Instituto Politécnico Nacional

Escuela Superior de Cómputo

Arquitectura de Computadoras

Practica 12: Cartas ASM

Nombre: *Sampayo Hernández Mauro*

Grupo: *3CV8*

Profesor: *Nayeli Vega García*

Fecha de entrega: *18 de abril del 2020*

Elementos de la Arquitectura

Código de Implementación

- **Unidad de Control**

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

entity UnidadControl is
    Port ( clk, clr, INI, z, A0 : in STD_LOGIC;
           LA, LB, EA, EB, EC : out STD_LOGIC);
end UnidadControl;

Architecture Behavioral of UnidadControl is

type estados is (e0, e1, e2);
signal edo_act, edo_sig : estados;

begin

    process (clk, clr)
    begin
        if (clr ='1') then
            edo_act <= e0;
        elsif (rising_edge(clk)) then
            edo_act <= edo_sig;
        end if;
    end process;

    process (edo_act, INI, z, A0)
    begin
        LA <= '0';
        EA <= '0';
        LB <= '0';
        EB <= '0';
        EC <= '0';

        case edo_act is
            when e0 =>--Estado 0
                LB <= '1';

                if (INI = '1') then
                    edo_sig <= e1;
                else
                    LA <= '1';
                    edo_sig <= e0;
                end if;
            when e1 =>--Estado 1
                EA <= '1';

                if (z = '1') then
                    edo_sig <= e2;
                else
                    if (A0 = '1') then
                       EB <= '1';
                    end if;
```

```
                edo_sig <= e1;
            end if;
        when e2 =>--Estado 2
            EC <= '1';

            if (INI = '1') then
                edo_sig <= e2;
            else
                edo_sig <= e0;
            end if;
        end case;
    end process;
end Behavioral;
```

- **Arreglo**

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

entity Arreglo is
    Port ( EA, LA, clk, clr : in  STD_LOGIC;
           DA : in  STD_LOGIC_VECTOR (8 downto 0);
           QA : out  STD_LOGIC_VECTOR (8 downto 0));
end Arreglo;

Architecture Behavioral of Arreglo is
begin
    process (clr, clk)
        variable arr : std_logic_vector(8 downto 0);
    begin
        if (clr = '1') then
            arr := (others => '0');
        elsif (rising_edge(clk)) then
            if LA = '1' then
                arr := DA;
            elsif EA = '1' then
                for i in 0 to 8 loop
                    if (i > 7) then
                        arr(i):= '0';
                    else
                       arr(i):= arr(i+1);
                    end if;
                end loop;
            end if;
        end if;

        QA <= arr;
    end process;
end Behavioral;
```

- **Contador**

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

```
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.all;

entity Contador is
    Port ( clk, clr, LB, EB : in  STD_LOGIC;
           QB : out  STD_LOGIC_VECTOR (3 downto 0));
end Contador;

Architecture Behavioral of Contador is
begin
    process (clk, clr)
        variable DB : STD_LOGIC_VECTOR (3 downto 0);
    begin
        if (clr = '1') then
            DB := "0000";
        elsif (rising_edge(clk)) then
            if LB = '1' then
                DB := "0000";
            elsif EB = '1' then
                DB := DB + 1;
            end if;
        end if;

        QB <= DB;
    end process;
end Behavioral;
```

- **Decodificador**

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

entity Decodificador is
    Port ( QB : in  STD_LOGIC_VECTOR (3 downto 0);
           deco : out  STD_LOGIC_VECTOR(6 downto 0));
end Decodificador;

Architecture Behavioral of Decodificador is

    constant d0: STD_LOGIC_VECTOR :="0000001";
    constant d1: STD_LOGIC_VECTOR :="1001111";
    constant d2: STD_LOGIC_VECTOR :="0010010";
    constant d3: STD_LOGIC_VECTOR :="0000110";
    constant d4: STD_LOGIC_VECTOR :="1001100";
    constant d5: STD_LOGIC_VECTOR :="0100100";
    constant d6: STD_LOGIC_VECTOR :="0100000";
    constant d7: STD_LOGIC_VECTOR :="0001111";
    constant d8: STD_LOGIC_VECTOR :="0000000";
    constant d9: STD_LOGIC_VECTOR :="0000100";
    constant di: STD_LOGIC_VECTOR :="0110110";--ERROR

begin
    deco <=  d0 when QB = x"0" else
             d1 when QB = x"1" else
             d2 when QB = x"2" else
             d3 when QB = x"3" else
```

```
                d4 when QB = x"4" else
                d5 when QB = x"5" else
                d6 when QB = x"6" else
                d7 when QB = x"7" else
                d8 when QB = x"8" else
                d9 when QB = x"9" else
                di;
end Behavioral;
```

- **Multiplexor**

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

entity Mux is
    Port ( deco : in  STD_LOGIC_VECTOR (6 downto 0);
           OP: in STD_LOGIC;
           Q : out  STD_LOGIC_VECTOR (6 downto 0));
end Mux;

Architecture Behavioral of Mux is
begin
    Q <= deco when OP = '1' else "1111110";--guion "-"
end Behavioral;
```

- **Bandera Z**

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

entity BanderaZ is
    Port ( Cz : in  STD_LOGIC_VECTOR (8 downto 0);
           z : out  STD_LOGIC);
end BanderaZ;

Architecture Behavioral of BanderaZ is
begin
    z <= NOT(Cz(8) OR Cz(7) OR Cz(6) OR Cz(5) OR Cz(4) OR Cz(3) OR Cz(2)
OR Cz(1) OR Cz(0));
end Behavioral;
```

Código de Simulación

- **Unidad de Control**

```
    signal A0 : STD_LOGIC := '0';
    signal z : STD_LOGIC := '0';

    --Outputs
    signal LA : STD_LOGIC;
    signal LB : STD_LOGIC;
    signal EA : STD_LOGIC;
    signal EB : STD_LOGIC;
    signal EC : STD_LOGIC;
```

```
    -- Clock period definitions
    constant CLK_period : time := 10 ns;

begin
    -- Instantiate the Unit Under Test (UUT)
    uut: UnidadControl
        Port Map ( clk => clk,
                   clr => clr,
                   INI => INI,
                   A0 => A0,
                   z => z,
                   LB => LB,
                   LA => LA,
                   EA => EA,
                   EB => EB,
                   EC => EC);

    -- 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
        clr <= '1';
        wait for 30 ns;
        clr <= '0';
        wait for 10 ns;
        --estado 0 -> estado 1
        INI <= '1';
        A0 <= '1';
        z <= '0';
        wait for 50 ns;
        --estado 1 -> estado 2
        INI <= '1';
        A0 <= '0';
        z <= '1';
        wait for 50 ns;
        --estado 2 -> estado 0
        INI <= '0';
        wait for 50 ns;
        wait;
   end process;
end Behavioral;
```

- **Arreglo**

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
```

```
entity Arreglo_tb is
end Arreglo_tb;

Architecture Behavioral of Arreglo_tb is

    component Arreglo is
        Port ( EA, LA, clk, clr : in  STD_LOGIC;
               DA : in  STD_LOGIC_VECTOR (8 downto 0);
               QA : out  STD_LOGIC_VECTOR (8 downto 0));
    end component;

    --Inputs
    signal clk : STD_LOGIC := '0';
    signal clr : STD_LOGIC := '0';
    signal EA : STD_LOGIC := '0';
    signal LA : STD_LOGIC := '0';
    signal DA : STD_LOGIC_VECTOR (8 downto 0) := (others => '1');

    --Outputs
    signal QA : STD_LOGIC_VECTOR(8 downto 0);

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

begin
    -- Instantiate the Unit Under Test (UUT)
    uut: Arreglo
        Port Map ( clk => clk,
                   clr => clr,
                   EA => EA,
                   LA => LA,
                   DA => DA,
                   QA => QA);

    -- 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
        LA <= '1';
        wait for 10 ns;
        LA <= '0';
        wait for 10 ns;
        clr <= '1';
        wait for 10 ns;
        clr <= '0';
        wait for 10 ns;
        LA <= '1';
        wait for 10 ns;
        LA <= '0';
```

```
        wait for 10 ns;
        EA <= '1';
        wait for 30 ns;
        EA <= '0';
        wait for 10 ns;
        EA <= '1';
        wait for 30 ns;
        EA <= '0';
        wait for 10 ns;
        wait;
   end process;
end Behavioral;
```

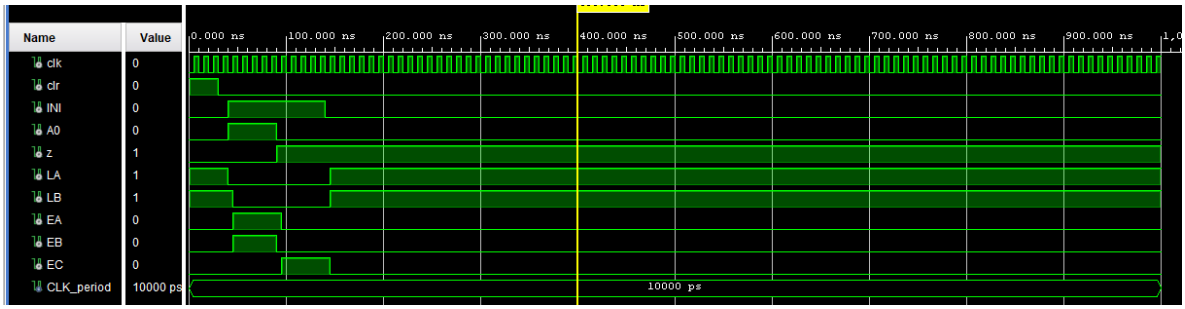
- **Contador**

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

entity Contador_tb is
end Contador_tb;

Architecture Behavioral of Contador_tb is

    component Contador is
        Port ( clk, clr, LB, EB : in STD_LOGIC;
               QB : out STD_LOGIC_VECTOR (3 downto 0));
    end component;

    --Inputs
    signal clk : STD_LOGIC := '0';
    signal clr : STD_LOGIC := '0';
    signal LB : STD_LOGIC := '0';
    signal EB : STD_LOGIC := '0';

    --Outputs
    signal QB : STD_LOGIC_VECTOR(3 downto 0);

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

begin
    -- Instantiate the Unit Under Test (UUT)
    uut: Contador
        Port Map ( clk => clk,
                   clr => clr,
                   LB => LB,
                   EB => EB,
                   QB => QB);

    -- Clock process definitions
    CLK_process :process
    begin
        clk <= '0';
        wait for CLK_period/2;
        clk <= '1';
        wait for CLK_period/2;
```

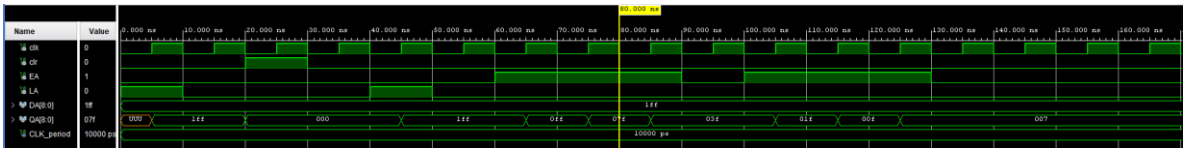
```
    end process;


    -- Stimulus process
    stim_proc: process
    begin
        clr <= '1';
        wait for 30 ns;
        clr <= '0';
        wait for 30 ns;
        LB <= '0';
        EB <= '1';
        wait for 100 ns;
        EB <= '1';
        wait for 100 ns;
        EB <= '0';
        LB <= '1';
        wait for 100 ns;
        clr <= '1';
        wait for 30 ns;
        clr <= '0';
        wait;
   end process;
end Behavioral;
```

Simulación:

- **Unidad de Control**



- **Arreglo**



- **Contador**

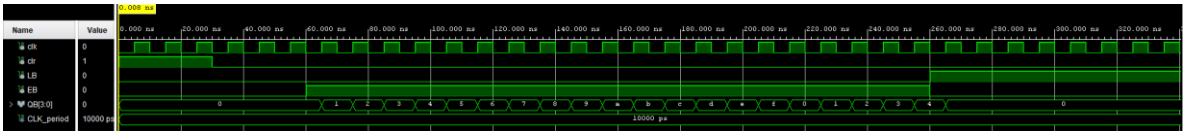
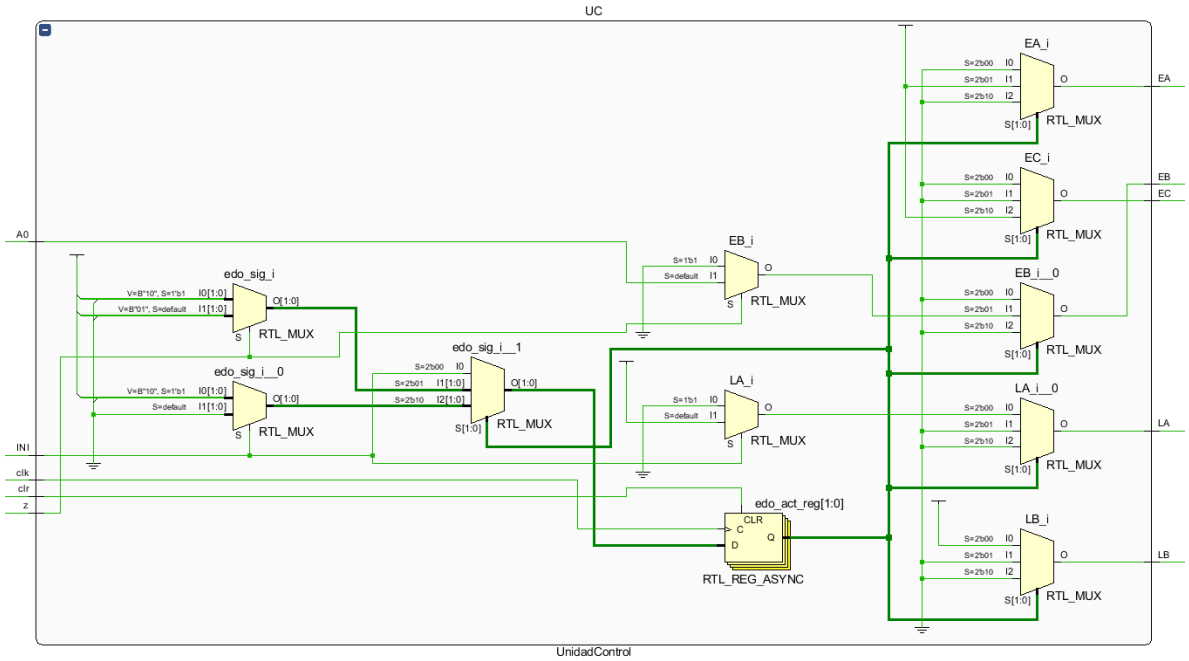
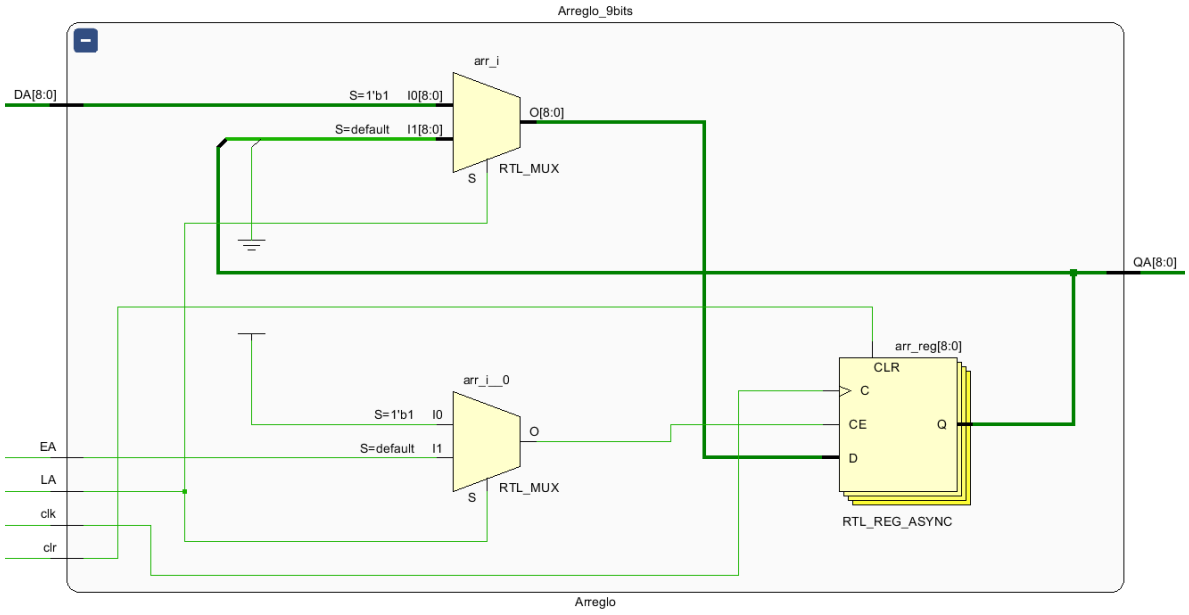


Diagrama RTL:

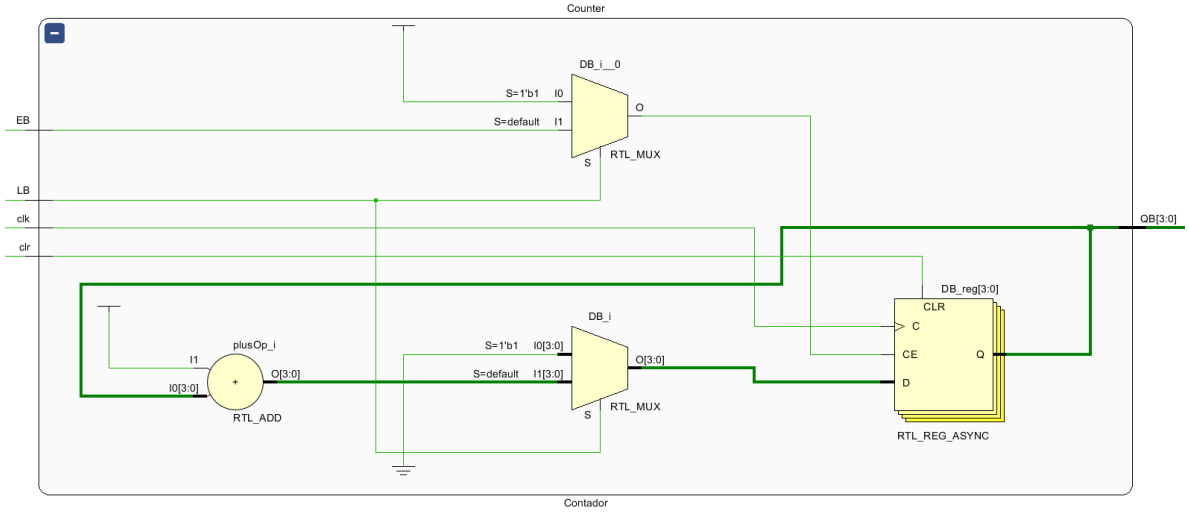
- **Unidad de Control**



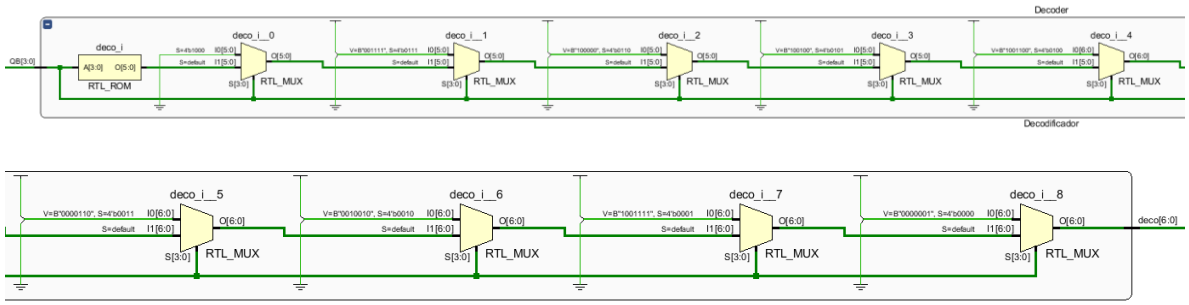
- **Arreglo**



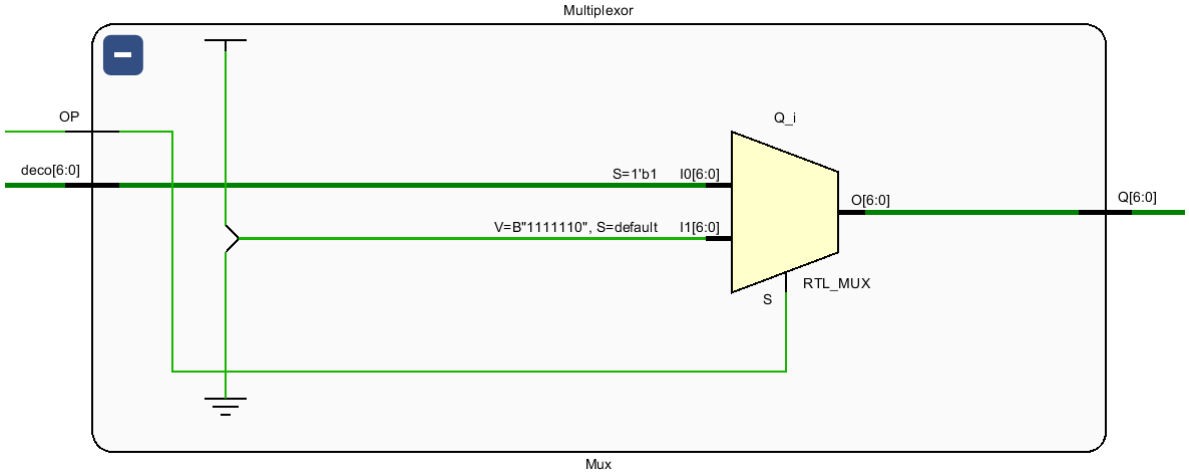
- **Contador**



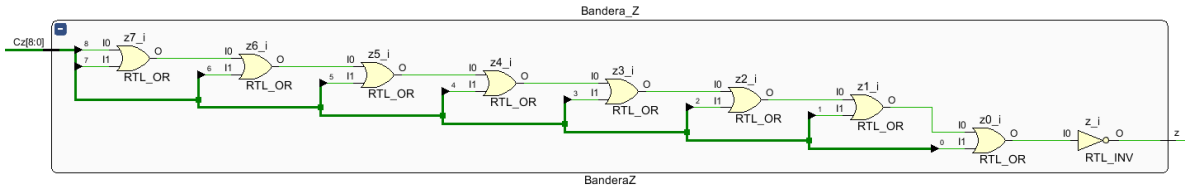
- **Decodificador**



- **Multiplexor**



- **Bandera Z**



Arquitectura Completa

Código de Implementación

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

entity cartaASM is
    Port ( INI, clk, clr : in  STD_LOGIC;
           D : in  STD_LOGIC_VECTOR (8 downto 0);
           A : out STD_LOGIC_VECTOR (8 downto 0);--Salida Arreglo
           B : out STD_LOGIC_VECTOR (6 downto 0));--Salida Display(Mux)
end cartaASM;

Architecture Behavioral of cartaASM is

    component Arreglo is
        Port ( EA, LA, clk, clr : in  STD_LOGIC;
               DA : in  STD_LOGIC_VECTOR (8 downto 0);
               QA : out  STD_LOGIC_VECTOR (8 downto 0));
    end component;

    component Contador is
        Port ( clk, clr, LB, EB : in  STD_LOGIC;
               QB : out  STD_LOGIC_VECTOR (3 downto 0));
    end component;

    component Decodificador is
        Port ( QB : in  STD_LOGIC_VECTOR (3 downto 0);
               deco : out  STD_LOGIC_VECTOR(6 downto 0));
    end component;

    component Mux is
        Port ( deco : in  STD_LOGIC_VECTOR (6 downto 0);
               OP: in STD_LOGIC;
               Q : out  STD_LOGIC_VECTOR (6 downto 0));
    end component;

    component UnidadControl is
        Port ( clk, clr, INI, z, A0 : in STD_LOGIC;
               LA, LB, EA, EB, EC : out STD_LOGIC);
    end component;

    component BanderaZ is
        Port ( Cz : in  STD_LOGIC_VECTOR (8 downto 0);
               z : out  STD_LOGIC);
    end component;
```

```
    signal A0, z, LA, LB, EA, EB, EC: STD_LOGIC;
    signal B1: STD_LOGIC_VECTOR(3 DOWNTO 0);
    signal Cz: STD_LOGIC_VECTOR(8 downto 0);
    signal B_aux: STD_LOGIC_VECTOR(6 downto 0);--auxiliar de la salida
del Dsiplay(B) para evitar corto circuito uwu

begin
    A <= Cz;
    A0 <= Cz(0);

    Bandera_Z : BanderaZ
       Port map( Cz => Cz,
                 z => z);

    UC : UnidadControl
        Port Map( clk => clk,
                  clr => clr,
                  INI => INI,
                  A0 => A0,
                  z => z,
                  LA => LA,-- A=D
                  LB => LB,-- B=0
                  EA => EA,-- A>>1
                  EB => EB,-- B++
                  EC => EC);-- Mostrar B

    Counter : Contador
        Port Map( clk => clk,
                  clr => clr,
                  LB => LB,
                  EB => EB,
                  QB => B1);

    Arreglo_9bits : Arreglo
        Port MAP( clk => clk,
                  clr => clr,
                  DA => D,
                  EA => EA,
                  LA => LA,
                  QA => Cz);

    Decoder: Decodificador
        Port Map( QB => B1,
                  deco => B_aux);

    Multiplexor: Mux
        Port Map( deco => B_aux,
                  OP => EC,
                  Q => B);
end Behavioral;
```

Código de Simulación

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
```

```vhdl
entity cartaASM_tb is
end cartaASM_tb;

Architecture Behavioral of cartaASM_tb is

    component cartaASM is
        Port ( INI, clk, clr : in  STD_LOGIC;
               D : in  STD_LOGIC_VECTOR (8 downto 0);
               A : out STD_LOGIC_VECTOR (8 downto 0);
               B : out STD_LOGIC_VECTOR (6 downto 0));
    end component;

    --Inputs
    signal clk : STD_LOGIC := '0';
    signal clr : STD_LOGIC := '0';
    signal INI : STD_LOGIC := '0';
    signal D : STD_LOGIC_VECTOR(8 downto 0) := (others => '0');

    --Outputs
    signal A : STD_LOGIC_VECTOR(8 downto 0);
    signal B : STD_LOGIC_VECTOR(6 downto 0);

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

begin
    -- Instantiate the Unit Under Test (UUT)
    uut: cartaASM
        Port Map ( clk => clk,
                   clr => clr,
                   INI => INI,
                   D => D,
                   A => A,
                   B => B);

    -- 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
        clr <= '1';
        wait for 10 ns;
        clr <= '0';
        d <= "101101011";-- inciso a
        wait for 50 ns;
        ini <= '1';
        wait for 150 ns;
        ini <= '0';
        d <= "000011101";--inciso b
```

```
        wait for 50 ns;
        ini <= '1';
        wait for 150 ns;
        ini <= '0';
        d <= "000010000";--inciso c
        wait for 50 ns;
        ini <= '1';
        wait for 150 ns;
        ini <= '0';
        d <= "100001000";--inciso d
        wait for 50 ns;
        ini <= '1';
        wait for 150 ns;
        ini <= '0';
        d <= "000000000";--inciso e
        wait for 50 ns;
        wait;
    end process;
end Behavioral;
```

Simulación:

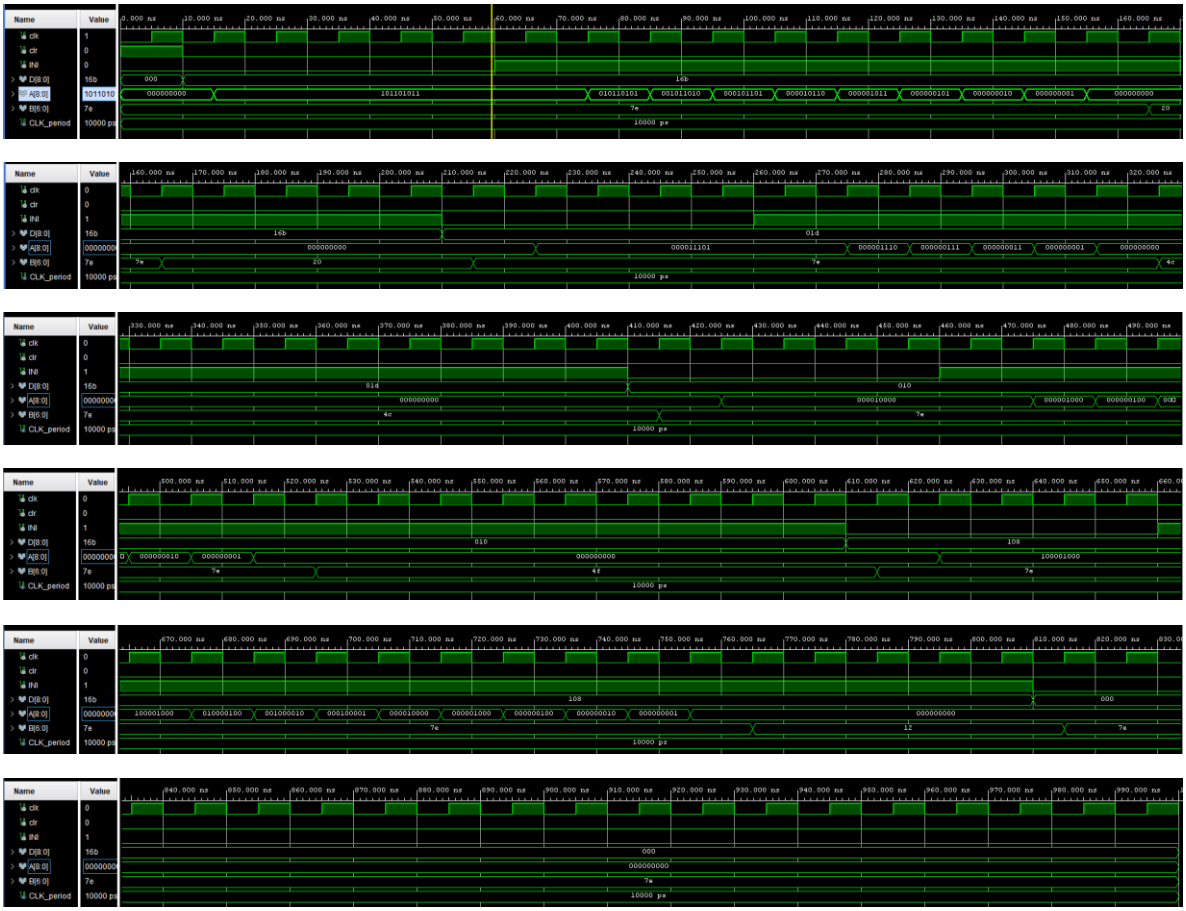


Diagrama RTL: