

Universidad de Costa Rica  
Escuela de Ingeniería Eléctrica  
IE0523 Circuitos Digitales II

Avance I

Proyecto I: Diseño de la capa PHY de la interfaz PCIe

Felipe Badilla Marchena - B70848

Jose Mario Gonzales Abarca - B83362

Rolando Murillo Perez - B54966

Karen Dayana Tovar Parra - B77673

Prof. Jorge Soto

25 de mayo

I semestre - 2021

## Plan de Trabajo

Para el primer avance del proyecto, nos enfocamos en el bloque de recirculación y en la lógica de muxes y demuxes.

### Creación Mux2x1 de 8 bits

Para esta parte se crea un módulo de Multiplexor 2x1 conductual con el objetivo de poder realizar instancias de este módulo más adelante para formar la lógica de Multiplexores. Este multiplexor 2x1 de 8 bits se realizó a partir de lógica combinacional utilizando `always(*)` y también a partir de lógica secuencial utilizando `always(posedge clk)`, básicamente con la lógica combinacional se define para que en los instantes en que los `valids` de cada entrada estén activos poder asociar la salida a una de estas entradas, esto a su vez se hace posible con el selector interno el cual nos elige cual de las dos entradas se quiere colocar en la salida.

Resumiendo lo que recibe este bloque serían: 2 Entradas con 1 válida cada entrada, 3 clock para posteriormente poder usarse de manera estratégica en la lógica de multiplexores, un selector que hace toggle (no se recibe como parámetro) y finalmente 1 salida junto con un `valid`.

En la figura 1 se muestra un diagrama de tiempos luego de elaborar un probador y un banco de pruebas individual para comprobar la correcta funcionalidad del mux2x1 de 8 bits.

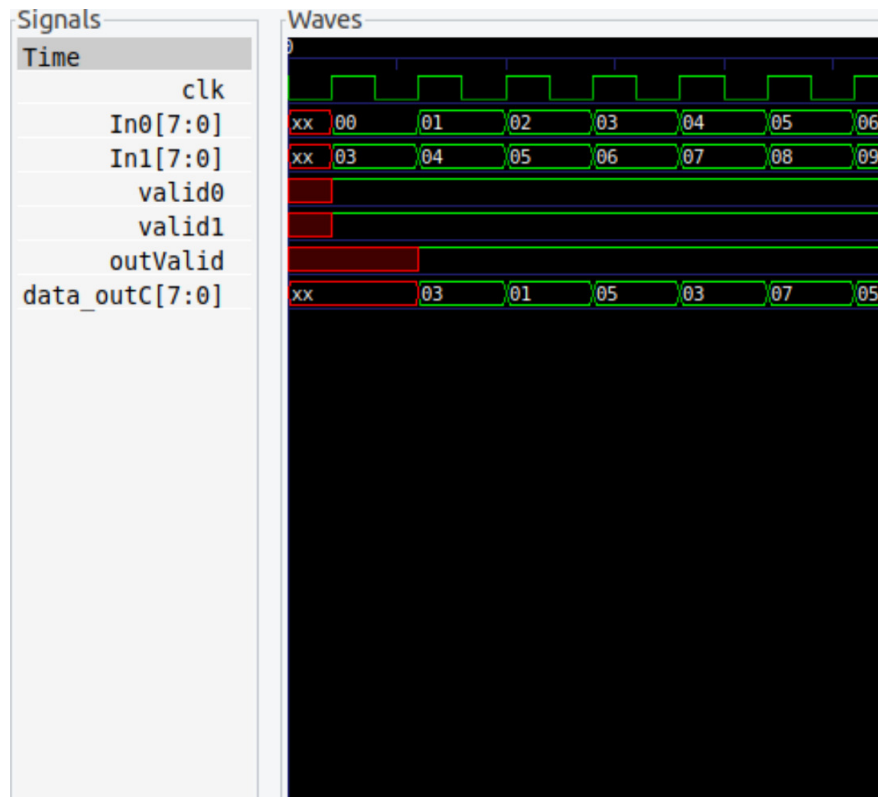


Figure 1: Diagrama de tiempos del multiplexor 2x1 de 8 Bits.

### Lógica de multiplexores dentro de Phy\_tx

La lógica de multiplexores la podemos pensar como un Mux4x1 de 8 bits formado de 3 Muxes2x1 de 8 Bits como se muestra en la Figura 2.

### Mux 4x1 de 8Bits con selector interno

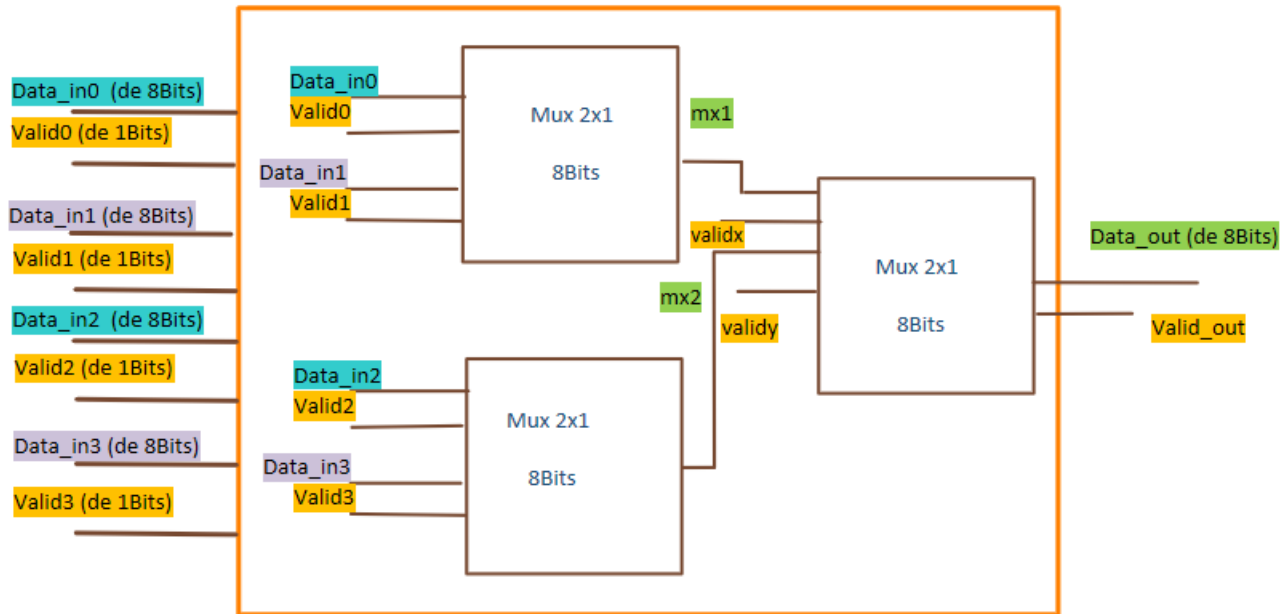


Figure 2: Diagrama de la lógica de multiplexores con base a 3 instancias de multiplexores 2x1 de 8 Bits.

Es importante, acotar que cada nivel de muxes trabaja con frecuencias de clk diferente, por lo que la salida final de esta lógica se verá afectada por estas frecuencias y por los dos niveles de flip flops que comparte.

Para mostrar la funcionalidad del módulo de multiplexor 4x1 se muestra la Figura 3 en la cuál se tiene el diagrama de tiempos hecho para el módulo ante un validIn igual a 1.

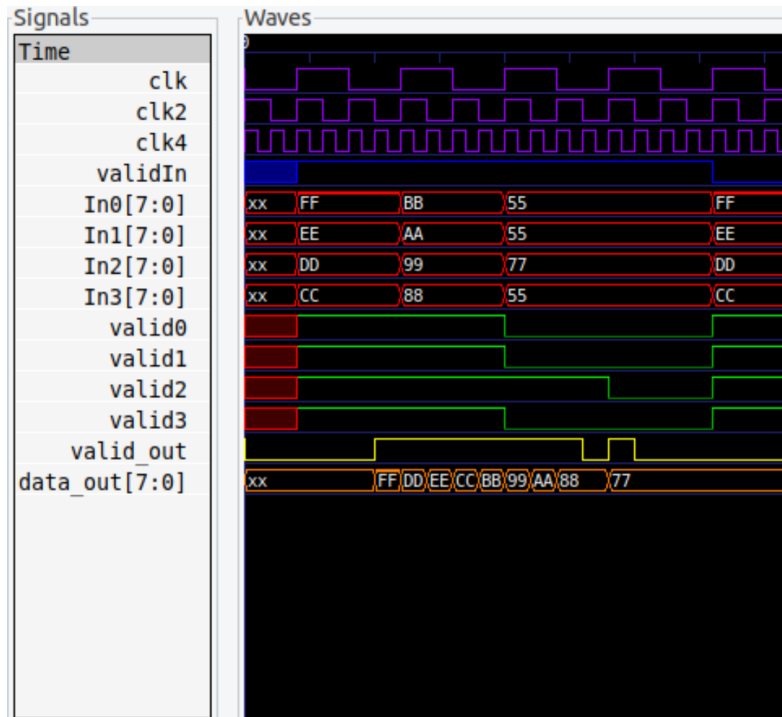


Figure 3: Diagrama de tiempos para la lógica de multiplexores.

### Creación Demux1x2 de 8 bits

Para crear el Demultiplexor 1x2 de 8 Bits hay que plantear algo similar a lo que se hizo para el Multiplexor 2x1 de 8 bits pero ahora teniendo 2 entradas (junto con 2 valids) y una salida (con un valid), la idea sigue siendo la misma, se quiere apartir del Demultiplexor 1x2 de 8 Bits formar la lógica de Demultiplexores.

En resumen se tendría 1 Entrada y esta asociada a un valid, un selector automático que hace toggle en cada flanco de reloj (no se recibe como parámetros) y por último 2 salidas con un valid cada una. Cabe mencionar que para el funcionamiento se le decide ingresar también 3 clocks que son pensando en la lógica de Demultiplexores, ahí tendrán mayor sentido.

A continuación en la Figura 4 se muestra un diagrama de tiempos del módulo creado para el demultiplexor 1x2 de 8 Bits ante un validIn igual a 1.

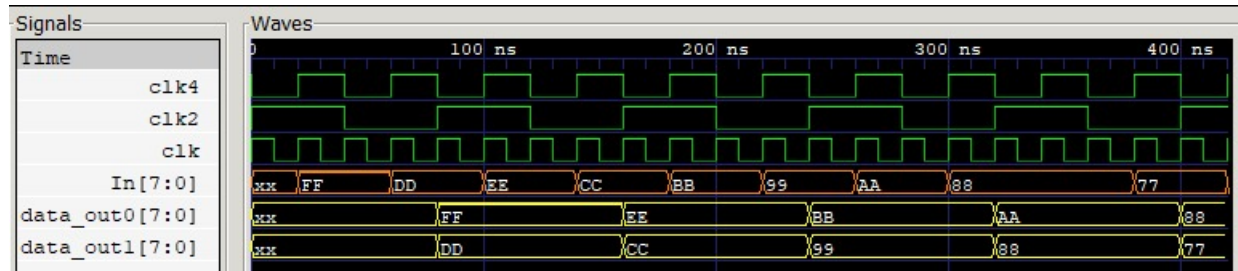


Figure 4: Diagrama de tiempos para el demultiplexor 1x2 de 8 Bits.

## Lógica de Demultiplexores dentro de Phy\_rx

La lógica de demultiplexores la podemos pensar como un Deux1x4 de 8 bits formado de 3 Deuxes1x2 de 8 Bits como se muestra en la Figura 5.

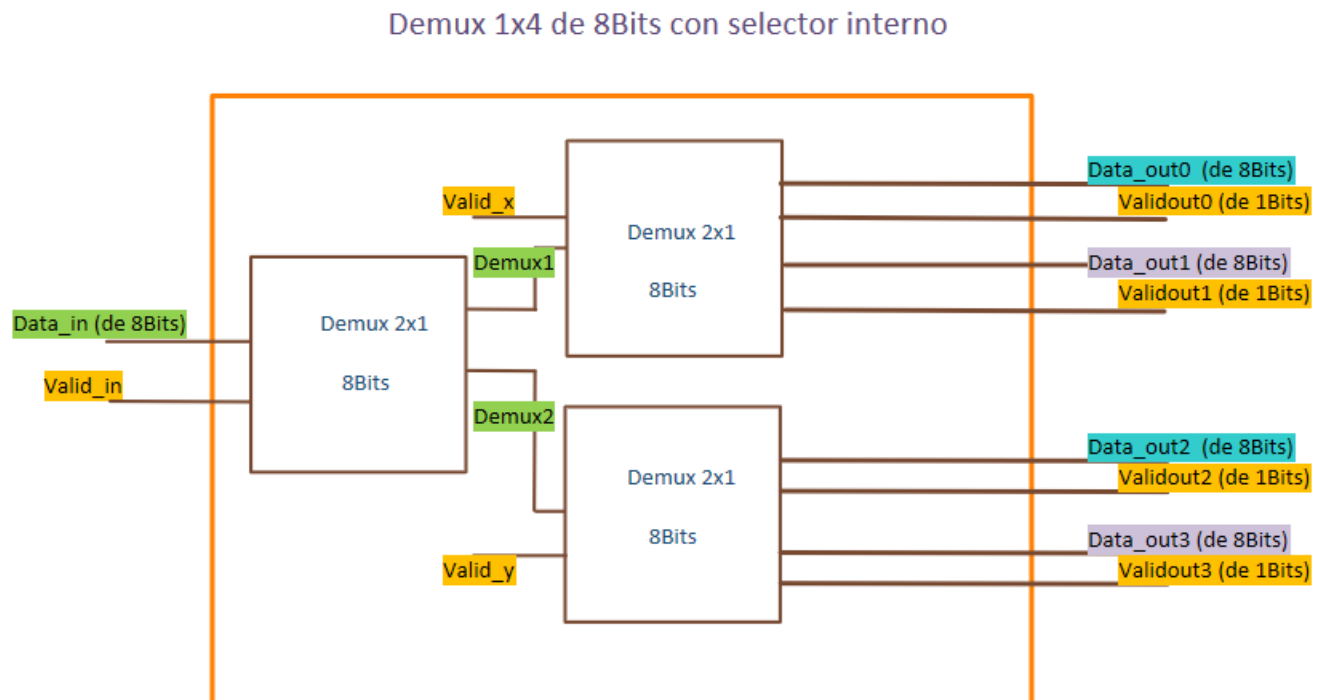


Figure 5: Diagrama de la lógica de demultiplexores con base a 3 instancias de demultiplexores 1x2 de 8 Bits.

De igual manera es importante, recalcar que cada nivel de muxes trabaja con frecuencias de clk diferente, por lo que la salida final de esta lógica se verá afectada por estas frecuencias y por los dos niveles de flip flops que comparte.

De igual manera para esta lógica se elaboró un probador individual y un banco de pruebas para observar el comportamiento del bloque como tal, en la figura 6 se muestra el diagrama de tiempos asociado.

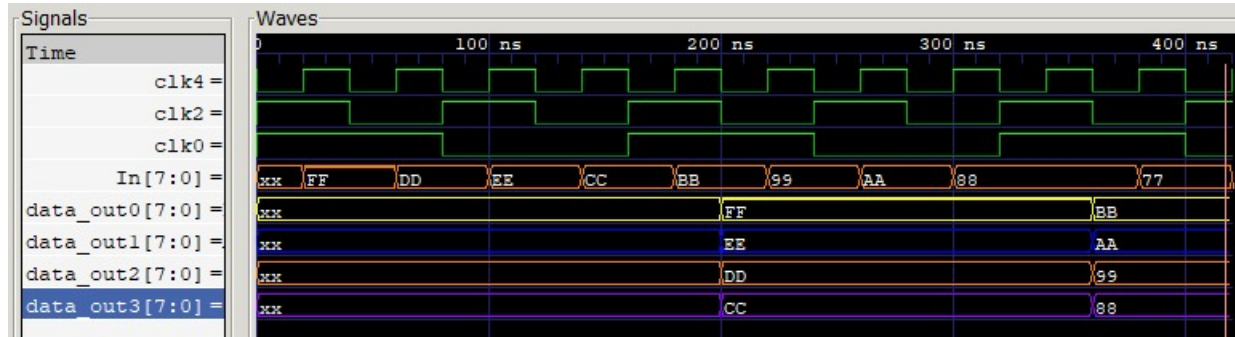


Figure 6: Diagrama de tiempos de la lógica de demultiplexores.

## Bloque de Recirculación

Este bloque funciona como una especie de demultiplexor, específicamente se utilizan 4 demultiplexores 1x2 de 8 Bits, todos los demuxes en este caso comparten el ValidIn, este validIn proviene de la señal que envía el bloque serial paralelo del phy\_tx.v. En base al valor recibido proveniente de ese bloque decide a donde enviar las salidas si a la lógica de multiplexores o directamente enviar esas salidas al probador.

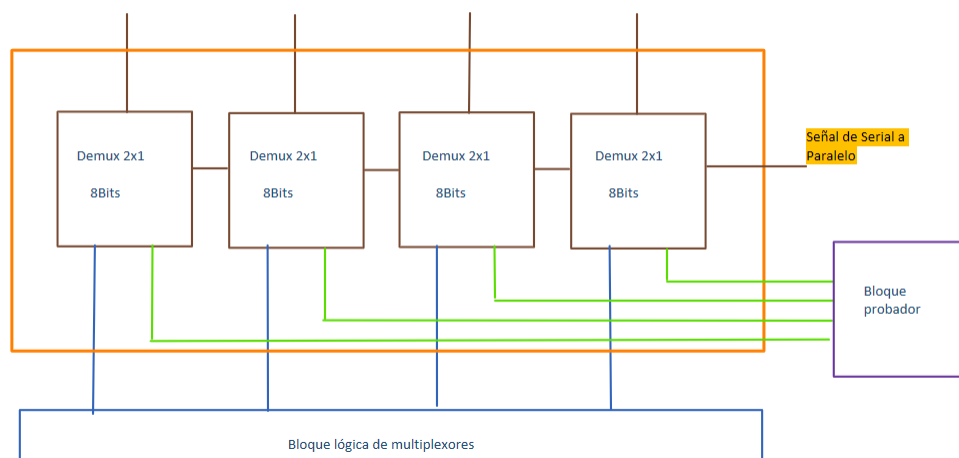


Figure 7: Diagrama descriptivo del bloque de Recirculación.

En el diagrama de la figura 8 se muestra la manera en que estaría funcionando este bloque de recirculación ante ciertos valores de validIn.

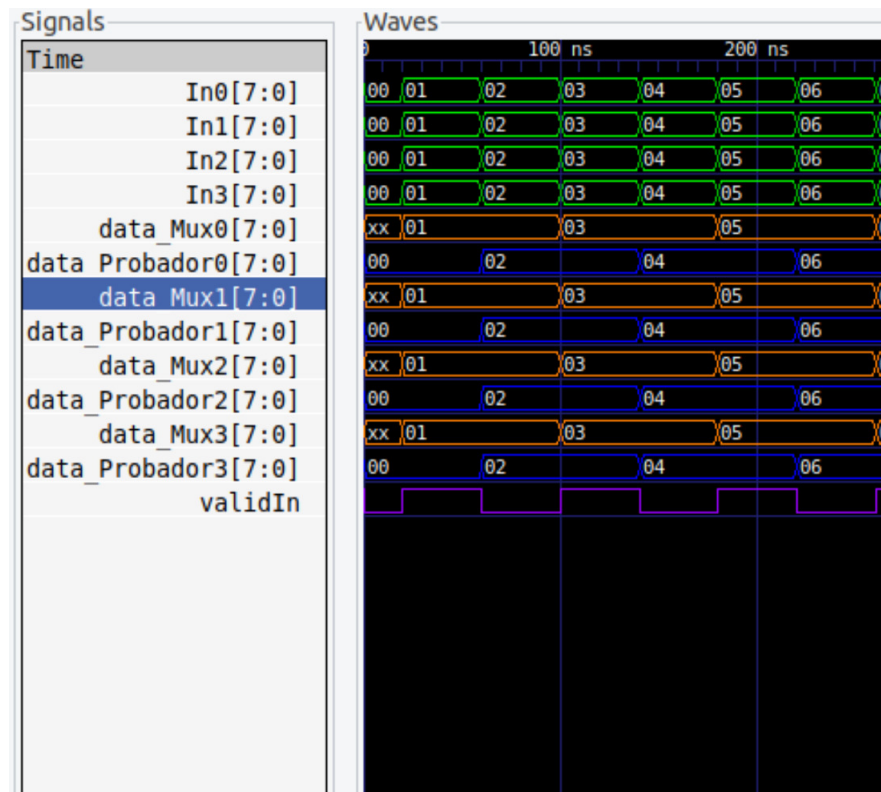


Figure 8: Diagrama de tiempos elaborado para el bloque de recirculación.