un camino crítico a partir de las señales de entrada `data_in` hasta la salida del segundo multiplexor, lo cual implica un tiempo de propagación total (lo cual es lo mismo que el periodo de trabajo mínimo)  $t_{pd} = 18.2$  ns. A su vez, esto conlleva a una frecuencia máxima de operación de  $f_{max} \approx 54.95$  MHz.

## 2.2. Checker y contador de transiciones

El *checker* indica al usuario cuando una señal de la descripción estructural discrepa de la conductual, mediante un mensaje de texto en la terminal. Esto se hace mediante la comparación constante en cada flanco creciente del reloj entre ambas señales.

El contador de transiciones consiste en un registro dentro del probador que cuenta la cantidad de flancos crecientes en la salida del circuito de la descripción estructural y la imprime en la terminal junto a la información de las señales.

## 3. Plan de pruebas

### 3.1. Compuertas y Flip-Flop D(FFD)

Para probar las compuertas básicas y el FFD simplemente se realizan tres señales (se podrían utilizar solo 2, pero se utilizan 3 para tener un poco más de claridad), las cuales se introducen en los circuitos para verificar tanto su comportamiento como sus características temporales. Para revisar el comportamiento lógico, se comparan las entradas introducidas en los circuitos con sus respectivas salidas y se revisa si el comportamiento tiene sentido según la función del componente utilizado. Cabe notar que se realizan pruebas con señales de 1 bit para el FFD (el cual se considera de 2 bits para la tarea), solamente para verificar su funcionamiento. Verilog asume automáticamente que el bit faltante del FFD será 0, aunque se obtendrán advertencias en la terminal, pero esto es esperado.

Para el comportamiento temporal, se revisó el comportamiento del circuito según los cambios en las señales utilizadas. Si el ancho de un pulso es menor al tiempo de propagación del circuito, no se verán los efectos del cambio de esta entrada en la salida del circuito, por lo que se estudiará esto mediante la aplicación de pulsos con duraciones tanto menores como mayores a los tiempos de propagación utilizados, lo cual debería dejar claras sus características por medio del estudio de la salida.

### 3.2. Descripción estructural

Para estudiar el comportamiento lógico de la descripción estructural implementada, se puede comparar su salida con la salida de la descripción conductual realizada en la tarea anterior, además del uso de un checker. El checker implementado imprime un mensaje en la terminal si hay una discrepancia.

Para la verificación de las características temporales, se revisa la señal de salida del circuito cuando se utiliza un periodo de reloj mayor al tiempo de propagación máximo (calculado con el

camino crítico en la sección de descripción arquitectónica) y cuando se utiliza uno menor, de modo que se pueden estudiar las diferencias entre ambos casos.

## 4. Instrucciones de simulación

Para realizar las pruebas, simplemente se abre una terminal en el directorio donde se encuentra el Makefile y usar el comando `make` para correr el código. El código y el Makefile se encuentran adjuntos dentro del zip junto al presente reporte.

Para modificar la frecuencia de las pruebas se ingresa al probador y se modifican los anchos de los pulsos de reloj utilizados. Estos se encuentran al final del archivo.

## 5. Resultados

### 5.1. Pruebas de las compuertas

Se utilizó una señal de entrada `gate_stim_indiv` para las pruebas de la compuerta NOT, mientras que para las demás compuertas y el FFD se utilizaron otras dos señales, `gate_stim1` y `gate_stim2`. Para el FFD se introdujo la señal `gate_stim1` a la entrada de reloj, mientras que `gate_stim2` se introdujo a la entrada D.

Primero se utilizaron tiempos de ancho de señal de 6 ns para verificar el funcionamiento lógico de cada componente. El resultado de esta prueba se presenta en la Figura 4.

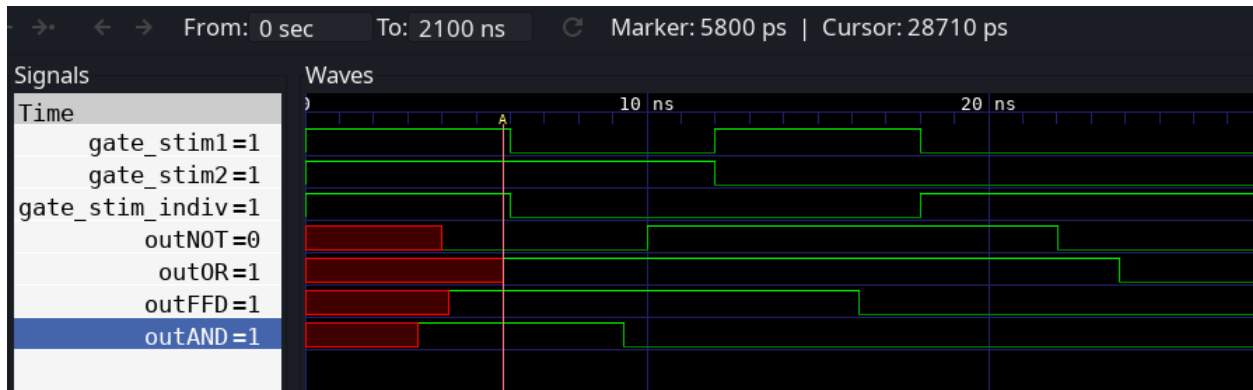


Figura 4: Funcionamiento correcto de las compuertas y el FFD con pulsos de ancho de 6 ns

Se ve que el comportamiento lógico es el que se espera, aunque bien hay retardos con respecto a la aplicación de las señales de las entradas debido a las características temporales aplicadas. Inicialmente verilog les asigna barras rojas a las señales dado que no conoce sus condiciones iniciales. Estas se corrigen rápido después de que se aplican las señales de entrada.

Cuando tanto `gate_stim1` como `gate_stim2` son 1, la salida del AND `outAND` es 1, mientras que cuando `gate_stim1` baja a 0, `outAND` lo hace también.

La salida del FFD, denotada por `outFFD`, guarda el valor inicial de `gate_stim2` por el primer flanco positivo de `gate_stim1`, mientras que después del segundo flanco positivo de `gate_stim1` `outFFD` se vuelve 0 para reflejar el cambio de `gate_stim2`, con el respectivo retardo del componente.

La salida del OR, denotada por `outOR`, es la que presenta el retardo más grande ya que es la compuerta que se escogió con mayor tiempo de propagación. Se puede ver que su valor se mantiene

en 1 hasta que tanto `gate_stim1` como `gate_stim2` son 0, después de lo cual `outOR` también se vuelve 0.

Finalmente, se ve que la salida del NOT, denotada por `outNOT`, siempre sigue el opuesto de la entrada `gate_stim_indiv`.

En la Figura 5 se ve que mientras están variando las señales de entrada, las salidas de los componentes se encuentran indeterminadas. No es hasta que todas las señales dejan de variar que se obtienen salidas determinadas, lo cual contrasta a las gráficas señaladas en la Figura 4.

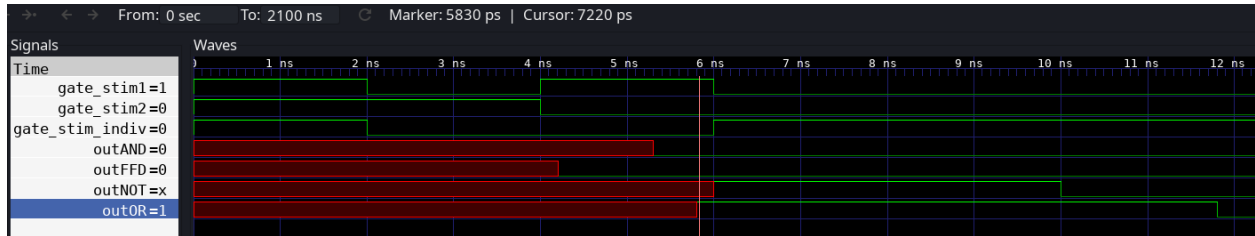


Figura 5: Funcionamiento erróneo de los componentes por sus tiempos de propagación.

## 5.2. Pruebas del diseño estructural

Se inició con un ancho de pulso de reloj relativamente grande para poder comparar directamente las salidas de la descripción conductual con la estructural, reduciéndolo gradualmente hasta llegar a un punto en el que las salidas difirieran, esto para hallar el tiempo de retardo de propagación experimental. El último ancho de pulso de reloj para el que la salida del módulo estructural coincidía con la del conductual fue para un ancho de pulso de reloj de 9 ns, es decir un periodo de 18 ns. Este resultado se muestra en la Figura 6.

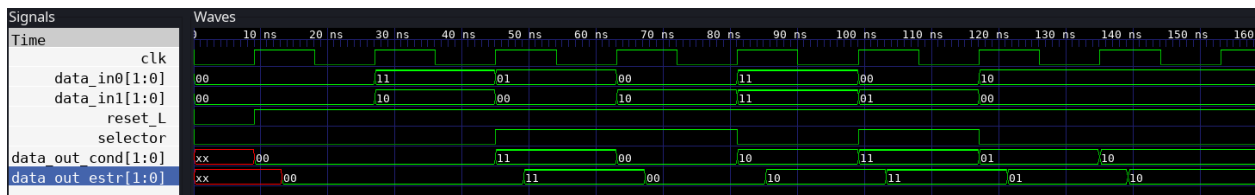


Figura 6: Salidas de la descripción conductual y la descripción estructural coincidentes

Si bien hay un pequeño desfase entre las señales, esto es de esperarse debido al retardo del FFD en la descripción estructural.

Después, se redujo el ancho de pulso de reloj a 8.9 ns, o sea un periodo de 17.8 ns y se obtuvo el resultado mostrado en la Figura 7.

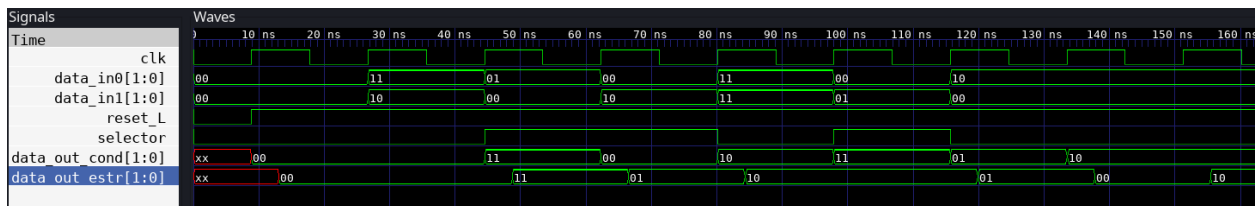


Figura 7: Salidas de la descripción conductual y la descripción estructural con discrepancias

Aquí se notan diferencias entre las señales de salida, por ejemplo en el segundo bucle que se observa en la señal de salida del modelo estructural. Por esto, se determina que el periodo mínimo de trabajo para el circuito es de aproximadamente  $T_{min,exp} \approx 18$  ns, y de manera análoga, se tiene que la frecuencia máxima experimental es de  $f_{max,exp} \approx 55.56$  MHz.

Con el ancho de pulso de tiempo utilizado para generar la Figura 7 se pudo verificar el funcionamiento del *checker* y el contador de transiciones positivas implementados, como se muestra en la Figura 8.

Inconsistencia a los										1513
151	1	1	0	10	00	00	10	0000	4	
156	1	1	0	10	00	10	10	0000	5	
160	0	1	0	10	00	10	10	0000	5	
169	1	1	0	10	00	10	10	0000	5	
178	0	1	0	10	00	10	10	0000	5	
187	1	1	0	10	00	10	10	0000	5	
gtkwave ./estr.vcd										

Figura 8: Verificación del *checker* y el contador de transiciones positivas

La última columna de datos numéricos corresponde a la cuenta de transiciones positivas en la salida del modelo estructural y se ve la notificación de inconsistencias en la pantalla. Esta notificación indica el tiempo al que se dio la inconsistencia.

## 6. Análisis y Conclusiones

En general, se puede decir que se obtuvieron los resultados esperados con el diseño. El comportamiento de cada componente, así como el del circuito completo, fue aceptable en cuanto a su funcionamiento lógico y se verificó la importancia de respetar los tiempos de propagación de los componentes lógicos para poder realizar circuitos lógicos realizables a nivel físico.

Se nota que el periodo mínimo de trabajo obtenido experimentalmente es distinto al calculado teóricamente. De hecho, el periodo mínimo experimental es menor al calculado teóricamente por 0.2 ns, ( $t_{pd} = 18.2$  ns,  $T_{min,exp} = 18$  ns). Esto se puede deber a un error de precisión producido por verilog, ya sea por algoritmos de división numérica o por aspectos desconocidos por el usuario, dado que la diferencia realmente no es grande.