

Proyecto 1 - PHY PCle

Grupo 03:

Gokeh Ávila Blanco - B50747

David Campos Espinoza - B51479

Isaac Fonseca Segura - B52786

Andrés Vega Zamora - B57739

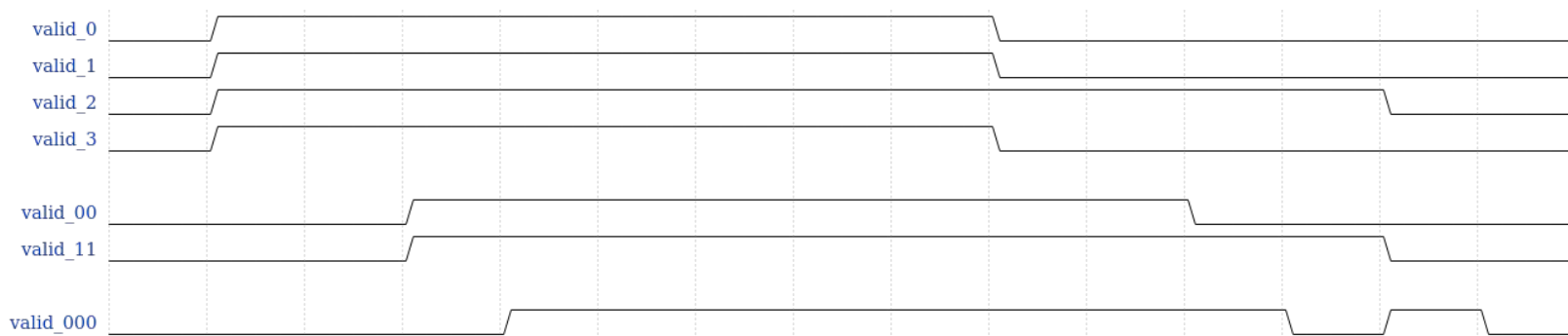


UNIVERSIDAD DE
COSTA RICA

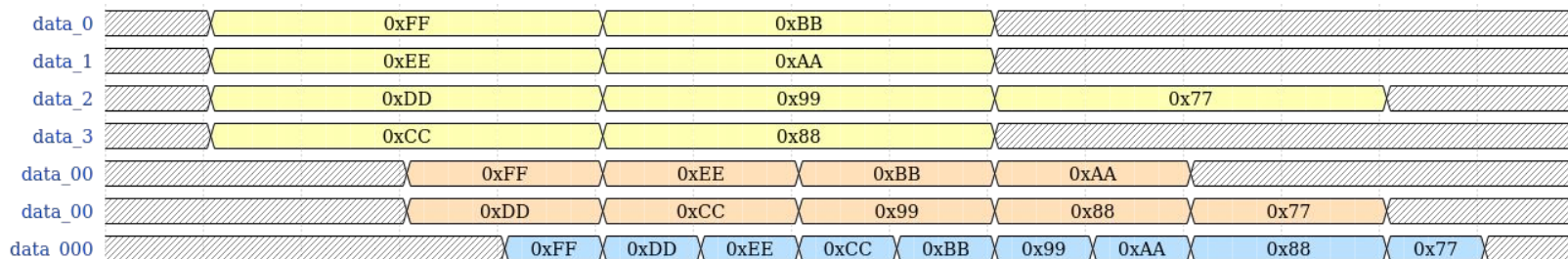
EIE

Problema: Lógica de multiplexores

¿Cómo propagar el valid de manera correcta entre los niveles de multiplexores?



¿Cómo asegurar el orden correcto de los datos en la salida?

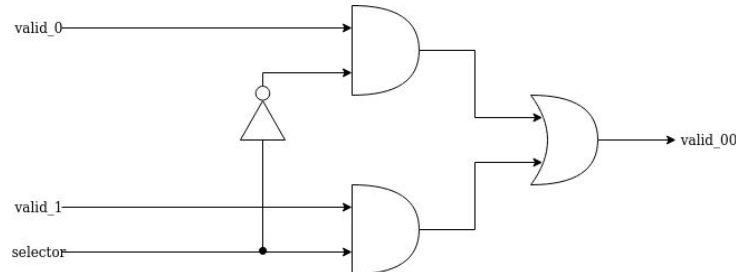


Solución: Selector automático

Tanto para propagar los valid de manera correcta, como para ordenar los datos de salida, la solución está en el selector automático.

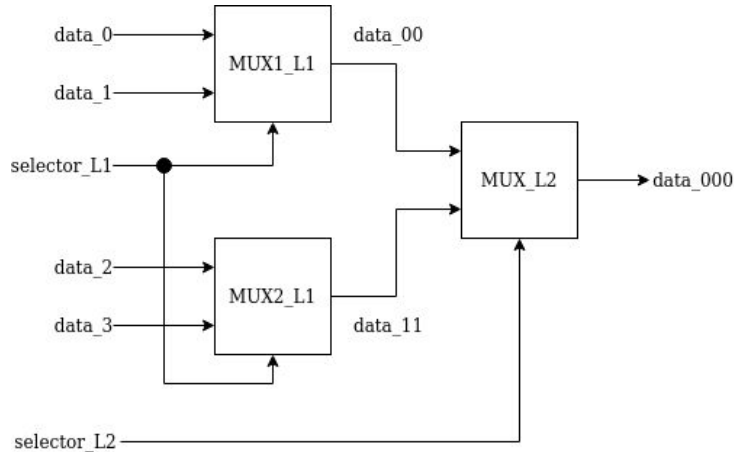
El selector automático alterna entre 0 y 1 al doble de la frecuencia de los datos de entrada del nivel de multiplexor.

La lógica del valid de salida depende del selector y de los valid de entrada. Simplemente propaga el valor del valid correspondiente al valor del selector.



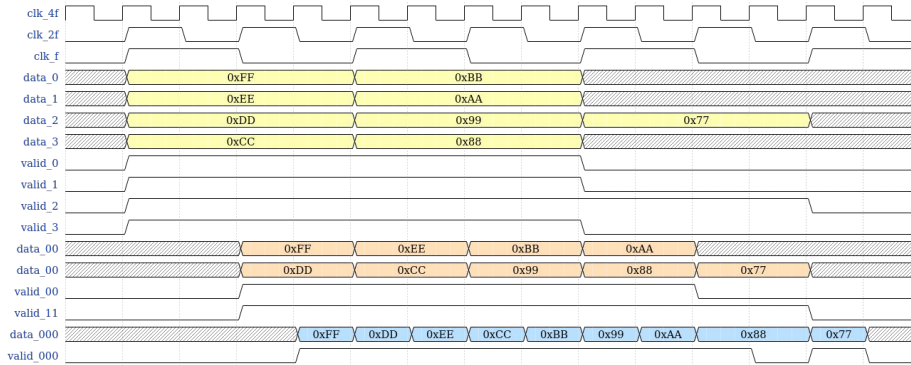
Solución: Selector automático

Con respecto al orden de los datos de salida, también dependen del selector automático y del valid de entrada.



El diagrama busca ilustrar la lógica de muxes y el flujo de datos. Sin embargo hay que considerar que los selectores son señales internas y que el valid correspondiente a cada entrada define si se actualiza el nuevo dato.

Demostración:

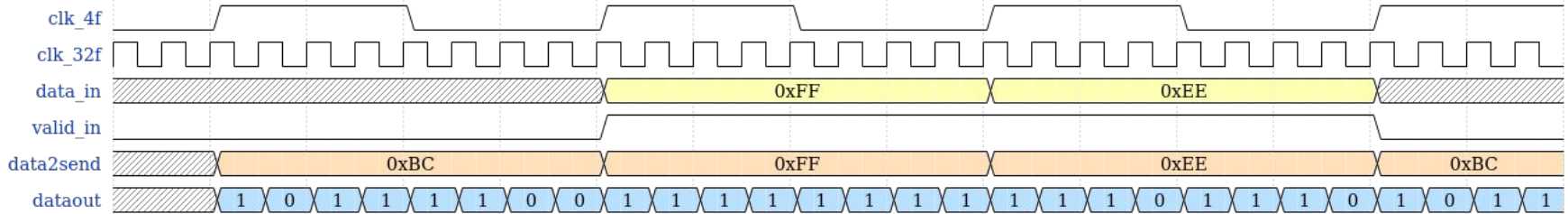


The diagram shows the timing for the first part of the test. It includes signals for **Datos entrada** (data_0, data_1, data_2, data_3, valid_0, valid_1, valid_2, valid_3, reset_L) and **Salida L1/ Entrada L2** (data_00, data_11, valid_00, valid_11). The data signals are shown as horizontal bars with values in hexadecimal. The valid signals are shown as green bars indicating when the data is valid. The reset_L signal is shown as a green bar indicating when the reset is active.

Signal	Value
data_0[7:0]	xx FF BB EA CC 15
data_1[7:0]	xx EE AA DE FF 20
data_2[7:0]	xx DD 99 77 55 33
data_3[7:0]	xx CC 88 66 44 22

Problema: Paralelo a serial (tx)

¿Cómo asegurar el orden correcto de los datos al pasar de bus de 8 bits a una señal serial?



Ejemplo:

- BC -> 10111100 (Correcto)
- BC -> 00111101 (Incorrecto)

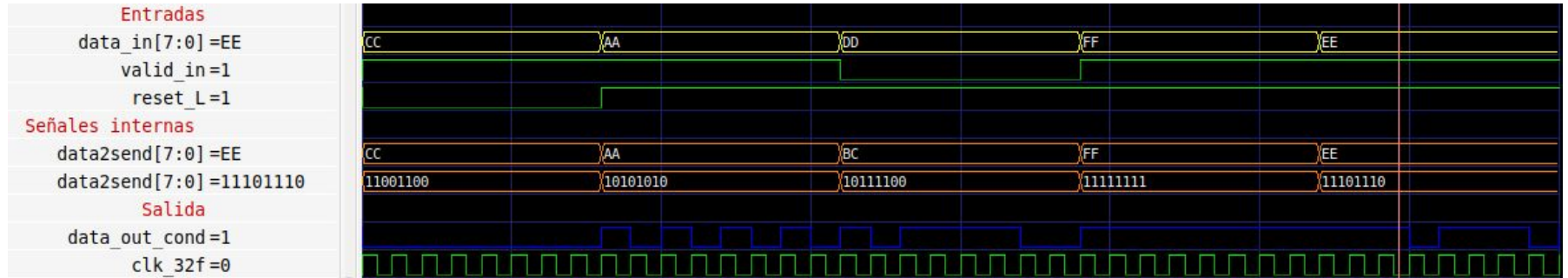
Solución: Contador automático de 3bits

El bloque de `paralelo_serial_tx` se puede describir también como un multiplexor de 8 a 1, con un selector automático.

La diferencia está en que para lograr que la salida serial vaya en el orden deseado, el valor del selector está dado por:

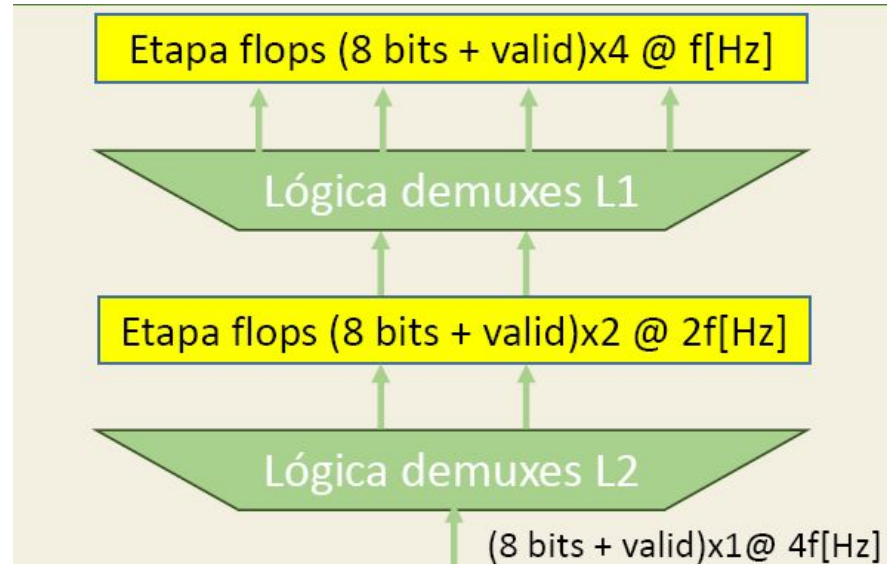
- `selector = 7-index`
- Donde el `index` corresponde a un contador de 3bits que incrementa su valor cada flanco creciente de `clk_32f`.

Demostración:

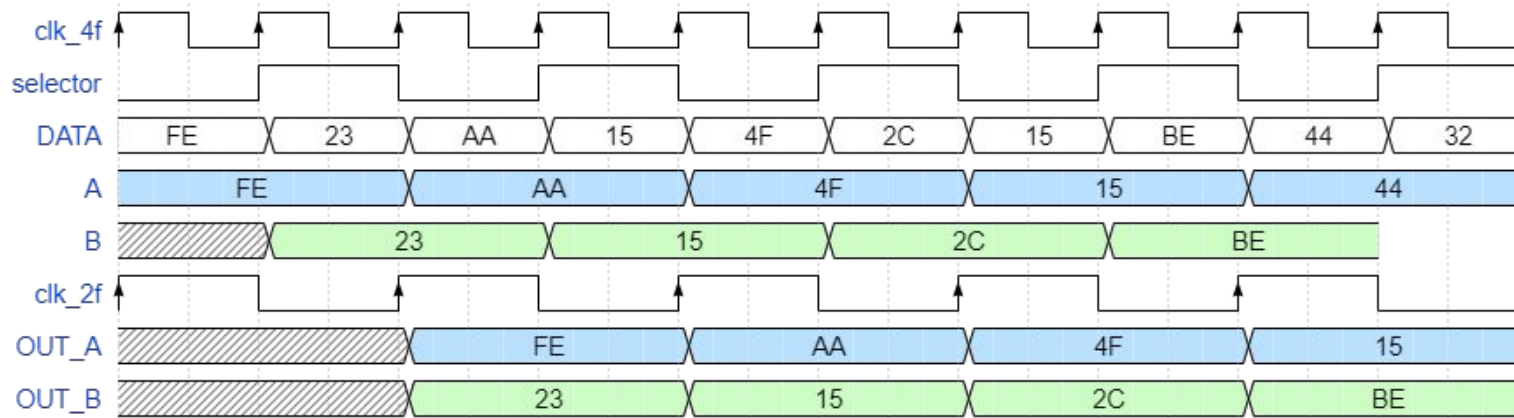


Problema: Lógica de demultiplexores

¿Cómo lograr convertir una entrada de datos en varias salidas sin dejar datos atrás?

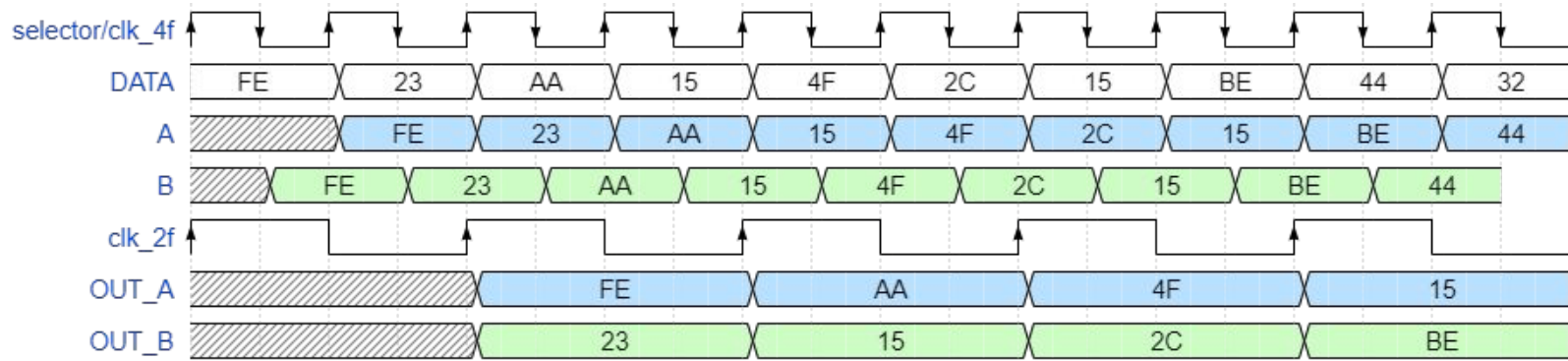


Solución 1 de la etapa de demultiplexores



Usar un selector que cambie en el flanco positivo del reloj de entrada. Luego guardar el dato hasta el siguiente flanco positivo del reloj de entrada para poder ser leído por el reloj de salida correctamente.

Solución 2 de la etapa de demultiplexores

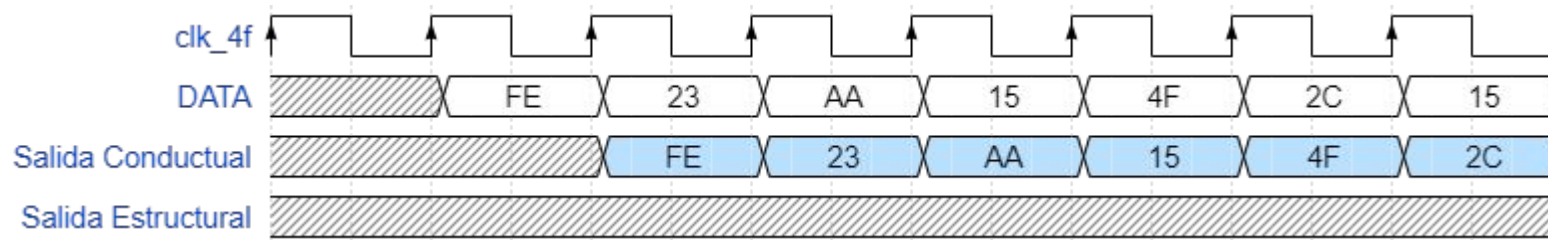


Usar ambos flancos del reloj de entrada de datos como selector permite acomodar los datos para poder capturarlos correctamente con el reloj de salida de datos.

Demostración

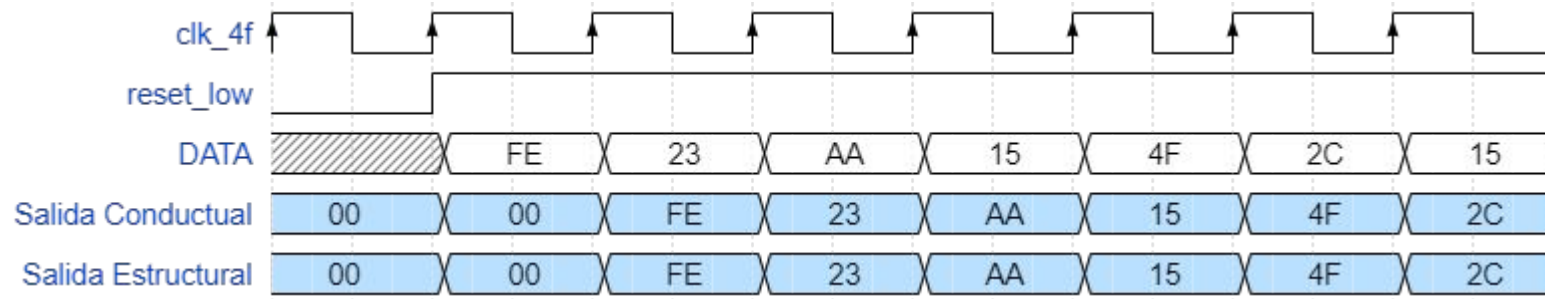


Problema: Síntesis de lógica secuencial



Cuando no hay reset el flip flop no conoce su estado anterior, por lo que si hay condición de memoria no sé sabe que se guardó. La salida se vuelve indeterminada.

Solución: Agregar una señal de reset



Al agregar una señal de reset el flip flop ahora sí conoce su estado anterior, por lo que podrá trabajar correctamente.

Espacio para consultas:

