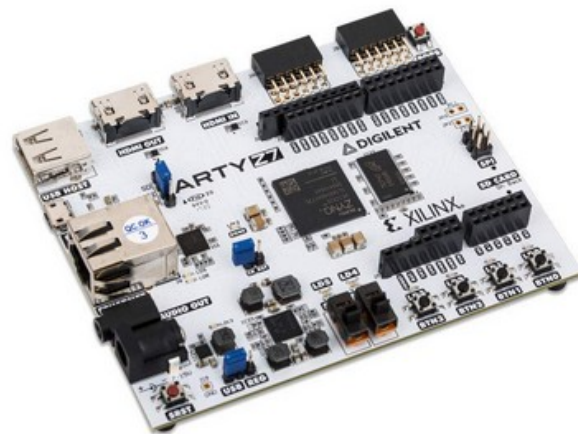


Microarquitecturas y Softcores

Carrera de Especialización en Sistemas Embebidos

Trabajo final
Raúl Romero



Contador BCD

Características principales

- Comandos por micro: “a”, “d”, “p” y “r”
- Frecuencia de actualización de la cuenta 1Hz

Diagrama en bloques

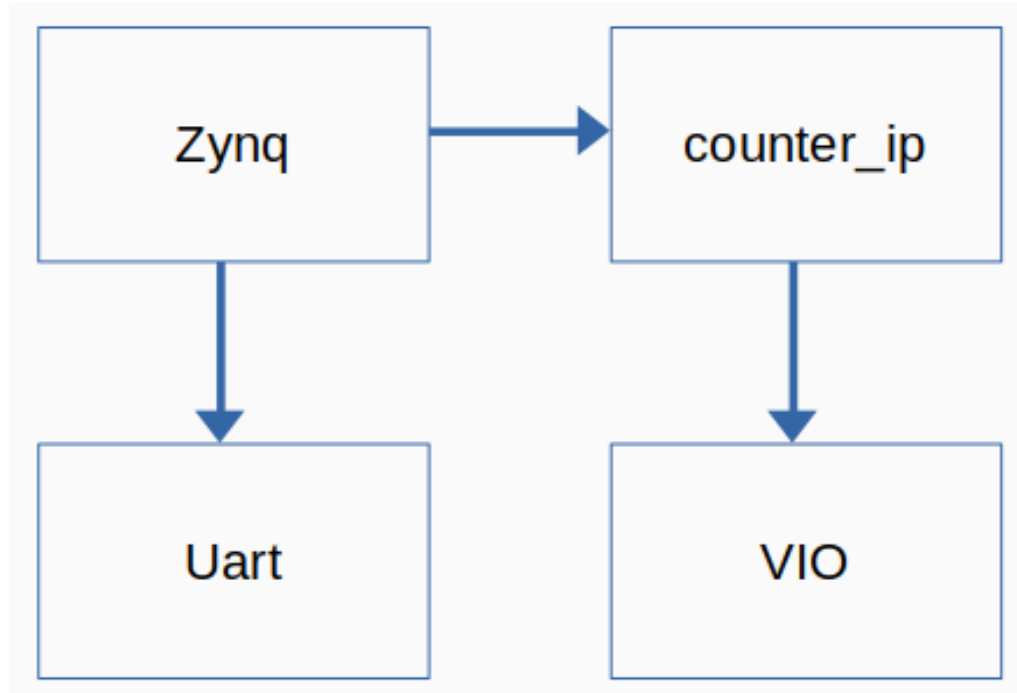


Diagrama en bloques CLP

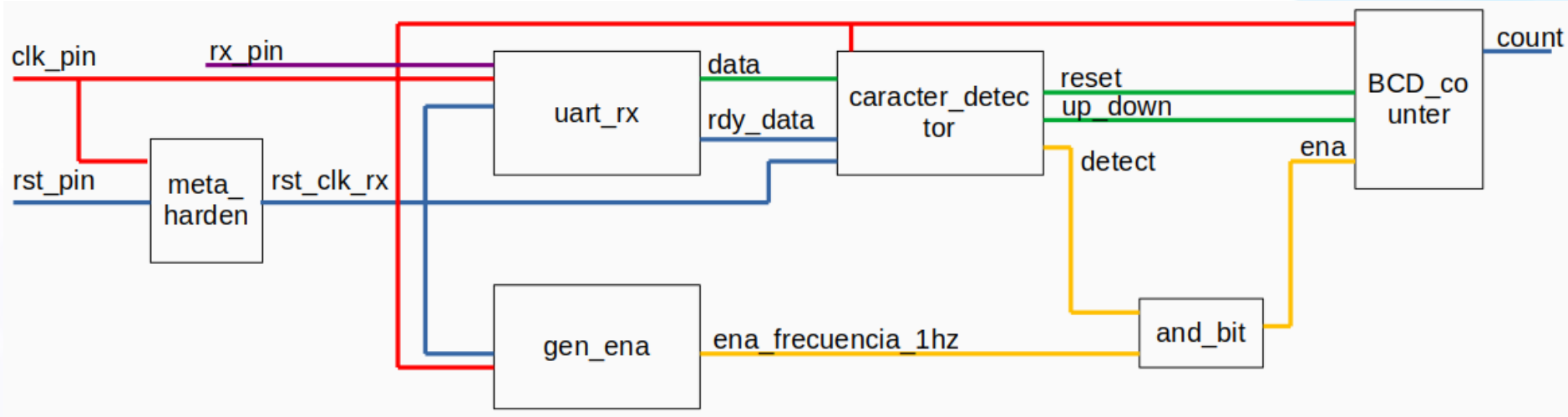


Diagrama en bloques MyS (counter_ip)

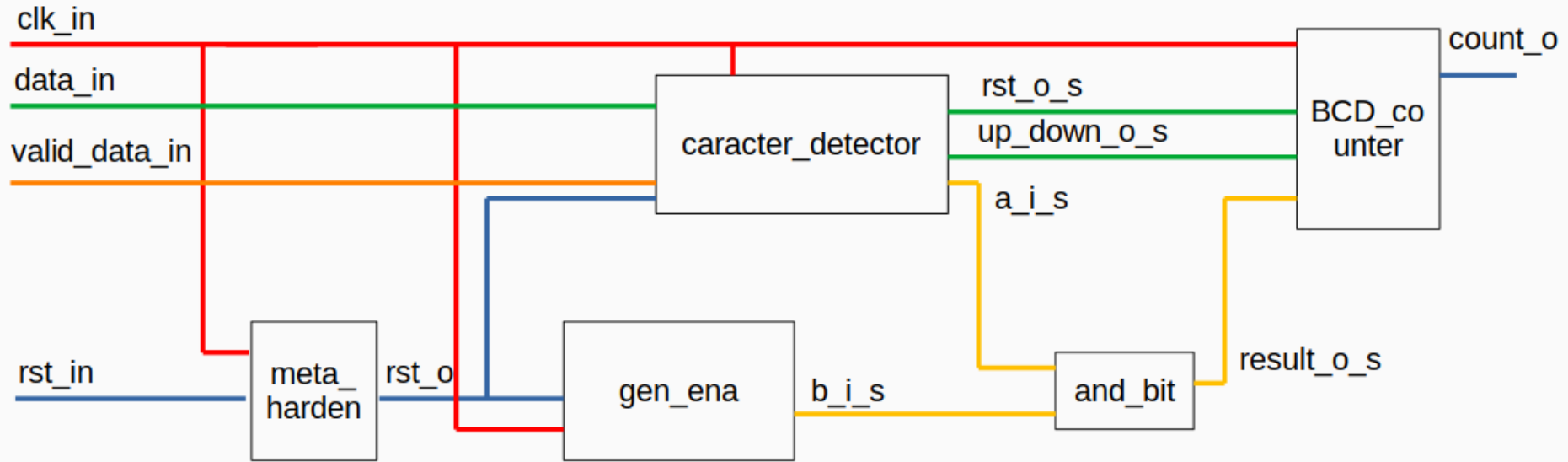


Diagrama en bloques vivado

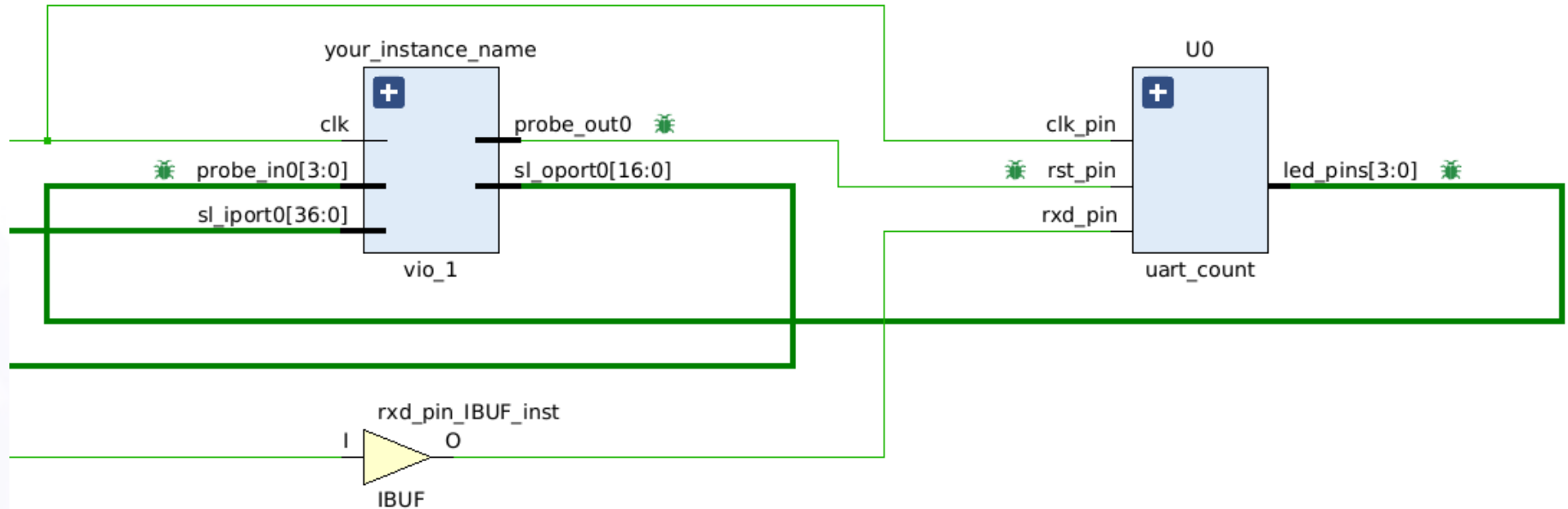
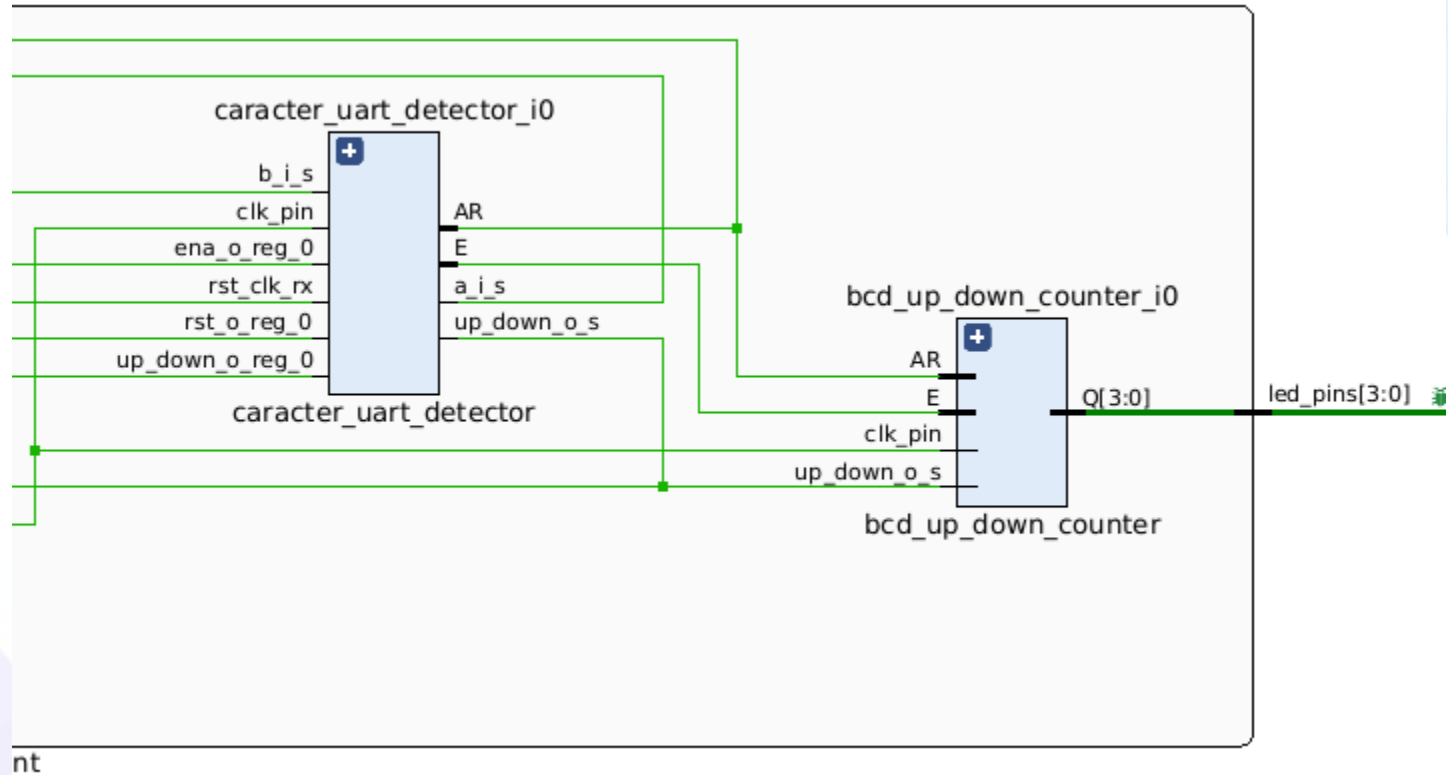
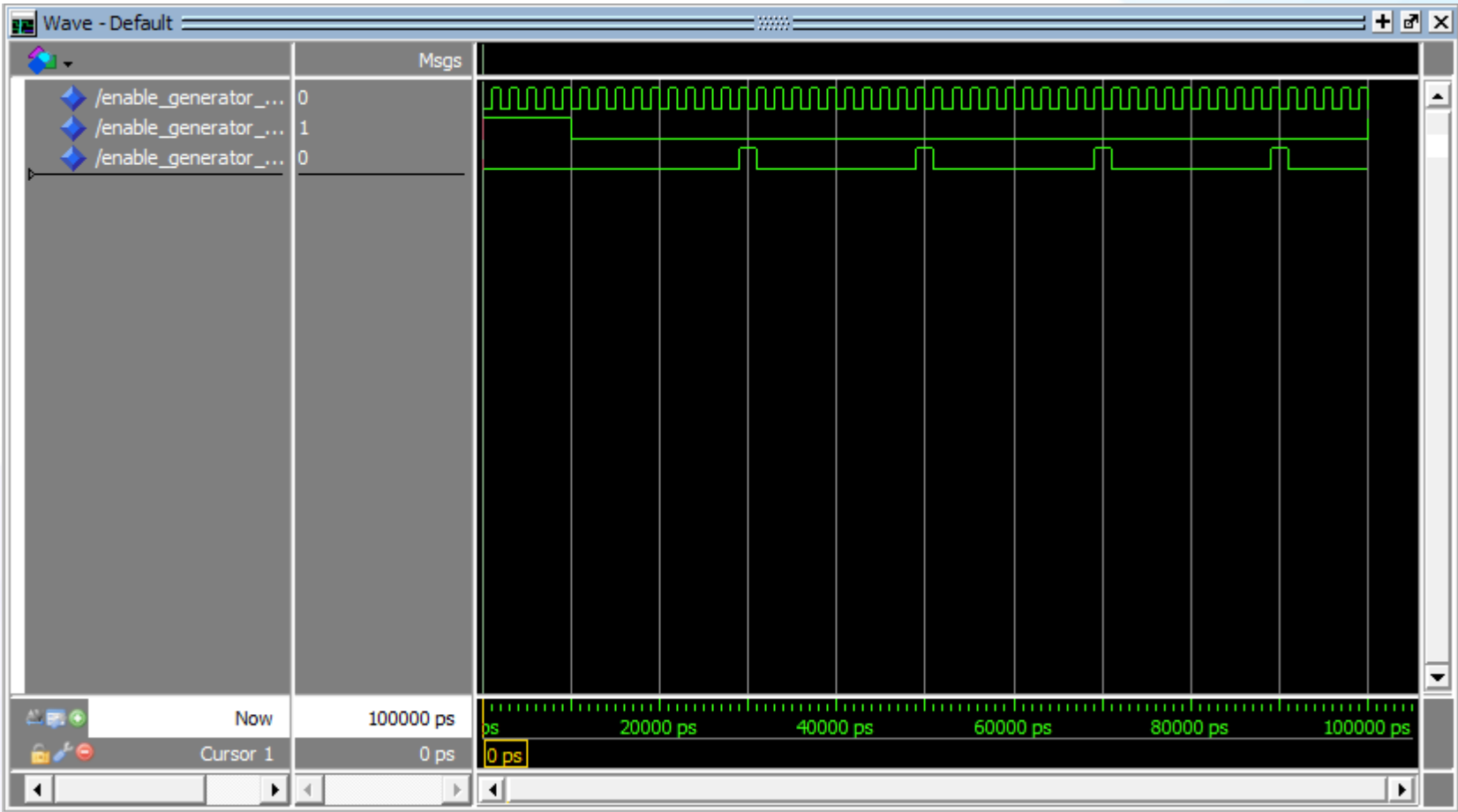


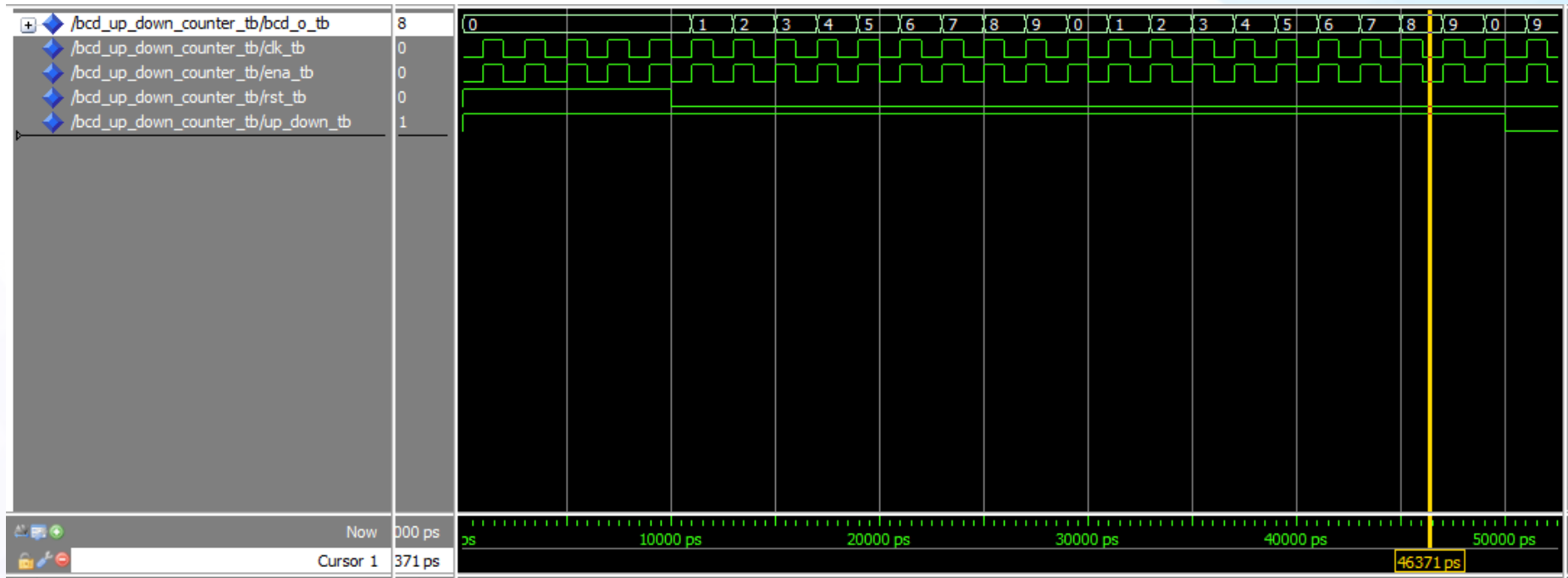
Diagrama en bloques vivado



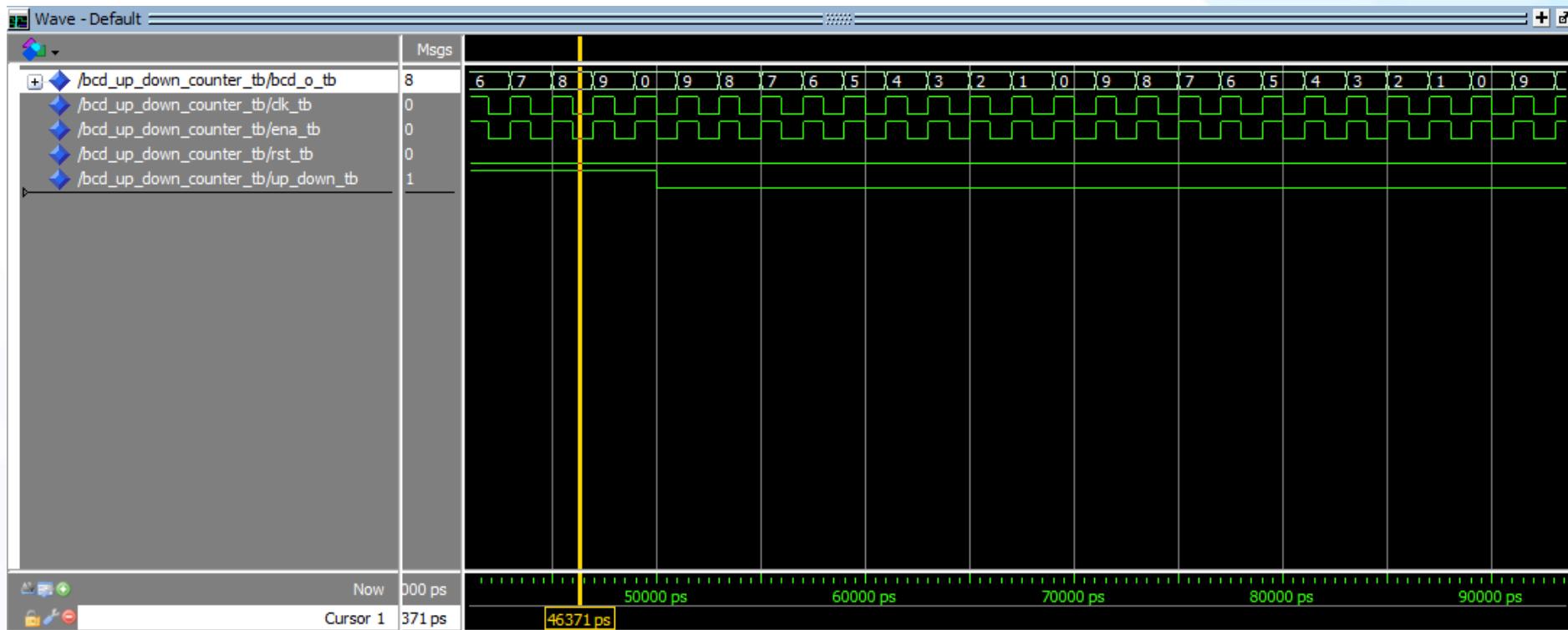
Generador de habilitación



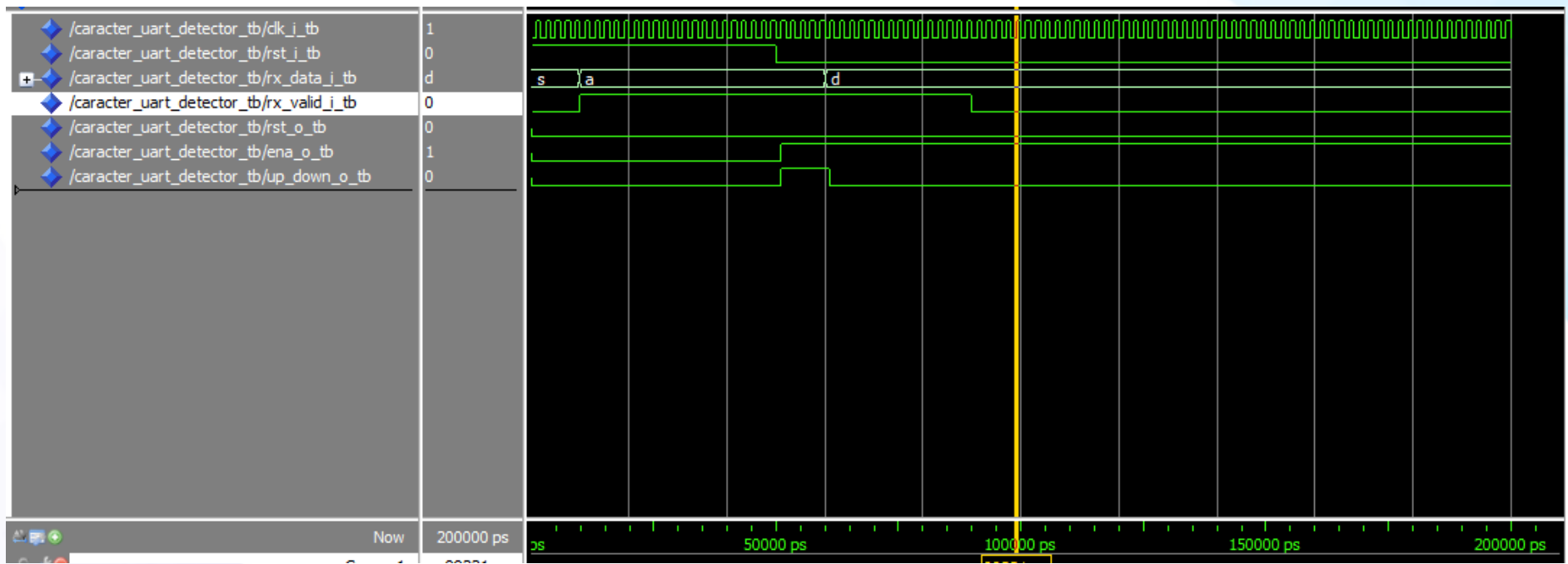
bcd_up_down_counter



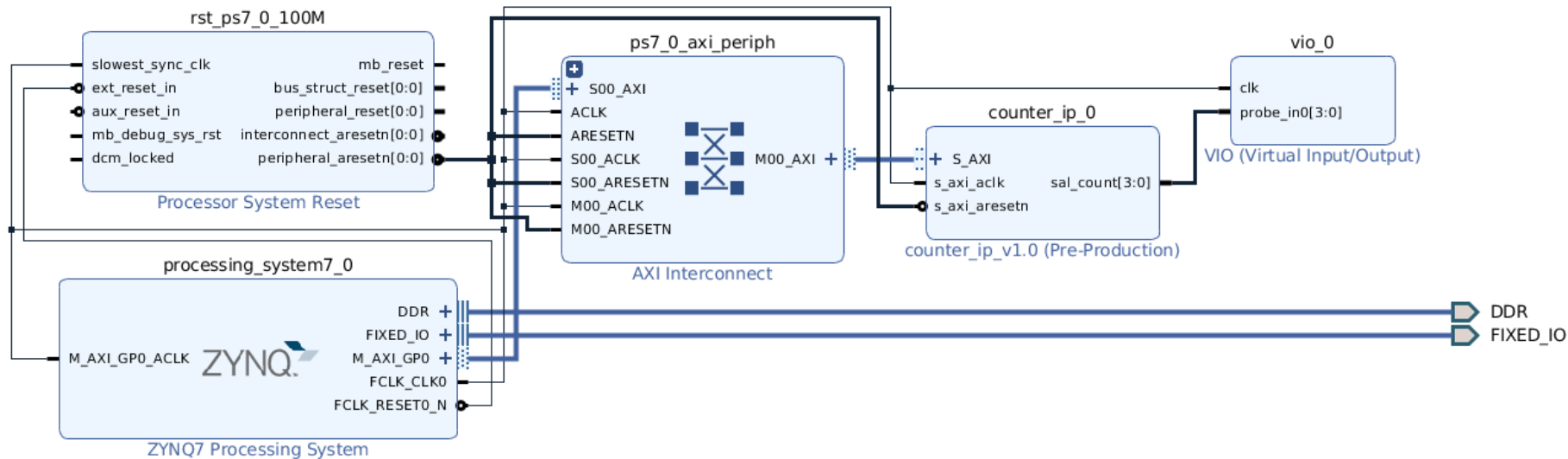
bcd_up_down_counter



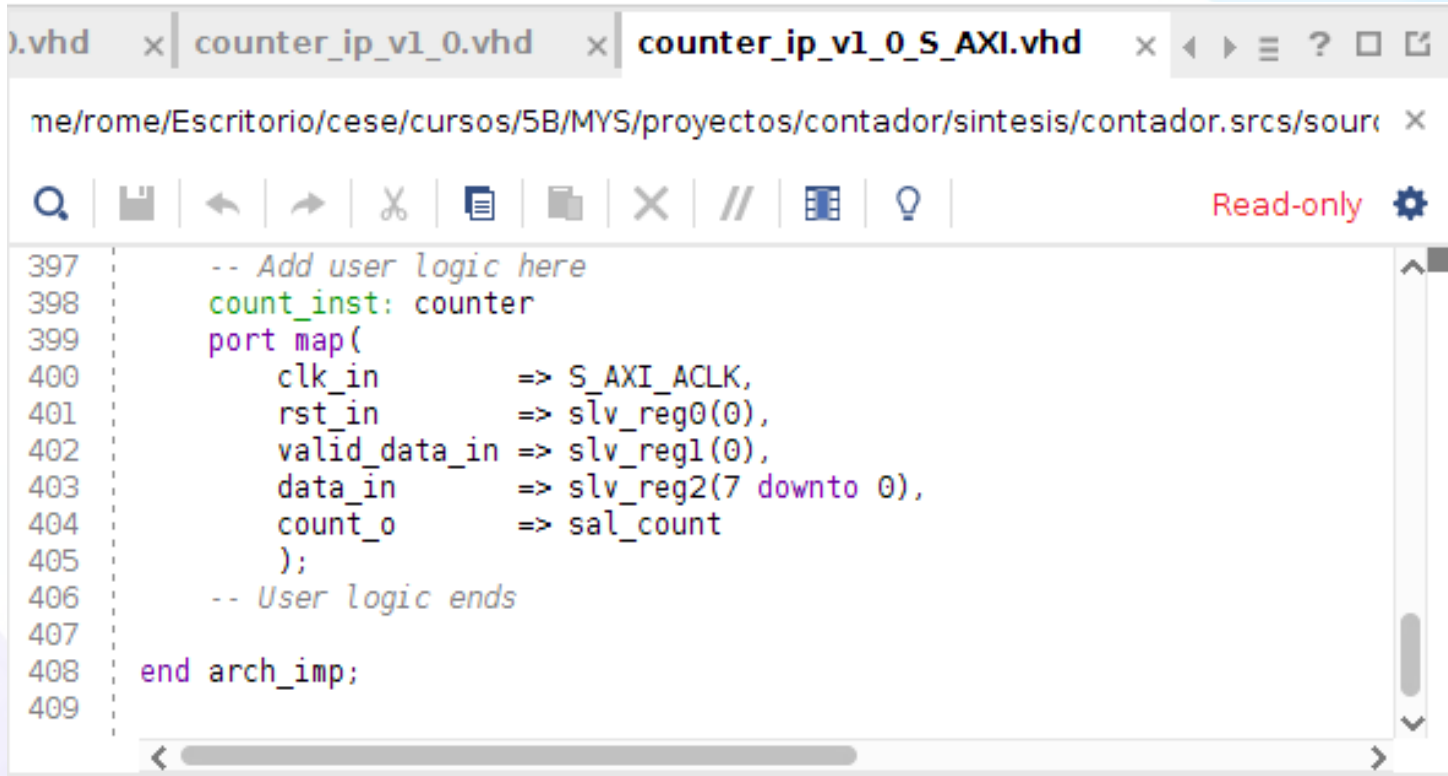
Character_detector



Block design



counter_ip. Señales y registros



The image shows a screenshot of a VHDL editor window. The title bar at the top displays three open files: `...vhd`, `counter_ip_v1_0.vhd`, and `counter_ip_v1_0_S_AXI.vhd`. The active file is `counter_ip_v1_0_S_AXI.vhd`. The editor's address bar shows the file path: `me/rome/Escritorio/cese/cursos/5B/MYS/proyectos/contador/sintesis/contador.srcc/sourc`. The toolbar includes icons for search, save, undo, redo, cut, copy, paste, delete, comment, and help, along with a "Read-only" status and a settings gear icon. The code editor displays the following VHDL code:

```
397      -- Add user logic here
398      count_inst: counter
399      port map(
400          clk_in      => S_AXI_ACLK,
401          rst_in      => slv_reg0(0),
402          valid_data_in => slv_reg1(0),
403          data_in      => slv_reg2(7 downto 0),
404          count_o      => sal_count
405      );
406      -- User logic ends
407
408  end arch_imp;
409
```

Board-Address

Sources | **Design** | **Signals** | **Board x** | ? _ □ ▢

🔍 | ⚙️ | ⚙️ | ⚙️ | ⚙️ | ⚙️ | ⚙️

Arty Z7-10

- 📁 Clocks (0 out of 1 connected)
 - 🕒 System Clock
- 📁 GPIO (0 out of 5 connected)
 - 🔦 2 RGB LEDs

Source File Properties | ? _ □ ▢ X

🔵 system.vhd | ⬅️ | ➡️ | ⚙️

< | _____ | >

General | Properties

Address Editor | ? _ □ ▢ X

🔍 | ⚙️ | ⚙️ | ⚙️ | ⚙️ | ⚙️ | ⚙️

Cell	Slave Interface	Base Name	Offset Address	Range	High Ac
📁 processing_system7_0					
📁 Data (32 address bits : 0x40000000 [1G])					
🔍 counter_ip_0	S_AXI	S_AXI_reg	0x43C0_0000	64K	0x43C0




Jerarquía

Sources

Q [?] + [?] 0

- Design Sources (1)
 - system_wrapper(STRUCTURE) (system_wrapper.vhd) (1)
 - system_i : system (system.bd) (1)
 - system(STRUCTURE) (system.vhd) (6)
 - counter_ip_0 : system_counter_ip_0_0 (system_counter_ip_0_0.xci) (1)
 - system_counter_ip_0_0(system_counter_ip_0_0_arch) (system_counter_ip_0_0.vhd) (1)
 - U0 : counter_ip_v1_0(arch_imp) (counter_ip_v1_0.vhd) (1)
 - counter_ip_v1_0_S_AXI_inst : counter_ip_v1_0_S_AXI(arch_imp) (counter_ip_v1_0_S_AXI.vhd) (1)
 - processing_system7_0 : system_processing_system7_0_0 (system_processing_system7_0_0.xci)
 - ps7_0_axi_periph : system_ps7_0_axi_periph_0(STRUCTURE) (system.vhd) (1)
 - ps7_0_axi_periph : system_ps7_0_axi_periph_0
 - rst_ps7_0_100M : system_rst_ps7_0_100M_0 (system_rst_ps7_0_100M_0.xci)
 - vio_0 : system_vio_0_0 (system_vio_0_0.xci)
- Constraints
- Simulation Sources (1)
 - sim_1 (1)
 - system_wrapper(STRUCTURE) (system_wrapper.vhd) (1)
- Utility Sources

Proyecto de software. Estructura de carpetas

- ▶  counter_app
- ▶  counter_app_bsp
- ▶  system_wrapper_hw_platform_0

Archivo system.hdf

contador.c system.hdf				
system_wrapper_hw_platform_0 Hardware Platform Specification				
Design Information				
Target FPGA Device: 7z010				
Part: xc7z010clg400-1				
Created With: Vivado 2018.3				
Created On: Thu Oct 3 11:36:31 2024				
Address Map for processor ps7_cortexa9_[0-1]				
Cell	Base Addr	High Addr	Slave I/f	Mem/Reg
ps7_intc_dist_0	0xf8f01000	0xf8f01fff		REGISTER
ps7_gpio_0	0xe000a000	0xe000afff		REGISTER
ps7_scutimer_0	0xf8f00600	0xf8f0061f		REGISTER
ps7_slcr_0	0xf8000000	0xf8000fff		REGISTER
ps7_scuwdt_0	0xf8f00620	0xf8f006ff		REGISTER
ps7_l2cachec_0	0xf8f02000	0xf8f02fff		REGISTER
ps7_scuc_0	0xf8f00000	0xf8f000fc		REGISTER
counter_ip_0	0x43c00000	0x43c0ffff	S_AXI	REGISTER
ps7_qspi_linear_0	0xfc000000	0xfcffffff		FLASH

Archivo contador.c

contador.c

```
#include "xparameters.h"
#include "xil_io.h"
#include "counter_ip.h"
⊕ // Comandos para trabajar con counter_ip "r", "a", "d", "p"

#define RESET_REG          0 // reset por registro
#define AVAILABLE_DATA     1 // Dato disponible para ser leído
#define NOT_AVAILABLE_DATA 0 // Dato no disponible

#define RESET_COMAND       114 // reset de la cuenta por comando
#define PAUSE               112 // pausa la cuenta
#define ASC_COUNT           97  // habilitación y cuenta ascendente
#define DES_COUNT          100  // habilitación y cuenta descendente
#define INVALID_COMAND     120 // comando inválido
```

Archivo contador.c

```
int main (void)
{
    int i = 0; // variable auxiliar para recorrer la secuencia de prueba
    int data[5]={ASC_COUNT,DES_COUNT,PAUSE,ASC_COUNT,RESET_COMAND}; // secuencia de prueba
    char ComandDataName[5][14]={"ASC_COUNT","DES_COUNT","PAUSE","ASC_COUNT","RESET_COMAND"}; // secuencia de prueba. Etiquetas

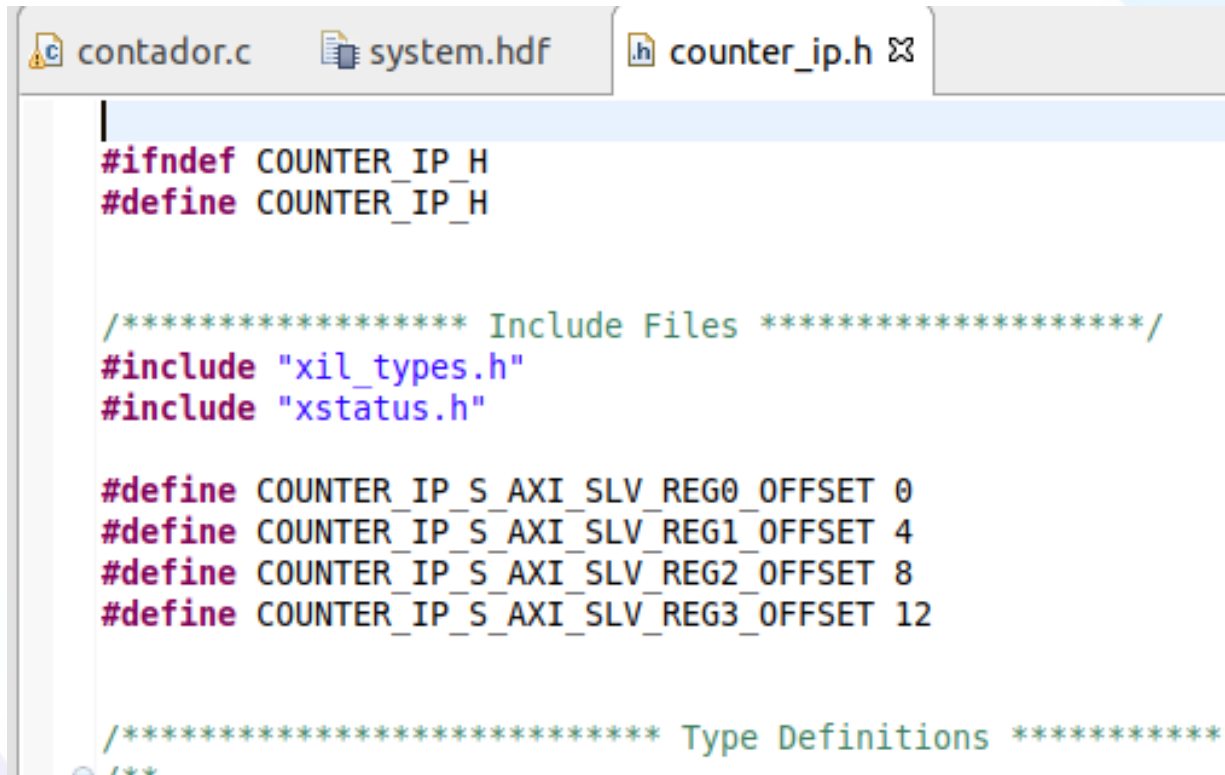
    xil_printf("-----\r\n");
    xil_printf("-- START PROGRAM COUNTER --\r\n");
    xil_printf("-- ALUMNO: RAUL ROMERO --\r\n");
    xil_printf("-----\r\n");

    // INICIALIZACIÓN
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG0_OFFSET, RESET_REG); // reset count
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG1_OFFSET, NOT_AVAILABLE_DATA); // dato no disponible
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG2_OFFSET, INVALID_COMAND); // dato inválido
    xil_printf("-- INICIALIZACIÓN --> RESET_REG --%d\r\n",RESET_REG);
    xil_printf("-- INICIALIZACIÓN --> NOT_AVAILABLE_DATA --%d\r\n",NOT_AVAILABLE_DATA);
    xil_printf("-- INICIALIZACIÓN --> INVALID_COMAND --%d\r\n",INVALID_COMAND);
```

Archivo contador.c

```
while (1)
{
    if(i>4){
        i=0;
    }
    xil_printf("-----\r\n");
    // Dato válido disponible
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG2_OFFSET, data[i]);
    xil_printf("-- Comando enviado-dato: %s --> %d\r\n", ComandDataName[i],data[i]);
    // habilitación de dato disponible
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG1_OFFSET, AVAILABLE_DATA);
    xil_printf("-- Dato habilitado --\r\n");
    // disponibilidad del dato por 1 segundos
    sleep(1);
    // dato de entrada no disponible
    COUNTER_IP_mWriteReg(XPAR_COUNTER_IP_0_S_AXI_BASEADDR,COUNTER_IP_S_AXI_SLV_REG1_OFFSET, NOT_AVAILABLE_DATA);
    xil_printf("-- Dato deshabilitado --\r\n");
    // delay para visualizar el comportamiento del contador
    xil_printf("-- sleep --\r\n");
    sleep(10);
    i++;
}
```

Archivo counter_ip.h



```
#ifndef COUNTER_IP_H
#define COUNTER_IP_H

/***** Include Files *****/
#include "xil_types.h"
#include "xstatus.h"

#define COUNTER_IP_S_AXI_SLV_REG0_OFFSET 0
#define COUNTER_IP_S_AXI_SLV_REG1_OFFSET 4
#define COUNTER_IP_S_AXI_SLV_REG2_OFFSET 8
#define COUNTER_IP_S_AXI_SLV_REG3_OFFSET 12

/***** Type Definitions *****/
```

Archivo xparameters.h

contador.c

system.hdf

counter_ip.h

xparameters.h

```
/* *****  
/* Definitions for driver COUNTER_IP */  
#define XPAR_COUNTER_IP_NUM_INSTANCES 1  
  
/* Definitions for peripheral COUNTER_IP_0 */  
#define XPAR_COUNTER_IP_0_DEVICE_ID 0  
#define XPAR_COUNTER_IP_0_S_AXI_BASEADDR 0x43C00000  
#define XPAR_COUNTER_IP_0_S_AXI_HIGHADDR 0x43C0FFFF  
  
/* *****
```

