

# Trabajo Práctico Final

## Microarquitecturas y Softcores

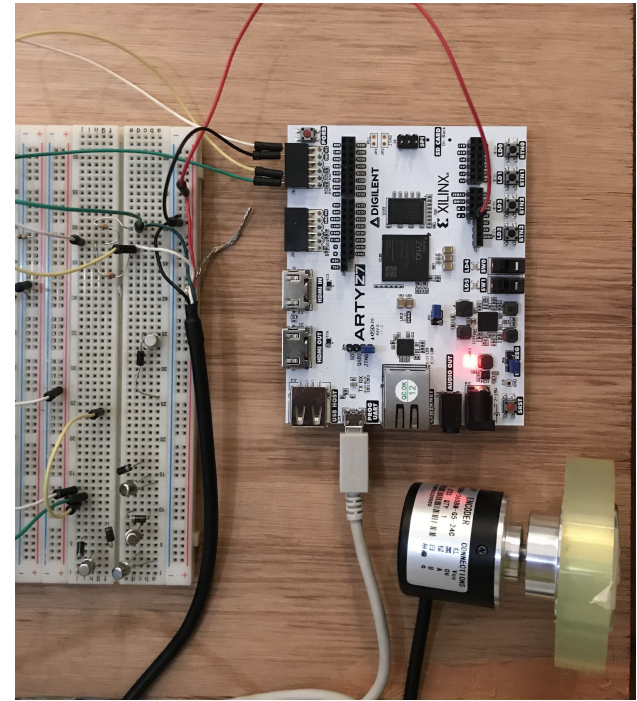
### **IP Core para manejo de encoder rotativo incremental con FPGA**

Autor: Ing. Ignacio Majul

Agosto de 2020

# Objetivos del TP

- Crear un sistema incorporando el procesador basado en ARM Cortex A9 (PS).
- Sintetizar un IP Core personalizado vinculado al PS.
- Reutilizar la configuración VHDL para manejo de encoder.
- Configuración del uC.
- Visualización de parámetros por UART.
- Uso de la placa Arty Z7-10.



Encoder LPD3806-360BM:

- Rotativo incremental
- 360 PPR dos canales
- 5000 rpm max
- Alimentación 5-24V

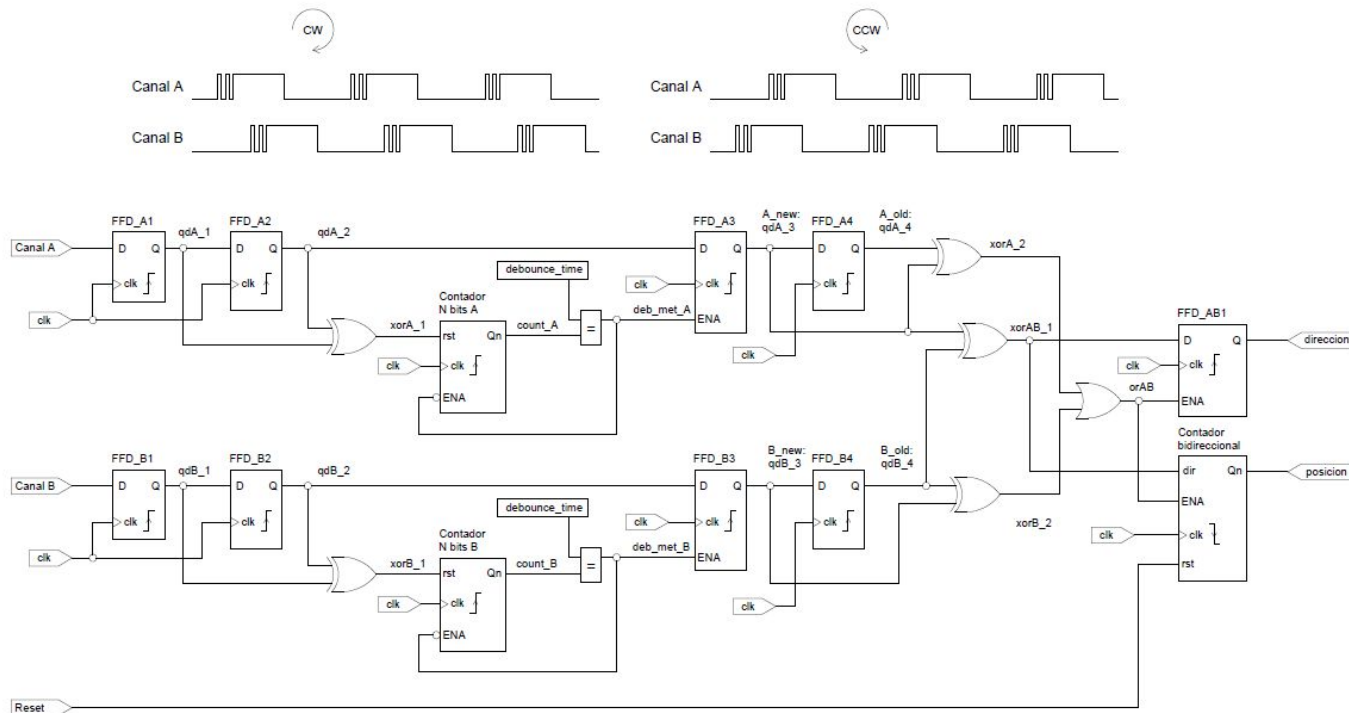
# Configuración PL

## Señales de Entrada:

- Canal A: 1 bit
- Canal B: 1 bit
- Reset: 1 bit

## Señales de Salida:

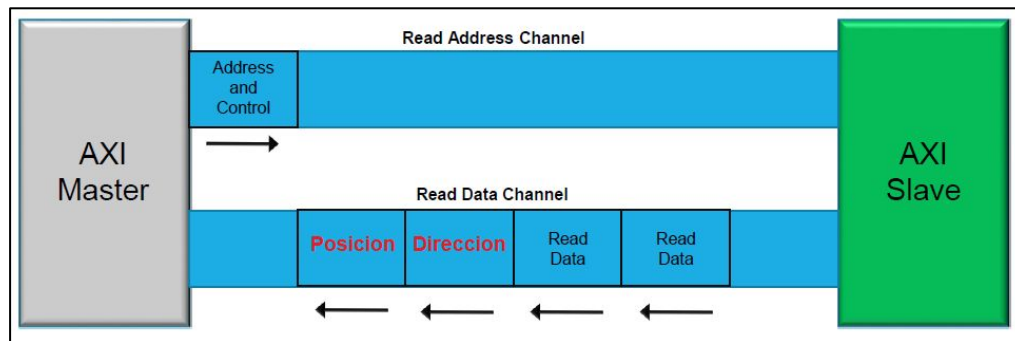
- Posición: 32 bit
- Dirección: 1 bit



# IP Integrator

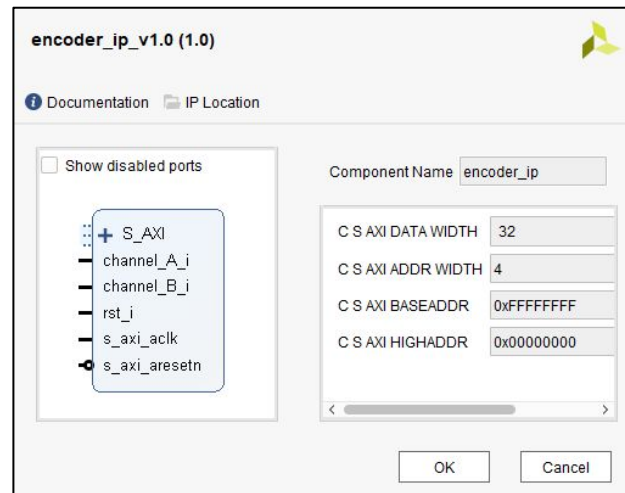
Sistema base:

- ZYNQ7 Processing System.
- Periféricos habilitados: UART\_0.
- Comunicación PS - PL: AXI4 Lite (Read)



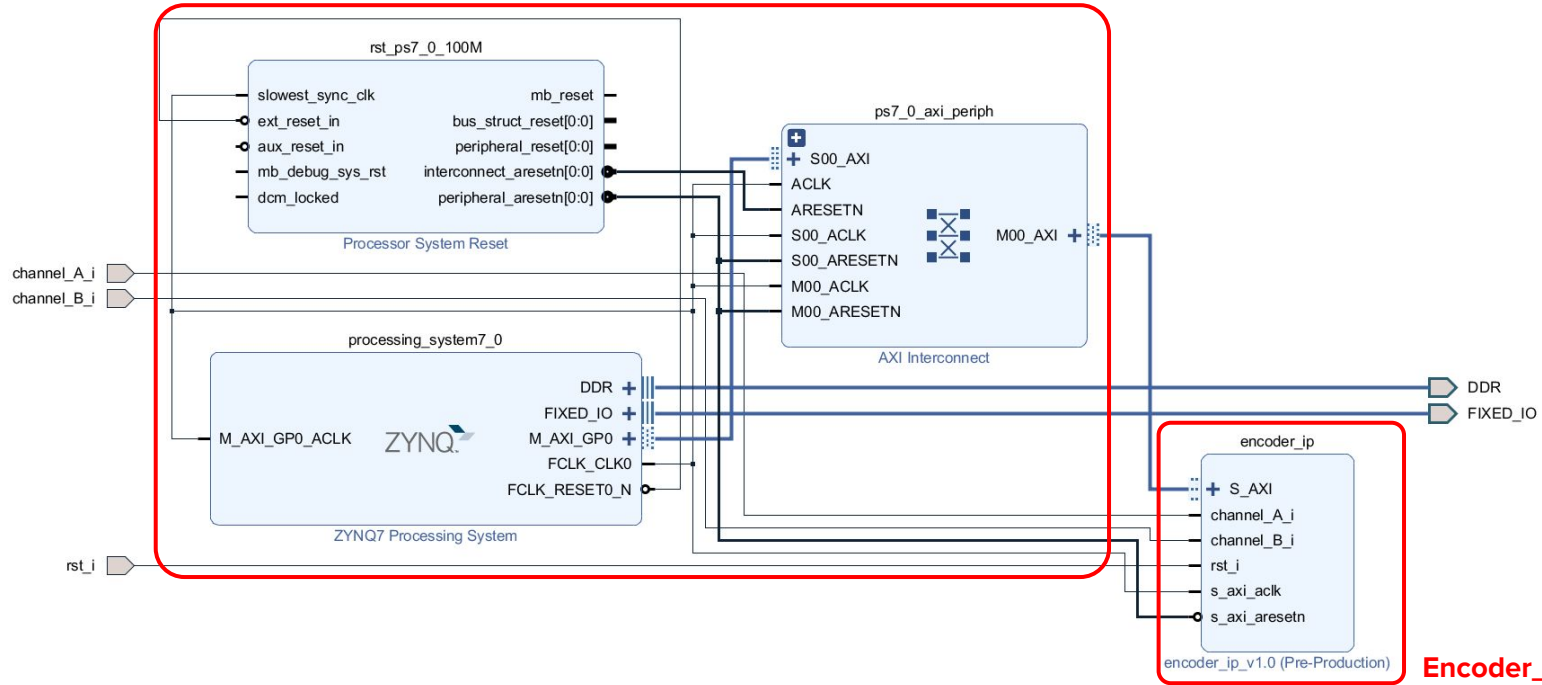
IP propio (Encoder\_IP):

- Entradas digitales mapeadas a JB1\_P, JB1\_N y BTNO
- Salidas sobre bus AXI.
- 4 registros de 32 bit (2 utilizados)
- AXI CLK: 50 MHz



# Diagrama de bloques

Sistema base



Encoder\_IP

Señales de  
entrada

Sistema de Procesamiento

Matriz de interconexión AXI

IP propio

# Configuración VHDL

## Se agregan los puertos en la lógica de la interfaz AXI

```
-- component declaration
component encoder_ip_v1_0_S_AXI is
  generic (
    C_S_AXI_DATA_WIDTH  : integer := 32;
    C_S_AXI_ADDR_WIDTH  : integer := 4
  );
  port (
    CHANNEL_A_I : in std_logic;
    CHANNEL_B_I : in std_logic;
    RST_I : in std_logic;

    S_AXI_ACLK : in std_logic;
    S_AXI_ARESETN : in std_logic;
    S_AXI_AWADDR : in std_logic_vector(C_S_AXI_ADDR_WIDTH-1 downto 0);
    S_AXI_AWPROT : in std_logic_vector(2 downto 0);
    S_AXI_AWVALID : in std_logic;
    S_AXI_AWREADY : out std_logic;
    S_AXI_WDATA : in std_logic_vector(C_S_AXI_DATA_WIDTH-1 downto 0);
    S_AXI_WSTRB : in std_logic_vector(C_S_AXI_DATA_WIDTH-1 downto 0);
```

## Se modifican los registros de salida

```
variable loc_addr : std_logic_vector(OPT_MEM_ADDR_WIDTH-1 downto 0);
begin
  -- Address decoding for reading registers
  loc_addr := axi_araddr(ADDR_LSB + OPT_MEM_ADDR_OFFSET);
  case loc_addr is
    when b"00" =>
      reg_data_out <= pos_reg;
    when b"01" =>
      reg_data_out <= dir_reg;
    when b"10" =>
      reg_data_out <= slv_reg2;
    when b"11" =>
      reg_data_out <= slv_reg3;
    when others =>
      reg_data_out <= (others => '0');
  end case;
end process;
```

## Lógica de usuario

```
-- Add user logic here
U1: entity work.encoder
  generic map(
    N => 32,
    debounce_time => 10,
    freq_clk => 50
  )
  port map(
    clk_i      => S_AXI_ACLK,
    pos_o => pos_reg,
    dir_o => dir_aux,
    channel_A_i => CHANNEL_A_I,
    channel_B_i => CHANNEL_B_I,
    rst_i => RST_I
  );
-- User logic ends
```



# Código C y Terminal serie

```
#include "xparameters.h"
#include "xil_io.h"
#include "encoder_ip.h"

#define PRESCALER 1440 // 1440 pulsos por revolucion

//=====

int main (void) {

    int posicion, direccion;
    char *sentido[2]={"antihorario", "horario"};

    xil_printf("-- Microarquitecturas y Softcores - TP Final - Ignacio Majul --\r\n");

    while(1)
    {
        posicion = ENCODER_IP_mReadReg(XPAR_ENCODER_IP_S_AXI_BASEADDR, ENCODER_IP_S_AXI_SLV_REG0_OFFSET);
        direccion = ENCODER_IP_mReadReg(XPAR_ENCODER_IP_S_AXI_BASEADDR, ENCODER_IP_S_AXI_SLV_REG1_OFFSET);
        xil_printf("Avance relativo: %d revoluciones. Sentido: %s \r\n", posicion/PRESCALER, sentido[direccion]);
        usleep(50000);
    }
}
```

SDK Log Terminal 1

Serial: (COM7, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)

Avance relativo: -10 revoluciones. Sentido: horario  
Avance relativo: -10 revoluciones. Sentido: horario  
Avance relativo: -9 revoluciones. Sentido: horario  
Avance relativo: -9 revoluciones. Sentido: horario  
Avance relativo: -8 revoluciones. Sentido: horario  
Avance relativo: -8 revoluciones. Sentido: antihorario  
Avance relativo: -9 revoluciones. Sentido: antihorario  
Avance relativo: -9 revoluciones. Sentido: antihorario  
Avance relativo: -10 revoluciones. Sentido: antihorario  
Avance relativo: -11 revoluciones. Sentido: antihorario  
Avance relativo: -11 revoluciones. Sentido: antihorario  
Avance relativo: -12 revoluciones. Sentido: horario  
Avance relativo: -11 revoluciones. Sentido: horario  
Avance relativo: -11 revoluciones. Sentido: horario  
Avance relativo: -11 revoluciones. Sentido: horario  
Avance relativo: -11 revoluciones. Sentido: horario  
Avance relativo: -10 revoluciones. Sentido: antihorario  
Avance relativo: -11 revoluciones. Sentido: antihorario  
Avance relativo: -12 revoluciones. Sentido: antihorario  
Avance relativo: -12 revoluciones. Sentido: antihorario  
Avance relativo: -13 revoluciones. Sentido: antihorario  
Avance relativo: -13 revoluciones. Sentido: antihorario  
Avance relativo: -14 revoluciones. Sentido: antihorario  
Avance relativo: -14 revoluciones. Sentido: antihorario  
Avance relativo: -14 revoluciones. Sentido: antihorario  
Avance relativo: -14 revoluciones. Sentido: antihorario  
Avance relativo: -15 revoluciones. Sentido: antihorario  
Avance relativo: -15 revoluciones. Sentido: antihorario

# Preguntas?



# Gracias!