

## Bitácora de Laboratorio 3

Roberto Sánchez Cárdenas - B77059

Gabriel Jiménez Amador - B73895

San José, 11 de julio de 2021

Laboratorio de Circuitos Digitales

### Resumen

Se realizaron implementaciones de hardware en una FPGA Nexys4, de modo que se aprendió a utilizar esta plataforma en conjunto con software Vivado. Se presenta un contador binario por medio de leds, contador con pantallas de 7 segmentos, un contador hexadecimal con pantallas de 7 segmentos y un alternador de cuentas en bases. Se suben videos sobre las implementaciones en FPGA a la plataforma Youtube.

### Índice

1. Introducción . . . . .	1
2. Correcciones realizadas . . . . .	2
3. Resultados y análisis . . . . .	4
3.1. Ejercicio 1 . . . . .	4
3.2. Ejercicio 2 . . . . .	4
3.3. Ejercicio 3 . . . . .	5
3.4. Ejercicio 4 . . . . .	5
4. Repositorio . . . . .	5
5. Conclusiones y recomendaciones . . . . .	5
5.1. Conclusiones . . . . .	5
5.2. Recomendaciones . . . . .	5

### 1. Introducción

A lo largo de este documento se presentan los resultados obtenidos de la implementación de una caché asociativa que políticas de reemplazo. En las secciones 2.1 y 2.2 se presenta una política de reemplazo aleatoria y en la secciones 2.3 y 2.4 se presenta la política LRU. en ambos casos se mantuvo habilitado la escritura write back.

## 2. Correcciones realizadas

Se tuvo que corregir el archivo de restricción de pines, debido a que estaban conectados de forma incorrecta con respecto al orden de los cátodos y ánodos físicamente en la tarjeta Nexys 4 DDR. El archivo corregido se encuentra en la Figura 1

```

set_property CFGBVS VCC0 [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]

# clk
set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }];
create_clock -add -name sys_clk_pin -period 10.00 [get_ports {clk}];

# switches
set_property -dict { PACKAGE_PIN V10 IOSTANDARD LVCMOS33 } [get_ports { resetn }];

# leds
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { out_byte[0] }];
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { out_byte[1] }];
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { out_byte[2] }];
set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33 } [get_ports { out_byte[3] }];
set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports { out_byte[4] }];
set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports { out_byte[5] }];
set_property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 } [get_ports { out_byte[6] }];
set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 } [get_ports { out_byte[7] }];

set_property -dict { PACKAGE_PIN V12 IOSTANDARD LVCMOS33 } [get_ports { out_byte_en }];
set_property -dict { PACKAGE_PIN V14 IOSTANDARD LVCMOS33 } [get_ports { trap }];

#7 segment display
# cathodes
set_property -dict { PACKAGE_PIN H15 IOSTANDARD LVCMOS33 } [get_ports { c_out[7] }];
set_property -dict { PACKAGE_PIN L18 IOSTANDARD LVCMOS33 } [get_ports { c_out[6] }];
set_property -dict { PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports { c_out[5] }];
set_property -dict { PACKAGE_PIN P15 IOSTANDARD LVCMOS33 } [get_ports { c_out[4] }];
set_property -dict { PACKAGE_PIN K13 IOSTANDARD LVCMOS33 } [get_ports { c_out[3] }];
set_property -dict { PACKAGE_PIN K16 IOSTANDARD LVCMOS33 } [get_ports { c_out[2] }];
set_property -dict { PACKAGE_PIN R10 IOSTANDARD LVCMOS33 } [get_ports { c_out[1] }];
set_property -dict { PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports { c_out[0] }];

# anodes
set_property -dict { PACKAGE_PIN U13 IOSTANDARD LVCMOS33 } [get_ports { an_out[7] }];
set_property -dict { PACKAGE_PIN K2 IOSTANDARD LVCMOS33 } [get_ports { an_out[6] }];
set_property -dict { PACKAGE_PIN T14 IOSTANDARD LVCMOS33 } [get_ports { an_out[5] }];
set_property -dict { PACKAGE_PIN P14 IOSTANDARD LVCMOS33 } [get_ports { an_out[4] }];
set_property -dict { PACKAGE_PIN J14 IOSTANDARD LVCMOS33 } [get_ports { an_out[3] }];
set_property -dict { PACKAGE_PIN T9 IOSTANDARD LVCMOS33 } [get_ports { an_out[2] }];
set_property -dict { PACKAGE_PIN J18 IOSTANDARD LVCMOS33 } [get_ports { an_out[1] }];
set_property -dict { PACKAGE_PIN J17 IOSTANDARD LVCMOS33 } [get_ports { an_out[0] }];

# switch
set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { base_sel }];

```

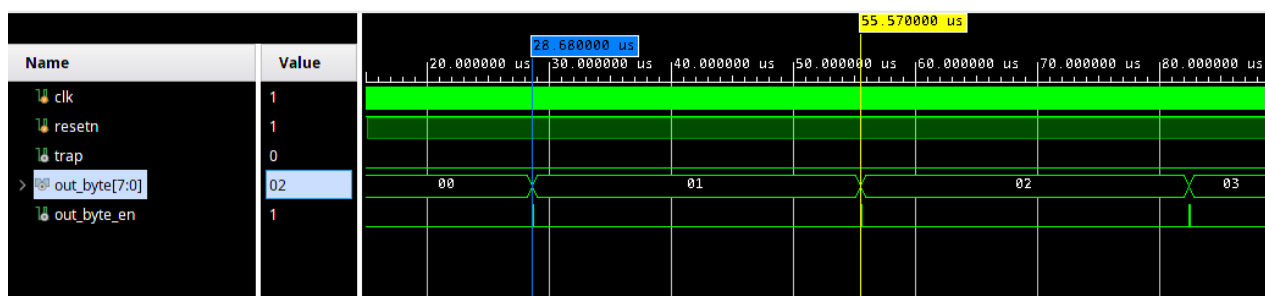
**Figura 1:** Archivo de restricciones de localización de pines corregido.

Además se trabajó la conversión de binario a BCD de una manera distinta. Se utilizó un tipo de desplazamiento de decimal, de modo que se puede extraer cada uno de los dígitos decimales de un número y realizar la actualización de cada pantalla.

### 3. Resultados y análisis

#### 3.1. Ejercicio 1

Utilizando el firmware base provisto por el profesor, se procedió a calcular el periodo al que funciona la simulación.



**Figura 2:** Periodo de las salidas de firmware original

De puede observar que el tiempo es aproximado a:

$$T = 26.89 \mu s$$

Por regla de 3, podemos hallar una equivalencia del tiempo de espera original respecto a un nuevo tiempo de espera para otro tiempo

$$\text{LOOP\_WAIT\_LIMIT} = \frac{T}{26,89 \cdot 10^{-6}} \cdot 100$$

Por lo tanto, para un tiempo de 100ms, se obtendría LOOP\_WAIT\_LIMIT=371885

La implementación del diseño se puede visualizar en el siguiente enlace:

<https://youtu.be/P3Ne2CIAem0>

#### 3.2. Ejercicio 2

Para este ejercicio se trajo una pequeña máquina de estados que se encarga de aislar cada uno de los dígitos de un número por medio de divisiones enteras y módulos. Se logró implementar el contador hasta un valor máximo de 99999999. Los resultados se pueden ver en el siguiente enlace de Youtube.

<https://youtu.be/qb7S7mX0KoM>

### 3.3. Ejercicio 3

En este ejercicio se implementó un contador hexadecimal en la pantalla 7 segmentos de la tarjeta Nexys 4 DDR. La implementación del diseño se puede visualizar en el siguiente enlace:

<https://youtu.be/fq7rHudYh9U>

### 3.4. Ejercicio 4

En este ejercicio se permitió alternar entre bases del contador desplegado en los *displays* de 7 segmentos variando el valor del SW0 (pin J15) de la tarjeta Nexys 4 DDR.

- J15 en 1: cuenta en decimal
- J15 en 0: cuenta en hexadecimal

Cuando se alterna la base no se modifica la cuenta, como se observa en el video en este enlace:

<https://youtu.be/tz3-YKI8nUg>

## 4. Repositorio

El código necesario para emular los resultados de este laboratorio se pueden encontrar en Github utilizando el siguiente enlace:

<https://github.com/ucr-ie-0424/laboratorio-3-jimenez-sanchez>

El ID del último commit es: f5868f8b608e5e9722bf990a2d744488b5d62136

## 5. Conclusiones y recomendaciones

### 5.1. Conclusiones

- Se pudo experimentar por primera vez sintetizando e implementando para la tarjeta Nexys 4 DDR en físico, a diferencia de los laboratorios anteriores que sólo se simulaba.
- Se verificó el diseño RTL en hardware así como la correlación entre el I/O del diseño con los pines físicos según lo dictado en el archivo de restricción de pines.

### 5.2. Recomendaciones

- Se recomienda verificar el archivo de restricción de pines, antes que concentrarse en verificar los archivos RTL, si la simulación por comportamiento parece funcionar bien, es probable que el error esté en este primer archivo.