

Universidad de Costa Rica
 Escuela de Ingeniería Eléctrica
 Circuitos Digitales II
 Prof. Jorge Soto
 II Ciclo 2020
 IE-0523

Tarea #3

1. Diagrama

Primeramente se presenta toda la información acerca cada compuerta de la biblioteca.

AND: Se buscó el modelo SN74LVC1G08IDCKRQ1, el cual posee un costo de \$0.172 por unidad. En la figura 1 se muestran las características de temporización del componente, donde se selecciona una tensión de 1.8V, con temperatura que va desde -40 hasta 80 grados. En la figura 2 se observa el consumo de potencia.

6.6 Switching Characteristics, $T_A = -40^\circ\text{C}$ to 125°C

over recommended operating free-air temperature range (unless otherwise noted) (see Figure 3)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	-40°C to 125°C								UNIT
			V _{CC} = 1.8 V ± 0.15 V		V _{CC} = 2.5 V ± 0.2 V		V _{CC} = 3.3 V ± 0.3 V		V _{CC} = 5 V ± 0.5 V		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t _{pd}	A or B	Y	1	10	1	7.5	1	6.5	1	6	ns

Figura 1: Características de temporización de la AND

6.8 Operating Characteristics

$T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	$V_{CC} = 1.8\text{ V}$	$V_{CC} = 2.5\text{ V}$	$V_{CC} = 3.3\text{ V}$	$V_{CC} = 5\text{ V}$	UNIT
		TYP	TYP	TYP	TYP	
C_{pd} Power dissipation capacitance	$f = 10\text{ MHz}$	21	24	26	31	pF

Figura 2: Consumo de potencia de la AND

OR: Se buscó el modelo SN74LVC1G32-EP, el cual posee un costo de \$1.693 por unidad. En la figura 3 se muestran las características de temporización del componente, donde se selecciona una tensión de 1.8V, con $C_L = 30\text{ pF}$ o $C_l = 50\text{ pF}$. En la figura 4 se observa el consumo de potencia.

Switching Characteristics

over recommended operating free-air temperature range, $C_L = 30$ pF or 50 pF (unless otherwise noted) (see [Figure 2](#))

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CC} = 1.8\text{ V}$ $\pm 0.15\text{ V}$		$V_{CC} = 2.5\text{ V}$ $\pm 0.2\text{ V}$		$V_{CC} = 3.3\text{ V}$ $\pm 0.3\text{ V}$		$V_{CC} = 5\text{ V}$ $\pm 0.5\text{ V}$		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t_{pd}	A or B	Y	2.8	8	1.2	6.5	1.1	5.7	1	4.5	ns

Figura 3: Características de temporización de la OR

6.8 Operating Characteristics

$T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	$V_{CC} = 1.8\text{ V}$	$V_{CC} = 2.5\text{ V}$	$V_{CC} = 3.3\text{ V}$	$V_{CC} = 5\text{ V}$	UNIT
		TYP	TYP	TYP	TYP	
C_{pd} Power dissipation capacitance	$f = 10\text{ MHz}$	21	24	26	31	pF

Figura 4: Consumo de potencia de la OR

Inversor: Se buscó el modelo SN74LVC1G04, el cual posee un costo de \$0.063 por unidad. En la figura 5 se muestran las características de temporización del componente, donde se selecciona una tensión de 1.8V, con con temperatura que va desde -40 hasta 85 grados y $C_L = 30\text{pF}$ o $C_L = 50\text{pF}$, adaptandose a las 2 compuertas anteriores. En la figura 6 se observa el consumo de potencia.

7.7 Switching Characteristics, $C_L = 30$ pF or 50 pF, -40°C to 85°C

over recommended operating free-air temperature range, $C_L = 30$ pF or 50 pF (unless otherwise noted) (see [Figure 4](#))

PARAMETER	FROM (INPUT)	TO (OUTPUT)	-40°C to 85°C								UNIT
			V _{CC} = 1.8 V ± 0.15 V		V _{CC} = 2.5 V ± 0.2 V		V _{CC} = 3.3 V ± 0.3 V		V _{CC} = 5 V ± 0.5 V		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
t _{pd}	A	Y	3	7.5	1.4	5.2	1	4.2	1	3.7	ns

Figura 5: Características de temporización del inversor

7.10 Operating Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	$V_{CC} = 1.8\text{ V}$	$V_{CC} = 2.5\text{ V}$	$V_{CC} = 3.3\text{ V}$	$V_{CC} = 5.0\text{ V}$	UNIT
		TYP	TYP	TYP	TYP	
C_{pd} Power dissipation capacitance	$f = 10\text{ MHz}$	16	18	18	20	pF

Figura 6: Consumo de potencia del inversor

Flip Floop: Se buscó el modelo SN74AUP2G79, el cual posee un costo de \$0.273 por unidad. En la figura 7 se muestran las características de temporización del componente, donde se selecciona una tensión de 1.8V, con $C_L = 30\text{pF}$, adaptandose a las 3 compuertas anteriores. En la figura 8 se observa el consumo de potencia.

SWITCHING CHARACTERISTICS

over recommended operating free-air temperature range, $C_L = 30 \text{ pF}$ (unless otherwise noted) (see [Figure 3](#) and [Figure 4](#))

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V_{CC}	$T_A = 25^\circ\text{C}$			$T_A = -40^\circ\text{C}$ to 85°C		UNIT
				MIN	TYP	MAX	MIN	MAX	
f_{\max}			0.8 V		24		20		MHz
			$1.2 \text{ V} \pm 0.1 \text{ V}$		72		80		
			$1.5 \text{ V} \pm 0.1 \text{ V}$		100		100		
			$1.8 \text{ V} \pm 0.15 \text{ V}$		127		140		
			$2.5 \text{ V} \pm 0.2 \text{ V}$		185		210		
			$3.3 \text{ V} \pm 0.3 \text{ V}$		266		260		
t_{pd}	CLK	Q	0.8 V		27.2				ns
			$1.2 \text{ V} \pm 0.1 \text{ V}$	7	11.5	17.3	5.9	24	
			$1.5 \text{ V} \pm 0.1 \text{ V}$	5.7	8.3	13.3	4.6	15.9	
			$1.8 \text{ V} \pm 0.15 \text{ V}$	4.7	6.7	11.3	3.8	13	
			$2.5 \text{ V} \pm 0.2 \text{ V}$	3.7	4.9	7.8	2.9	9	
			$3.3 \text{ V} \pm 0.3 \text{ V}$	3.2	4.1	6.3	2.6	7.2	

Figura 7: Características de temporización del flip flop

OPERATING CHARACTERISTICS

$T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	V_{CC}	TYP	UNIT
C_{pd} Power dissipation capacitance	$f = 10 \text{ MHz}$	0.8 V	2.5	pF
		$1.2 \text{ V} \pm 0.1 \text{ V}$	2.5	
		$1.5 \text{ V} \pm 0.1 \text{ V}$	2.5	
		$1.8 \text{ V} \pm 0.15 \text{ V}$	2.5	
		$2.5 \text{ V} \pm 0.2 \text{ V}$	3	
		$3.3 \text{ V} \pm 0.3 \text{ V}$	3	

Figura 8: Consumo de potencia del flip flop

En cuanto al mux 2:1, este se armó con las compuertas mencionadas anteriormente que son parte de la biblioteca solicitada (AND, OR e inversor). En la figura 9 se muestra el diagrama del mux 2:1 de 1 bit. Para la implementación del mux 2:1 de 2 bits se utilizan 2 mux 2:1 de 1 bit, como se ve en la figura 10.

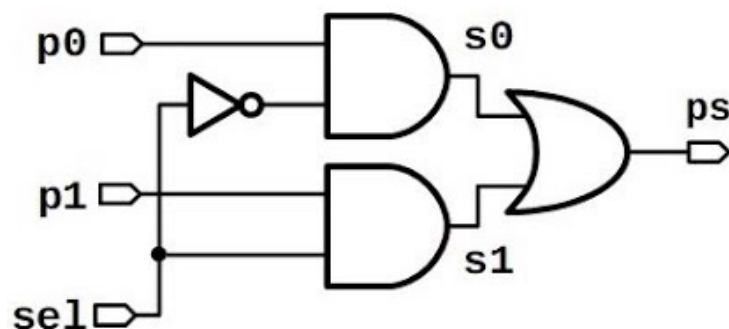


Figura 9: Diagrama mux 2:1 de 1 bit

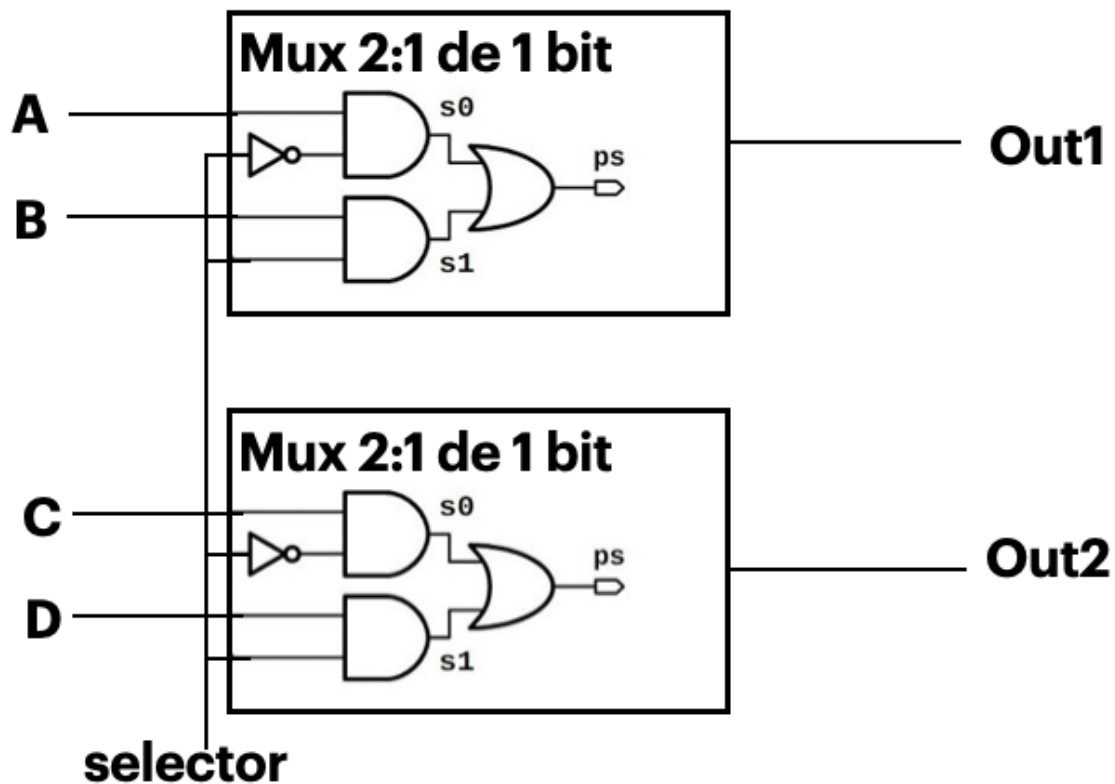


Figura 10: Diagrama mux 2:1 de 2 bits

Cabe mencionar que debido a que las hojas de fabricante no dan un valor típico, se saca la media entre el mínimo y el máximo para obtener este. En la figura 11 se muestra el diagrama del multiplexor estructural utilizado anteriormente en la tarea 2. El cual posee 4 entradas, las cuales son `data_in0`, `data_in1`, `reset_L` y `clk`, además de una salida `data_out`. EL diagrama utiliza 2 multiplexores de 2:1 con dos entradas y una salida de datos de 2 bits y un flip-flop tipo D.

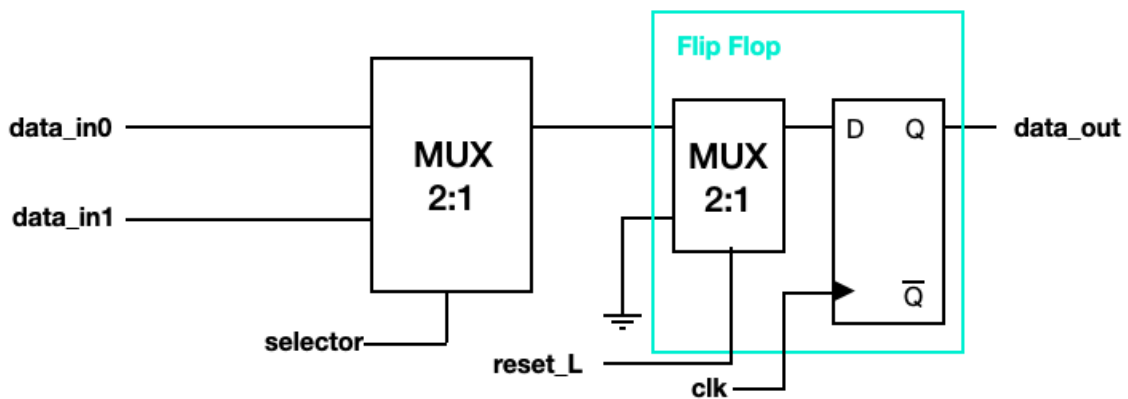


Figura 11: Diagrama mux estructural

2. Plan de Pruebas

Para realizar las pruebas se realizaron diversos probadores y bancos de prueba. Primeramente con respecto a la biblioteca lógica, esta se dividió en 2, en los componentes con 1 bit de entrada como AND, OR, inversor y el primer mux 2:1. La segunda división corresponden a los componentes con 2 bits de entrada los cuales corresponden al flip flop y el segundo mux. Cabe mencionar que mediante autoinst se realizaron las conexiones de manera automática en cada banco de pruebas. Entonces se realizó un probador con su respectivo banco de pruebas para cada sección de 1 bit de entrada o 2 bits de entrada, esto de modo que se pueda comprobar mediante el uso de gtkwave el óptimo funcionamiento de cada compuerta, por medio de las gráficas. Además de visualizar que la memoria se actualice con respecto al flanco creciente de la señal clk y comprobar los tiempos de delay que posee cada compuerta.

Posteriormente en la segunda parte de la tarea, dado que se solicita crear una descripción estructural del mux creado en la tarea 2, se modifica el probador de modo que cuente con la descripción estructural además de la descripción conductual que se tenía. Al probador se le agrega un contador conductual y estructural, además de un checker que funciona como un comparador entre ambos modelos. Las pruebas que se le hacen corresponden a las mismas de la vez pasada, como que si la señal de reset no es cero, depende de la señal del selector. En caso de que la señal del selector sea de 0, data_out=data_in0 y si selector es 1, data_out=data_in1. En caso de ser el reset 0 se pone a la salida 0. Además que se busca comprobar que todos los cambios se realicen únicamente cuando haya un flanco creciente en la señal del clock(reloj). Para cumplir con el objetivo D de la segunda parte, se aumenta la frecuencias, esto disminuyendo el periodo en que hace toggle, hasta que se llega al punto donde ambas descripciones son distintas, esto mediante el comando de iverilog que se muestra en la terminal con el checker en caso de que ambas descripciones sean distintas.

3. Instrucciones de utilización de la simulación

Para la automatización se crea el siguiente makefile:

```
1
2 iverilogt2:
3     iverilog -o tarea2 BancoPruebaTarea2.v
4     vvp tarea2
5 gtkwavet2:
6     gtkwave muxTarea2.vcd
7
8 tarea2: iverilogt2 gtkwavet2
9
10 iverilog1bit:
11     iverilog -o 1bit BancoPrueba1bit_tarea34.v
12     vvp 1bit
13 gtkwave1bit:
14     gtkwave probador1bit.vcd
15
16 1bit: iverilog1bit gtkwave1bit
17
18 iverilog2bit:
```

```
19         iverilog -o 2bit BancoPrueba2bit_tarea34.v
20         vvp 2bit
21 gtkwave2bit:
22         gtkwave probador2bit.vcd
23
24 2bit: iverilog2bit gtkwave2bit
```

Para utilizar el makefile primeramente hay que dirigirse en la terminal a la carpeta que contiene estos archivos. Para cada una de las 3 simulaciones correspondientes se crearon 3 archivos, uno de iverilog el cual va a crear el archivo .vcd correspondiente, el de gtkwave para que se abra el programa y poder ver las gráficas y el que une las 2 funciones. Para el caso de probar los componentes de la biblioteca de 1 bit de entrada se introduce el comando **make 1bit**, en caso de solo querer ver el comando de Icarus Verilog se introduce el comando **make iverilog1bit** y en caso de solo querer utilizar el comando de gtkwave se introduce **make gtkwave1bit**. Para el caso de probar los componentes de la biblioteca de 2 bits de entrada se introduce el comando **make 2bit**, en caso de solo querer ver el comando de Icarus Verilog se introduce el comando de **make iverilog2bit** y en caso de solo querer utilizar el comando de gtkwave se introduce **make gtkwave2bit**. Finalmente para ver el mux de la tarea 2 con las nuevas implementaciones se introduce el comando **make tarea2**, en caso de solo querer ver el comando de Icarus Verilog se introduce el comando de **make iverilogt2** y en caso de solo querer utilizar el comando de gtkwave se introduce **make gtkwavet2**.

4. Ejemplos de los resultados:

Como medio de comprobación de resultados, se muestran las diversas gráficas de la biblioteca con 1 bit, con 2 bits y además el mux de la tarea 2 con los implementos de esta tarea. Además que se destacan los tiempos de delay y se mencionan las razones del por qué la salida presenta dicho comportamiento.

Primeramente están los elementos de 1 bit de entrada de la biblioteca. Cabe mencionar que el periodo en este caso corresponde a 10ns. La compuerta AND depende de las señales de A y B, estas cambian cada flanco creciente del reloj. De la compuerta AND se sabe que su señal es de 1 únicamente cuando las 2 señales son 1, por ende en la figura 12 se muestra que cuando las señales A y B son 1, la señal de la AND es de 1, en los demás casos es de 0. Además que se muestra un tiempo de delay de 5ns, el cual corresponde al atraso típico. De la compuerta OR se sabe que esta únicamente posee salida 0 si las dos entradas son 0. En la figura 13 se muestra este comportamiento esperado, además que se observa un tiempo de delay de 5.4 ns el cual es el valor esperado dado que es el tiempo de delay típico. En cuanto a la compuerta del inversor, esta tiene relación únicamente con la entrada A, por ende esta señal debe ser el opuesto de la entrada. Tal como se ve en la figura 14, además se muestra el tiempo de delay de 5.3ns el cual es aproximadamente igual a su valor típico de 5.25ns. En cuanto al mux de 2:1 de 1 bit se tiene que si el selector esta en 0 este debe de tener el valor de A, en caso contrario de que el selector de 1 a la salida tiene el valor de B. Este comportamiento se comprueba en la figura 15, donde además se observa un tiempo de delay de 10.04ns esto es de esperar dado que los componentes con los que se fabricó poseen tiempos de delay.

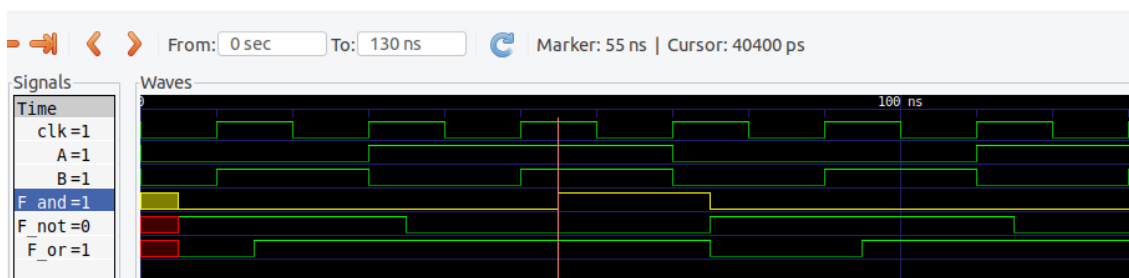


Figura 12: Ejemplo compuerta AND

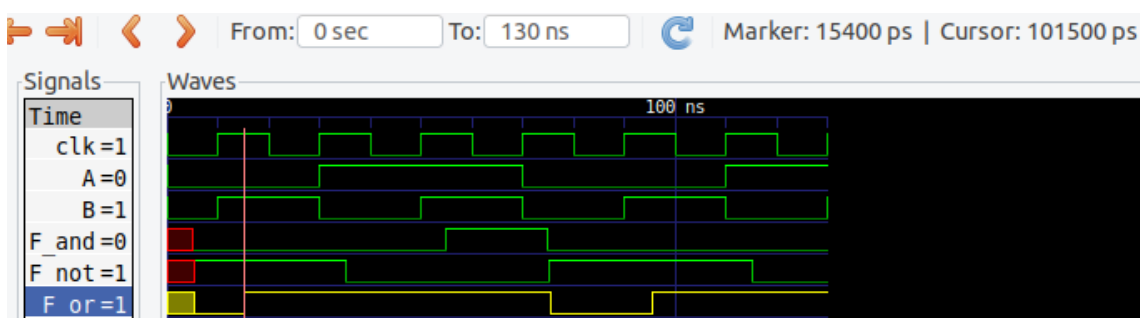


Figura 13: Ejemplo compuerta OR

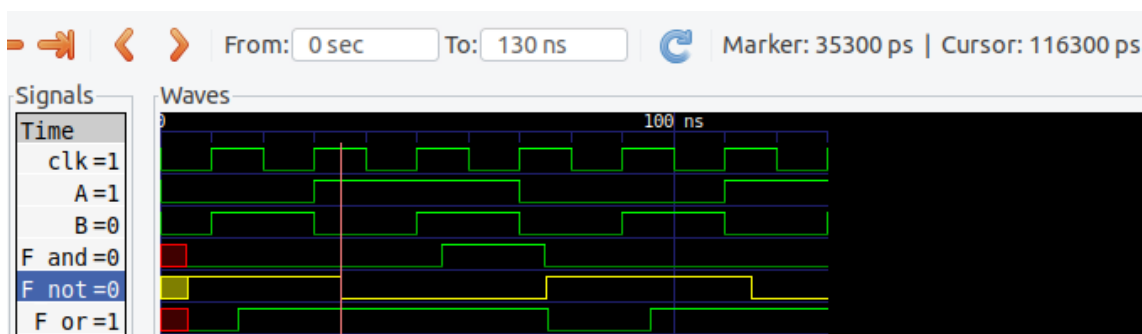


Figura 14: Ejemplo compuerta inversor

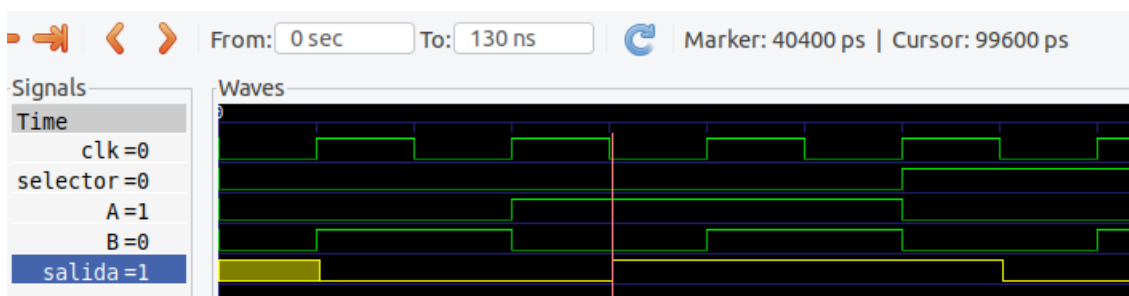


Figura 15: Ejemplo mux 2:1 de 1 bit

Posteriormente están los elementos de 2 bit de entrada de la biblioteca. Cabe mencionar que el periodo en este caso también corresponde a 10ns. En el floop se tiene que cada flanco

creciente del reloj, su salida debe tener el valor anterior de la señal de A, además de un retraso de 6.7, tal como se ve en la figura 16. En cuanto al mux 2:1 de 2 bits se espera que si el selector está en 0, cada flanco creciente tenga el valor de la salida del mux debe ser el anterior al de la señal de A, y en caso de que el selector esté en 1, la señal de salida sea la anterior de B, tal como se observa en la figura 17.

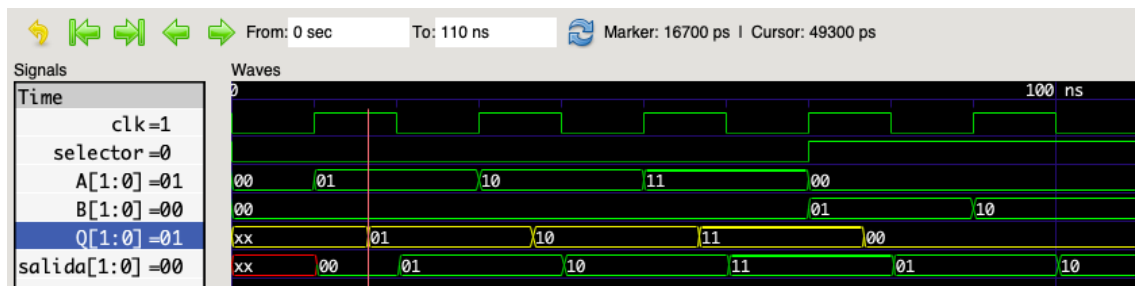


Figura 16: Ejemplo flip flop tipo d

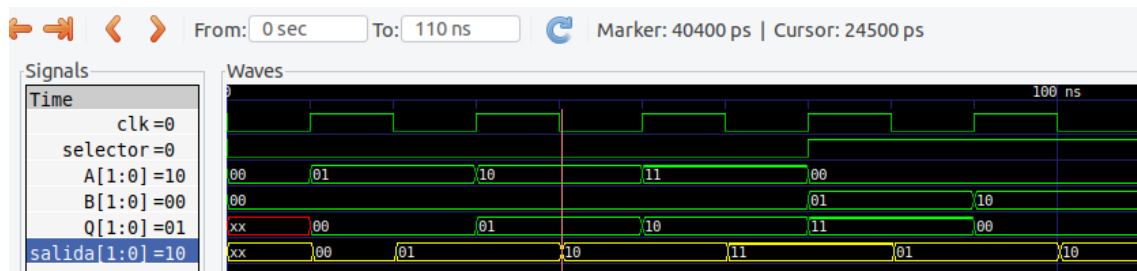


Figura 17: Ejemplo mux 2:1 de 2 bit

Por último está el ejemplo del mux de la tarea 2 con su diseño estructural. Cabe mencionar que el periodo en este caso corresponde a 15ns. Se espera que los resultados sean similares a los de la tarea 2, ya que a esa frecuencia de $1/15\text{ns}$ se supone que el diseño estructural y conductual se comporten de la misma manera y se están utilizando las mismas señales correspondientes. Tal como se ve en la figura 18. Se disminuye el valor del período hasta sobrepasar la frecuencia máxima que corresponde a $1/15\text{ns}$, por ende disminuyendo el periodo a 13ns se tiene que el diseño estructural y conductual varían como se ve en la figura 19. Además en la figura 20 se ve mediante el comando de **iverilog** donde están cambiando ambos diseño debido al checker.

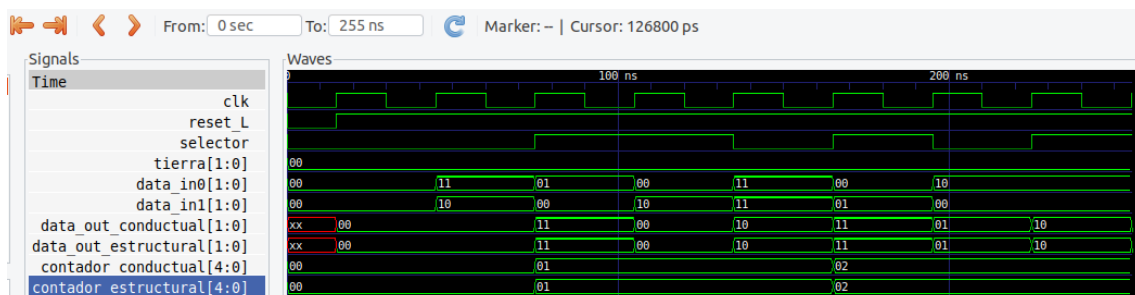


Figura 18: Ejemplo mux tarea 2, con frecuencia por debajo de la máxima

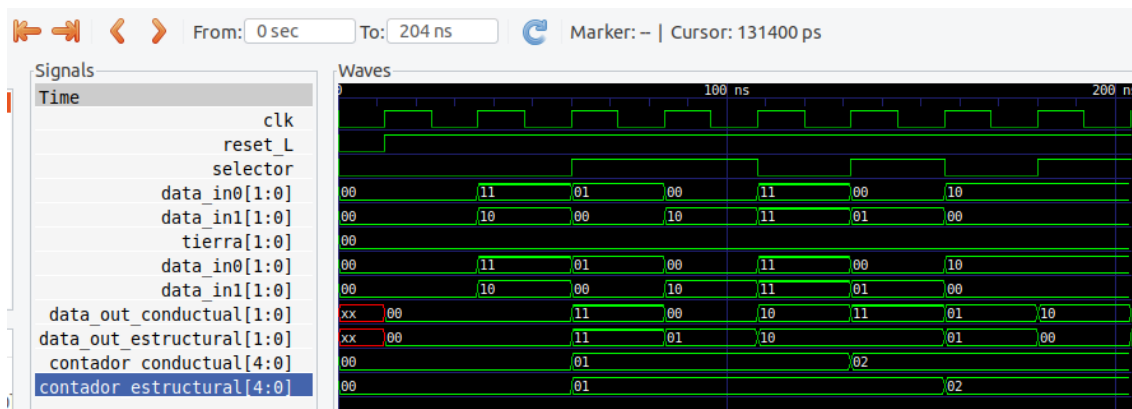


Figura 19: Ejemplo mux tarea 2, con frecuencia por encima la máxima

```

alberto@alberto-VirtualBox: ~/Descargas/Digitales2/Tarea3
[238000] end time.
WM Destroy
alberto@alberto-VirtualBox:~/Descargas/Digitales2/Tarea3$ make tarea2
iverilog -o tarea2 BancoPruebaTarea2.v
./AND_tarea34.v:7: warning: choosing typ expression.
./AND_tarea34.v:7: warning: choosing typ expression.
./OR_tarea34.v:7: warning: choosing typ expression.
./OR_tarea34.v:7: warning: choosing typ expression.
./NOT_tarea34.v:6: warning: choosing typ expression.
./NOT_tarea34.v:6: warning: choosing typ expression.
vvp tarea2
VCD info: dumpfile muxTarea2.vcd opened for output.
      clk,      out,      reset,      selector,      in0,      in1
      0      0      xx      00      0      0      000
      13      1      00      00      0      1      000
      26      0      00      00      0      1      000
      39      1      00      00      0      1      112
      52      0      00      00      0      1      112
      65      1      11      11      1      1      010
      78      0      11      11      1      1      010
      91      1      00      01      1      1      002
     104      0      00      01      1      1      002
Diferencia modulo conductual y estructural
     117      1      10      10      0      1      113
     130      0      10      10      0      1      113
     143      1      11      10      1      1      001
     156      0      11      10      1      1      001
Diferencia modulo conductual y estructural
     169      1      01      01      0      1      100
     182      0      01      01      0      1      100
     195      1      10      00      1      1      100
     208      0      10      00      1      1      100
Diferencia modulo conductual y estructural
     221      1      00      10      1      1      100
gtkwave muxTarea2.vcd
Gtk-Message: 19:42:40.979: Failed to load module "canberra-gtk-module"

```

Figura 20: Ejemplo checker

5. Análisis y conclusiones

Analizando los ejemplos de la sección anterior, se puede notar que los distintos diseños cumplen con lo solicitado. Ya que como se ve en el ejemplo 1, para cada compuerta de entrada

de 1 bit se obtiene la simulación esperada, con cambios inmediatos retrasados únicamente por el tiempo del delay. En cuanto al segundo ejemplo de compuerta de entrada de 2 bits se obtienen los resultados esperados igualmente, con la señal anterior en la salida cada vez que ocurre un flanco creciente. Con respecto a la frecuencia máxima esta teóricamente se saca utilizando la ecuación (1), la cual corresponde a sumar los tiempos de delay de los componentes de los cuales se conforma el mux 2:1 de 1 bit, como se ve en la figura 9. Cabe mencionar que para esta parte no se consideró el retraso del flip flop debido a que no intervenía en toda la lógica anterior, simplemente en la salida. En la simulación con un periodo de 26,6ns el modelo conductual y estructural seguían siendo iguales por lo que la frecuencia máxima experimental está por encima de 37.59Mhz, el cual infiere que el valor teórico pero se puede considerar aproximadamente igual en comparación al nivel práctico.

$$T_{minimo} = T_{pd_{AND}} + T_{pd_{OR}} + T_{pd_{NOT}} + T_{pd_{AND}} + T_{pd_{OR}} \quad (1)$$

$$T_{min} = 5 + 5,4 + 5,25 + 5 + 5,4 = 26,05ns \quad (2)$$

$$F_{max} = \frac{1}{T_{min}} = 38,39Mhz \quad (3)$$

Se puede concluir que la simulación cumple con lo esperado. Se obtuvo una biblioteca con compuertas con entrada de 1 bit y 2 bits, con sus respectivos probadores y bancos de prueba. Se utilizó esa biblioteca para el desarrollo del modulo estructural, donde se le agregó un contador de flancos positivos y un checker para ver si los modelos conductuales o estructurales cambiaban en algún momento. Cabe mencionar que se reutilizó código de la tarea 2, por ende cada archivo perteneciente a este posee el nombre al final de tarea2 o t2, a pesar de que esté modificado. Además se puede destacar la importancia de manejar de forma correcta los tiempos de delay ya que a nivel del curso nos garantizamos que está correcto el resultado, y a nivel laboral se va creando una idea de que en la vida real los cambios no suceden de inmediato, sino que debido a propiedades de los semiconductores estos tienden a atrasar los tiempos.

6. Distribución del tiempo invertido en la tarea

En total se duró 12 horas con 50 minutos realizando la tarea

- Buscar información: 2 horas debido a que se buscó información general y específica acerca verilog, además de las hojas de fabricante.
- Estudiando la información tomó 3 horas entender las distintas opciones que verilog ofrece
- Ejecutando lo que decidió hacer y probándolo se tomó 5 horas debido a que me costó bastante saber como acomodar y unir cada elemento.
- Realizar el reporte tomó 2 horas con 50 minutos.