

# Ejercicio

## - Implementación de un sistema de comunicaciones básico

El objetivo es aplicar los conceptos de Python, Verilog y Aritmética de Punto Fijo aplicados en un sistema de comunicaciones básico.

### - Ejercicio 1

Diseñar un sistema de comunicaciones básico que contenga los bloques PRBS9, filtro transmisor, diezmador con selección de fase óptima de muestreo, contador de Bit Error Rate y módulo de Control tal como se observa en la Fig. 1.

#### ■ Características

- **Modulación:** QPSK
- **Frecuencia de Reloj:** 100MHz
- **Over Sampling Factor:** 4
- **Tipo de Filtro:** Raised Cosine (6 baudios)
- **Roll Off:** 0.5
- **Control:** Controla el funcionamiento a diferentes velocidades de los módulos del sistema.
- **BER:** El contador de BER debe realizar la etapa de inicialización cada vez que se cambia la fase de muestreo.
- **Seed PRBS9:** PRBS9I (9'h1AA) - PRBS9Q (9'h1FE).

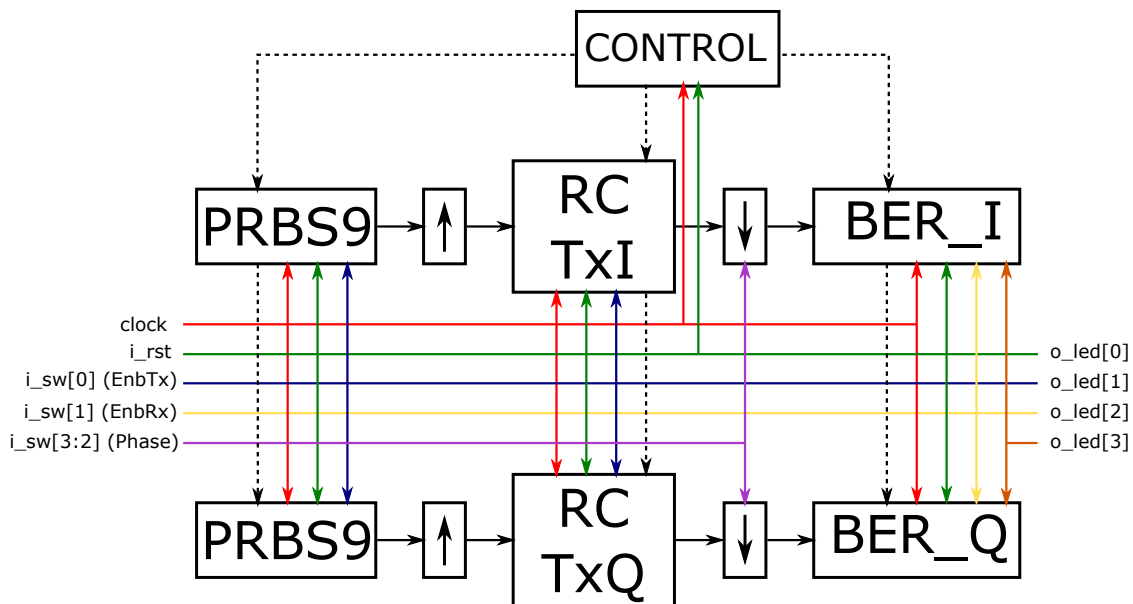


Figura 1: Diagrama en bloques del sistema de comunicaciones.

El diseño completo deberá seguir el siguiente flujo de trabajo

1. Diseñar en Python un **simulador en punto flotante** que contemple todo el diseño en donde la representación de la PRBS9 es una secuencia aleatoria y la estimación de la BER es una comparación vectores.
  - a) Realizar los siguientes gráficos
    - Bits transmitidos
    - Respuesta al impulso y frecuencia del filtro Tx.
    - Salida y diagrama de ojo del filtro Tx.
    - Diagrama de constelación a la salida del filtro Tx (por cada fase).
2. Diseñar un **simulador en punto fijo** partiendo del simulador anterior utilizar la librería *fixedInt.py*.
  - a) Definir las resoluciones de las diferentes etapas del diseño.
  - b) Para el caso de la PRBS9 y el contador de BER aplicar el concepto de diseño de hardware.
  - c) Buscar una metodología de trabajo para simular los retardos del sistema.
  - d) Utilizar el modelo para generar los archivos de estímulos para el Vector Matching (VM).
  - e) Realizar los siguientes gráficos
    - Bits transmitidos
    - Respuesta al impulso y frecuencia del filtro Tx.
    - Salida y diagrama de ojo del filtro Tx.
    - Diagrama de constelación a la salida del filtro Tx (por cada fase).
3. Implementar en Verilog el modelo de punto fijo.
  - a) Verificar el diseño utilizando TestBench, donde los estímulos se obtienen del modelo anterior.
4. Implementación en FPGA.
  - a) Aplicar el flujo completo de implementación con el fin de aplicar el diseño en FPGA.
  - b) El diezmado de la salida del filtro Tx se realizará con los switchs del kit
  - c) Para BER igual a cero (0) se prenderá un led.