

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

Tarea #6

1. Diagrama

En la figura 1 se muestra el diagrama del circuito Mux 2:1 (2 bits) con bit de valid. El cual es similar al que se a usado anteriormente, sin embargo este posee las señales de valid. Este circuito posee 6 entradas, las cuales son data_in0, data_in1, reset_L, clk, valida_in0 y valida_in1, 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. En la figura 2 se muestra el diagrama del circuito Mux 2:1 (4 bits) con bit de valid, donde se utilizan 2 circuitos Mux 2:1 (2 bits) de la figura 1. En la figura 3 se muestra el diagrama del circuito Mux 4:1 (4 bits) con bit de valid, donde se utilizan 3 circuitos Mux 2:1 (4 bits) de la figura 2.

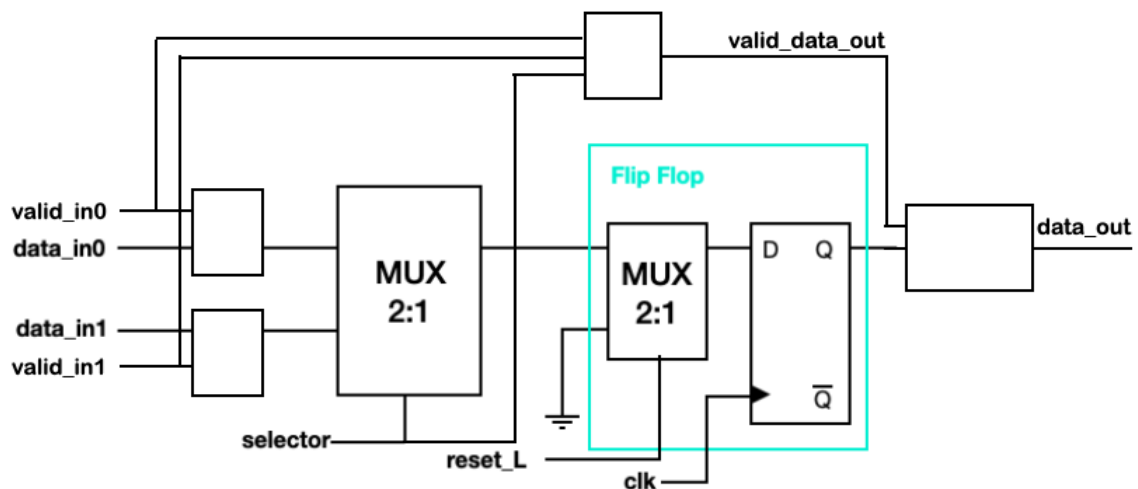


Figura 1: Diagrama Mux 2:1 (2 bits) con bit de valid

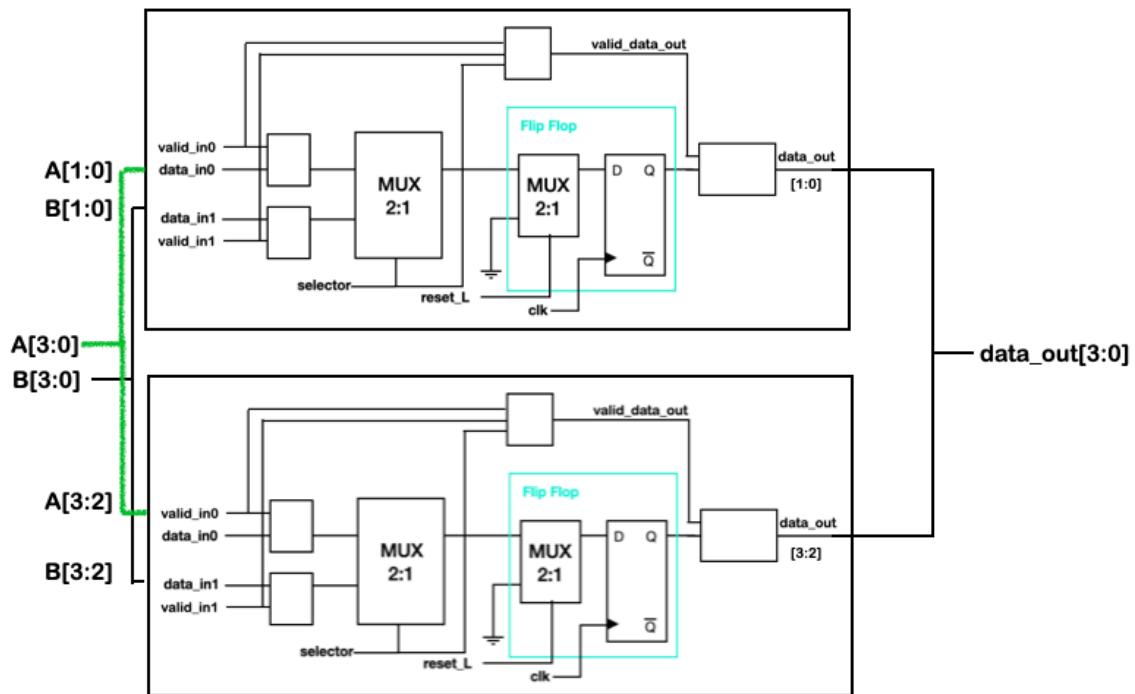


Figura 2: Diagrama Mux 2:1 (4 bits) con bit de valid

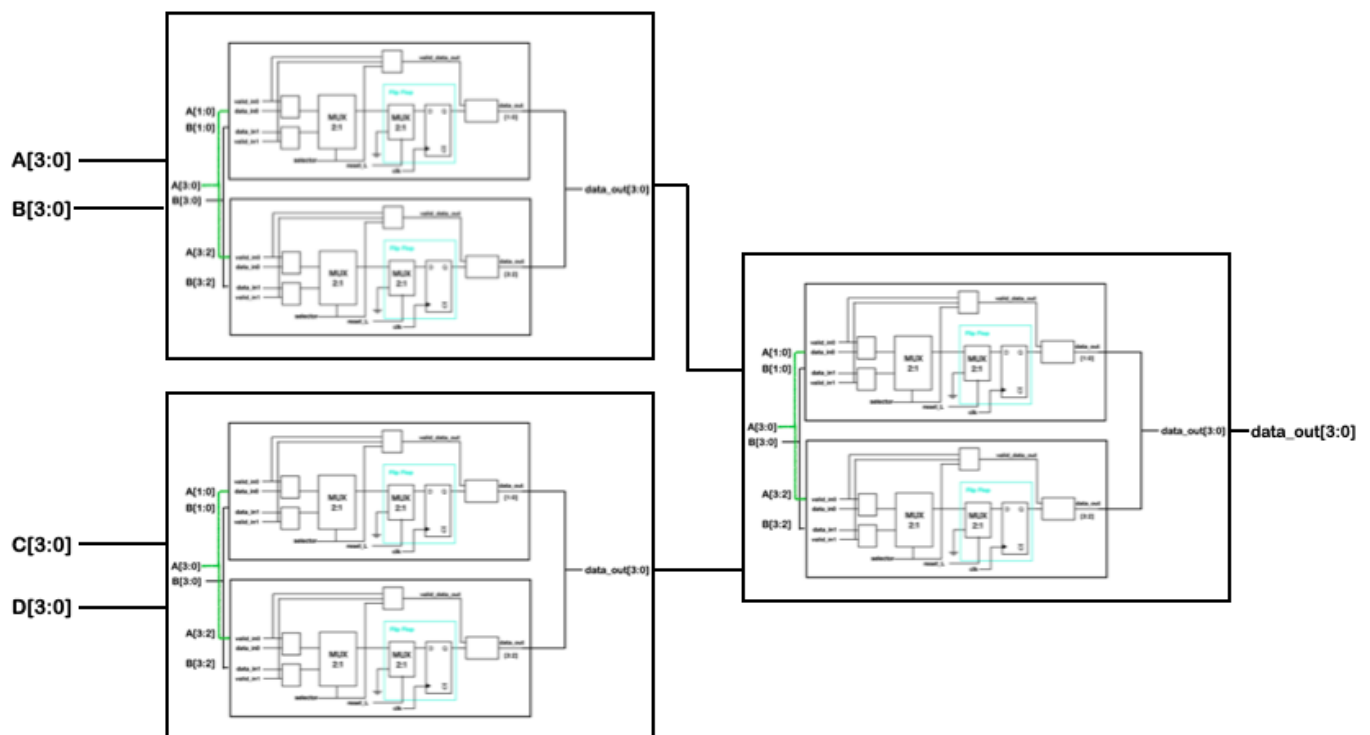


Figura 3: Diagrama Mux 4:1 (4 bits) con bit de valid

2. Plan de Pruebas

Se crean los archivos de descripción conductual, probador y banco de pruebas para cada circuito solicitado. Para el circuito Mux 2:1 (2 bits) se realiza de manera similar a las veces anteriores donde se hace la descripción conductual de manera lógica de modo 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. A diferencia de los casos anteriores se añaden las señales de `valid_in` a la entrada, y una señal de salida `valid_out`. En caso de que selector sea 0 y `valid_in0` sea 1 o cuando selector sea 1 y `valid_in1` sea 1, la presentará en `data_out` el valor correspondido, en caso contrario se mostrará el valor anterior. En el caso del probador se añadieron manualmente todas las señales correspondidas como las de `reset_L`, `selector`, `data_in0`, `data_in1` y los `valid_in`. Cabe mencionar que debido a que para formar los siguientes circuitos como Mux 2:1 (4 bits) y Mux 4:1 (4 bits) se hacen a partir del circuito Mux 2:1 (2 bits), lo único que se modifica es el tamaño de estos como por ejemplo en el Mux 2:1 de 4 bits que en vez de poseer 2 bits, posee 4 a la entrada, o se aumentan como en el caso del Mux 4:1 que se cuentan con 4 entradas (A;B;C;D) en vez de 2 (A;B) como en los casos anteriores. Por último se instancian los archivos de conductual y probador al archivo de banco de prueba. Mediante `autoinst` se realizaron las conexiones de manera automática en el banco de prueba. Para comprobar que la salida dé lo esperado, se utiliza el programa `gtkwave` que muestra estas señales de manera gráfica. Además para cada descripción estructural se realiza una síntesis para generar la descripción estructural, por medio del checker se comprueba que este posea el mismo comportamiento que el diseño conductual, además se comprueba de manera gráfica con la herramienta del `gtkwave`. Todos los cambios deben suceder en un flanco creciente del reloj.

3. Instrucciones de utilización de la simulación

Para la automatización se crea el siguiente `makefile`:

```

1  yosys1:
2      yosys -s script2a1_2b.y
3
4
5  sed1:
6      sed -i '5s/mux2a1_2b_desc_conductual/sintetizado2a1_2b/'
7          sintetizado2a1_2b.v
8
9  iverilog1:
10     iverilog -o tarea6 BancoPrueba2a1_2b.v cmos_cells.v
11     vvp tarea6
12
13 gtwave1:
14     gtwave mux2a1_2b.vcd
15
16 all1: yosys1 sed1 iverilog1 gtwave1
17
18 yosys2:
19     yosys -s script2a1_4b.y

```

```

19 sed2:
20     sed -i '4_s/mux2a1_2b_desc_conductual/sintetizado2a1_2b/'
21         sintetizado2a1_4b.v
22     sed -i '174_s/mux2a1_4b_desc_conductual/sintetizado2a1_4b/'
23         sintetizado2a1_4b.v
24
25 iverilog2:
26     iverilog -o tarea6 BancoPrueba2a1_4b.v cmos_cells.v
27     vvp tarea6
28
29 gtkwave2:
30     gtkwave mux2a1_4b.vcd
31
32 all2: yosys2 sed2 iverilog2 gtkwave2
33
34 yosys3:
35     yosys -s script4a1_4b.ys
36
37 sed3:
38     sed -i '4_s/mux2a1_2b_desc_conductual/sintetizado2a1_2b/'
39         sintetizado4a1_4b.v
40     sed -i '173_s/mux2a1_4b_desc_conductual/sintetizado2a1_4b/'
41         sintetizado4a1_4b.v
42     sed -i '222_s/mux4a1_4b_desc_conductual/sintetizado4a1_4b/'
43         sintetizado4a1_4b.v
44
45 iverilog3:
46     iverilog -o tarea6 BancoPrueba4a1_4b.v cmos_cells.v
47     vvp tarea6
48
49 gtkwave3:
50     gtkwave mux4a1_4b.vcd
51
52 all3: yosys3 sed3 iverilog3 gtkwave3

```

Para utilizar el makefile primeramente hay que dirigirse en la terminal a la carpeta que contiene estos archivos. Se utiliza el comando **make yosys** para sintetizar el modelo conductual, en el caso del Mux 2:1 de 2 bits corresponde a yosys1, el Mux 2:1 de 4 bits yosys2 y el Mux 4:1 de 4 bits yosys3. Posteriormente mediante el comando **make sed** se modifica el nombre del modulo al sintetizado para su respectivo numero. Después utilizando el comando **make iverilog** se crea el archivo **mux.vcd** respectivo para cada mux, además que se abre Icarus Verilog. Una vez creado el archivo mux.vcd correspondiente, se utiliza el comando **make gtkwave** con su respectivo numero y se abre el programa gtkwave, se seleccionan las señales que se quieren observar y finalmente se presenta la simulación deseada. En caso de utilizar el comando **make all1** se realizan juntos todos los comando necesarios para la síntesis y gráficas del Mux 2:1 de 2 bits, con **make all2** se realizan juntos todos los comando necesarios para la síntesis y gráficas del Mux 2:1 de 4 bits y por último con **make all3** se realizan juntos todos los comando necesarios para la síntesis y gráficas del Mux 4:1 de 4 bits

4. Ejemplos de los resultados:

Como parte de comprobación de que los resultados obtenidos sean los esperados, las siguientes figuras muestran los cambios en el tiempo que sufren las distintas señales, siendo el

flanco positivo de la señal de clk el único momento donde se puedan realizar cambios. Debido a que se utilizó el comando de **posedge clock** \leq , los valores anteriores se asignaran a las variables después del flanco creciente.

En la figura 4 se observa que con la señal reset_L en alto, el selector igualmente en 1 y el valid de la entrada 1 también en 1, esto significa que a la salida debe salir el valor de la entrada data_in12a1_2b, tal como se muestra en la imagen. Además cabe destacar que el valid_data_out esta en alto lo que significa que permite cambios en la salida. Por último también se puede apreciar mediante la gráfica que el modelo estructural y sintetizado poseen el mismo comportamiento y se verifica el funcionamiento, además que el checker no emite ninguna señal de que hubiera algún cambio. En la figura 5 a pesar que se tiene la señal reset_L en alto, el selector igualmente en 1 y el valid de la entrada 1 no corresponde a 1, por ende el valid_data_out se pone en bajo y eso significa que no se producen cambios. Por ende la salida data_out va a conservar su valor anterior y no el de 01 que le correspondería.

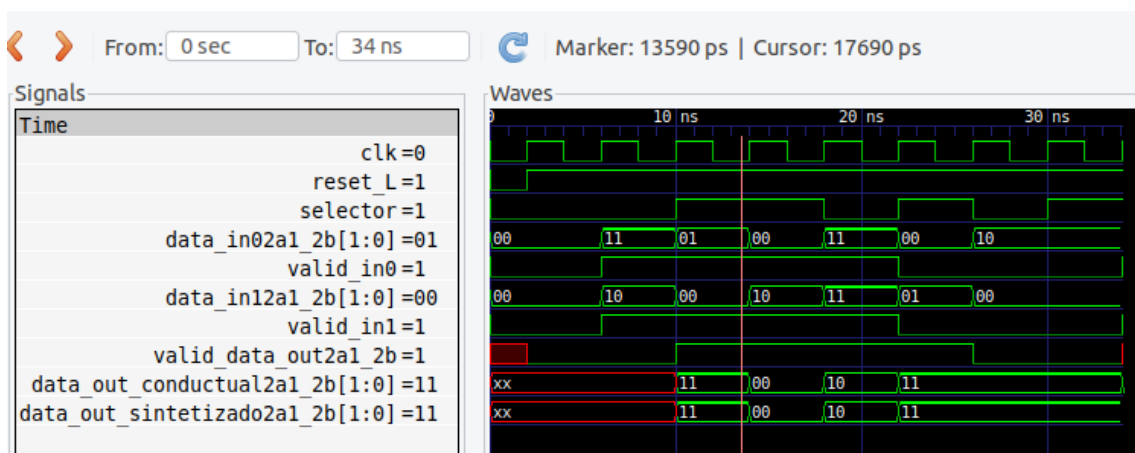


Figura 4: Ejemplo 1 Mux 2:1 con 4 bits

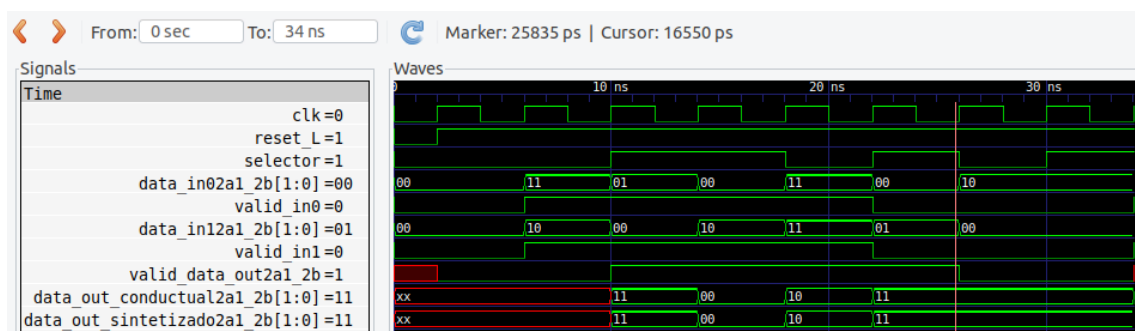


Figura 5: Ejemplo 2 Mux 2:1 con 2 bits

En la figura 6 pasa similar a lo que sucede en la figura 4, pero con salida de 4 bits. Ya que se cuenta con salida 1 las señales de reset, selector y el valid de entrada 1, por ende el valid_data_out está con señal alta y significa que la salida si puede cambiar al valor esperado, como sucede en la imagen que cambia su valor al esperado que es 0. En la figura 7 pasa un caso similar al de la figura 5, dado que a pesar de que se tiene la señal reset_L en alto, el selector igualmente en 1 y el valid de la entrada 1 no corresponde a 1, por ende

el valid_data_out se pone en bajo y eso significa que no se producen cambios. Por ende la salida data_out va a conservar su valor anterior y no el de 5 que le correspondería. Además se observa comportamiento idéntico entre modelo conductual y síntesis

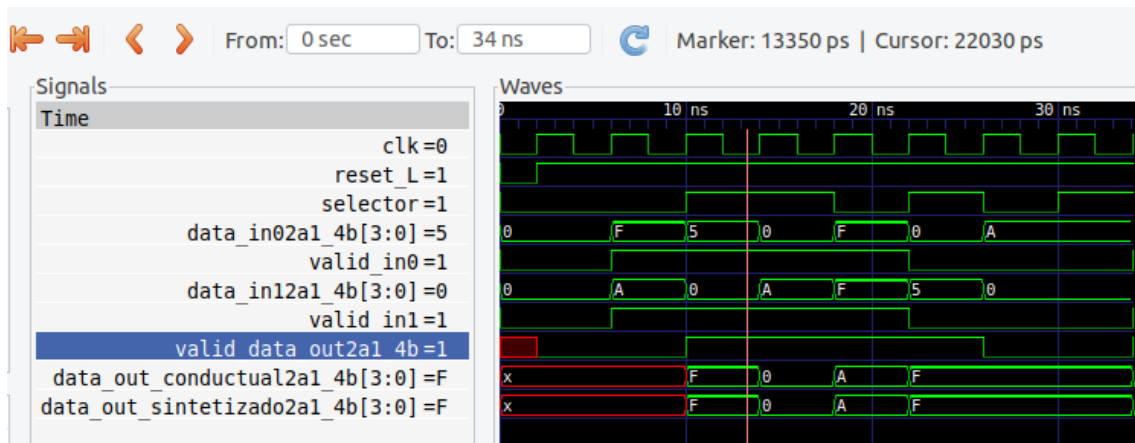


Figura 6: Ejemplo 1 Mux 2:1 con 4 bits

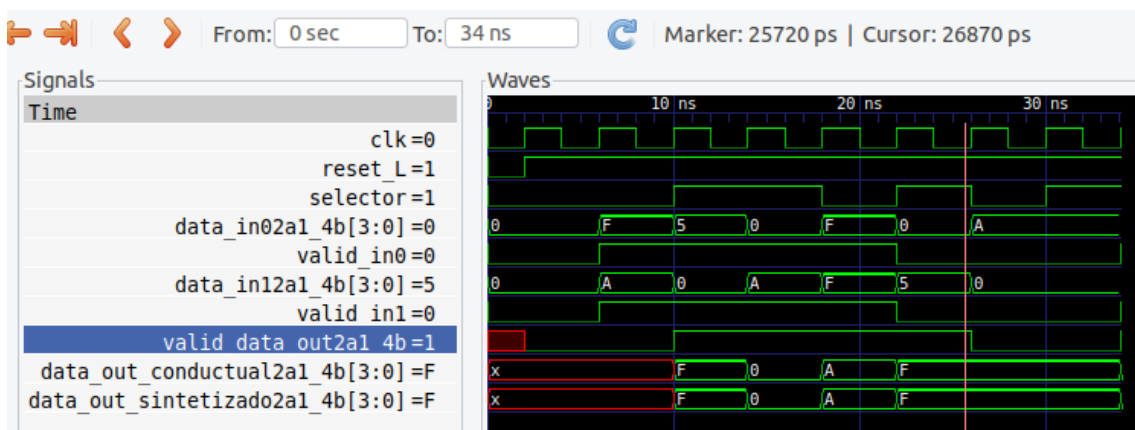


Figura 7: Ejemplo 2 Mux 2:1 con 4 bits

En la figura 8 pasa similar a lo que sucede en las figuras 4 y 6, pero con un Mux de 4:1. Cabe mencionar que el selector funciona de manera que se tiene ABCD, significa que los divide en 2 donde A=0 y B=1 en la primera señal, C=0 y D=1 en la segunda señal y por último en la tercera señal sería A|B=0 y C|D=1, dependiendo de las señales que se tengan. Por ejemplo si se tiene 001, se escogerá la letra C. Se cuenta con salida 1 las señales de reset, en selector se sabe que la señal de interés es la data_in2y el valid de entrada 2 corresponde a 1, por ende el valid_data_out está con señal alta y significa que la salida si puede cambiar al valor esperado, como sucede en la imagen que cambia su valor al esperado que es 0. En la figura 9 pasa un caso similar al de la figura 5 y 7, dado que a que se tiene que la señal del valid de la entrada 1 no corresponde a 1, por ende el valid_data_out se pone en bajo y eso significa que no se producen cambios. Por ende la salida data_out va a conservar su valor anterior y no el de 9 que le correspondería.

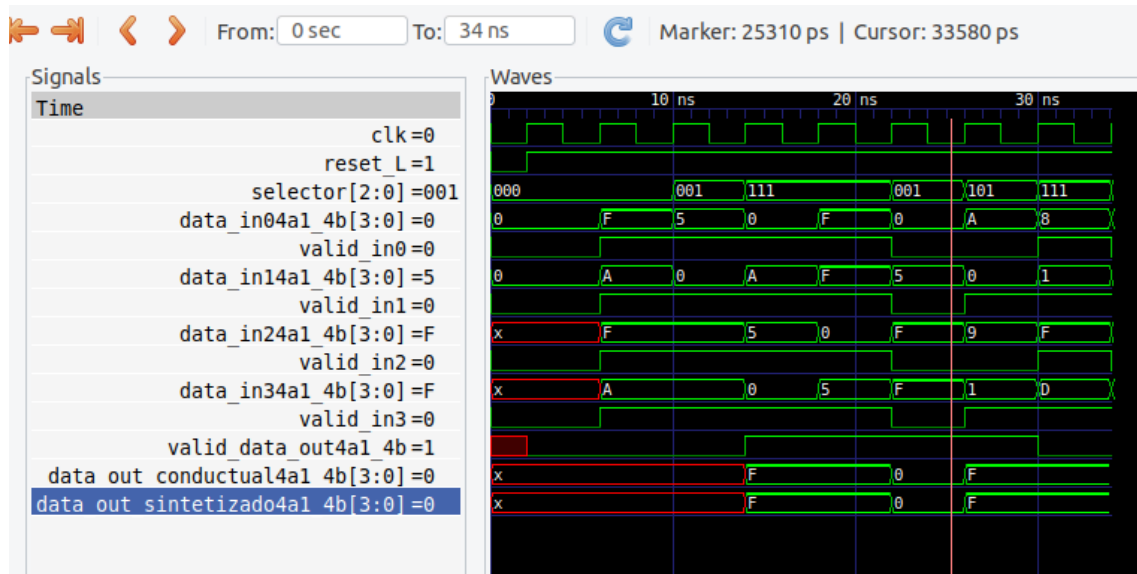


Figura 8: Ejemplo 1 Mux 4:1 con 4 bits

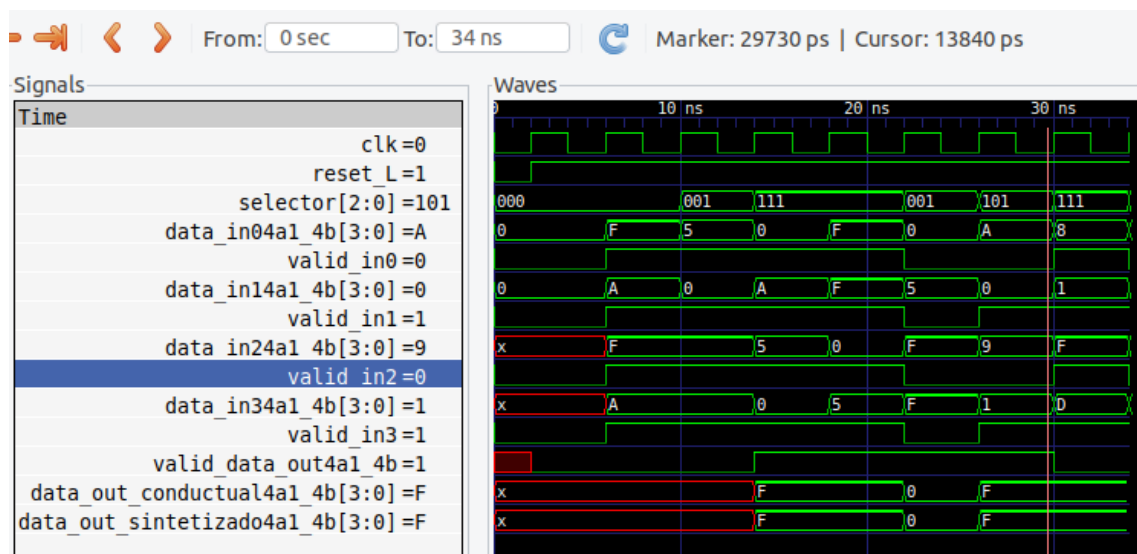


Figura 9: Ejemplo 2 Mux 4:1 con 4 bits

5. Análisis y conclusiones

Analizando los ejemplos de la sección anterior, se puede notar que el diseño cumple con lo solicitado para cada circuito, osea el Mux 2:1 con 2 y 4 bits y el Mux 4:1. Donde se verificó el funcionamiento del valid en las entradas y se reflejó esto en el valid de salida. Además analizando todos los ejemplos se destaca el mismo comportamiento entre el modelo conductual y el sintetizado, lo que garantiza el óptimo funcionamiento de estos diseños.

Se puede concluir que la simulación cumple con lo esperado. Los archivos de descripción conductual, probador y el banco de pruebas son muy similares entre cada circuito debido a

que para hacer el circuito c, se utilizó el circuito b y de igual manera para contruir el b se utilizó el a. La conexión del banco de pruebas se hizo de manera automatica con autoinst, sin embargo hubo que corregir manualemente la salida del archivo sinteizado. Finalmente se hizo un makefile para automatizar el proceso, y como resultado de ingresar el comando make all1, make all2 y make all3 y se obtienen todas las gráficas buscadas como se observa en los ejemplos. Cabe mencionar que para el circuito a se utilizaron los mismos archivos de descripción conductual, probador y Banco de pruebas, sin embargo se le agregó el valid. Además se comprobó el funcionamiento correcto de estas entradas valid y su relación con la salida valid, además del comportamiento de cada circuito.

6. Distribución del tiempo invertido en la tarea

En total se duró 7 horas con 10 minutos realizando la tarea

- Buscar información: 2 minutos ya que todo lo que se necesito fue repasar tareas anteriores
- Estudiando la información tomó 2:08 horas entender que era exactamente lo que se pedía
- Ejecutando lo que decidió hacer y probándolo se tomó 4 horas.
- Realizar el reporte tomó 1 hora.