Adrián Montero Bonilla

B85092

Tarea 2

Distribución de tiempo:

-Búsqueda de información: 2 horas

-Realización del módulo: 2 horas y media

-Realización de pruebas/testbench: 45 minutos

Diagrama del circuito:

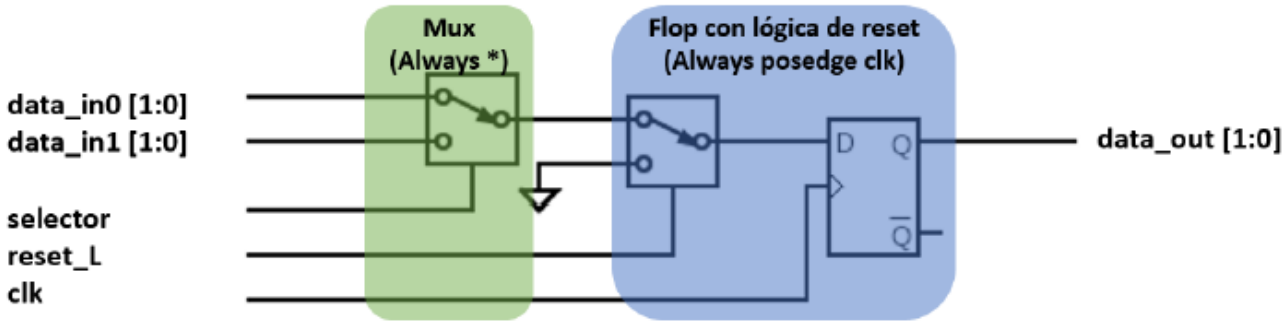

Figura 1. Esquemático del diseño.

Plan de pruebas:

Primeramente, se diseñó un archivo testbench la cual es el generador de señales y dichas señales cambian cada cierto tiempo con el fin de probar el diseño del multiplexor realizado; cuando llegue un flanco creciente del reloj, basándose en lo que se tenga en la entrada *selector*, el circuito llevará a la salida el valor de *data_in0* o de *data_in1*. Todas las señales se inicializaron con el valor 0 por default y los cambios de estas fueron sincronizadas con el reloj para evitar los casos de metaestabilidad.

Instrucciones de utilización de la simulación:

Para poder correr la simulación, simplemente se guarda en una carpeta aparte los archivos adjuntos en el .zip y se abre una consola con la dirección de dicha carpeta; una vez hecho esto, se corre en la terminal el comando “make” e inmediatamente se desplegará GTKWave para verificar los resultados.

Resultados obtenidos:

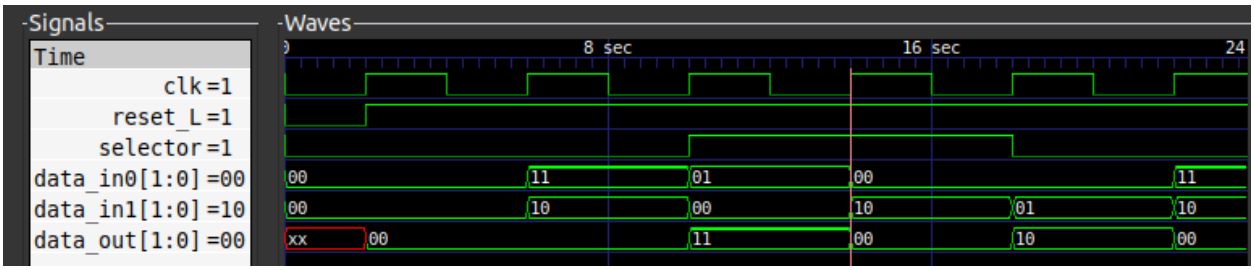

Figura 2. Resultados obtenidos en GTKWave

La figura 2 muestra el comportamiento del diseño creado; se muestran los comportamientos de las entradas clk, reset_L, selector, data_in0 y data_in1 como también el comportamiento de la salida data_out. Nótese que al puro principio se aplica un reset para inicializar de manera correcta la simulación.

Análisis y conclusiones:

Como se puede ver en la figura 2, los resultados adquiridos son correctos; cada vez que ocurre un flanco positivo del reloj, si se tiene selector = 0 a la salida se mostrará la entrada data_in0 y, en caso contrario, si se tiene selector = 1 a la salida se muestra la entrada data_in1. Cabe destacar que, en los casos en los que las señales cambien justo en el flanco positivo del reloj, se tomará el valor de la señal justo antes de que ocurra el flanco; véase la línea roja vertical de la figura 2, en ese caso se tiene a selector = 1 y, por lo tanto, a la salida se debe reflejar el dato de data_in1. Nótese que data_in1 cambia justo en el flanco positivo del reloj y, como se menciona anteriormente, se toma el valor de data_in1 como 00 (valor de dicha entrada justo antes del flanco) y dicho valor se refleja en la salida data_out.