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

Tarea #2

1. Diagrama

En la figura 1 se muestra el diagrama del circuito. 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.

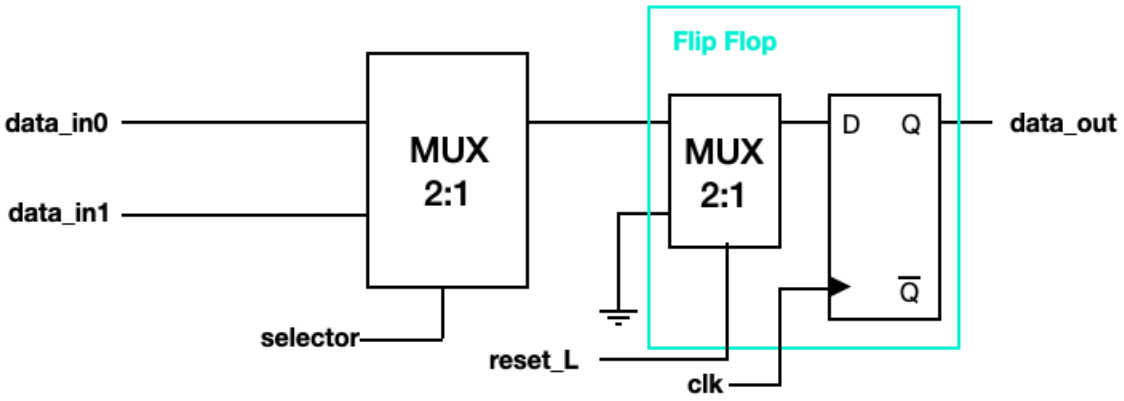

Figura 1: Diagrama

2. Plan de Pruebas

Se crean los archivos de descripción conductual y probador basándose en el ejemplo de alarma disponible en mediación virtual.Para el archivo de descripción conductual se realizó una 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. En el caso del probador se añadieron manualmente las señales de reset_L, selector, data_in0 y data_in1 indicadas en el enunciado. 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. Entre las verificaciones se deben asegurar que se cumplan las condiciones de que en reset sea 0,se posea una señal de salida de 00. De que en caso de que reset sea 1, dependa del selector(0=data_in0 y 1=data_in1) y además

comprobar que todos los cambios se realicen únicamente cuando haya un flanco creciente en la señal del clock(reloj).

3. Instrucciones de utilización de la simulación

Para la automatización se crea el siguiente makefile:

```
1
2 iverilog:
3         iverilog -o tarea2 BancoPrueba.v
4         vvp tarea2
5 gtkwave:
6         gtkwave mux.vcd
7
8 all: iverilog gtkwave
```

Para utilizar el makefile primeramente hay que dirigirse en la terminal a la carpeta que contiene estos archivos.Posteriormente utilizando el comando **make iverilog** se crea el archivo **mux.vcd**, además que se abre Icarus Verilog. Una vez creado el archivo mux.vcd, se utiliza el comando **make gtkwave** 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 all** se realizan juntos los comandos de iverilog y gtkwave.

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** <=, los valores anteriores se asignaran a las variables después del flanco creciente. En la figura 2 se muestra que debido a que la señal del reset está en 0 antes del flanco creciente, esta no depende de data_in0 y data_in1 ,osea la salida debe de dar 00, tal como se ve. En la figura 3 se ve que el valor anterior del reset es 1, entonces depende del selector, dado que este posee un valor anterior de 0 se le asigna a la salida el valor anterior de data_in0 osea 11 como se muestra. En la figura 4 el reset anterior sigue en 1, el selector anterior posee un valor de 1 por ende a la salida debe estar el valor anterior de data_in1 osea 00 como se puede apreciar en la figura. Solo se utilizan estos 3 ejemplos ya que representan las 3 condiciones en las que depende la salida.

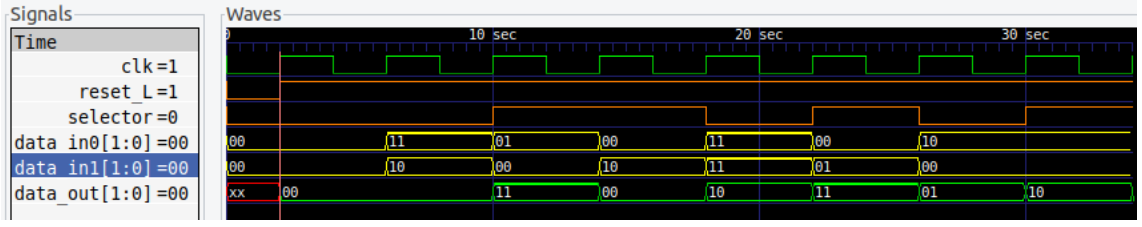

Figura 2: Ejemplo 1

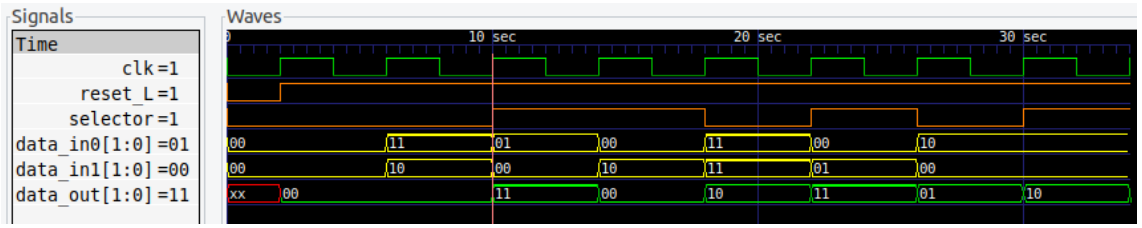

Figura 3: Ejemplo 2

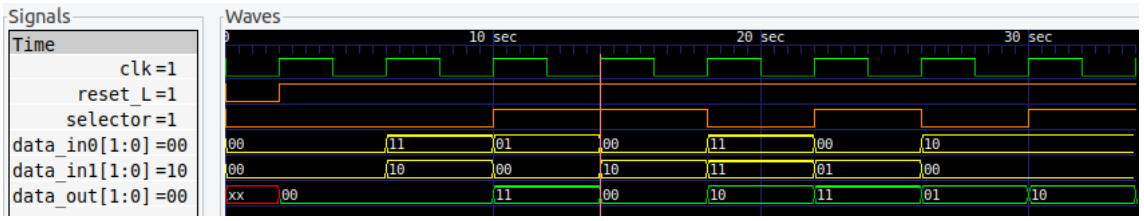

Figura 4: Ejemplo 3

5. Análisis y conclusiones

Analizando los ejemplos de la sección anterior, se puede notar que el diseño cumple con lo solicitado. Ya que como se ve en el ejemplo 1, si el reset antes del flanco creciente está en 0, la salida estará en 0. En cambio en los ejemplos 2 y 3 se muestra cuando el reset anterior está en 1, que en el caso del ejemplo 2 el selector está en 0 por ende a la salida sale el valor anterior de data_in0 y en el ejemplo 3 el selector está en 1 por ende a la salida sale el valor anterior de data_in1.

Se puede concluir que la simulación cumple con lo esperado. Los archivos de descripción conductual, probador y el banco de pruebas fueron basados en el ejemplo de alarma disponible en mediación virtual, es por eso que la estructura se parece bastante. Cabe mencionar que toda la lógica en el archivo conductual se hizo únicamente a través del **always(posedge clk)**, sin necesidad de utilizar **always(*)**, razón por la cual todos los cambios se realizaron únicamente en los flancos crecientes. En el probador se introdujeron todas las señales de manera manual a excepción de la salida, de acuerdo a las especificaciones del enunciado.La conexión del banco de pruebas se hizo de manera automatica con autoinst. Finalmente se hizo un makefile para automatizar el proceso, y como resultado de ingresar el comando make all se obtiene la gráfica buscada como se observa en los ejemplos.

6. Distribución del tiempo invertido en la tarea

En total se duró 6 horas con 32 minutos realizando la tarea

- Buscar información: 2 minutos ya que todo lo que se necesito fue el ejemplo de la alarma
- Estudiando la información tomó 2 horas entender el código
- Ejecutando lo que decidió hacer y probándolo se tomó 3 horas.

- Realizar el reporte tomó 1 hora con 20 minutos.