

Tarea #2
Multiplexor con memoria
(Entrega 27 de agosto)

*Al igual que en la **Tarea #1** tome el tiempo que demora en hacer cada una de las cosas solicitadas: búsqueda de información, diseño, elaboración de las pruebas, ejecución de las simulaciones, etc.*

Evaluación

- | | |
|-------------------------------|-----|
| 1. Funcionamiento del diseño: | |
| a. Simulación | 40% |
| b. Pruebas | 40% |
| 2. Documentación | 10% |
| 3. <i>Makefile</i> | 10% |

Trabajo a realizar sobre el dispositivo a diseñar

Para esta tarea se deben completar los siguientes puntos:

1. Realice un diagrama del circuito propuesto.
2. Escriba una descripción **conductual** del multiplexor usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado. Además servirá como patrón para hacer las pruebas sobre los diseños estructurales que se vayan a hacer. La descripción en Verilog deberá tener al menos un módulo de banco de pruebas, un módulo probador, y un módulo con la descripción conductual. Use Icarus Verilog para hacer esto.

Recuerde que esta descripción conductual deberá ser sintetizada en el futuro. No se permite el uso de bloques *initial* u operadores “#” en la descripción conductual (en el probador sí se pueden usar). Tenga cuidado con los *latches*. No utilice *negedge clk* en ninguna parte de la tarea.

3. Defina un plan de pruebas para garantizar el funcionamiento del diseño. Desarrolle las pruebas y automatizarlas mediante el *Makefile*. Diseñe un archivo banco de pruebas y un archivo probador. Utilice como guía el probador del ejemplo Alarma, respetando las reglas de sincronizar con el reloj y utilizar asignaciones no bloqueantes.

Especificaciones

Estructura del bloque conductual:

```
module mux (
    input clk,
    input reset_L,
    input selector,
    input [1:0] data_in0,
    input [1:0] data_in1,
    output reg [1:0] data_out
);
```

El bloque consiste en un multiplexor de 2:1 con dos entradas y una salida de datos de 2 bits, además de un flip-flop a la salida para almacenar el resultado. Todo esto es parte de la descripción conductual, dentro del “*module mux*”.

Recuerde aplicar reset_L a los flip-flops para borrar su estado inicial no determinado. Cuando el bloque está en reset (reset_L == 0), los fops mostrarán valor de 0. Cuando no está en reset (reset_L == 1), mostrará el resultado del multiplexor.

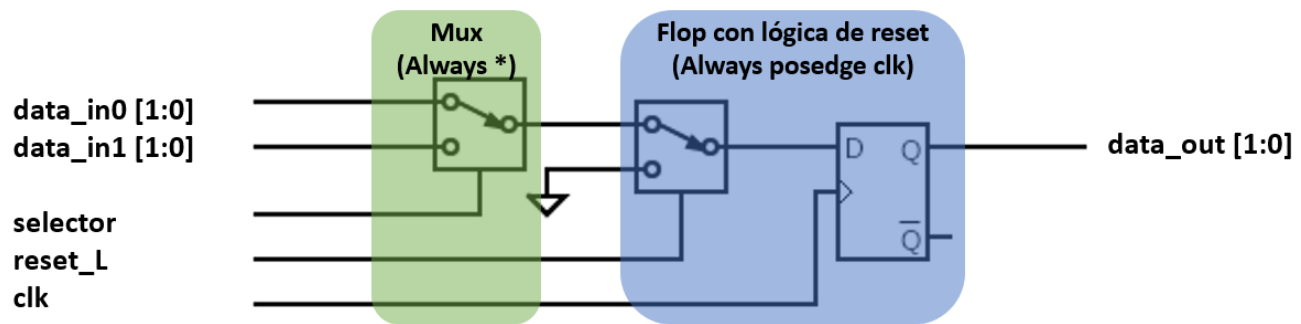


Figura 1. Esquemático del diseño.

Consejos para el diseño

- Recuerde que esta tarea se trata de un diseño **conductual**, en el que se utilizan bloques *always* con lógica *if/else*, entre otros.
- Utilice como guía el ejemplo de Alarma (descripción, probador y testbench).
- Cuando se habla de que un dato debe pasar sin retrasos de un punto al otro, tiene que utilizar lógica combinacional. Utilice un *always @(*)* o un *assign* (el retardo combinacional de las compuertas se desprecia en esta tarea).
- Cuando debe almacenar o sostener un dato, tiene que utilizar lógica secuencial (flip-flops). Utilice un *always @(posedge clk)*
- Recuerde el paralelismo en el diseño de hardware, distintas lógicas o distintos bloques *always* pueden ejecutarse en el mismo tiempo de simulación.
- Dos *drivers* o asignadores no pueden manipular la misma señal. Por ejemplo, dos bloques *always* no pueden tratar de asignar un valor a la misma señal. Esto provocaría un corto en el circuito.
- La lógica combinacional en un bloque *always @(*)* siempre debe tener un valor por defecto al inicio del bloque.

Propuesta de Plan de Pruebas Mínimo

Construya un probador que genere como mínimo el estímulo siguiente sobre las entradas del circuito. Además, en la captura se observa cómo deben reaccionar las salidas ante los cambios en las entradas.

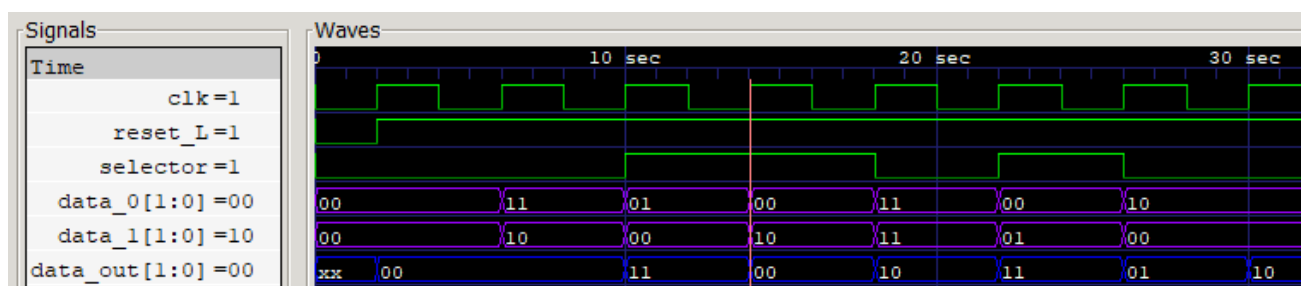


Figura 1. Captura de entradas y salidas esperadas del multiplexor.

Guía para el reporte

Se debe entregar en forma electrónica un documento que incluya los siguientes puntos en a lo sumo 10 páginas de longitud:

1. **Distribución del tiempo invertido en la tarea.**
2. **Descripción arquitectónica o diagrama del circuito.**
3. **Plan de Pruebas:** Descripción de las pruebas realizadas.
4. **Instrucciones de utilización de la simulación:** Describir cómo correr las pruebas mediante el *makefile*.
5. **Ejemplos de los resultados:** Una descripción de los resultados más importantes acompañados de los diagramas temporales de la simulación (GTKWave) o cualquier otra salida que demuestre claramente el comportamiento descrito. No es necesario incluir una muestra exhaustiva de resultados, sino que los más representativos del diseño. El punto es mostrarle al lector los comportamientos más sobresalientes para formarle una idea clara del funcionamiento del diseño.
6. **Análisis y conclusiones:** Analice los resultados y enumere los inconvenientes durante la tarea.