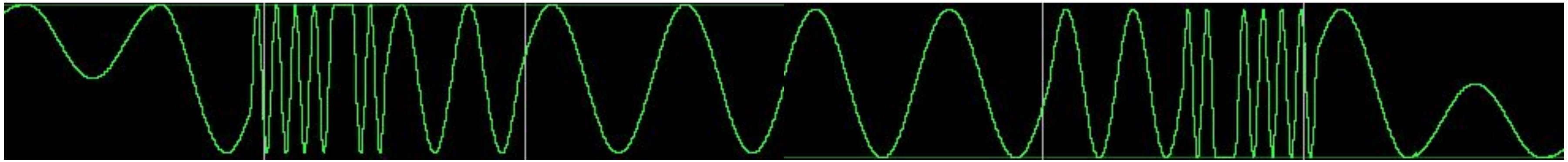
Trabajo práctico final
Circuitos Lógicos Programables

Especialización en Sistemas Embebidos

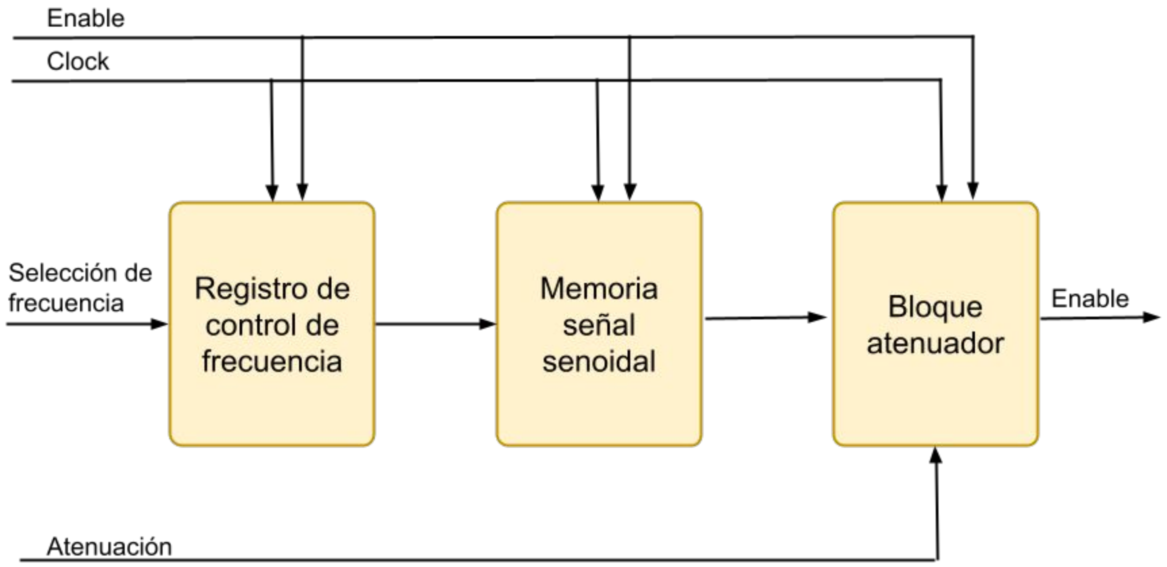
NCO

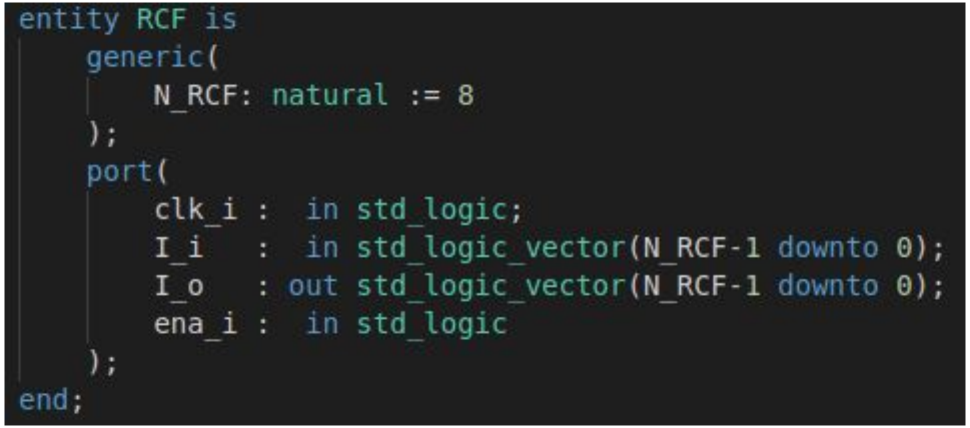
(Oscilador controlado numéricamente)

Alumno: Ing. Leandro Arrieta
Profesor: Ing. Nicolás Álvarez
Fecha: 18/04/2022



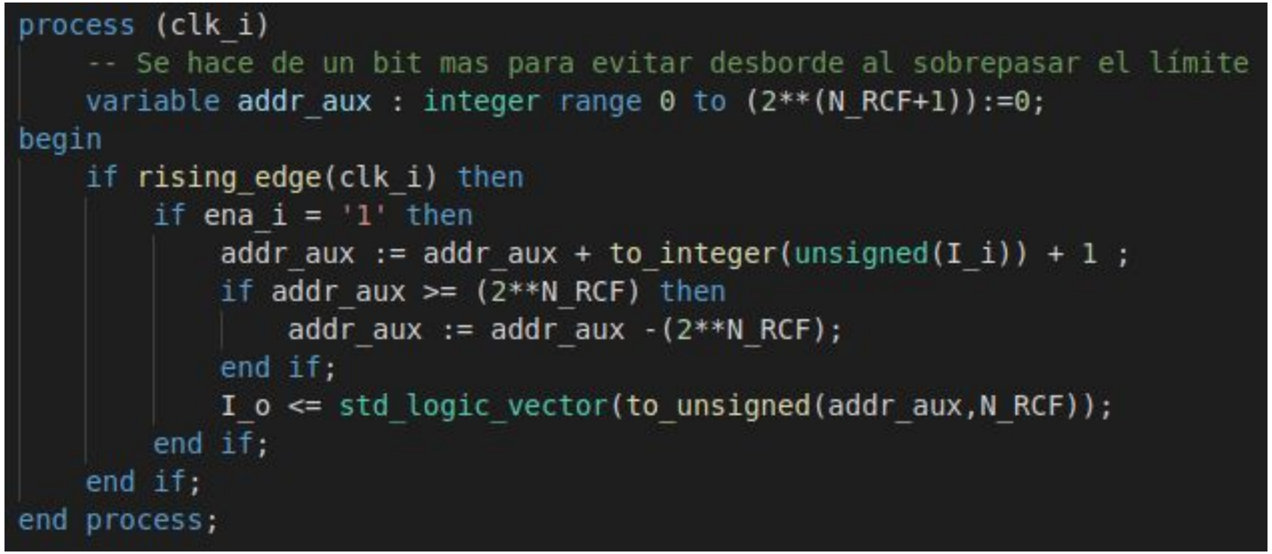
1. Diagrama en bloques





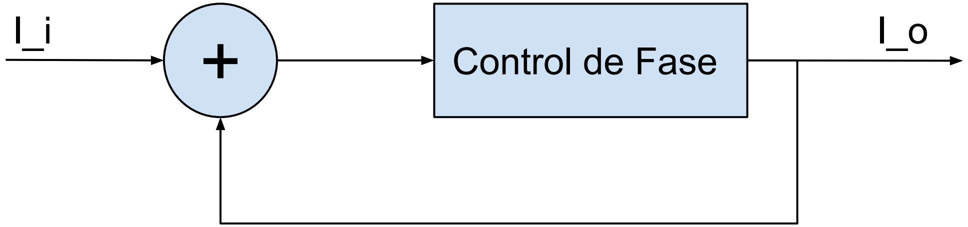
```vhdl
entity RCF is
    generic(
        N_RCF: natural := 8
    );
    port(
        clk_i :  in std_logic;
        I_i   :  in std_logic_vector(N_RCF-1 downto 0);
        I_o   : out std_logic_vector(N_RCF-1 downto 0);
        ena_i :  in std_logic
    );
end;
```

Registro de control de frecuencia

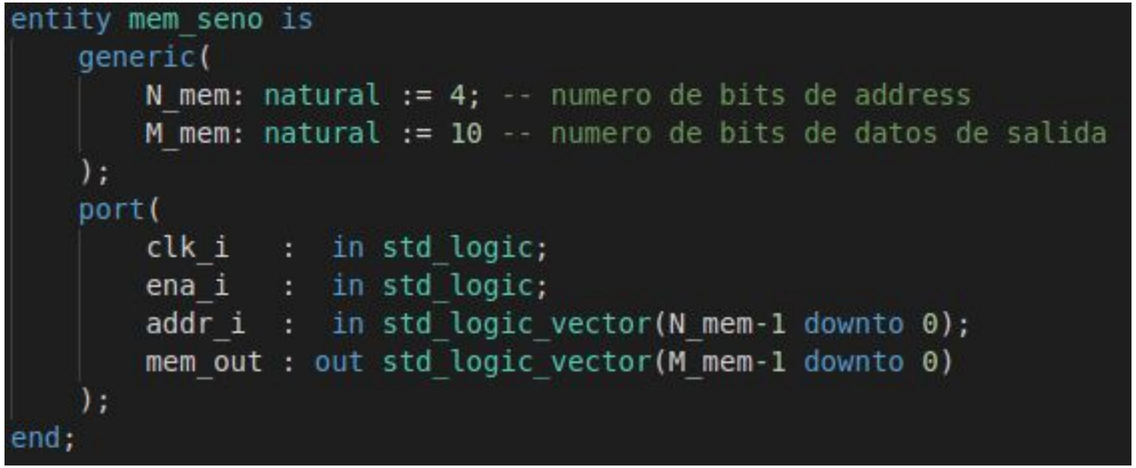


```vhdl
process (clk_i)
    -- Se hace de un bit mas para evitar desborde al sobrepasar el límite
    variable addr_aux : integer range 0 to (2**(N_RCF+1)):=0;
begin
    if rising_edge(clk_i) then
        if ena_i = '1' then
            addr_aux := addr_aux + to_integer(unsigned(I_i)) + 1 ;
            if addr_aux >= (2**N_RCF) then
                addr_aux := addr_aux -(2**N_RCF);
            end if;
            I_o <= std_logic_vector(to_unsigned(addr_aux,N_RCF));
        end if;
    end if;
end process;
```

Registro de control de frecuencia

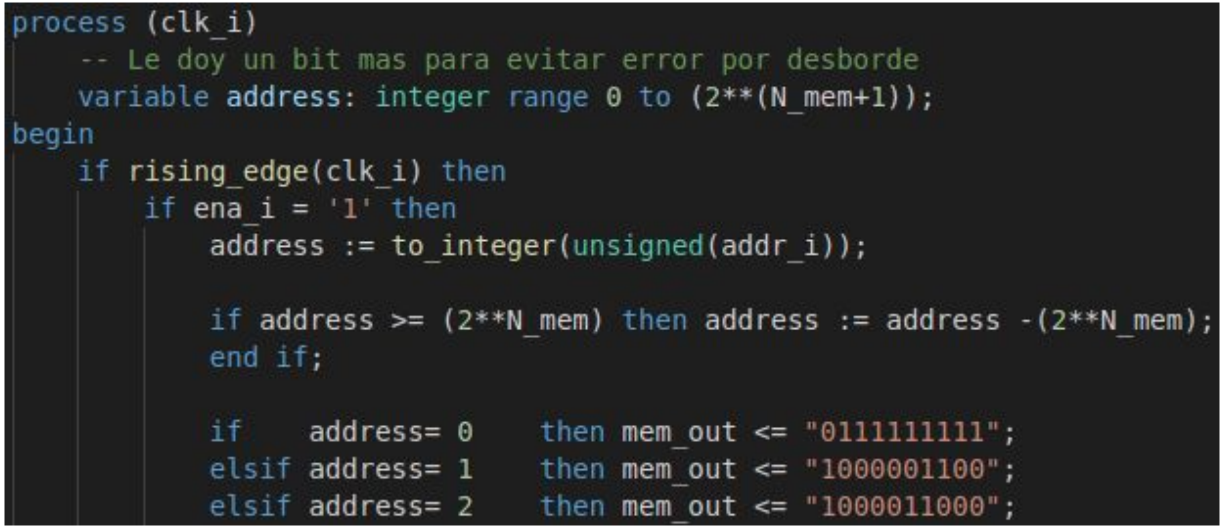


$$Fo = \frac{Fclk \times I\_i}{M}$$

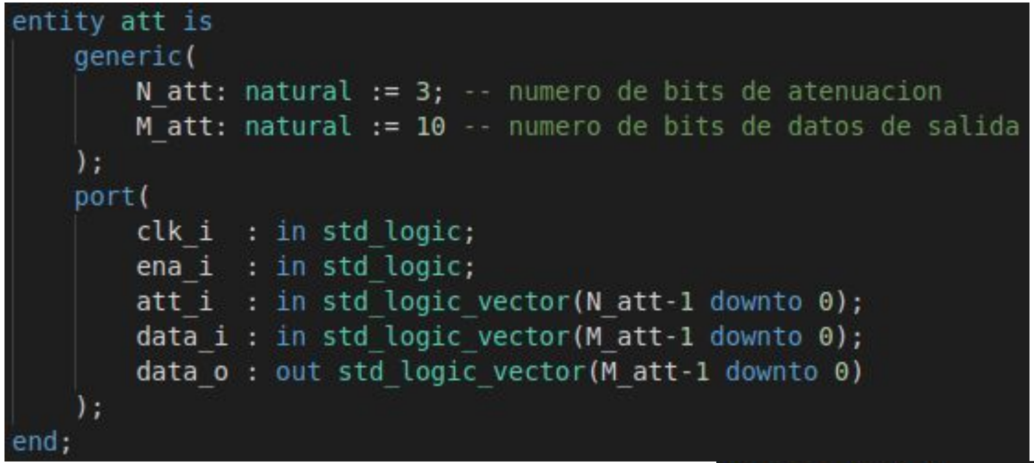


```vhdl
entity mem_seno is
    generic(
        N_mem: natural := 4; -- numero de bits de address
        M_mem: natural := 10 -- numero de bits de datos de salida
    );
    port(
        clk_i   :  in std_logic;
        ena_i   :  in std_logic;
        addr_i  :  in std_logic_vector(N_mem-1 downto 0);
        mem_out : out std_logic_vector(M_mem-1 downto 0)
    );
end;
```

Memoria señal senoidal

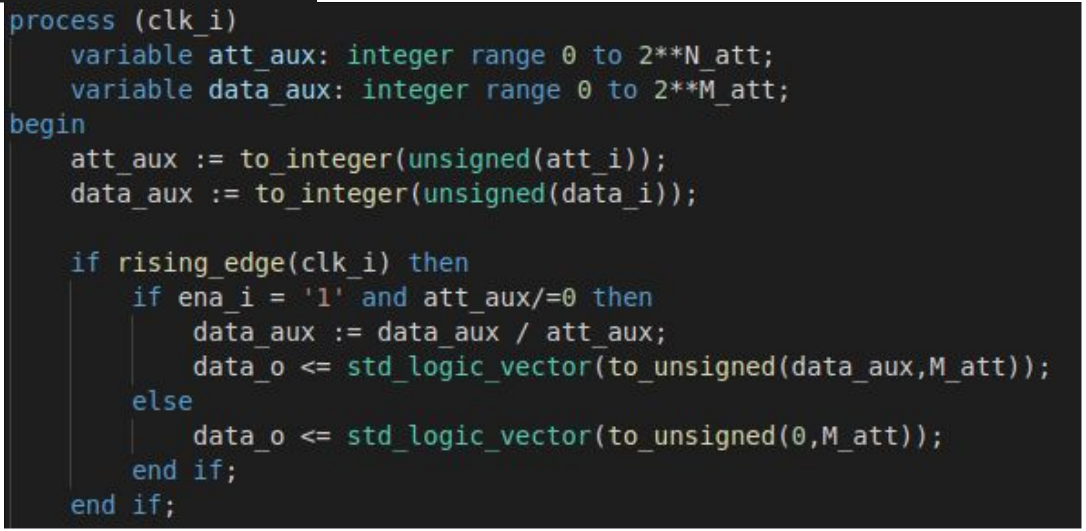


```vhdl
process (clk_i)
    -- Le doy un bit mas para evitar error por desborde
    variable address: integer range 0 to (2**(N_mem+1));
begin
    if rising_edge(clk_i) then
        if ena_i = '1' then
            address := to_integer(unsigned(addr_i));

            if address >= (2**N_mem) then address := address -(2**N_mem);
            end if;

            if    address= 0    then mem_out <= "0111111111";
            elsif address= 1    then mem_out <= "1000001100";
            elsif address= 2    then mem_out <= "1000011000";
```



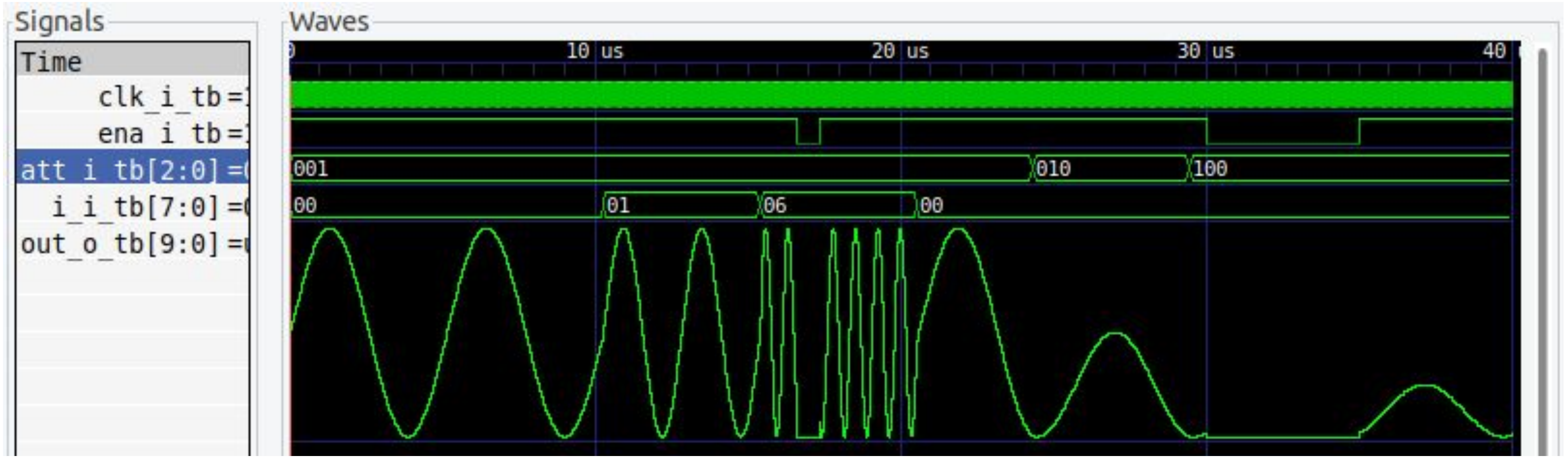
```vhdl
entity att is
    generic(
        N_att: natural := 3; -- numero de bits de atenuacion
        M_att: natural := 10 -- numero de bits de datos de salida
    );
    port(
        clk_i  : in std_logic;
        ena_i  : in std_logic;
        att_i  : in std_logic_vector(N_att-1 downto 0);
        data_i : in std_logic_vector(M_att-1 downto 0);
        data_o : out std_logic_vector(M_att-1 downto 0)
    );
end;
```

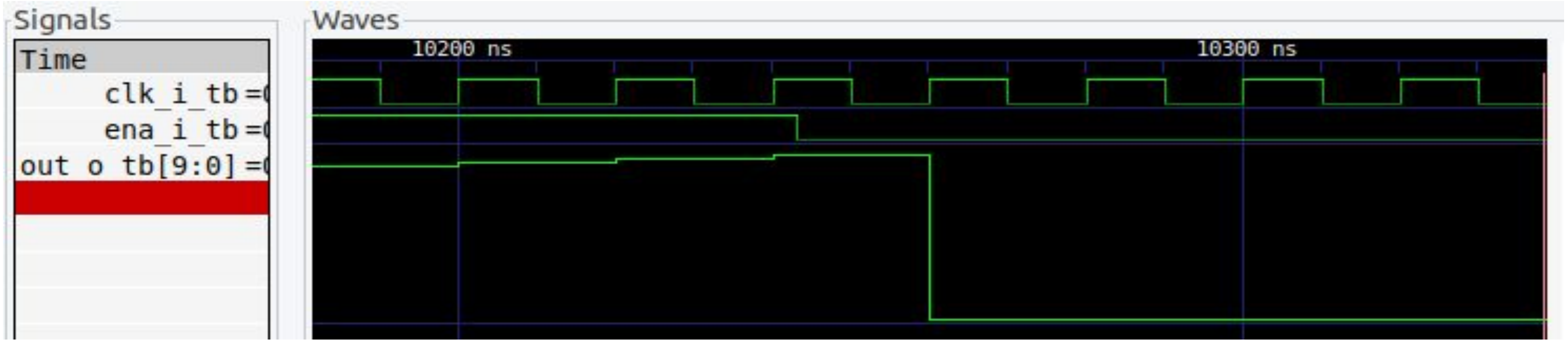
Bloque atenuador



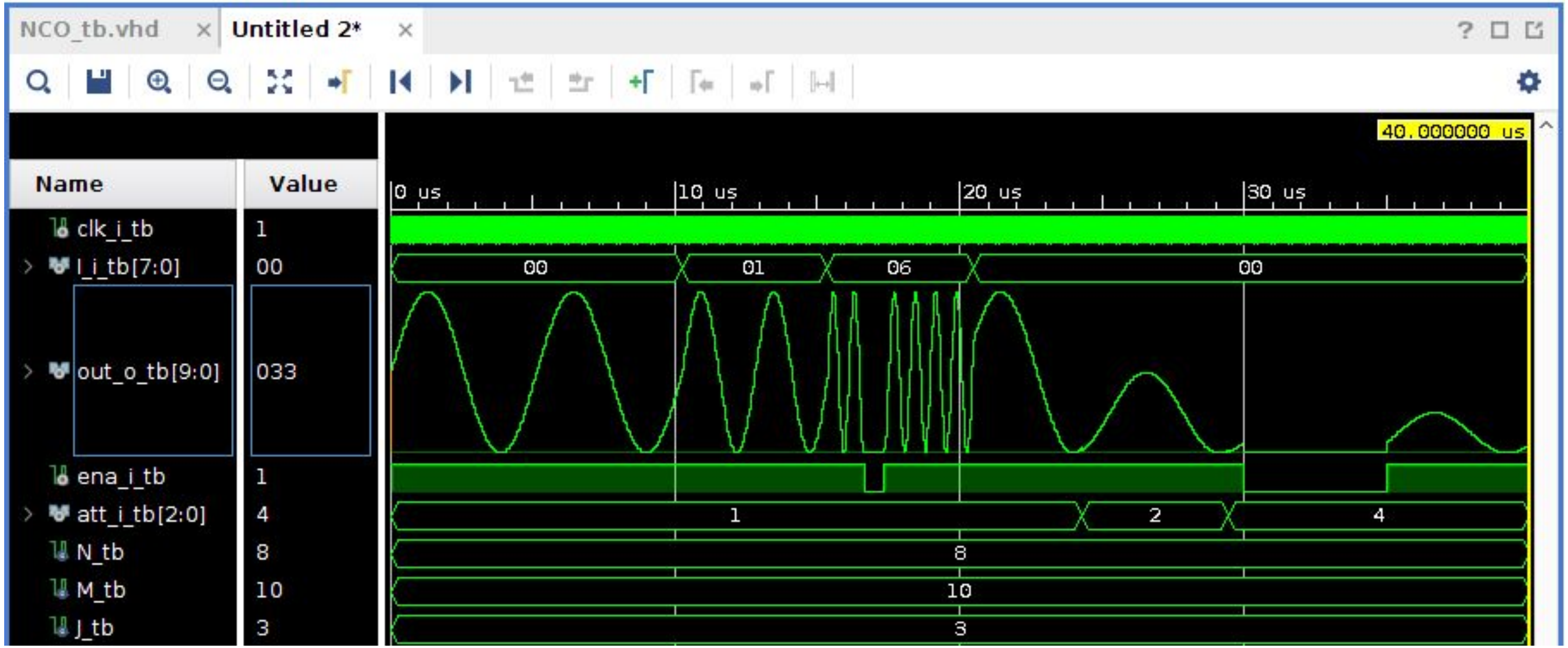
```vhdl
process (clk_i)
    variable att_aux: integer range 0 to 2**N_att;
    variable data_aux: integer range 0 to 2**M_att;
begin
    att_aux := to_integer(unsigned(att_i));
    data_aux := to_integer(unsigned(data_i));

    if rising_edge(clk_i) then
        if ena_i = '1' and att_aux/=0 then
            data_aux := data_aux / att_aux;
            data_o <= std_logic_vector(to_unsigned(data_aux,M_att));
        else
            data_o <= std_logic_vector(to_unsigned(0,M_att));
        end if;
    end if;
```

Simulación con GTKWave



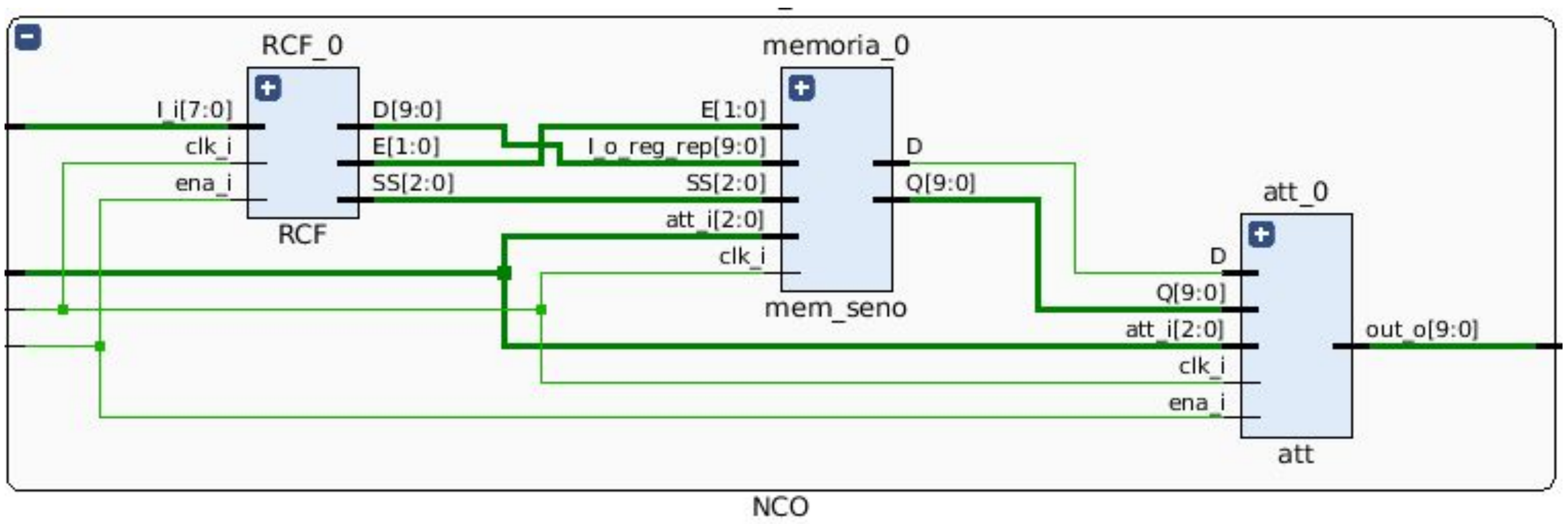


Simulación en Vivado

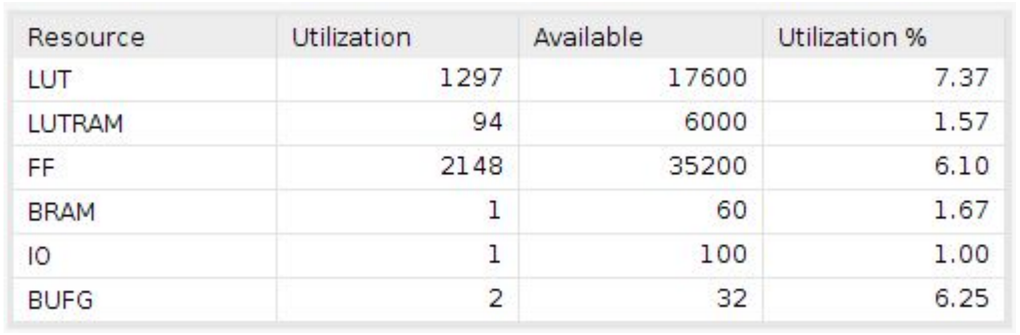


Name	Value
clk_i_tb	1
I_i_tb[7:0]	00
out_o_tb[9:0]	033
ena_i_tb	1
att_i_tb[2:0]	4
N_tb	8
M_tb	10
J_tb	3

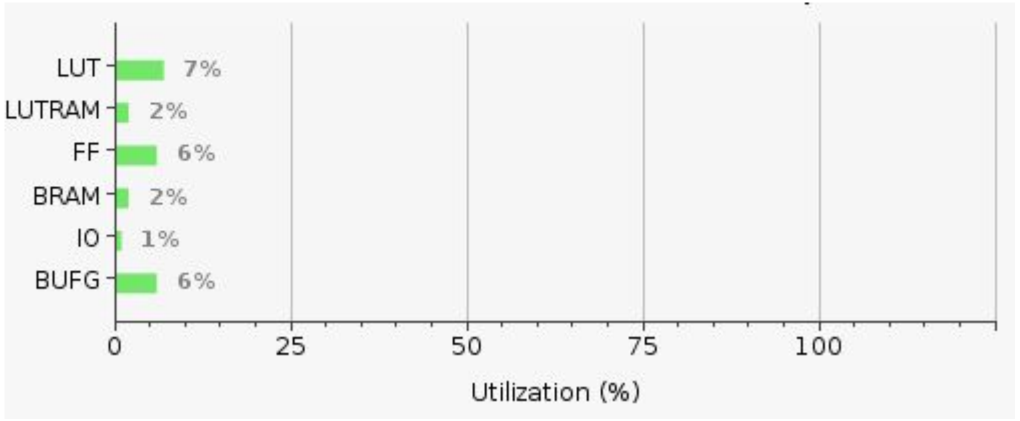
Esquemático con Vivado



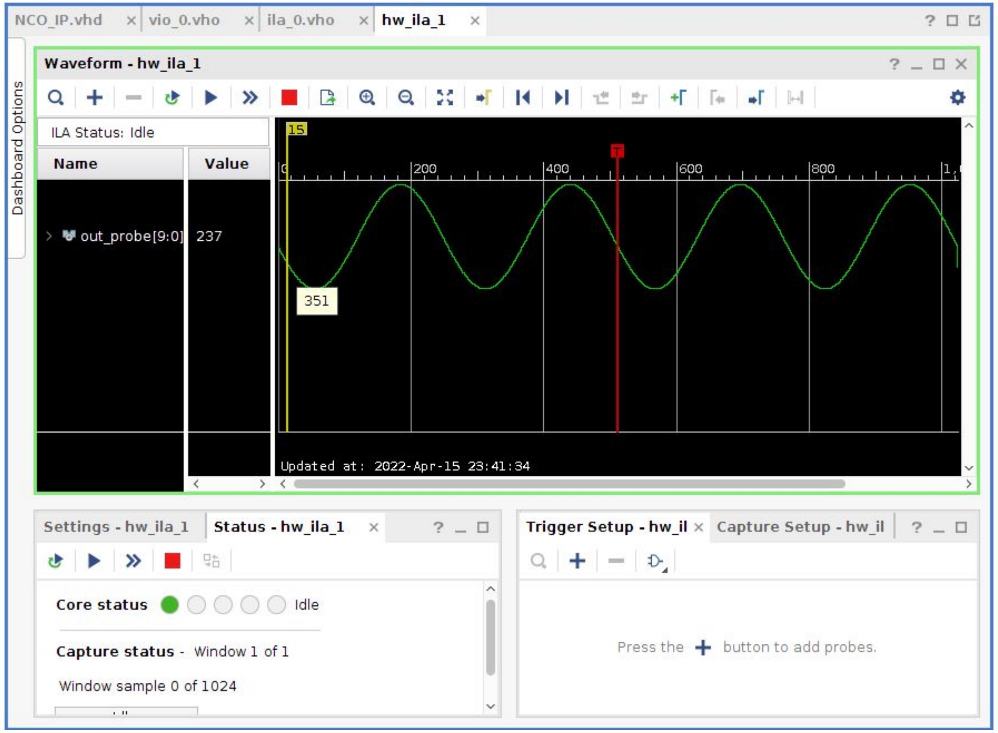
Uso de recursos de la FPGA



Resource	Utilization	Available	Utilization %
LUT	1297	17600	7.37
LUTRAM	94	6000	1.57
FF	2148	35200	6.10
BRAM	1	60	1.67
IO	1	100	1.00
BUFG	2	32	6.25



Prueba con ILA y VIO



ILA Status: Idle

Name	Value
out_probe[9:0]	237

Updated at: 2022-Apr-15 23:41:34

Settings - hw_ila_1 | Status - hw_ila_1

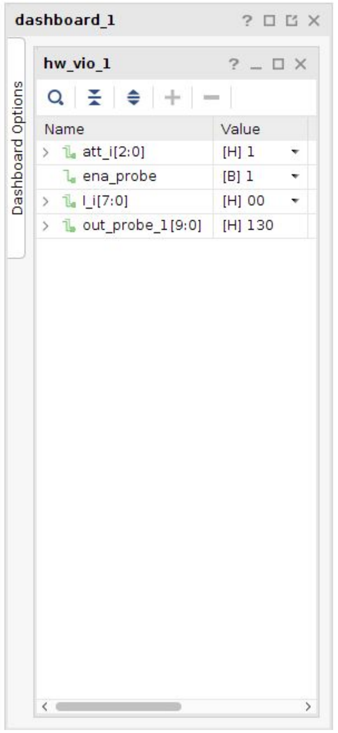
Core status ● Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

Trigger Setup - hw_il | Capture Setup - hw_il

Press the + button to add probes.



Name	Value
att_i[2:0]	[H] 1
ena_probe	[B] 1
I_i[7:0]	[H] 00
out_probe_1[9:0]	[H] 130