

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 5

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

I-2020

1. Tiempo invertido en cada sección

- 30 minutos para leer lo solicitado en la tarea y entender el funcionamiento de Yosys
- 20 minutos en el Makefile
- 30 minutos confeccionando scripts de yosys y bancos de prueba para cada síntesis
- 30 minutos realizando pruebas y revisando la consistencia de los resultados
- 40 minutos elaborando el reporte

2. Descripción arquitectónica/Diagrama del circuito

Se utilizó como guía el circuito dado en el enunciado de la tarea 2, el cual se presenta en la Figura 1.

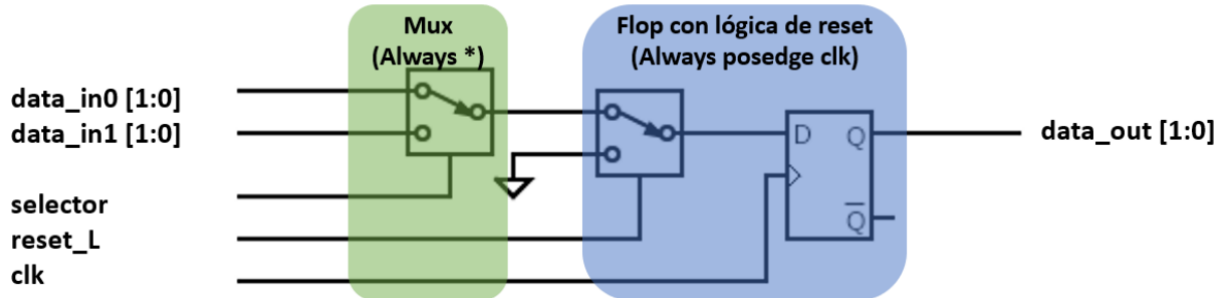


Figura 1: Esquemático del circuito a implementar, con entradas y salidas

Los componentes utilizados para la síntesis con tecnología comercial se presentan 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 máximo (μ A)
BUFFER	SN74LVC3G34	0,29	$t_{pd} = 4,1$	$I_{cc} = 10$
NOT	SN74LV1T04	0,22	$t_{pd} = 4,8$	$I_{cc} = 10$
NAND	SN74AUP2G00-Q1	0,66	$t_{pd} = 2,4$	$I_{cc} = 1,7$
NOR	SN74LVC1G02	0,08	$t_{pd} = 3,6$	$I_{cc} = 10$
FFD	SN74LVC1G80-Q1	0,2	$t_{su} = 1,3$ $t_h = 0,9$ $t_{pd} = 4,2$	$I_{cc} = 10$
FFD SR	SN74HCS74-Q1	0,17	$t_{su} = 9$ $t_h = 0$ $t_{pd} = 8$	$I_{cc} = 10$

Similar a como se mencionó en la tarea 3 y 4, para un diseño de la vida real se deben considerar los tiempos de setup y hold descritos para los flip-flops, pero se considerarán visualmente en las simulaciones, no se implementará en código algún tipo de checker para revisar esta característica dado que no es algo interno del diseño del circuito a realizar.

La frecuencia máxima de operación se obtendrá reduciendo el ancho de pulso del reloj, hasta que se halle la frecuencia a la que la descripción estructural sintetizado produce una salida distinta a la descripción conductual (lo cual se puede ver en salida de GTKWave y sería indicado en la terminal por el checker implementado). En teoría, únicamente la descripción estructural sintetizada con retrasos debería tener una frecuencia máxima, ya que los otros modelos son idealizados sin retrasos de tiempo, por lo que el comportamiento lógico en estos últimos no se ve afectado por la frecuencia del reloj.

3. Plan de pruebas

Se utilizó como base de diseño la simulación presentada en el enunciado de la tarea 2, la cual se puede observar en la Figura 2.

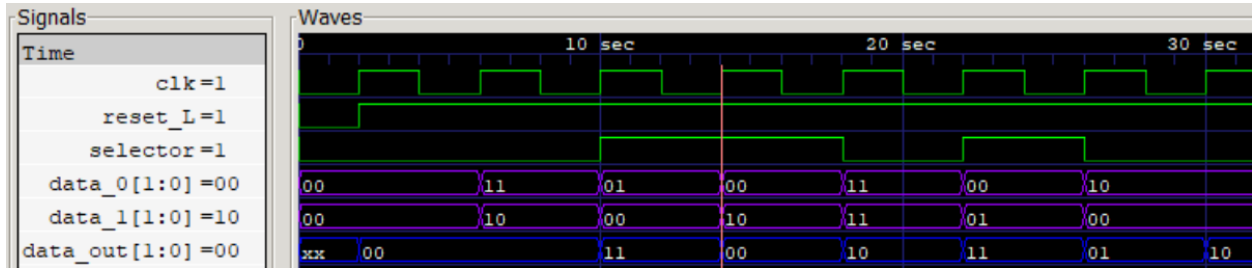


Figura 2: Simulación de prueba de la tarea 2

Se contrastan los resultados obtenidos con cada simulación; RTLIL (genérica con los componentes internos de Yosys), sin retrasos (con la biblioteca de componentes de `cmos_cells.v`) y con retrasos (con modificaciones en la biblioteca de componentes de `cmos_cells.v` para modelar los componentes comerciales escogidos).

4. Instrucciones de simulación

Para realizar las pruebas, se debe estar en el directorio donde se encuentra el Makefile. Si se desean realizar las pruebas de RTLIL, se ejecuta el comando `make rtlil`. Si se desean realizar las pruebas de compuertas comerciales sin retrasos, se ejecuta el comando `make nodelay`. Si se desean realizar las pruebas de compuertas comerciales con retrasos, se ejecuta el comando `make delay`. Si se desea limpiar los archivos y dejar el directorio únicamente con los archivos base (los mínimos para poder ejecutar todo), se ejecuta el comando `make clean`. Si solo se ejecuta `make`, se ejecutan las pruebas de RTLIL de manera predeterminada.

Si se desea cambiar la frecuencia de operación a la que corren las pruebas, se debe ingresar al módulo probador, dentro del archivo en la ruta `src/probador.v`, y modificar el ancho de pulso de reloj que se haya al final del módulo.

5. Resultados

En cada caso, el resultado “teórico” es presentado en la señal `data_out_cond`, la cual corresponde a la salida del módulo conductual. La salida del módulo estructural corresponde a la señal `data_out_estr`. Se espera que la forma de ambas señales sean iguales para un diseño que se considera correcto.

5.1. Síntesis RTLIL

Los resultados de la simulación con síntesis RTLIL se presentan en la Figura 3.

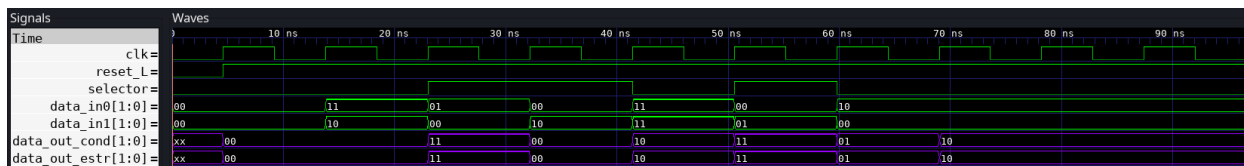


Figura 3: Simulación con síntesis RTLIL

Se verifica que la salida de la descripción estructural corresponde a la señal teórica deseada por la salida del checker en la terminal, como se observa en la Figura 4.

VCD info: dumpfile mux.vcd opened for output.

	time	clk	reset_L	selector	data_in0	data_in1	data_out_estr	data_out_cond	cuenta
	0	0	0	0	00	00	xx	xx	0
	5	1	1	0	00	00	00	00	0
	9	0	1	0	00	00	00	00	0
Todo bien	14	1	1	0	11	10	00	00	0
	18	0	1	0	11	10	00	00	0
Todo bien	23	1	1	1	01	00	11	11	2
	28	0	1	1	01	00	11	11	2

Figura 4: Salidas del checker en la terminal

El mensaje “todo bien” indica que no hay ninguna inconsistencia entre las salidas de la descripción conductual y la descripción estructural.

5.2. Síntesis sin retrasos

Al realizar una síntesis con mapeo a una biblioteca con tecnologías comerciales (en este caso, utilizando cmos_cells.lib y cmos_cells.v), Yosys imprime en terminal la cantidad de componentes utilizados, como se presenta en la Figura 5.

```
12.1.2. Re-integrating ABC results.
ABC RESULTS:      NAND cells:      4
ABC RESULTS:      NOR cells:       4
ABC RESULTS:      NOT cells:       2
ABC RESULTS:      internal signals: 3
ABC RESULTS:      input signals:    6
ABC RESULTS:      output signals:   2
```

Figura 5: Cantidad de componentes utilizados en la síntesis sin retrasos

Utilizando esta información, con los costos unitarios descritos en el Cuadro 1, se puede estimar que el costo de fabricación de uno de los dispositivos diseñados sería de \$3,4. Sin embargo, los costos individuales de cada componente usualmente es menor si se compran en lotes (aunque generalmente las reducciones de precio individual inician en la compra de un mínimo de 100 unidades), además de que varios de los componentes son de múltiples canales (es decir, son varias compuertas en un único componente), lo cual puede ser de interés para algunos desarrolladores. Considerando esto lo anterior, se podría poner un precio de venta entre \$5 y \$7, según lo el plan de negocios que se tenga.

Los resultados de la simulación realizada con esta síntesis se presentan en la Figura 6.

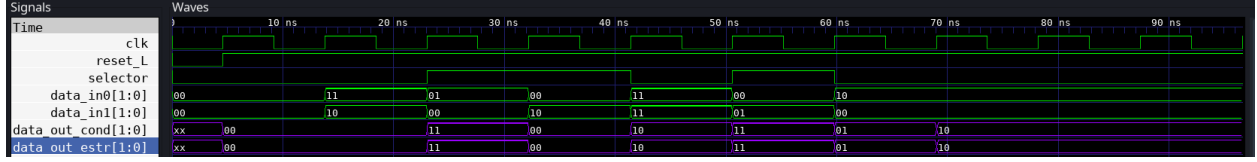


Figura 6: Resultados de la simulación con una síntesis sin retrasos

Aquí se observa que la salida de la descripción estructural es prácticamente idéntica a la obtenida de la Figura 3. Esto se debe a que, si bien se utilizaron componentes externos, estos siguen siendo modelos ideales sin retrasos y, por ende, su comportamiento lógico y temporal es igual al que se obtiene con el RTLIL. Considerando esto, se tendría que ese dispositivo tiene frecuencia máxima infinita, lo cual físicamente no es realizable.

5.3. Síntesis con retrasos

De igual manera que con la síntesis sin retrasos, Yosys imprime en terminal la cantidad de componentes utilizadas en la descripción estructural. Esto se muestra en la Figura 7.

```
12.1.2. Re-integrating ABC results.
ABC RESULTS:          NAND cells:      4
ABC RESULTS:          NOR cells:       4
ABC RESULTS:          NOT cells:        2
ABC RESULTS:          internal signals:  3
ABC RESULTS:          input signals:     6
ABC RESULTS:          output signals:    2
Removing temp directory.
Removed 0 unused cells and 8 unused wires.
```

Figura 7: Cantidad de componentes utilizados en la síntesis con retrasos

Se nota que los números son iguales a los de la Figura 5, lo cual es de esperarse ya que el comportamiento lógico es idéntico, lo que cambia es la respuesta en el tiempo. Como los componentes utilizados no difiere de los que se tienen de la Figura 5, el análisis del precio en mercado sería igual en este caso.

Mediante simulaciones, se determinó que la frecuencia máxima de operación f_{max} para este modelo más cercano a la realidad sería $f_{max} \approx 108,7 \text{ MHz}$, dado que se utilizó un ancho de pulso de reloj mínimo de 4,6 ns (y un ancho de pulso de reloj es medio periodo).

El resultado de la simulación con retrasos, con un pulso de reloj de 4,6 ns, se presenta en la Figura 8.

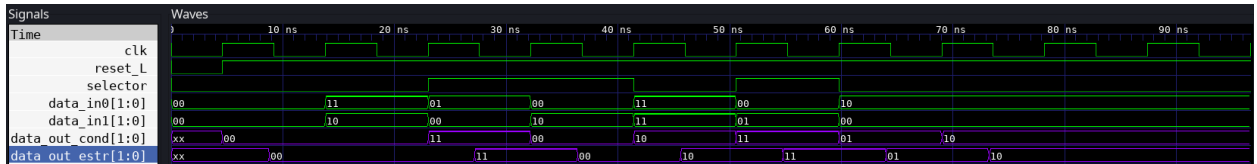


Figura 8: Simulación con síntesis con retrasos

Se puede ver que el comportamiento lógico de la descripción estructural es igual al teórico, solo que se ve desplazado en el tiempo (lo cual es de esperarse por los retrasos). Se verifica con el checker que el comportamiento lógico que la salida no tiene problemas, como se observa en la Figura 9.

VCD info: dumpfile mux.vcd opened for output.

	time	clk	reset_L	selector	data_in0	data_in1	data_out_estr	data_out_cond	cuenta
	0	0	0	0	00	00	xx	xx	0
	5	1	1	0	00	00	xx	00	0
	9	1	1	0	00	00	00	00	0
	9	0	1	0	00	00	00	00	0
Todo bien									
	14	1	1	0	11	10	00	00	0
	18	0	1	0	11	10	00	00	0
Todo bien									
	23	1	1	1	01	00	00	11	0
	27	1	1	1	01	00	11	11	2
	28	0	1	1	01	00	11	11	2
Todo bien									

Figura 9: Confirmación del checker de que no se hayan inconsistencias en las señales

También se debe notar que los datos en la salida del circuito cambian cada periodo, lo cual permite que se cumpla el tiempo de setup del FFD y se podría implementar el circuito en la vida real sin problemas. Cuando se reduce el ancho del pulso de reloj a 4,5 ns, se producen pulgas y la salida deja de asemejarse al comportamiento deseado. Esto se observa en la Figura 10, donde en el 4to flanco positivo de reloj la salida de la descripción estructural no es lógicamente equivalente a la teórica.

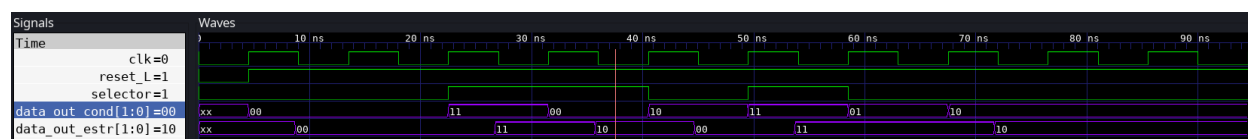


Figura 10: Simulación con síntesis con retrasos y comportamiento lógico erróneo

Esto se verifica con la salida del checker, como se ve en la Figura 11.

VCD info: dumpfile mux.vcd opened for output.

	time	clk	reset_L	selector	data_in0	data_in1	data_out_estr	data_out_cond	cuenta
	0	0	0	0	00	00	xx	xx	0
	5	1	1	0	00	00	xx	00	0
	9	1	1	0	00	00	00	00	0
	9	0	1	0	00	00	00	00	0
Todo bien									
	14	1	1	0	11	10	00	00	0
	18	0	1	0	11	10	00	00	0
Todo bien									
	23	1	1	1	01	00	00	11	0
	27	1	1	1	01	00	11	11	2
	28	0	1	1	01	00	11	11	2
Todo bien									
	32	1	1	1	01	00	11	11	2
	36	1	1	1	01	00	11	11	2
	36	0	1	1	01	00	11	11	2
Inconsistencia a los									

Figura 11: Errores marcados por el checker en terminal

5.4. Uso de sed

Para el uso de sed simplemente se utilizó el comando:

```
sed -i 's/mux/mux_synth/g' `$(SYNTH)`
```

Donde SYNTH guarda el path al archivo de síntesis generado por el script de yosys. La bandera -i le indica a sed que se guarden los cambios en el mismo archivo.

6. Análisis y Conclusiones

En general se observó que el proceso de síntesis con Yosys facilita bastante el proceso de diseño. Uno no se debe enfocar tanto en la implementación de los dispositivos básicos que conforman el circuito a diseñar y el tiempo que se invertiría en esta tarea se puede dirigir a un mejor análisis de diseño en cuanto a pruebas y precios, lo cual llega a ser muy significativo en el proceso de fabricación en la industria.

Los resultados obtenidos están de acuerdo a lo que se espera con la implementación estructural de un dispositivo, además de que la optimización automatizada produce mejores resultados en términos de rendimiento (la frecuencia máxima alcanzada con este esquema de diseño es considerablemente mayor a la obtenida en la tarea anterior por casi el doble (con este diseño se obtiene una frecuencia máxima de 108,7 MHz, mientras que en la tarea anterior se obtuvo una frecuencia máxima experimental de 55,56 MHz). Si bien esta diferencia se debe a las compuertas utilizadas, se puede decir que la automatización de la síntesis por Yosys permitió invertir más tiempo en buscar componentes más eficientes y con mejores tiempos de propagación, lo cual resulta en un mejor producto.