

Proyecto Final de CLP

Implementacion de un Filtro Digital FIR pasa bajo

...

Alumno: Ing. Acerbo Ezequiel

Introducción

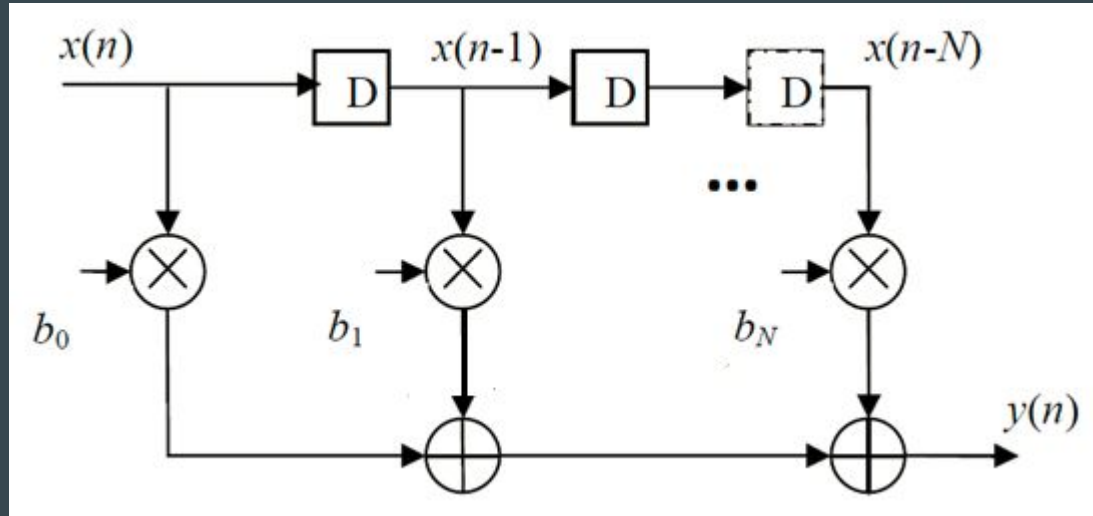
- El filtro FIR se implementa básicamente mediante el uso de Flip-Flops tipo D, multiplicadores y sumadores con signo.
- Un bloque básico incluye un registro de N bits(retardo), un multiplicador y un sumador.
- La declaración de generación en VHDL se usa para generar el diseño completo usando el bloque básico.
- El código VHDL para el filtro FIR de paso bajo es:

EL codigo tiene dos componentes:

- *Registro de Nbits(retardo con flip flop D).*
- *Filtro: contiene multiplicadores y sumadores.*

Estructura de Filtro a diseñar

*Entrada
de
Señal*



*Salida
de
Señal
Filtrada*

Filtros de Respuesta al Impulso Finita (FIR: Finite Impulse Response)

- *Filtro de orden 10.*
- *Coeficientes b_k son enteros con signo de 8bits.*
- *En la expresión siguiente N es el orden del filtro, b_k es el número de coeficientes del filtro.*

$$y_n = \sum_{k=0}^{N-1} b_k x(n - k)$$

Vhdl del Registro

```
Library IEEE;
USE IEEE.Std_logic_1164.all;

entity N_bit_Reg is
generic ( input_width : integer :=8 );

    port(
        D : in std_logic_vector(input_width-1 downto 0);
        Q : out std_logic_vector(input_width-1 downto 0);
        Clk : in std_logic;
        reset : in std_logic

    );

end N_bit_Reg;

architecture Behavioral of N_bit_Reg is

begin
    process(Clk,reset)
    begin
        if (reset = '1') then
            Q <= (others => '0');
        elsif ( rising_edge(Clk) ) then
            Q <= D;
        end if;
    end process;
end Behavioral;
```

Vhdl del filtro

```
entity FIR_RI is
generic (
    input_width  : integer :=8  ;-- para configurar el ancho de la entrada
    output_width : integer :=16 ;-- para configurar el ancho de salida
    coef_width   : integer :=8  ;-- para configurar el ancho del coeficiente
    tap          : integer :=11 ;-- para configurar el orden del filtro
    guard        : integer :=0  -- log2(tap)+1
) ;

port(
    Din: in  std_logic_vector(input_width-1 downto 0);-- entrada de la señal a filtrar
    Clk: in  std_logic;                                -- Clock
    reset: in std_logic;                                -- Reset
    Dout: out std_logic_vector(output_width-1 downto 0)-- salida de la señal filtrada

);

end;

architecture FIR_RI_arq of FIR_RI is

--Se llama al componente Registro de N bits (que es un retardo de la señal de entrada entre etapas)

    component N_bit_Reg
        generic ( input_width: integer:=8);
        port(
            D :in std_logic_vector(input_width-1 downto 0);
            Q :out std_logic_vector(input_width-1 downto 0);
            Clk :in std_logic;
            reset :in std_logic

        );
    end component;

end architecture;
```

Declaraciones

```
-----Se declara un tipo de variable para los coeficientes-----  
type Coefficient_type is array (1 to tap) of std_logic_vector(coef_width-1 downto 0);  
  
-----Coeficientes del filtro-----  
constant coefficient: coefficient_type := (X"F1",X"F3",X"07",X"26",X"42",X"4E",X"42",X"26",X"07",X"F3",X"F1");  
  
-----Se declaran tipos de variable para el multiplicador, sumador y para la señales de entrada entre etapas-----  
type shift_reg_type is array (0 to tap-1) of std_logic_vector(input_width-1 downto 0);  
signal shift_reg : shift_reg_type;  
type mult_type is array (0 to tap-1) of std_logic_vector(input_width+coef_width-1 downto 0);  
signal mult : mult_type;  
type ADD_type is array (0 to tap-1) of std_logic_vector(input_width+coef_width-1 downto 0);  
signal ADD: ADD_type;
```

Generate para las distintas etapas

-----Aquí se implementa la cantidad de etapas del filtro mediante un GENERATE-----

```
GEN_FIR:
  for i in 0 to tap-2 generate
    begin
      --Unidad del retardo de N Bits, Aquí le asigno las señales a los distintos puertos
      N_bit_Reg_unit : N_bit_Reg generic map (input_width => 8)
        port map (
          Clk => Clk,
          reset => reset,
          D => shift_reg(i),
          Q => shift_reg(i+1)
        );

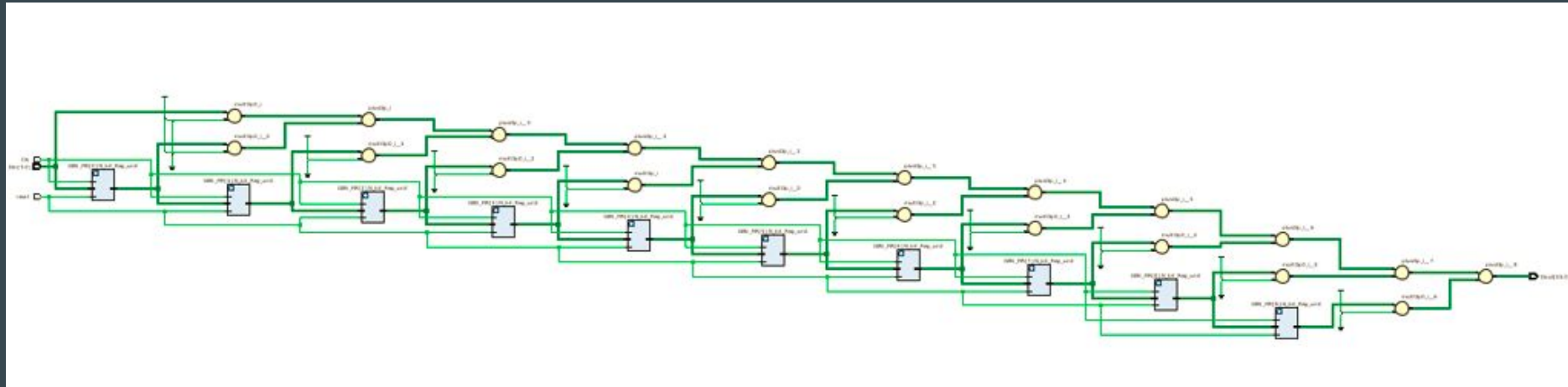
      --Aca implemento una multiplicacion segun la estructura del Filtro Digital

      mult(i+1) <= shift_reg(i+1) * coeficient(i+2);

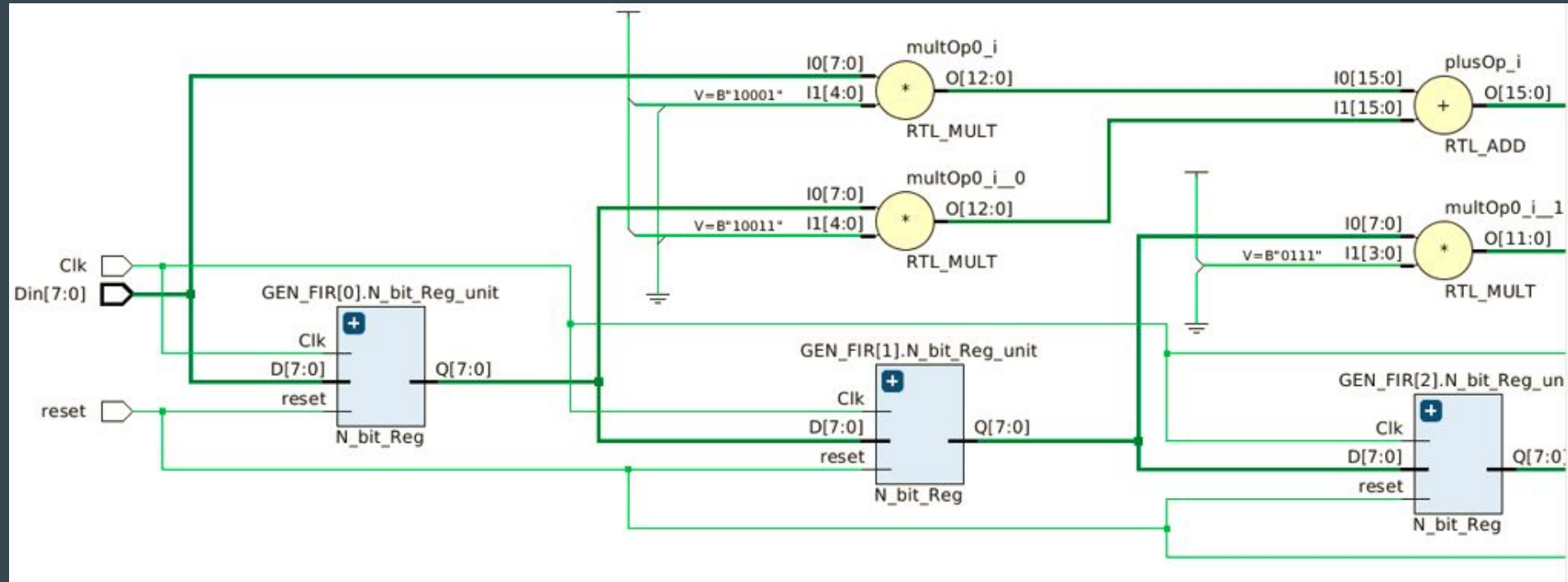
      --Aca implemento una suma segun la estructura del Filtro Digital

      ADD(i+1) <= ADD(i) + mult(i+1);
```

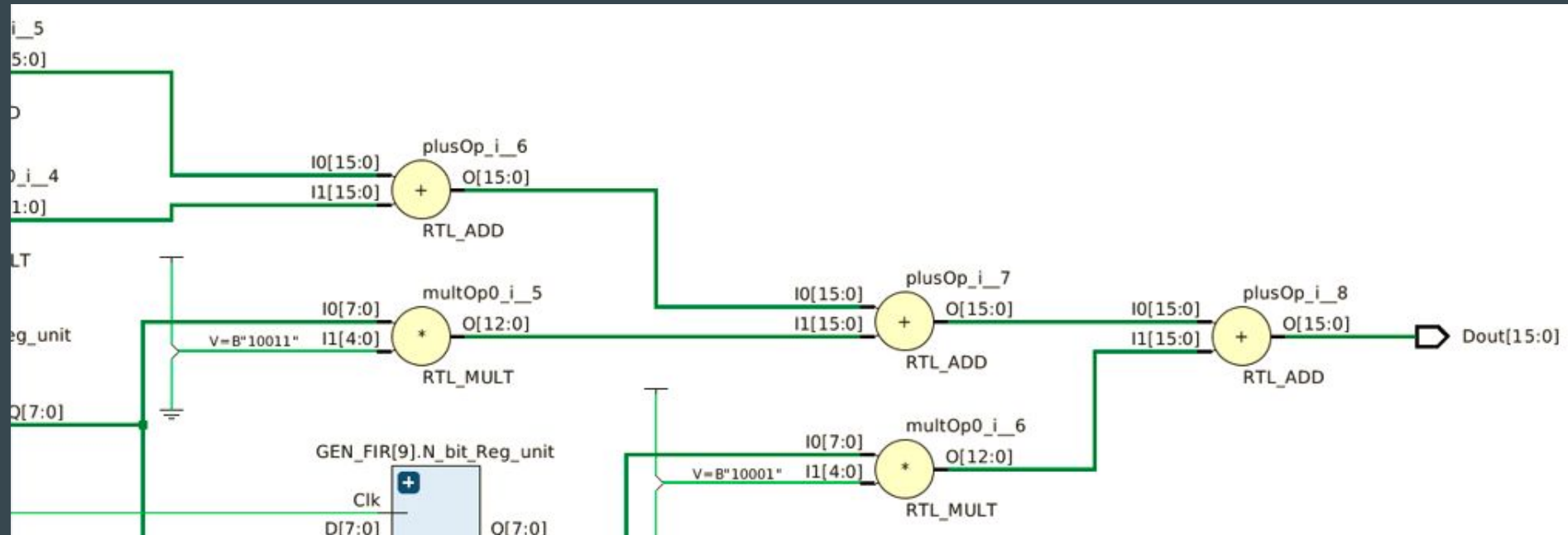

Esquemático del Filtro Completo



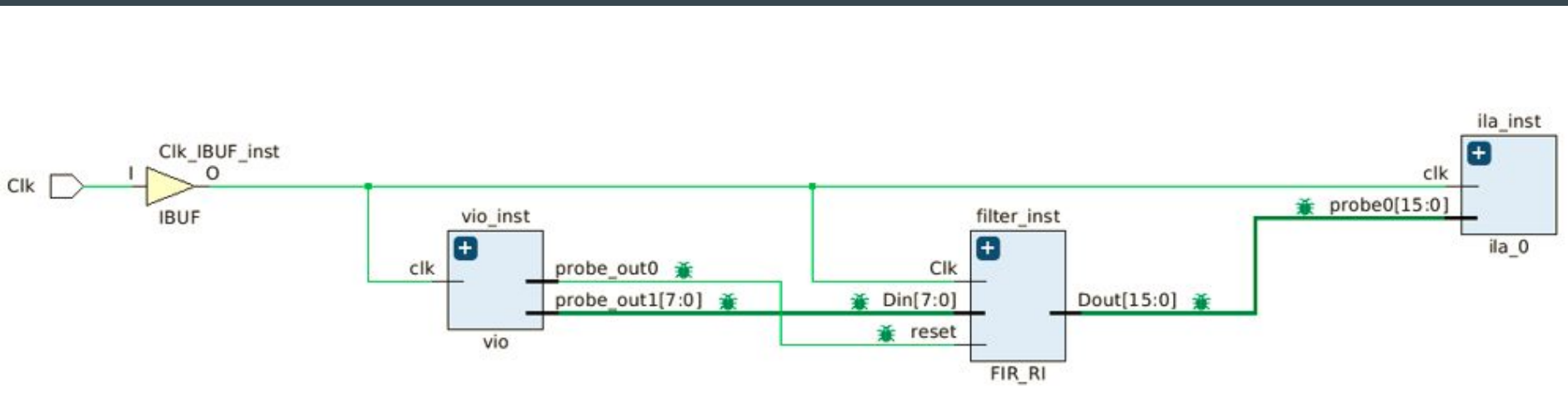
Zoom del Filtro Completo(Entrada)



Zoom del Filtro Completo(Salida)



Esquema de VIO-Filtro-ILA



Conexion Remota a la Placa

filter_vio_ila - [/home/ezequiel/CLP/filter_vio_ila/filter_vio_ila.xpr] - Vivado 2018.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access write_bitstream Complete

Dashboard

Flow Navigator

- Open Block Design
- Generate Block Design
- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Schematic
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG
 - Generate Bitstream
 - Open Hardware Manager
 - Open Target

HARDWARE MANAGER - 181.47.10.131/xilinx_tcf/Digilent/003017A4C81CA

Hardware

Name	Status
181.47.10.131 (3)	Connected
xilinx_tcf/Digilent/003017A...	Open
arm_dap_0 (0)	N/A
xc7z010_1 (3)	Programmed
XADC (System Monitor)	
hw_ila_1 (ila_inst)	Idle
hw_vio_1 (vio_inst)	Outputs out-of-sync
xilinx_tcf/Digilent/003017A...	Closed
xilinx_tcf/Digilent/003017A...	Closed

ILA Core Properties

hw_ila_1

Name: hw_ila_1

Cell: ila_inst

Device: xc7z010_1

HW core: core_1

Capture sample count: 0 of 1024

Core status: Idle

General Properties

Tcl Console Messages Serial I/O Links Serial I/O Scans

hw_ila_1 x hw_vios x

Waveform - hw_ila_1

ILA Status: Idle

Name	Value
prueba_dout[15:0]	
[15]	
[14]	
[13]	
[12]	
[11]	
[10]	

Settings - hw_ila_1 Status - hw_ila_1

Trigger Mode Settings

Trigger mode: BASIC_ONLY

Capture Mode Settings

Trigger Set Capture Set

Name	Operator
prueba_dout[15:0]	==

ILA esperando el Trigger

filter_vio_ila - [/home/ezequiel/CLP/filter_vio_ila/filter_vio_ila.xpr] - Vivado 2018.1

File Edit Flow Tools Reports Window Layout View Help Q- Quick Access write_bitstream Complete ✓ Default Layout

Flow Navigator

- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Schematic
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG**
 - Generate Bitstream
 - Open Hardware Manager
 - Open Target
 - Program Device
 - Add Configuration Mem

HARDWARE MANAGER - 181.47.10.131/xilinx_tcf/Digilent/003017A4C81CA

Hardware

	Status
0.131 (3)	Connected
_tcf/Digilent/003017A...	Open
_dap_0 (0)	N/A
z010_1 (3)	Programmed
ADC (System Monitor)	
hw_ila_1 (ila_inst)	Waiting For Tri...
hw_vio_1 (vio_inst)	Outputs out-of-sy...
_tcf/Digilent/003017A...	Closed

ILA Core Properties

hw_ila_1

Name: hw_ila_1
Cell: ila_inst
Device: xc7z010_1
HW core: core_3
Capture sample count: 512 of 1024

Waveform - hw_ila_1

ILA Status: Waiting For Trigger (512 out of 1024)

Name	Value
prueba_dout[15:0]	
[15]	
[14]	
[13]	
[12]	
[11]	
[10]	
[9]	
[8]	

Settings - hw_ila_1

Trigger Mode Settings

Trigger mode: BASIC_ONLY

Trigger Set

Name	Operator
prueba_dout[15:0]	==

ILA Core: hw_ila_1

!!!MUCHAS GRACIAS!!!