

Práctica 1. Lab2

Uso de Vivado: Síntesis e implementación

Objetivos

- Realizar el flujo de diseño con un ejemplo más complejo que el del *lab1*.
- Revisar con mayor nivel de detalle los conceptos de Síntesis e Implementación y, opcionalmente, ver el uso del depurador hardware.

Metodología

Este laboratorio está organizado en tres etapas, dos de ellas obligatorias y una tercera opcional. Los pasos a desarrollar están descritos en los siguientes documentos:

Lab2a - Synthesizing a RTL Design

Lab2b - Implementing the Design

Lab2c - Hardware Debugging

De cada uno de ellos deberás seguir los pasos y entender conceptualmente que se está haciendo. Como ayuda a la comprensión, tienes una serie de preguntas de cada apartado que no debes entregar, pero sí ser capaz de responderte.

Circuito utilizado

En este ejercicio se implementa sobre la placa *Zybo* un circuito que recibe por puerto serie (RS232) información serial del teclado, la paraleliza y visualiza en los leds de la placa. Dado que la placa tiene solo 4 leds, con el botón BTN0 se puede seleccionar la parte alta y baja del byte ASCII recibido. La figura 1 muestra el diseño.

El módulo *uart_rx_io* es el encargado de recibir la entrada serie y convertirla en 8 bits en paralelo.

Los módulos *meta_harden* minimizan la probabilidad de metaestabilidad en las señales asíncronas de entrada.

El módulo *led_ctrl* registra la entrada proveniente del puerto serie y en función del botón de entrada muestra la parte alta o baja del byte recibido.

En este diseño el top-level (*uart_led*) está descrito en Verilog. Existe una versión equivalente en VHDL (*uart_led2*) que podéis utilizar equivalentemente. El resto de los módulos están descritos en Verilog.

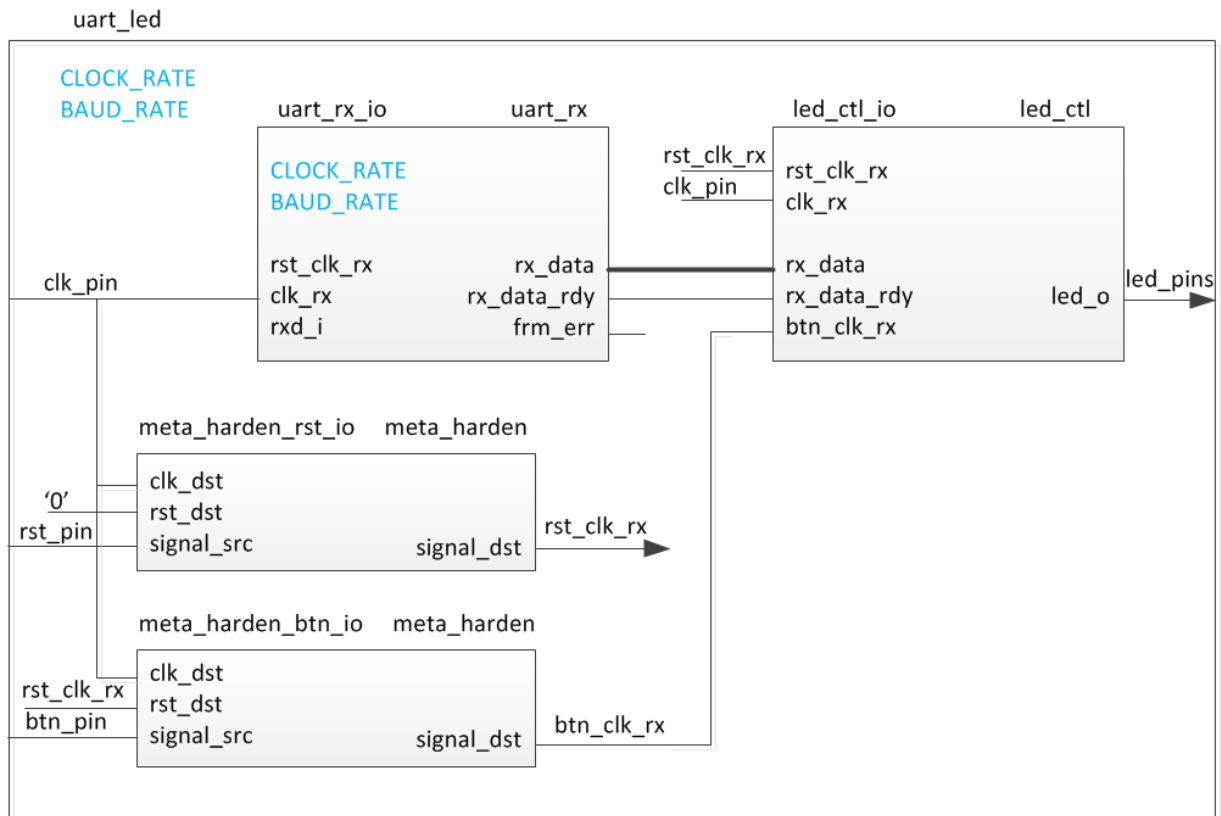


Figura 1. Diseño `uart_led` a implementar.

Tareas a realizar

1. Realizar los tutoriales Lab2a - Synthesizing a RTL Design y Lab2b - Implementing the Design. Opcionalmente realizar el Lab2c - Hardware Debugging.
2. Mostrar al profesor el resultado del punto anterior.
3. Escribir un módulo que se llame `led_controller.vhd`, descrito en VHDL y que realice la misma funcionalidad que el módulo `led_ctl`. Reemplazar el módulo `led_ctl.v` por `led_controller.vhd`. Reemplazar la ubicación de la señal `btn_pin` y ubicarlo en un switch de la placa en vez de un botón.
4. Mostrar al profesor el resultado del punto anterior.
5. Leer las preguntas y razonar las respuestas. Deberías ser capaz de responderte las preguntas. No debes entregar nada de este punto

Preguntas de cada parte:

Lab2a - Synthesizing a RTL Design:

- ¿Entiendes que hace el circuito? ¿Qué función realiza el módulo *uart_rx* ?
- ¿Qué se restringe en el fichero de restricciones *uart_led_timing.xdc*?
- ¿Cuándo se hace la elaboración del diseño (*elaborate design*)?
- ¿Qué significa hacer un análisis RTL?
- ¿Qué información ofrece la vista del diseño elaborado?
- ¿Qué significan los acrónimos FF, LUT, I/O y BUFG en el resumen del proyecto (*Project summary*)?
- ¿Por qué la vista del esquema tras la síntesis se ve distinta que en el diseño elaborado?
- ¿Qué elementos puedes identificar en el esquema post-synthesis?
- ¿Dónde sabremos si el circuito tiene violaciones de tiempo?
- ¿Dónde puedo obtener información detallada de cuantos recursos de bajo nivel requiere cada módulo del diseño?

Lab2b - Implementing the Design

- ¿Que realiza el proceso de “*run implementation*”?
- ¿Por qué hay diferentes valores en los informes tiempos (*timing report*) de la síntesis y de la implementación? ¿Qué valores son los más fiables?
- ¿Qué realiza la etapa “*generate bitstream*”?
- ¿Qué realiza la opción “*open target*” del menú “Program and Debug”?
- ¿Cómo te conectas al sistema para realizar las pruebas? ¿Cómo sabes que realmente funciona?

Lab2c - Hardware Debugging

- ¿Qué es el depurador Hardware?
- ¿Qué son los core ILA?
- ¿Qué nivel de observabilidad se posee con esta herramienta?
- ¿Cómo se pueden agregar puntas de prueba en un diseño RTL?