

### **GUÍA MÓDULO 4: Circuitos de recuperación de reloj**

Al comunicar dos sistemas entre sí pasa a ser de importancia acordar los momentos de sincronización en el intercambio de datos: en forma amplia, el “reloj de los datos”. En algunos casos es posible transmitir ese reloj junto a los datos, en otros casos conviene embeber la señal de reloj junto a los datos, creando una nueva señal.

Este agregado (del lado transmisor) implica la necesidad de poder separar el reloj de los datos en el lado receptor (un “circuito de recuperación de reloj”).

En este módulo se analizarán los siguientes temas:

- Procesos básicos de recuperación de reloj, el caso de las UART
- Circuitos de recuperación de reloj en conexiones sincrónicas
- Uso de PLLs para recuperar reloj desde los datos
- Recuperación de reloj en enlaces de alta velocidad
- Circuitos CDR en enlaces de alta velocidad: data retiming
- Enlaces pseudo-sincrónicos: Bit Stuffing



#### **Actividad 4.1 (Entrega Obligatoria)**

1. Describa las técnicas de autoaprendizaje de la frecuencia de transmisión que se emplea en el protocolo LIN para disminuir los requisitos de tolerancia de frecuencia del receptor de una UART.
2. En base a la transparencia “Uso de PLLs para recuperar reloj desde los datos” y los bloques ya vistos, trate de imaginar y luego diseñe los bloques de un circuito que (operando a una frecuencia mucho más alta que la de los datos) reciba una señal sincrónica, y separe datos, reloj/enable de datos, y un múltiplo de ese reloj/enable, para su uso dentro de la lógica interna.
3. En BitStuffing la tasa efectiva de transferencia de datos depende de cuántos bits de “stuff” se agreguen. Proponga alguna forma de mejorar la tasa efectiva para aproximarla lo más posible a la tasa máxima, sin cambiar la longitud de secuencia de unos o ceros.
4. Describa cómo redefiniría el método de BitStuffing si se desea evitar largas secuencias de ‘0’s y ‘1’s ?



Utilice el recurso [Actividad 4.1](#) para enviar



### Actividad 4.2 (Entrega Obligatoria)

1. Implemente en VHDL los módulos codificador y decodificador de Bit-Stuffing incluyendo la inserción y detección de violaciones de código:
  - En el módulo codificador las señales serán: Datos, Reloj, SndSync y EnaTX.
  - El módulo decodificador operará con una señal de reloj de valor “aproximado” a 8 veces la del reloj transmisor (Reloj8x) y generará las señales Datos, Reloj, EnaRX, y Sync, sincrónicas con Reloj8x



Utilice el recurso [Actividad 4.2](#) para enviar

### Consideraciones finales

La combinación de este módulo con los previos habilita un enorme abanico de soluciones para procesos de comunicación entre sistemas, atendiendo desde los problemas de metaestabilidad a algunos de coordinación entre distintos dominios de reloj.

### Bibliografía sugerida:

- [http://en.wikipedia.org/wiki/Clock\\_recovery](http://en.wikipedia.org/wiki/Clock_recovery)
- [http://en.wikibooks.org/wiki/Clock\\_and\\_Data\\_Recovery/Structures\\_and\\_types\\_of\\_CDRs](http://en.wikibooks.org/wiki/Clock_and_Data_Recovery/Structures_and_types_of_CDRs)
- [http://en.wikipedia.org/wiki/Bit\\_stuffing](http://en.wikipedia.org/wiki/Bit_stuffing)
- <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1523394>