

UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE0523 – Circuitos Digitales II

Proyecto 1. Plan de trabajo

Yeison Rodríguez, B56074
Pablo Vargas B57564
Luis Soto B57082

15/05/19

1. Cronograma de trabajo

El proyecto se divide por módulos y cada módulo será asignado a uno de los integrantes del equipo. En caso de que se tengan problemas en el diseño de algún módulo los demás integrantes ayudarían a resolver el problema.

Se espera cumplir con el siguiente cronograma de trabajo:

■ Avance 1, 15 de mayo:

- **Luis Soto:** Creación de módulos Generador de relojes, Lógica mux y lógica de demux
- **Pablo Vargas:** Creación del módulo Paralelo a serial
- **Yeison Rodríguez:** Seguimiento de bitácora y plan de trabajo. Ayudar con problemas en los módulos de trabajo

■ Avance 2, 22 de mayo:

- **Luis Soto:** Ayudar con problemas en los módulos de trabajo.
- **Pablo Vargas:** Creación de módulos Byte Striping y Un-striping
- **Yeison Rodríguez:** Creación de módulos Serial a Paralelo y Módulo phy_tx.v

■ Avance 3, 29 de mayo:

- **Luis Soto:** Aporte en reacción de Módulo phy_rx.v y módulo phy.v
- **Pablo Vargas:** Aporte en reacción de Módulo phy_rx.v y módulo phy.v
- **Yeison Rodríguez:** Aporte en reacción de Módulo phy_rx.v y módulo phy.v

■ Avance 4, 1° de junio: Todos los integrantes trabajarán en el reporte, específicamente en cada uno en los módulos que realizó

■ Presentación final: 5 de junio: Todos los integrantes trabajarán en la presentación final, utilizando la bitácora para describir los principales problemas que se tuvo.

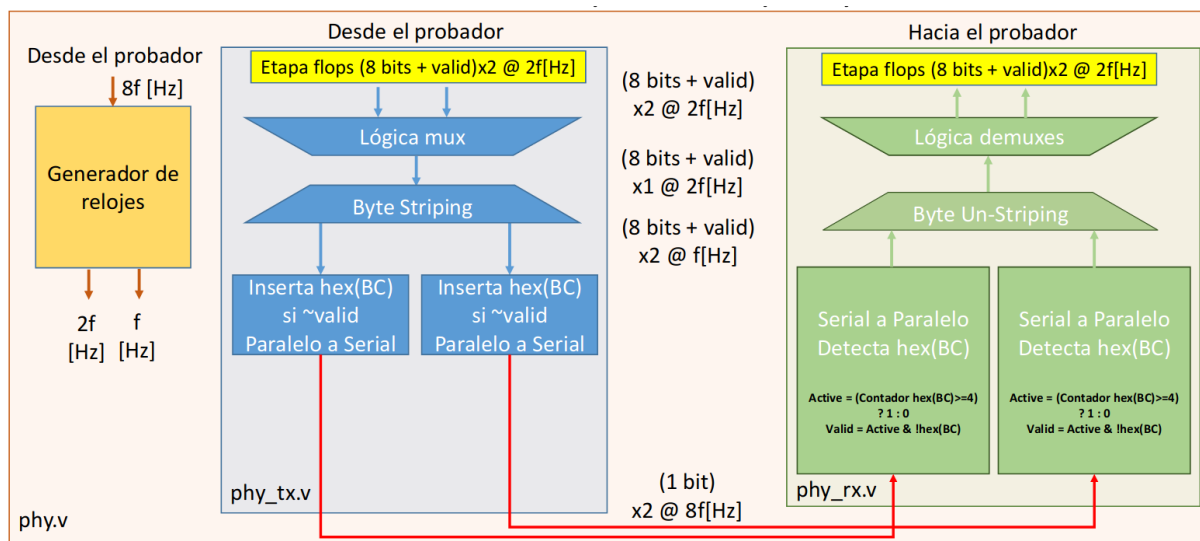


Figura 1: Diagrama de los módulos del proyecto

2. Plan de pruebas

2.0.1. clkgen

El generador de relojes debe tener como entrada un reloj de frecuencia 8f, y tener como salidas un reloj de frecuencia 2f y uno de f. Se realizarán pruebas como:

- Verificar que todos los relojes compartan al menos un flanco positivo en algún momento
- Verificar que el reloj de 2f cambie cada 2 ciclos de 8f, y el de f que cambie cada 4 ciclos de 8f

2.0.2. Lógica de mux y demux

El multiplexor debe pasar paquetes a su salida de un canal u otro, siguiendo varias condiciones listadas más adelante. Para la lógica del mux se le ingresarán varios paquetes buscando entradas específicas para que se cumplan las siguientes condiciones:

- Si se está recibiendo un paquete en un canal, el otro sea ignorado.
- Que pase el paquete completo y no se pierda información
- Si recibe dos paquetes al mismo tiempo, toma sólo uno, intercambia canal por cada paquete completo si vuelven a llegar dos juntos.
- Dejar dos ciclos libres entre paquetes para sincronización.

El demultiplexor debe mandar los paquetes a una de sus salidas a la vez, alternando conforme llegan más paquetes. En la prueba se debe cumplir con lo siguiente

- Que salgan todos los paquetes que entraran al multiplexor
- Los paquetes no deben salir por el mismo canal que ingresaron
- Se debe alternar las salidas del demux con la llegada de cada paquete

2.0.3. Paralelo a serial

Convierte el bus paralelo con reloj clk_fa una señal serial con reloj clk_8f. Las pruebas ingresadas buscarán que se cumpla con lo siguiente:

- Envía la señal paralela hex(BC) cuando valid está en bajo
- Refleja en su salida bit por bit de la entrada paralela

Ninguno de los datos ingresados podrá ser BC.

2.0.4. Serial a paralelo

Convierte datos seriales a paralelos. En las pruebas se buscará que se cumpla con lo siguiente:

- Al enviar 4 hex(BC), se habilita la señal active y éstase mantiene arriba
- La señal valid_out estará arriba siempre que active esté arriba y no se tenga un hex(BC) en los datos
-

2.0.5. Byte Un-striping

Recopila datos de dos carriles en uno solo. En las pruebas se verificará que

- La frecuencia de salida de los datos debe ser el doble de la frecuencia de entrada
- Se toman siempre primero los datos del lane0

2.0.6. Módulo phy_tx.v

Módulo que consiste en instancias de los módulos Lógica de mux, Byte striping y dos instancias de paralelo a serial. Se probará que se cumpla con lo siguiente

- Que la lógica de todas las etapas se cumpla de manera correcta, ya que podrían existir desfases entre los dispositivos a la hora de activarse
- que ambas salidas de datos estén propagando datos
- Que los paquetes que deberían salir, lo estén haciendo
- Que los pquetes que no deberían salir, no lo estén haciendo

2.0.7. Módulo phy_rx.v

Módulo que consiste en instancias de los módulos lógica de demux, Byte un Striping y dos intancias de serial a paralelo, en las pruebas se buscará que se cumpla:

- Que todos los paquetes que entren logren salir por el demux
- Que el demux esté sacando los paquetes en una salida y luego en la otra
- Que no existan desfases entre el funcionamiento de los componentes

2.0.8. Módulo phy.v

Este módulo consiste en instancias de los módulos Generador de relojes, phy_rx.v y phy_tx.v . Se buscará que cumpla con lo siguiente en las pruebas:

- Que todos los paquetes que entren y deban salir, lo hagan
- Que los paquetes que no deban salir, no lo hagan.

2.0.9. Uso de autoinst

El uso de autoins se realizará en los módulos más grandes. Será sencillo de utilizar ya que tenemos experiencia.