

# Informe del Trabajo Práctico Final

Nombre del Alumno: Ing. José Roberto Castro

10 de diciembre de 2024

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Breve Explicación de lo Implementado</b>	<b>2</b>
2.1. Descripción del Contador en VHDL . . . . .	2
2.2. Aplicación en C . . . . .	3
<b>3. Diagramas en Bloques del Circuito</b>	<b>4</b>
<b>4. Captura de Simulación</b>	<b>4</b>
<b>5. Tabla de Uso de Recursos de la FPGA</b>	<b>4</b>
<b>6. Conclusión</b>	<b>5</b>

# 1. Introducción

El presente documento detalla la implementación de un bloque de hardware digital desarrollado como parte del trabajo práctico final de la materia Microarquitecturas y Softcores. El objetivo fue implementar un contador ascendente y descendente en VHDL, simular su comportamiento, sintetizarlo para una FPGA y establecer su conectividad con un microprocesador mediante una aplicación en lenguaje C.

## 2. Breve Explicación de lo Implementado

El sistema consta de dos componentes principales:

- **Bloque VHDL:** Un contador configurable para operar en modo ascendente o descendente.
- **Aplicación en C:** Un programa que interactúa con el bloque lógico a través del bus AXI, permitiendo configurar el modo de operación y leer el valor del contador.

### 2.1. Descripción del Contador en VHDL

El siguiente fragmento muestra el código del contador implementado en VHDL. Este bloque incluye una señal de control para definir el modo (ascendente o descendente) y una señal de reset para inicializar el conteo.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity contador_asc_desc is
7     Port (
8         clk : in STD_LOGIC;
9         reset : in STD_LOGIC;
10        control : in STD_LOGIC; -- '0' para ascendente, '1' para
            descendente
11        count_out : out STD_LOGIC_VECTOR(3 downto 0)
12    );
13 end contador_asc_desc;
14
15 architecture Behavioral of contador_asc_desc is
16     signal count : STD_LOGIC_VECTOR(3 downto 0) := "0000";
17 begin
18     process(clk, reset)
19     begin
20         if reset = '1' then
21             count <= "0000";
22         elsif rising_edge(clk) then
23             if control = '0' then
24                 count <= count + 1;
25             else
26                 count <= count - 1;
27             end if;
```

```

28     end if;
29 end process;
30
31     count_out <= count;
32 end Behavioral;

```

Listing 1: Código VHDL del contador ascendente/descendente

El comportamiento del contador está determinado por:

- **Señal de reloj (clk):** Sincroniza las operaciones del contador.
- **Señal de reset:** Reinicia el contador a cero.
- **Señal de control:** Define el modo de operación:
  - '0': Modo ascendente.
  - '1': Modo descendente.

## 2.2. Aplicación en C

La aplicación en C interactúa con el bloque VHDL a través del bus AXI. Permite configurar el modo del contador y leer su valor en tiempo real.

```

1 #include "xparameters.h"
2 #include "xil_io.h"
3
4 #define CONTADOR_BASE_ADDR XPAR_CONTADORIP_0_S00_AXI_BASEADDR
5 #define REG_RESET_OFFSET 0x00
6 #define REG_CONTROL_OFFSET 0x04
7 #define REG_COUNT_OUT_OFFSET 0x08
8
9 void Contador_SetReset(int reset) {
10     Xil_Out32(CONTADOR_BASE_ADDR + REG_RESET_OFFSET, reset);
11 }
12
13 void Contador_SetControl(int control) {
14     Xil_Out32(CONTADOR_BASE_ADDR + REG_CONTROL_OFFSET, control);
15 }
16
17 int Contador_ReadCount() {
18     return Xil_In32(CONTADOR_BASE_ADDR + REG_COUNT_OUT_OFFSET) & 0xF;
19 }

```

Listing 2: Fragmento del código C para interactuar con el bloque VHDL

La funcionalidad de esta aplicación incluye:

- Configuración del modo (ascendente o descendente) del contador.
- Lectura del valor actual del contador.
- Generación de un retardo simulado para observar los cambios en el conteo.

### 3. Diagramas en Bloques del Circuito

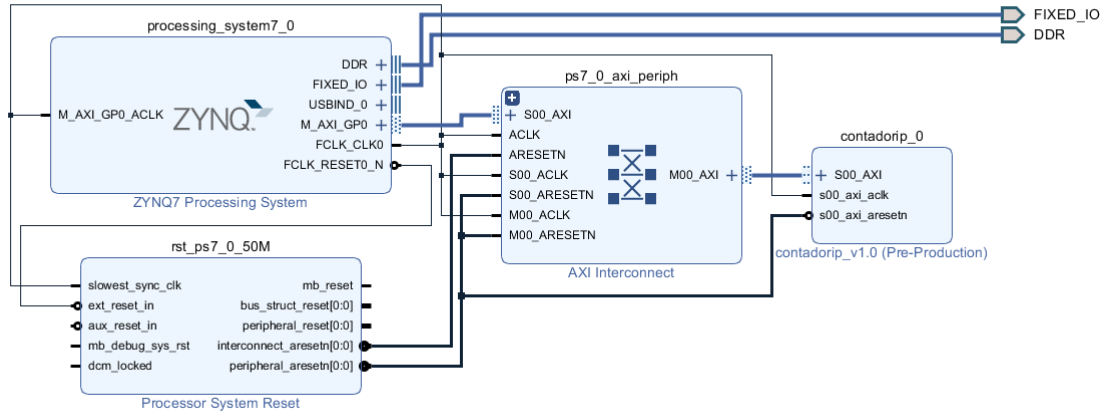


Figura 1: Diagrama en bloques del sistema implementado.

### 4. Captura de Simulación

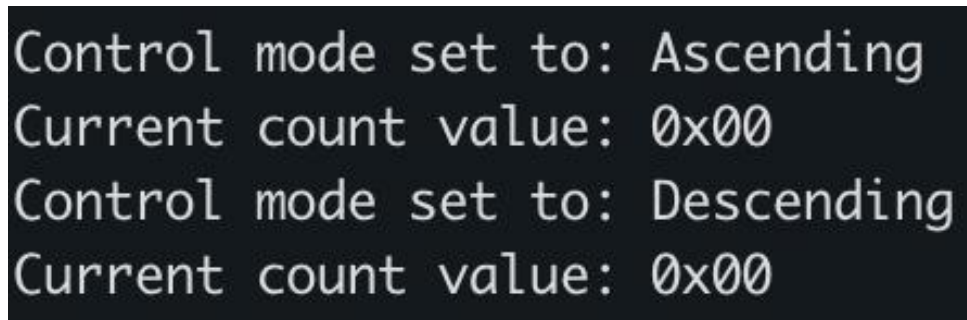


Figura 2: Simulación del contador en modo ascendente y descendente.

### 5. Tabla de Uso de Recursos de la FPGA

A continuación se presentan los recursos utilizados en la FPGA para el diseño implementado:

Cuadro 1: Recursos detallados por módulo

Nombre	Slice LUTs	Slice Registers	Bonded IOPADs	BUFGCTRL
design_1.wrapper	608	809	130	1
design_1.i	608	809	0	1
contadorip_0	57	169	0	0
processing_system7_0	112	0	0	0
ps7_0_axi_periph	420	600	0	0
rst_ps7_0_50M	19	40	0	0

Cuadro 2: Resumen de utilización de recursos			
Recurso	Utilización	Disponible	Porcentaje de utilización
LUT	608	17600	3.45 %
LUTRAM	66	6000	1.10 %
FF	809	35200	2.30 %

## 6. Conclusión

El desarrollo del contador ascendente y descendente permitió comprender los conceptos fundamentales del diseño de hardware digital y la interacción entre bloques lógicos y sistemas embebidos. La implementación en FPGA y la conectividad con el Cortex demostraron ser efectivos para alcanzar los objetivos planteados.

## Referencias

- Código fuente VHDL y C: Anexado en la entrega.