

## Tarea 6

### Distribución de tiempo:

- Realización del módulo: 12 horas y media
- Realización de pruebas/testbench: 10 horas con 48 minutos

### Diagrama del circuito

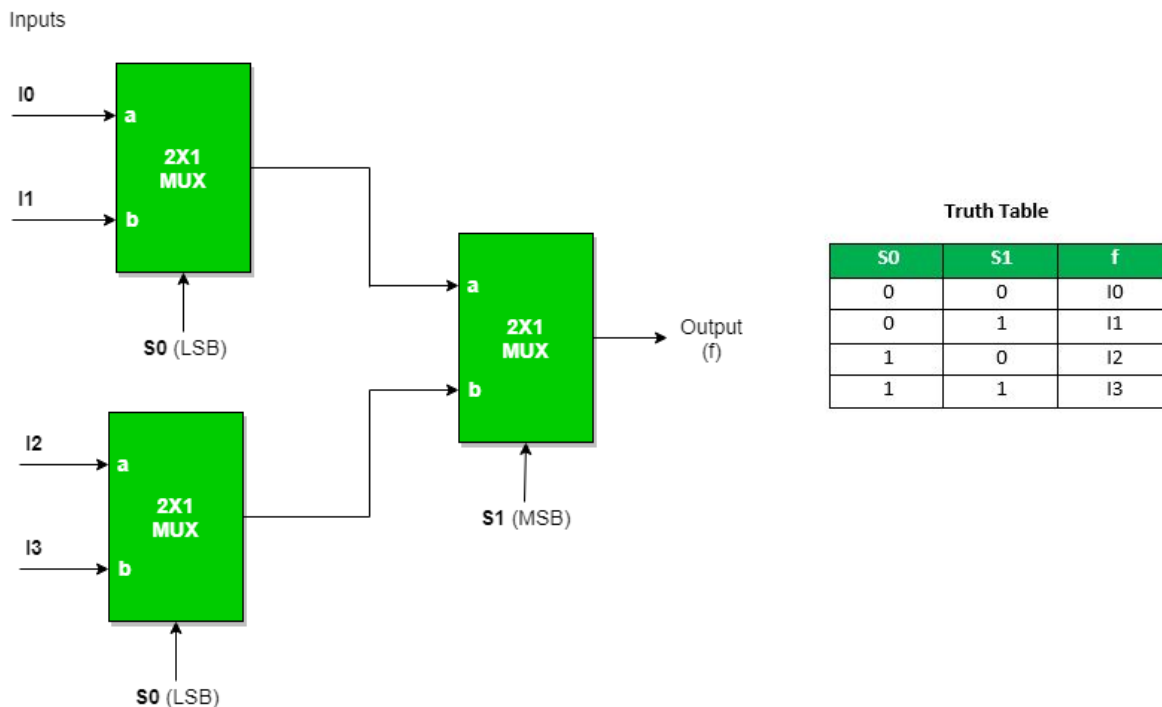


Figura 1. Mux 4x1 de 4 bits a partir de muxes 2x1 de 4 bits.

### Plan de pruebas:

Para cada uno de los muxes creados, se hace un testbench y un probador para comprobar el funcionamiento de los mismos. Se cambian periódicamente las entradas y se observan los resultados.

### Instrucciones para simulación:

Descargar .zip y en terminal (dependiendo de cuál carpeta se esté viendo) escribir make para revisar el comportamiento de un MUX en particular.

## Resultados:

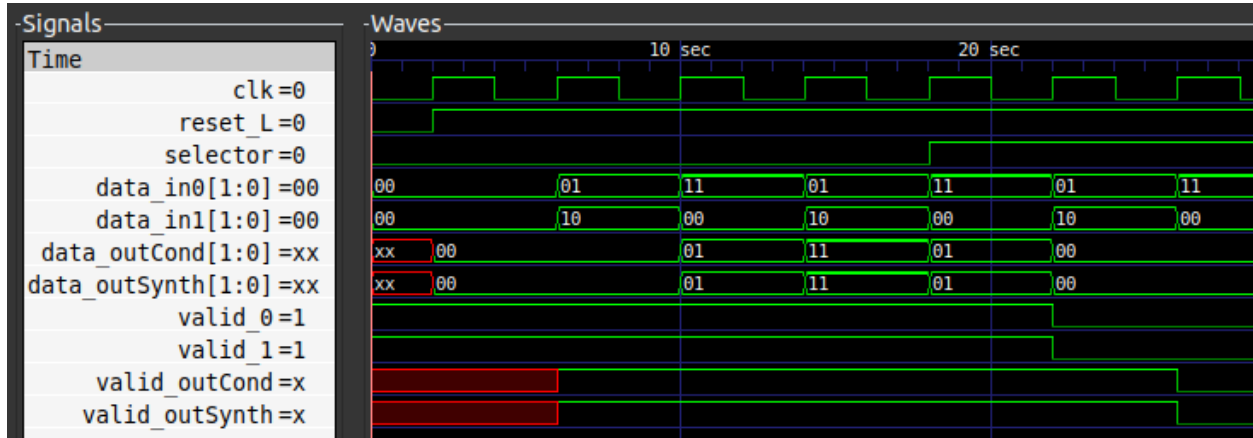


Figura 2. Mux 2x1 de 2 bits conductual vs síntesis.

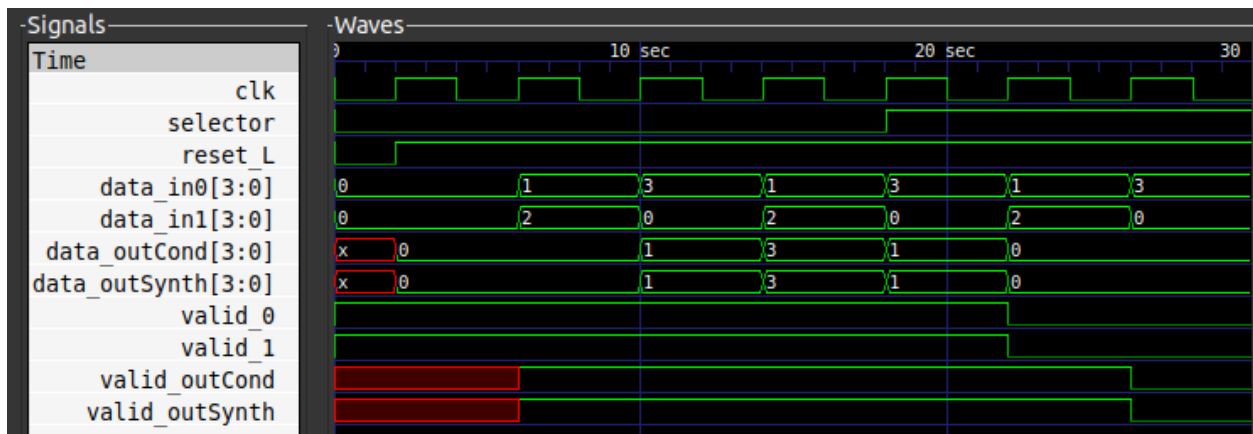


Figura 3. Mux 2x1 de 4 bits conductual vs síntesis.

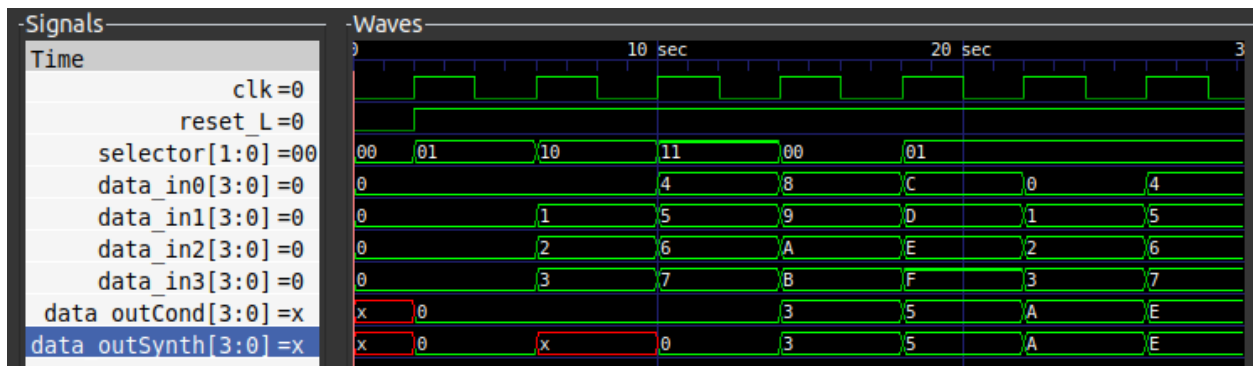


Figura 4. Mux 4x1 de 4 bits conductual vs síntesis.

## Análisis y conclusiones:

La figura 1 muestra el comportamiento del MUX 2x1 de 2 bits; nótese que el funcionamiento de dicho dispositivo es justo el esperado, tanto para el modelo conductual como para el estructural. Cuando selector = 0, a la salida sale data\_in0 y si selector = 1, a la salida sale data\_in1, esto solamente si dicha entrada es válida. En caso que la entrada no sea válida, el multiplexor sigue tirando el valor que tenía anteriormente en la salida hasta que se reciba una entrada válida.

Lo mismo pasa en la figura 2, solamente que esta vez es un MUX 2x1 de 4 bits, cabe destacar que este fue construido a partir de MUXES 2x1 de 2 bits. De misma manera, se tienen que los resultados son los esperados.

Por último, la figura 3 muestra el MUX 4x1 de 4 bits creado a partir de los otros 2 MUXES 2x1 mencionados anteriormente. Nótese que la descripción conductual y la estructural tiran las mismas salidas, sin embargo, véase que los valores que arroja a la salida no son los de las entradas; los MUXES 2x1 de 4 bits utilizados para crear el 4x1 tienen por dentro FlipFlops, y estos generan retardos a la hora de entregar la salida, es por esta razón que los valores de data\_out no coinciden con las entradas que se encuentran en el flanco que se está analizando. Si se quisiese arreglar este inconveniente, simplemente se debe respetar un cierto tiempo antes de cambiar las entradas para así tener el comportamiento del MUX un poco más evidente.

En esta tarea se gastó mucho tiempo haciendo troubleshooting, por lo que se recomienda ser más organizado con los files para tener menos problemas a la hora de buscar bugs en los códigos de Verilog.