

INTEGRANTES:

No	NOMBRE	CARNÉ
1	Cindy Melissa Gatica Arriola	201709692
2		
3		

PRACTICA No. 3 CONTROL DE SEÑALES Y SIMULACIÓN

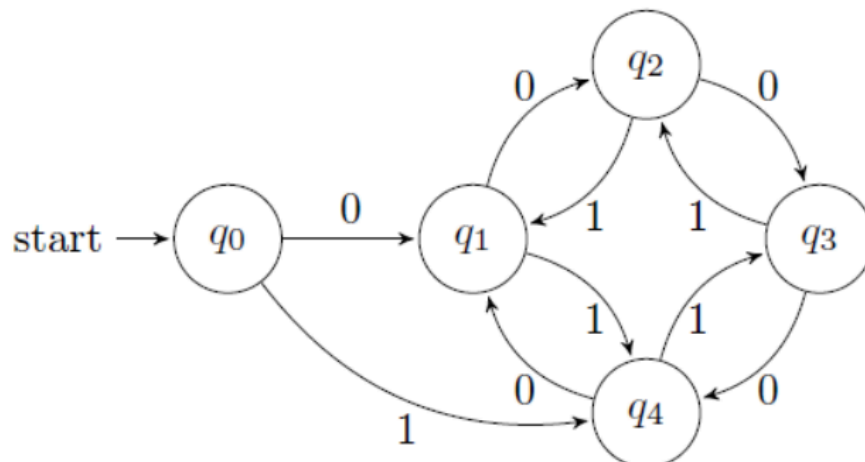
OBJETIVOS

- Que el estudiante se familiarice con el entorno ISE Desing y VHDL
- Utilizar el software de descripción de hardware.
- Implementar el conocimiento adquirido en una aplicación práctica.

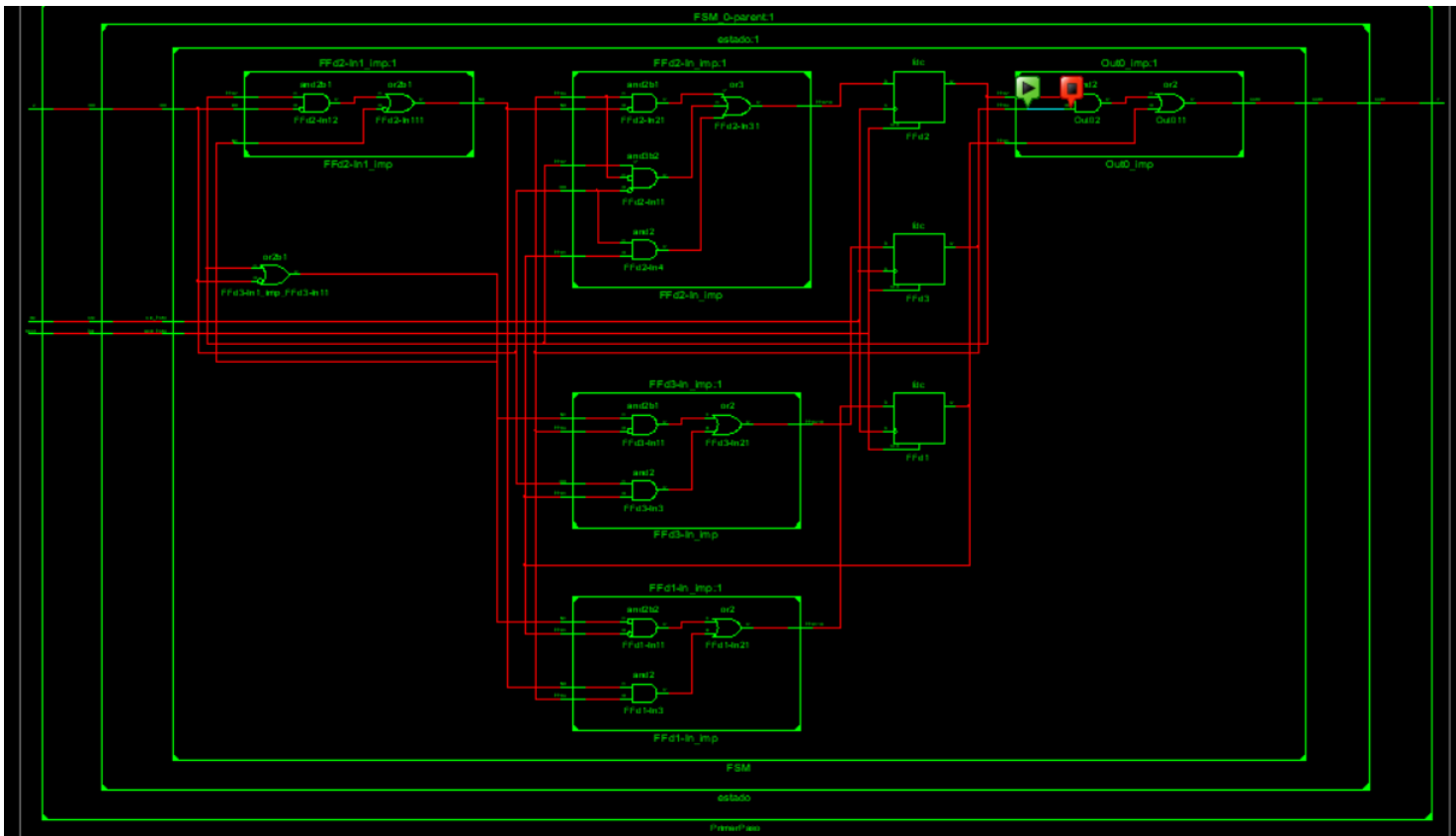
CONOCIMIENTOS PREVIOS

- Circuitos combinacionales
- ISE WebPACK 14.7 Xilinx
- ElbertV2Config
- ISIM
- User Constrains File (UFC).

A continuación se muestra la imagen de la máquina de estados a seguir

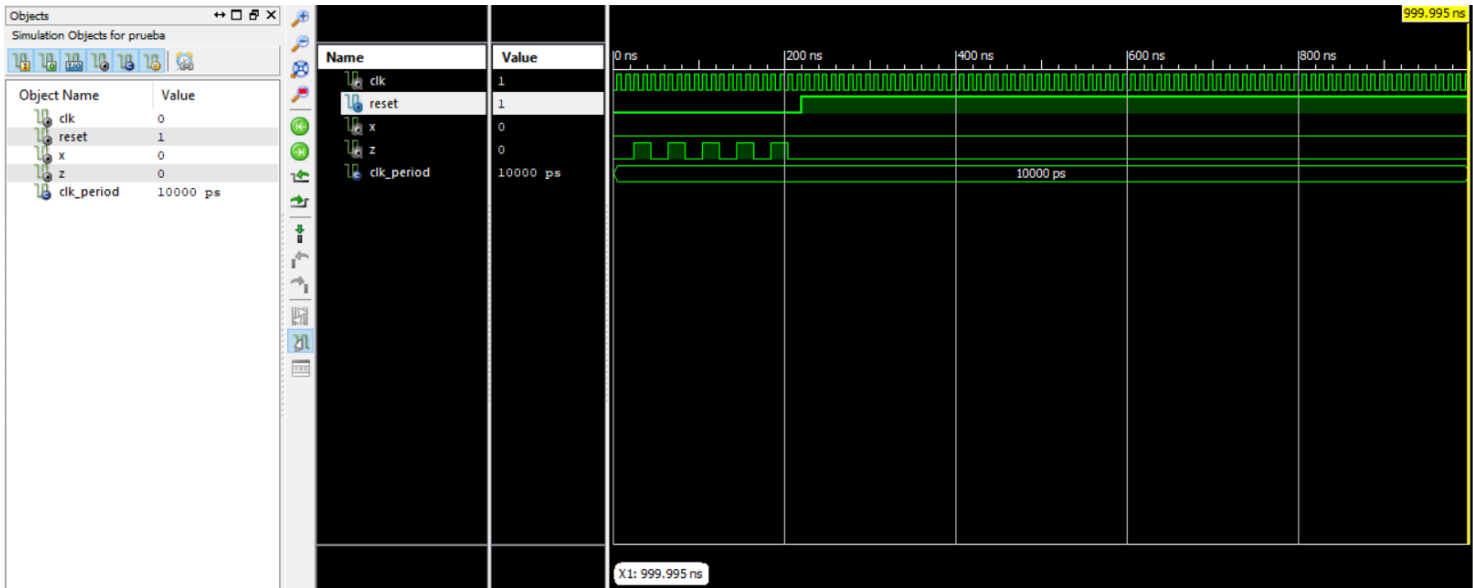


El esquemático RTL se muestra a continuación



Salida de reloj

Se muestra la señal digital de la salida de reloj y las demás variables consideradas x, z, reset



Código utilizado

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity PrimerPaso is
  Port ( clk : in  STD_LOGIC;
        reset : in  STD_LOGIC;
        x : in  STD_LOGIC;
        z : out STD_LOGIC);
end PrimerPaso;

architecture Behavioral of PrimerPaso is
  type estados_t is (q0,q1,q2,q3,q4);
  signal estado, estado_siguiete : estados_t;

begin

  process(x,estado)
  begin
    z <= '0';
    estado_siguiete <= estado;

    case(estado) is
      when q0 =>
        z <= '0';
        if(x='0') then estado_siguiete <= q1;
        else estado_siguiete <= q4;
        end if;
    end if;
  end process;
end Behavioral;
```

```

when q1 =>
  z <= '0';
  if(x='0') then estado_siguiete <= q2;
  else estado_siguiete <= q4;
  end if;
when q2 =>
  z <= '0';
  if(x='0') then estado_siguiete <= q3;
  else estado_siguiete <= q1;
  end if;
when q3 =>
  z <= '1';
  if(x='0') then estado_siguiete <= q4;
  else estado_siguiete <= q2;
  end if;
when q4 =>
  z <= '1';
  if(x='0') then estado_siguiete <= q1;
  else estado_siguiete <= q3;
  end if;
  when others => null;
end case;
end process;

process (clk,reset) begin
  if(reset='1') then
    estado <= q0;
  elsif rising_edge(clk) then
    estado <= estado_siguiete;
  end if;
end process;

end Behavioral;

```

UCF

Se declararon las siguientes variables de entrada y de salida, así como la salida del reloj

```
NET "clk"          LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
NET "z"           LOC = P46  | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "reset"       LOC = P80  | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "x"           LOC = P79  | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

Para el **VHDL Benchmark**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY prueba IS
END prueba;

ARCHITECTURE behavior OF prueba IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT PrimerPaso
    PORT(
        clk : IN  std_logic;
        reset : IN std_logic;
        x : IN std_logic;
        z : OUT std_logic
    );
    END COMPONENT;

    --Inputs
    signal clk : std_logic := '0';
    signal reset : std_logic := '0';
    signal x : std_logic := '0';
```

```

--Outputs
signal z : std_logic;

-- Clock period definitions
constant clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: PrimerPaso PORT MAP (
        clk => clk,
        reset => reset,
        x => x,
        z => z
    );

    -- Clock process definitions
    clk_process :process
    begin
        clk <= '0';
        wait for clk_period/2;
        clk <= '1';
        wait for clk_period/2;
    end process;

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        -- insert stimulus here
        reset <= '0';
        x <= '0';

        wait for 20 ns;
    end process;

```

```
        reset <= '1';  
  
    wait;  
end process;  
  
END;
```

link del funcionamiento

<https://drive.google.com/drive/folders/1E1l0ImXyqSwddFFIJdCTZsS4T3J7bUSM?usp=sharing>