Universidad de Buenos Aires

Facultad de Ingeniería

Trabajo Práctico 4

Autor:
Franco Mariotti 102223
2do Cuatrimestre 2020

1 Introducción

El trabajo consistió en la utilización de la herramienta IP catalog para generar un recurso de temporización que permita generar una señal de 50MHz a partir del oscilador de 125MHz disponible en la FPGA. Este recurso generado luego fue incorporado al receptor serie asincrónico(UART) de la practica 2.

2 Modificación los archivos fuente para adaptarlos a la placa Arty-Z7.

Inicialmente cree el proyecto a partir de los archivos de la practica 2. Luego en el archivo de restricciones **uart_led_timing_ArtyZ7.xdc** cambie la frecuencia del clock del sistema a 125MHz con la siguiente linea:

```
create_clock -period 8.000 -name clk_pin -waveform {0.000 4.000} [get_ports clk_pin]
```

También cambie la frecuencia del reloj virtual con la siguiente linea:

```
create_clock -period 20.000 -name virtual_clock
```

Luego agregue el pin **txd_pin** al componente **UART_Top** el cual va conectado al pin de entrada **rxd_pin**.Para realizar esto agregue el siguiente código al archivo fuente **uart_top.vhd**

```
#En la sección declarativa de la arquitectura uart_top_arq
txd_pin: out std_logic; --Uart output
#En la seccion descriptiva de la arquitectura  uart_top_arq
loopback <= not rxd_pin;
txd_pin <= not loopback;

#En la sección descriptiva de la arquitectura  uart_top_arq
#Frecuencia de clock del componente uart_led.
CLOCK_RATE => 50E6
#conexión de la señal clk50 al pin de entrada de clock clk_pin del componente uart_led.
clk_pin => clk50
```

3 Agregado de IP del Catalogo

Para conseguir esto inicie el asistente de temporización **Clocking Wizard** el cual generara el subsistema de temporización. Configure la frecuencia del pin entrada **clk_in1** de dicho subsistema a 125MHz y posteriormente la frecuencia del pin de salida **clk_out1** a 50Mhz. Luego el asistente genero la plantilla de instaciado en VHDL del subsistema configurado, el cual incorpore al proyecto.

Figure 1: Plantilla de instaciado componente clk_wiz_0

C:/Users/Franco_M_32/Documents/sistemasDigitales/tps/Practica04/Practica04.ip_user_files/ip/clk_wiz_0/clk_wiz_0_stub.vhdl

```
1  -- Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
2  -- ------------------------------------------------------------------------------
3  -- Tool Version: Vivado v.2018.3 (win64) Build 2405991 Thu Dec  6 23:38:27 MST 2018
4  -- Date         : Wed Dec 16 13:13:17 2020
5  -- Host         : Franco running 64-bit major release  (build 9200)
6  -- Command      : write_vhdl -force -mode synth_stub
7  --                c:/Users/Franco_M_32/Documents/sistemasDigitales/tps/Practica04/Practica04.srcs/sou
8  -- Design       : clk_wiz_0
9  -- Purpose      : Stub declaration of top-level module interface
10 -- Device       : xc7z010clg400-1
11 -- ------------------------------------------------------------------------------
12 library IEEE;
13 use IEEE.STD_LOGIC_1164.ALL;
14
15 entity clk_wiz_0 is
16   Port (
17     clk_out1 : out STD_LOGIC;
18     reset : in STD_LOGIC;
19     locked : out STD_LOGIC;
20     clk_in1 : in STD_LOGIC
21   );
22
23 end clk_wiz_0;
24
25 architecture stub of clk_wiz_0 is
26 attribute syn_black_box : boolean;
27 attribute black_box_pad_pin : string;
28 attribute syn_black_box of stub : architecture is true;
29 attribute black_box_pad_pin of stub : architecture is "clk_out1,reset,locked,clk_in1";
30 begin
31 end;
32
```

Una vez incorporado el componente al proyecto, el siguiente paso fue instanciar dicho componente en el fuente **uart_top.vhd** la cual se consiguió con el siguiente código:

```
component clk_wiz_0 is
    Port (
        clk_out1 : out STD_LOGIC;
        reset : in STD_LOGIC;
        locked : out STD_LOGIC;
        clk_in1 : in STD_LOGIC
    );
end component;

#Declaración de la señal clk50.
signal clk50: std_logic;
```

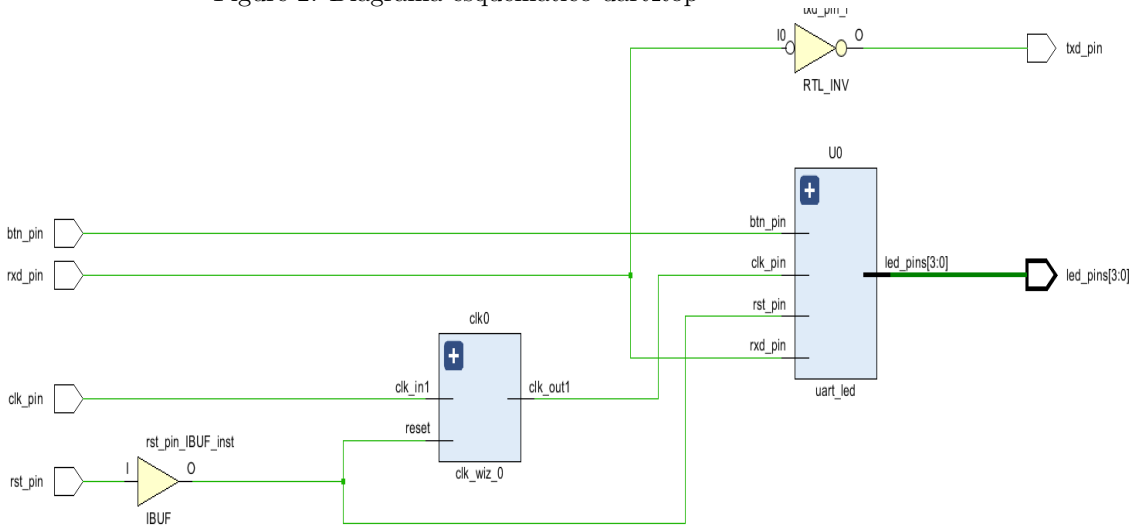
y posteriormente en la sección descriptiva de la arquitectura se definieron las conexiones de los pines del componente de la sigueinte manera:

```
clk0: clk_wiz_0
    port map(
        clk_out1 => clk50,
        reset => rst_pin,
        locked => open,
        clk_in1 => clk_pin
    );
```

Acá podemos observar que la señal clk50 se conecta a la salida clk_out1 del componte de temporización. A su vez la señal clk50 también se con anterioridad a la entrada clk_pin del componente uart_led

Finalmente genere el esquemático en diagramas de bloques del sistema completo el cual podemos ver en la siguiente imagen:

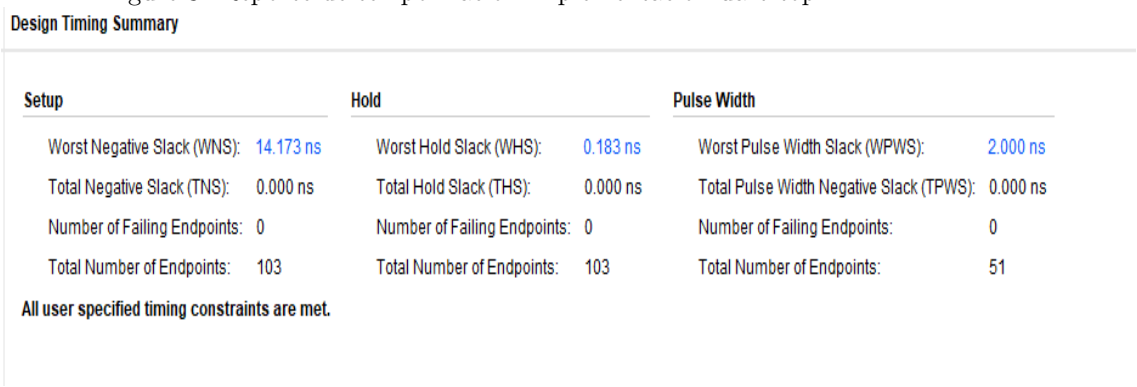
Figure 2: Diagrama esquemático uart_top



4 Implementación del Sistema

Para finalizar procedí a la implementación del sistema el cual genero el siguiente reporte de temporización:

Figure 3: Reporte de temporización implementación uart_top



Podemos observar que todas las restricciones de temporización impuestas se cumplen. Y por ultimo genere el **BitStream** para la posterior configuración de la FPGA.