UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE0523 – Circuitos Digitales II

Tarea 3 y 4

Yeison Rodríguez Pacheco, B56074

4/04/19

Resumen

En esta tarea se realiza una biblioteca de componentes con sus respectivos tiempos de propagación, setup y hold. Además se realizará la descripción estructural del demultiplexor de la tarea anterior, y se comprobará con un checker la equivalencia entre este modelo y el conductual.

1. Contabilización del tiempo

Tabla 1: Contabilización del tiempo

Sesiones	Sesión 1	Sesión 2	Sesión 3
Búsqueda y estudio de información	50 min	25 min	0 min
Diseño inicial del circuito	0 min	40 min	0 min
Programación del código	2h	3h	2h
Ejecución y pruebas	0min	25 min	20 min
Reporte	0 min	0 min	2.5h

2. Descripción arquitectónica del circuito

En la figura 1 vemos un diagrama estructural del circuito diseñado, es el mismo circuito de la tarea 2 solo que en este caso se debe realizar una descripción estructural del mismo. Podemos ver que con ayuda de Flip Flops, mux, y demux es posile resolver el problema propuesto. Estos componentes están hechos a partir de una biblioteca de compuertas que se crearon con sus retrasos correspondientes, por lo que al presentarse una entrada en el circuito se debe esperar un tiempo de propagación. También los Flip Flop cuentan con un detector de que se cumpla el tiempo de setup y hold.

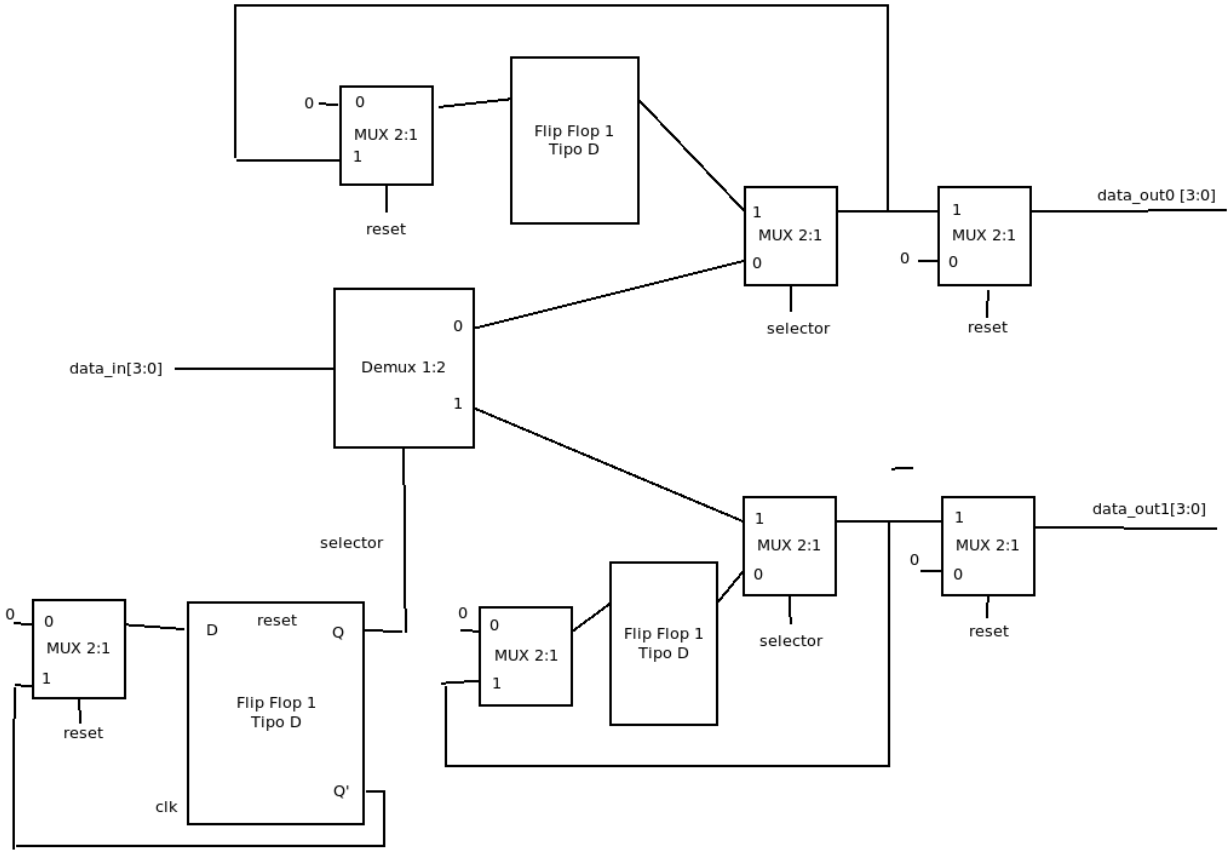


Figura 1: Diagrama arquitectónico del circuito diseñado. Hecho en Dia

3. Plan de pruebas

Para el diseño realizado se dividen las pruebas en dos secciones, la comprobación del correcto funcionamiento de la biblioteca de compuertas y la equivalencia lógica entre el modulo conductual y estructural del demux de la tarea anterior.

3.1. Tarea 3:Biblioteca de componentes.

Para la comprobación del correcto funcionamiento de la biblioteca de componentes se realizó una prueba a todos los componentes en la cual se vieran forzados a cambiar entre todos sus estados, esto para verificar su funcionamiento lógico. También se corroboró su funcionamiento temporal observando que una vez que se cambian las entradas de cada componente, su salida cambia hasta pasada cierta cantidad de tiempo que es establecida en la descripción de cada compuerta.

Para el componente Flip Flop se realizó un código de instrumentación, el cuál detecta si la entrada del Flip Flop cambió entre los tiempos de Set Up y Hold, si es así el programa genera un mensaje dependiento del tipo de violación y se detiene la ejecución. Todas las pruebas están implementadas por medio del makefile, y son posibles de observar en GTKwave. En la comprobación de resultados se podrán observar estas pruebas.

La biblioteca de componentes se encuentra en la carpeta Tarea3 en el archivo biblioteca_componentes.v

3.2. Tarea 4: Modelo conductual vs Modelo estructural

En esta ocasión se debe realizar el modelo estructural de manera manual con los componentes de la biblioteca creada anteriormente. Para corroborar el correcto funcionamiento se realizó un probador el cuál contiene un checker que hace la comparación entre la salida del modelo estructural y conductual. Al ejecutar la simulación se puede observar en GTKwave y en consola que el funcionamiento del circuito es el esperado. También se realizaron pruebas variando la lógica para que los modelos no fueran equivalentes, en este caso el ckecker saltó y detuvo el programa.

También se cuenta con dos pruebas adicionales en las cuales se varían las señales reset y entrada, para verificar que no existan casos esquina.

4. Instrucciones de utilización de la simulación

4.1. Tarea 3

Para ejecutar la tarea 3 se debe posicionar en la carpeta correspondiente llamada Tarea3 desde su terminal, y ejecutar los siguientes comandos.

```
make
make gtk
```

Al ejecutar GTKwave encontraremos las respuestas de las compuertas en el siguiente conjunto de señales:

```
Banco_pruebas_conductual
    biblioteca_prueba_comp
        CompuertaNand
        etc
```

4.2. Tarea 4

Para ejecutar las pruebas de esta tarea, debemos posicionarnos en nuestra terminal en Tarea4, aquí se encuentra un makefile el cuál contiene los comandos vistos en la tabla 2.

Tabla 2: Comandos de Makefile

Comandos de Makefile	
make	Compila y ejecuta la prueba0, la cuál es la que se dió en el enunciado original.
make gtk	Abre el GTKwave con los datos de la simulación compilada actualmente
make prueba0	Compila y ejecuta la prueba0, la cuál es la que se dió en el enunciado original.
make prueba1	Compila y ejecuta la prueba1.
make prueba2	Compila y ejecuta la prueba2.
make clean	Borra los archivos binarios y .vcd
make pdf	Abre el pdf del informe.

A continuación ejemplos para ejecutar las pruebas:

```
make
make gtk

make prueba1
make gtk

make prueba2
make gtk
```

En las señales generadas por el probador se podrá observar el comportamiento del modelo conductual y estructural.

5. Ejemplos de los resultados

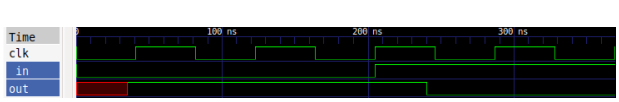
5.1. Tarea 3

En la tabla 3 se presenta una descripción de los componentes utilizados en las simulaciones. Los tiempos y potencia que aparecen son las máximas del circuito, en las simulaciones se utilizaron los valores típicos, que se pueden encontrar en las hojas del fabicante adjuntas en este trabajo.

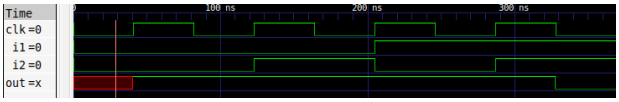
Tabla 3: Tiempos propagación, consumo de potencia y precio de componentes

Componente	NAND	NOT	NOR	Flip Flop
T_{setup}	—-	—-	—-	20
T_{hold}	—-	—-	—-	5
Precio unitario	0.3$	0.53$	0.3$	0.15$
Tiempo propagación	500	60	250	—-
Potencia	500mW	300uW	500mW	500mW

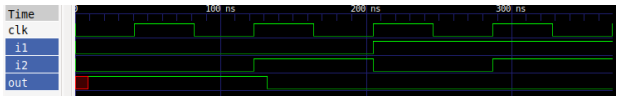
En las figuras 2 podemos observar el comportamiento temporal y lógico de cada una de los dispositivos creados en la biblioteca.



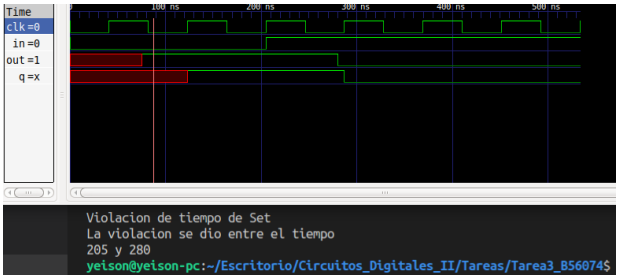
(a) Compuerta Not



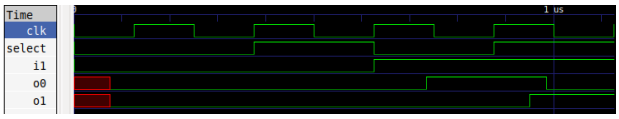
(b) Compuerta Nand



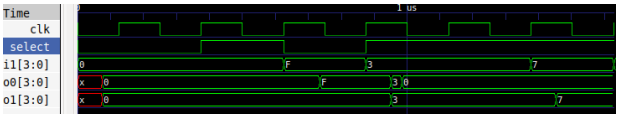
(c) Compuerta Nor



(d) Flip Flop con código de instrumentación



(e) Demultiplexor 1:2 de 1 bit1



(f) Demultiplexor 1:2 de 4 bits

Figura 2: Comportamiento lógico y temporal de la biblioteca de componentes.

5.2. Tarea 4

En las figuras 3 podemos observar el comportamiento de los modelos estructural y conductual ante estímulos en sus entradas. Se realizaron 3 pruebas que pueden ser repetidas desde el makefile.

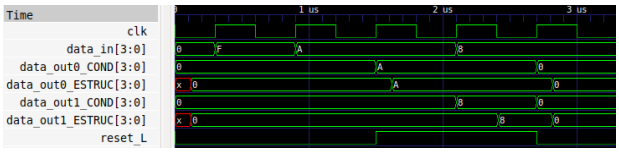
6. Análisis y conclusiones

6.1. Análisis de resultados

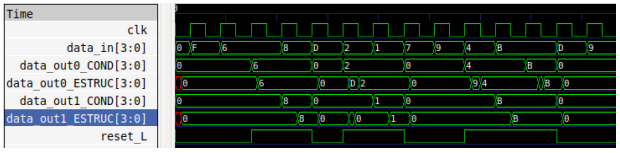
6.1.1. Tarea 3

Con respecto a la tarea 3, vemos en las figuras 2 el comportamiento lógico y temporal de cada compuerta, observamos que en cada caso la o las entradas cambian entre todos sus estados posibles, y la salida se comporta como debería, por ejemplo la Nand 2b solo presenta un 0 en su salida cuando ambas entradas son 1. Es posible verificar este análisis con todos los componentes. La salida en cada caso tarda un tiempo extra en propagarse a partir del estímulo de las entradas, esto ocurre por los retrasos programados en los componentes.

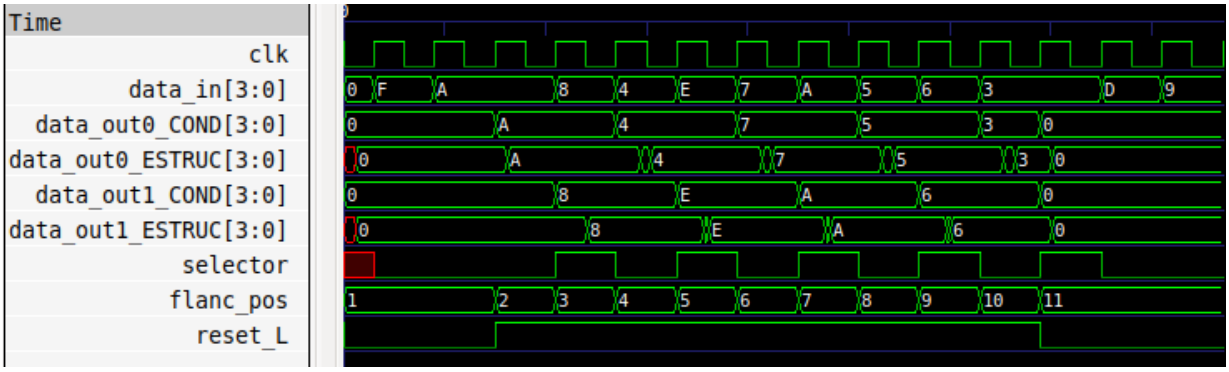
Es importante hacer énfasis en el comportamiento del Flip Flop (2d), notamos que en ese ejemplo se hizo que la señal de entrada del Flip Flop cambiara violando el tiempo de Set del componente, por lo que este despliega una alarma con el tiempo en el cuál se realizó la violación, y el tipo (set o hold).



(a) Prueba 1



(b) Prueba 2



(c) Prueba 0. Original de la tarea anterior

Figura 3: Comparación entre los modelos conductuales y estructurales de diferentes pruebas.

6.1.2. Tarea 4

Con respecto a la tarea 4, en las figuras 3 podemos observar el comportamiento estructural y conductual del demux 2:1 de la tarea 2. Vemos que el comportamiento de los modelos estructurales es más lento para responder ante el cambio positivo en la señal de reloj, esto ocurre porque está hecho con los componentes de la sección anterior, por lo que tiene un tiempo de propagacion mayor a 0, a diferencia del modelo conductual que actua de manera instantánea. Entre las señales de las figuras tenemos una llamada flanc_pos la cuál se encarga de contar las transiciones de flancos positivos que han ocurrido en la salida. También se agregó un checker al probador para corroborar cuando las señales conductuales y estructurales son distintas, en este caso finaliza la simulación, este fue probado y se verificó su correcto funcionamiento.

Podemos observar en la prueba2 3b que en algunos casos, el modelo estructural presenta variaciones en las salidas que parecen ser erráticas, esto es un coportamiento normal ya que al ser hecho con compuertas con retrasos se pueden presentar estas variaciones si un camino lógico llega primero que otro. Esto no es ningún inconveniente mientras no se viole los tiempos de Set y hold, que no es el caso, ya que sino hubiese saltado una alarma de que se violaron estas condiciones.

6.2. Inconvenientes

Al realizar este diseño se tuvieron algunos inconvenientes, los cuales fueron:

- Dificultad para utilizar la sintaxis de tiempos de propagación en compuertas estructurales, ya que el editor de texto mostraba una advertencia y no compilaba el archivo de manera correcta. Esto ocurria porque faltaba la bandera Ttyp a la hora de ejecutar Icarus verilog.
- Dificultad para crear el código de instrumentación del Flip Flop, ya que fue dificil de implementar y diseñar.

- Complicaciones a la hora de realizar las conexiones en el modelo estructural del Demux, principalmente porque es fácil cometer errores.

6.3. Conclusiones

- Se logró crear una biblioteca de componentes con tiempos de transición, además de un Flip Flop con código de instrumentación que se encargue de detener la simulación si se presenta una violación a los tiempos de Set y Hold.
- Se pudo verificar que el comportamiento del modelo estructural creado con la biblioteca de componentes es equivalente al modelo conductual de la tarea 2.
- Creación de un checker que detiene el programa si se presenta una inconsistencia entre el modelo conductual y estructural, también un contador de transiciones.
- Se realizó un Makefile que pudiera automatizar varias pruebas con la ayuda de sed, para facilitar el uso del diseño.
- Al realizar un modelo estructural, se puede notar el valor de una herramienta sintetizadora, ya que es un trabajo muy laborioso.