

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

## Tarea 5

Yeison Rodríguez Pacheco, B56074

11/04/19

### **Resumen**

En esta tarea se realiza una síntesis estructural mediante Yosys del mismo circuito generado en las tareas 2, 3 y 4. El demultiplexor de 4 bits.

# 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
Mejora del Makefile	1 h	50 min	0 min
Programación del código	40min	10min	0
Ejecución y pruebas	0min	35 min	40 min
Reporte	0 min	0 min	1.5h

# 2. Descripción arquitectónica del circuito

Debido a que es el mismo circuito de la tarea anterior, el mismo diagrama aplica. 1

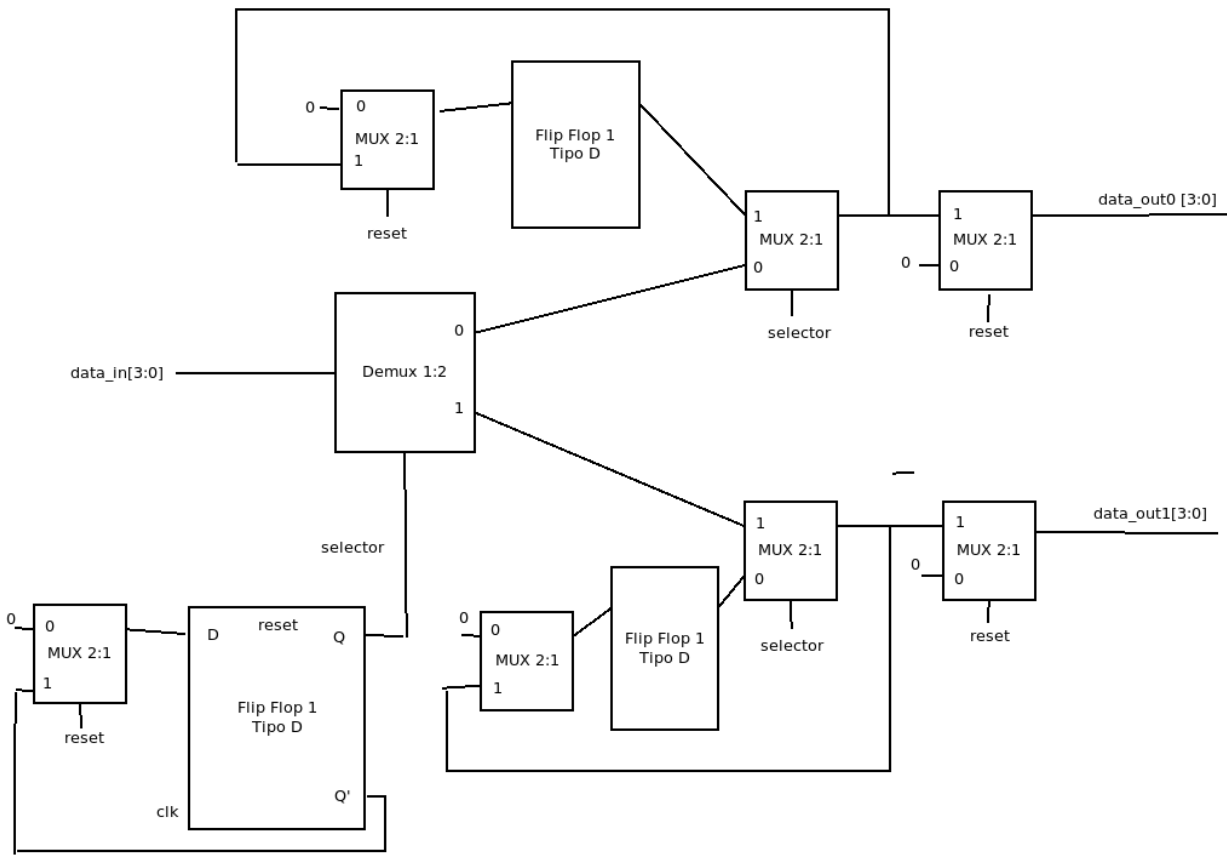


Figura 1: Diagrama arquitectónico del circuito diseñado. Hecho en Dia

# 3. Plan de pruebas

Las pruebas consisten en realizar una síntesis genérica de la descripción conductual hecha en la tarea 2 del demultiplexor con ayuda de Yosys, para proceder a verificar que su comportamiento con-

ductual sea igual al estructural.

El siguiente paso es generar una síntesis con una tecnología conocida, en este caso se utilizaron las compuertas Cmos\_cells.v de la documentación de Yosys.

Seguidamente se agregan retrasos a las compuertas del archivo Cmos\_cells.v para finalizar verificando que el diseño estructural aún es funcional con respecto al conductual, y hacer una comparación entre los modelos conductuales de la tarea anterior (generado con nuestra biblioteca) y el generado por la librería Cmos\_cells.v.

En todas las pruebas se hace una verificación con el checker para comprobar equivalencia lógica, además se realizó un cálculo de la frecuencia de operación máxima variando el periodo de la señal de reloj.

## 4. Instrucciones de utilización de la simulación

En esta tarea se presentarán 3 carpetas principales, en la que se llama sintesis\_generica se presentan los resultados de la síntesis estandar de Yosys.

En la carpeta sintesis\_cmos\_cells se presenta la síntesis con los comandos de yosys recomendados en el enunciado de la tarea, las compuertas aún sin retrasos y utilizando el archivo Cmos\_cells.v.

En la carpeta sintesis\_con\_retrasos se presenta el archivo Cmos\_cells.v con los retrasos correspondientes.

Para ejecutar cualquiera de las pruebas en distintas carpetas, simplemente se debe posicionar en la terminal en cualquiera de las mismas y ejecutar el comando:

```
make
```

En la tabla 2 se dejará una lista detallada de comandos del makefile, muchos de estos solo se programaron para la creación del laboratorio, y no para las pruebas en sí.

Tabla 2: Comandos Makefile

Comandos del Makefile para pruebas (usuario)	
make	Compila y ejecuta la prueba estandar, también abre el GTKwave con las simulaciones
make gtk	Ejecuta nuevamente el gtkWave, por si necesita abrirlo de nuevo sin compilar
make clean	Remueve todos los archivos .o y .vcd
make pdf	Abre el pdf del informe
Comandos de Makefile para facilitar generación de código (diseñador)	
make generar_archivos	Crea los archivos conductual, probador y banco de prueba según variables de los nombres deseados especificadas en el Makefile (Comando parametrizado)
make generar_archivo_yosys	Genera el archivo con los comandos de Yosys con respecto a los nombres elegidos por el diseñador en las variables del Makefile (Comando parametrizado)
make llenarBancoPrueba	Este comando hace los include e instancias necesarias para que el banco de pruebas funcione correctamente. (Comando parametrizado)
make ejecutar_yosys	Este comando ejecuta el archivo .ys y además se encarga de cambiar el nombre del módulo generado por la síntesis, y hace automaticamente el include del archivo que contiene la librería de compuertas a utilizar (Comando parametrizado)

## 5. Discusión y Análisis de resultados

### 5.1. a.

Para esta descripción estructural genérica se realizó la síntesis sin ningún tipo de archivo .lib, por lo que el resultado de la misma contiene en su mayoría assign, y también algunos always posedge clk para simular los FF

### 5.2. b.

Vemos en la figura 2 que tanto el comportamiento estructural como el conductual son iguales.

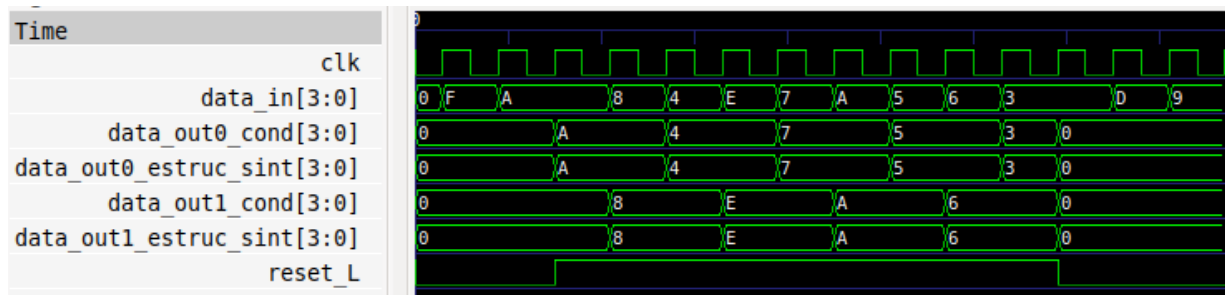


Figura 2: Respuesta conductual y estructural de la síntesis hecha con Yosys sin ningún tipo de archivo .lib. Hecho en GTKwave

### 5.3. c.

Se utilizó cmos\_cells.lib como librería para crear el archivo estructural con yosys.

### 5.4. d.

Al utilizar el archivo cmos\_cells.v para realizar la simulación, se obtuvo lo que vemos en la figura 3, podemos ver que la respuesta conductual y estructural son iguales, por lo que se concluye que la síntesis es correcta. También se verificó que no existiera ningún latch en el diseño. También se cuenta con el checker en el probador para saber si existen incongruencias entre la salida conductual y estructural.

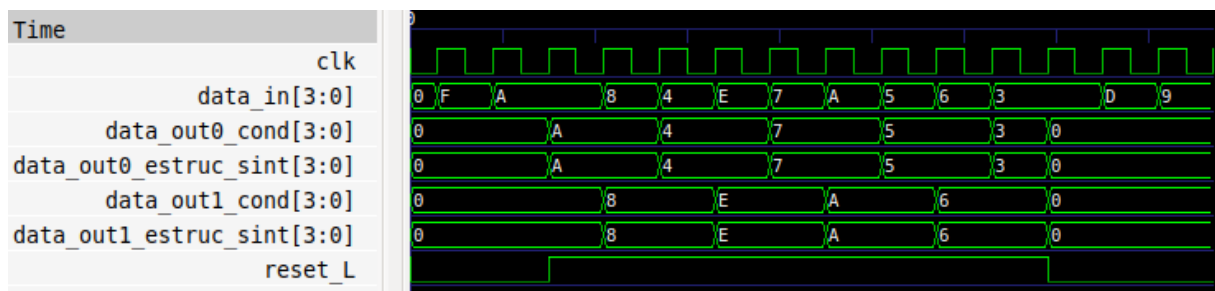


Figura 3: Respuesta conductual y estructural de la síntesis hecha con Yosys. Hecho en GTKwave

### 5.5. e.

Al modificar el archivo cmos\_cells.v para que se analicen los retardos, se obtiene lo que vemos en la figura 4. Vemos que aunque se presentan retrasos el circuito sigue comportándose como debería, ya que estos retrasos son pequeños con respecto al tamaño del ciclo de reloj. EL checker también se encuentra implementado en este probador.

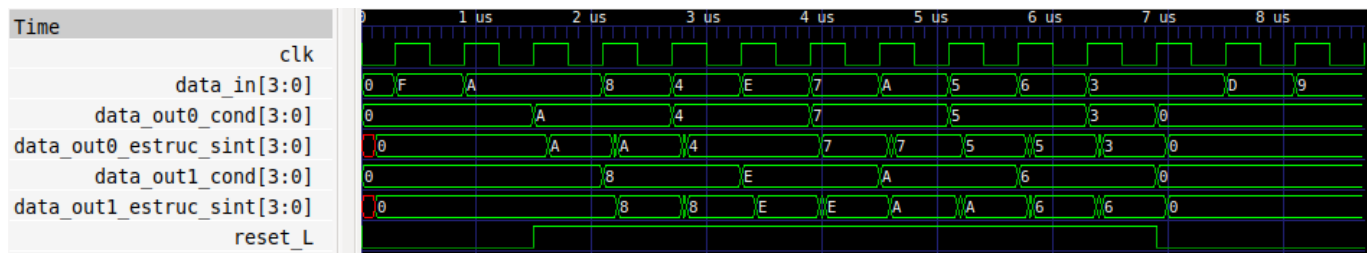


Figura 4: Respuesta conductual y estructural con retardos de la síntesis hecha con Yosys. Hecho en GTK-wave

## 5.6. f.

1. Para este diseño se utilizaron 32 NAND, 2NOR, 2NOT y 9 DFF, por lo que son un total de 45 componentes.
2. El costo solo por los componentes, usando los datos recopilados en la tarea pasada sería de 22,21 dolares (aproximadamente 13326 colones). Tomando en cuenta que la hora de trabajo de un estudiante de ingeniería podría ser de 3000 colones, y que se tomó aproximadamente 11h en realizar las tareas correspondientes, se puede concluir que el precio del dispositivo final debería ser de aproximadamente 40726 colones.
3. Para el modo -Tmin (tiempos mínimos de las compuertas) se encontró de manera manual (variando la frecuencia hasta que el checker avise que se presentó un error ya que las salidas conductual y estructural no son iguales) que la frecuencia máxima es de 38Mhz. Para -Ttyp se encontró que la frecuencia máxima es de 8.065MHz. Para -Tmax se obtuvo que la frecuencia máxima era de 4.9Mhz.
4. Ya que la frecuencia máxima de operación es de 38Mhz y que se utiliza la versión simplificada del cálculo de potencia (contando las transiciones de flancos positivos), se hace uso del contador de flancos para averiguar cuantos se presentan en esta frecuencia y se hace el cálculo. La potencia consumida es de 2000000. Claro está que el cálculo es una aproximación pésima de la realidad, debido a que si se quisiera calcular la potencia real se debería medir las transiciones de cada transistor.

## 5.7. g.

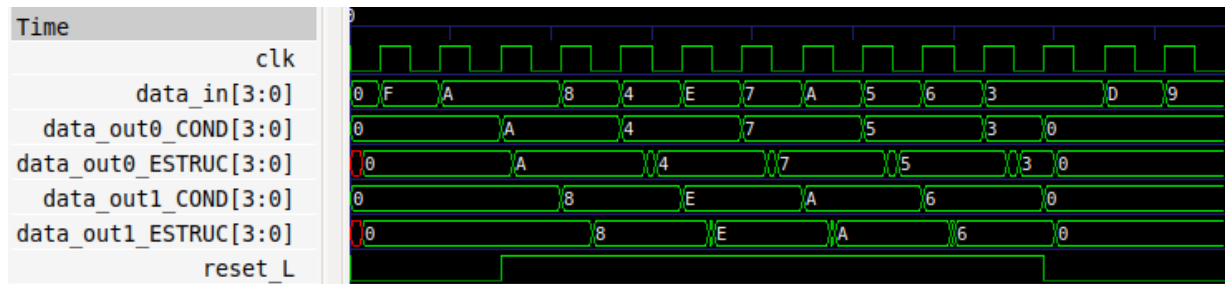
Para agregar esta instrucción al makefile se utilizaron parámetros (para hacer un makefile parametrizado que funcione en todas las tareas), el código es el siguiente:

```
yosys yosys_comandos.ys; sed -i 's/module
$(Nombre_archivo_conductual)/module $(Nombre_archivo_estructural)/g'
$(Nombre_archivo_estructural).v;
```

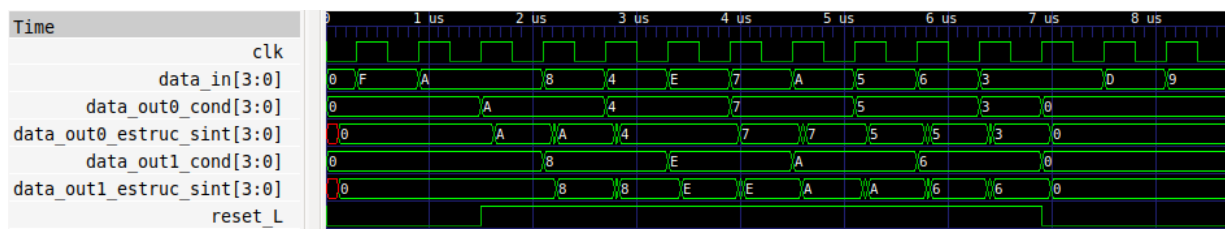
También se agregaron otras funcionalidades al makefile, como que haga el include de la librería elegida por el usuario, un ejemplo sería la cmos\_cells.v, y que genere el archivo de comandos de yosys de manera automática.

### 5.8. h.

En las figuras 5 vemos la comparación entre las simulaciones hechas con el modelo estructural de manera manual con la biblioteca que se creó también de manera manual y la síntesis hecha con Yosys con la librería cmos\_cells.v.



(a) Síntesis de la tarea 4 con biblioteca hecha de manera manual



(b) Síntesis hecha con Yosys y biblioteca cmos\_cells.v

Figura 5: Comparación entre el modelo sintetizado con Yosys y el modelo estructural hecho en la tarea anterior.

Al realizar los diseños de manera manual se aprende cómo es el funcionamiento de un archivo estructural, pero no es algo que sea factible en la industria ya que lleva mucho tiempo hacer los modelos estructurales a mano, y no tendría mucho sentido realizar el modelo conductual si de todas formas se debe ir a conectar compuerta por compuerta. En cambio con Yosys y bibliotecas ya programadas es mucho más sencillo ya que se genera automáticamente, además que si se programó de manera correcta, se podrá verificar la equivalencia entre el modelo conductual y estructural. Con Yosys se debe tener cuidado ya que podrían aparecer latches si se programó de manera inadecuada.

### 5.9. i.

Se utilizó el script completo de Yosys, es posible verificarlo en los archivos correspondientes.

## 6. Conclusiones

- Fue posible verificar la funcionalidad del programa Yosys, así como entender la importancia de tener a la mano herramientas sintetizadoras
- Se pudo comprobar que es posible agregar retardos a las bibliotecas, para asemejar las simulaciones a la realidad.
- Es importante tener un buen Makefile para no reinventar la rueda.