Código de Implementación:

Código Completa:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE work.paquete.all;

entity contadorUnos is
    Port ( D : in STD_LOGIC_VECTOR (8 downto 0);
           Ini,clr,clk : in STD_LOGIC;
           QA : out STD_LOGIC_VECTOR (8 downto 0);
           dispFin : out STD_LOGIC_VECTOR (6 downto 0));
end contadorUnos;

architecture Behavioral of contadorUnos is

SIGNAL UC_LA : STD_LOGIC;
SIGNAL UC_EA : STD_LOGIC;
SIGNAL UC_Z : STD_LOGIC;
SIGNAL UC_LB : STD_LOGIC;
SIGNAL UC_EB : STD_LOGIC;
SIGNAL UC_EC : STD_LOGIC;
--**SEÑALES UNIDAD DE CONTROL

SIGNAL salidaContador : STD_LOGIC_VECTOR(3 downto 0);
--**SEÑALES CONTADOR

SIGNAL salidaArreglo : STD_LOGIC_VECTOR(8 downto 0);
--**SEÑALES ARREGLO

SIGNAL salidaDecodificador : STD_LOGIC_VECTOR(6 downto 0);
--**SEÑALES DECODIFICADOR

SIGNAL reloj : STD_LOGIC;
begin
UC_z <= not(salidaArreglo(8) or salidaArreglo(7) or salidaArreglo(6) or salidaArreglo(5) or salidaArreglo(4)
or salidaArreglo(3) or salidaArreglo(2) or salidaArreglo(1) or salidaArreglo(0));

Divisor : divisorFrecuencia
    Port map(
        clr => clr,
        clk => clk,
        frecuenciaFin => reloj
    );
--**DIVISOR DE FRECUENCIA

UC : cartaASM
    Port map(
        Ini => Ini,
        clr => clr,
        clk => reloj,
        LA => UC_LA,
        EA => UC_EA,
```

```
        A0 => salidaArreglo(0),
        Z => UC_Z,
        LB => UC_LB,
        EB => UC_EB,
        EC => UC_EC
    );

Cont : Contador
    Port map (
    clk => reloj,
    clr => clr,
    LB => UC_LB,
    EB => UC_EB,
    QB => salidaContador
    );

Arregl : Arreglo
    Port map (
        DA => D,
        clk => reloj,
        clr => clr,
        LA => UC_LA,
        EA => UC_EA,
        QA => salidaArreglo
    );
--**ARREGLO

Decodificado : Decodificador
    Port map(
        salidaCont => salidaContador,
        display => salidaDecodificador
    );
--**DECODIFICADOR

mu : mux
    Port map(
    salidaDeco => salidaDecodificador,
    op => UC_EC,
    salidaMux => dispFin
    );
--**MUX
QA <= salidaArreglo;
end Behavioral;
```

Código Divisor:

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

entity divisorFrecuencia is
    Port ( clr, clk : in STD_LOGIC;
           frecuenciaFin : out STD_LOGIC);
end divisorFrecuencia;

architecture Behavioral of divisorFrecuencia is
```

```
    signal temp : std_logic := '0';
    signal conta : integer range 0 to 49999 := 0;
begin
    process(clr,clk) is
    begin
        if(clr = '0')then
            temp <= '0';
            conta <= 0;
        elsif(rising_edge(clk)) then
            if(conta = 49999) then
                temp <= not temp;
                conta <= 0;
            else
                conta <= conta + 1;
            end if;
        end if;
    end process;
frecuenciaFin <= temp;
end Behavioral;
```

Código UC:

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

entity cartaASM is
    Port ( clk,clr,ini,z,a0 : in STD_LOGIC;
           la,lb,ea,eb,ec : out STD_LOGIC);
end cartaASM;

architecture Behavioral of cartaASM 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;

UC : process(edo_act,ini,z,a0)
begin

    LA <= '0';
    EA <= '0';
    LB <= '0';
    EB <= '0';
    EC <= '0';
    case edo_act is

        when e0 =>
```

```
            LB <= '1';
            if(ini = '1') then
               edo_sig <= e1;
            else
               LA <= '1';
               edo_sig <= e0;
            end if;
         when e1 =>
            EA <= '1';
               if(z = '1') then
                  edo_sig <= e2;
               elsif(a0 = '1') then
                  EB <= '1';
                  edo_sig <= e1;
               else
                  edo_sig <= e1;
               end if;
          when e2 =>
            EC <= '1';
               if(ini = '1') then
                  edo_sig <= e2;
               else
                  edo_sig <= e0;
               end if;
      end case;

end process;

end Behavioral;
```

Código Contador:

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

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

architecture Behavioral of Contador is
signal DB : STD_LOGIC_VECTOR(3 downto 0);
begin
DB <= (others => '0');
process(clk,clr)
begin
    if(clr = '1') then
        QB <= (others => '0');
    elsif rising_edge(clk) then
        if(LB = '0' and EB = '0') then
            QB <= QB;
        elsif(LB = '1' and EB = '0') then
            QB <= DB;
        elsif(LB = '0' and EB = '1') then
```

```
            QB <= QB + 1;
        end if;
    end if;
end process;

end Behavioral;
```

Código Arreglo:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Arreglo is
    Port ( DA : in STD_LOGIC_VECTOR (8 downto 0);
           la,ea,clk,clr : in STD_LOGIC;
           QA : inout STD_LOGIC_VECTOR (8 downto 0));
end Arreglo;

architecture Behavioral of Arreglo is
begin
process ( clr,clk)
begin
    if( clr = '1') then
        QA <= (others => '0');
    elsif rising_edge(clk) then
        if(LA = '0' and EA = '0') then
            QA <= QA;
         elsif(LA = '1' and EA = '0') then
            QA <= DA;
         elsif(LA = '0' and EA = '1')then
            for i in 0 to 7 loop
                                QA(i)<=QA(i+1);
                               end loop;
                                QA(8)<='0';
        end if;
    end if;
end process;

end Behavioral;
```

Código MUX:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux is
    Port ( salidaDeco : in STD_LOGIC_VECTOR (6 downto 0);
           op : in STD_LOGIC;
           salidaMux : out STD_LOGIC_VECTOR (6 downto 0));
end mux;

architecture Behavioral of mux is
constant def : STD_LOGIC_VECTOR(6 DOWNTO 0) := "1111110";
begin
    salidaMux <= salidaDeco when op = '1' else def;
end Behavioral;
```

Códigos de simulación:

Código Completa:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_UC is
--  Port ( );
end TB_UC;

architecture Behavioral of TB_UC is

component contadorUnos is
    Port ( D : in STD_LOGIC_VECTOR (8 downto 0);
           Ini,clr,clk : in STD_LOGIC;
           QA : out STD_LOGIC_VECTOR (8 downto 0);
           dispFin : out STD_LOGIC_VECTOR (6 downto 0));
end component;
SIGNAL D : STD_LOGIC_VECTOR (8 downto 0);
SIGNAL Ini,clr,clk : STD_LOGIC;
SIGNAL QA : STD_LOGIC_VECTOR (8 downto 0);
SIGNAL dispFin : STD_LOGIC_VECTOR (6 downto 0);
constant CLK_period : time := 10 ns;
begin

u1 : contadorUnos
    Port map(
    D => D,
    ini => ini,
    clr => clr,
    clk => clk,
    QA => QA,
    dispFin => dispFin
    );

CLK_process :process
   begin
     CLK <= '0';
     wait for CLK_period/2;
     CLK <= '1';
     wait for CLK_period/2;
   end process;

  entradas: process
  begin
  clr<='1';
  d<="101101011";
  ini<='0';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  ini<='1';
  wait for 120 ns;
```

```
  clr<='1';
  d<="000011101";
  ini<='0';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  ini<='1';
  wait for 120 ns;


  clr<='1';
  d<="000010000";
  ini<='0';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  ini<='1';
  wait for 120 ns;


  clr<='1';
  d<="100001000";
  ini<='0';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  ini<='1';
  wait for 120 ns;


  clr<='1';
  d<="000000000";
  ini<='0';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  ini<='1';
  wait for 120 ns;

  wait;
end process;
end Behavioral;
```

Código UC:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_UC_S is
    --PORT
end TB_UC_S;

architecture Behavioral of TB_UC_S is
component cartaASM is
```

```
    Port ( clk,clr,ini,z,a0 : in STD_LOGIC;
           la,lb,ea,eb,ec : out STD_LOGIC);
end component;
SIGNAL clk,clr,ini,z,a0 : STD_LOGIC;
SIGNAL la,lb,ea,eb,ec : STD_LOGIC;
constant clk_period: time := 10 ns;

begin
u1 : cartaASM
    Port map(
    clk => clk,
    clr => clr,
    ini => ini,
    z => z,
    a0 => a0,
    la => la,
    lb => lb,
    ea => ea,
    eb => eb,
    ec => ec
    );

CLK_process :process
   begin
     CLK <= '0';
     wait for CLK_period/2;
     CLK <= '1';
     wait for CLK_period/2;
 end process;

 stimulus: process
  begin
    ini<='0';
    a0<='0';
    z<='0';
            clr<='1';
    wait for 30 ns;
    clr<='0';
            wait for 60 ns;
    ini<='1';
            wait for 10 ns;
    ini<='0';
    wait for 50 ns;
    a0<='1';
            wait for 10 ns;
    a0<='0';
            wait for 20 ns;
            a0<='1';
            wait for 10 ns;
    a0<='0';
            wait for 120 ns;
    z<='1';
    wait;
  end process;
end Behavioral;
```

Código Arreglo:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_Arreglo is
--  Port ( );
end TB_Arreglo;

architecture Behavioral of TB_Arreglo is
component Arreglo is
    Port ( DA : in STD_LOGIC_VECTOR (8 downto 0);
           la,ea,clk,clr : in STD_LOGIC;
           QA : inout STD_LOGIC_VECTOR (8 downto 0));
end component;

SIGNAL DA : STD_LOGIC_VECTOR (8 downto 0);
SIGNAL la,ea,clk,clr : STD_LOGIC;
SIGNAL QA : STD_LOGIC_VECTOR (8 downto 0);
constant clk_period: time := 10 ns;
begin
u2 : Arreglo
    Port map(
        DA => DA,
        la => la,
        ea => ea,
        clk => clk,
        clr => clr,
        QA => QA
    );

   CLK_process :process
   begin
     CLK <= '0';
     wait for CLK_period/2;
     CLK <= '1';
     wait for CLK_period/2;
   end process;

     stimulus: process
  begin
    LA <= '0';
    EA <= '0';
    DA <= "001001110";
            clr<='1';
    wait for 30 ns;
    clr<='0';
            wait for 60 ns;
    LA<='1';
            wait for 50 ns;
    LA <='0';
    EA <= '1';
    wait for 50 ns;
    DA <= "011101110";
            clr<='1';
    wait for 30 ns;
```

```
   clr<='0';
            wait for 60 ns;
   LA<='1';
            wait for 50 ns;
   LA <='0';
   EA <= '1';
   wait for 50 ns;
   wait;
  end process;

end Behavioral;
```

Código Contador:

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

entity TB_Contador is
--  Port ( );
end TB_Contador;

architecture Behavioral of TB_Contador is
component Contador is
    Port ( LB,EB,clr,clk : in STD_LOGIC;
           QB : inout STD_LOGIC_VECTOR (3 downto 0));
end component;
SIGNAL LB,EB,clr,clk : STD_LOGIC;
SIGNAL QB : STD_LOGIC_VECTOR (3 downto 0);
constant CLK_period : time := 10 ns;
begin
u3 : Contador
    Port map(
        LB => LB,
        EB => EB,
        clr => clr,
        clk => clk,
        QB => QB
    );
   CLK_process :process
   begin
     CLK <= '0';
     wait for CLK_period/2;
     CLK <= '1';
     wait for CLK_period/2;
   end process;


  stimulus: process
  begin
  clr<='1';
  wait for 30 ns;
  clr<='0';
  wait for 30 ns;
  LB<='0';
  EB<='0';
```

```
  wait for 30 ns;
  LB<='1';
  EB<='0';
  wait for 30 ns;
  LB<='0';
  EB<='1';
  wait for 30 ns;
  LB<='0';
  EB<='1';
  wait for 30 ns;
  LB<='0';
  EB<='1';
  wait for 30 ns;
  wait;
  end process;
end Behavioral;
```

Código Decodificador:

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

entity TB_Decodificador is
--  Port ( );
end TB_Decodificador;

architecture Behavioral of TB_Decodificador is
component Decodificador is
    Port ( salidaCont : in STD_LOGIC_VECTOR (3 downto 0);
           display : out STD_LOGIC_VECTOR (6 downto 0));
end component;
SIGNAL salidaCont : STD_LOGIC_VECTOR (3 downto 0) := "0000";
SIGNAL display : STD_LOGIC_VECTOR (6 downto 0) := "0000000";

begin

u4 : Decodificador
    Port map(
        salidaCont => salidaCont,
        display => display
    );


  stimulus: process
  begin
  salidaCont <= "0000";
  wait for 30 ns;
  salidaCont <= "0001";
   wait for 30 ns;
  salidaCont <= "0010";
   wait for 30 ns;
  salidaCont <= "0011";
   wait for 30 ns;
  salidaCont <= "0100";
   wait for 30 ns;
```

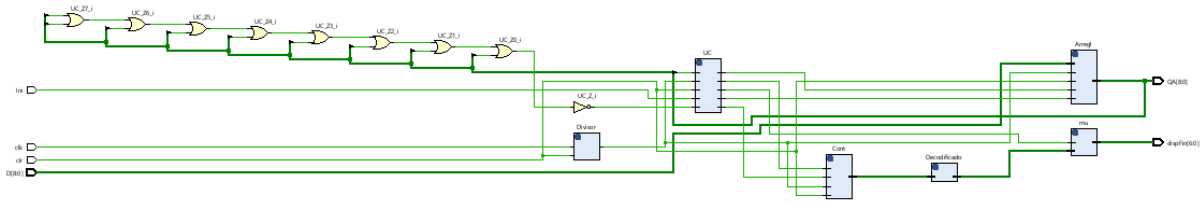
```
 salidaCont <= "0101";
  wait for 30 ns;
 salidaCont <= "0110";
 wait;
 end process;

end Behavioral;
```

Código Mux:

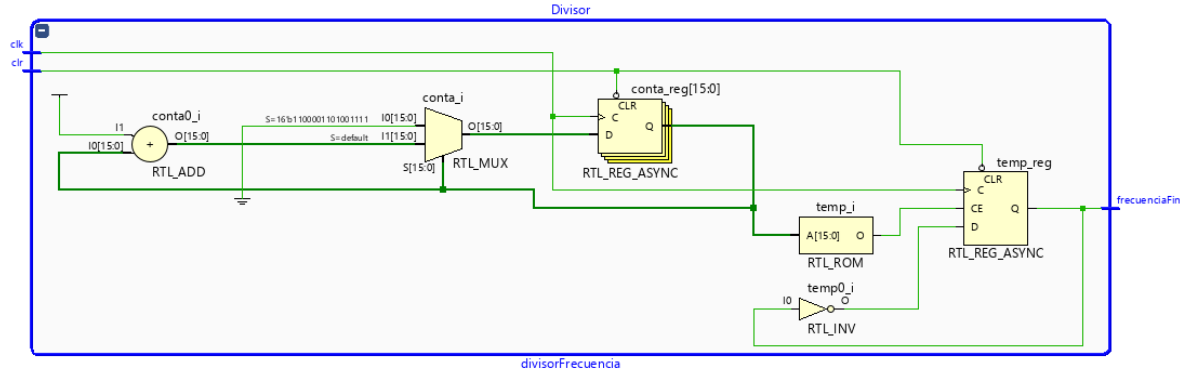
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TB_Mux is
--  Port ( );
end TB_Mux;

architecture Behavioral of TB_Mux is
component mux is
    Port ( salidaDeco : in STD_LOGIC_VECTOR (6 downto 0);
           op : in STD_LOGIC;
           salidaMux : out STD_LOGIC_VECTOR (6 downto 0));
end component;
SIGNAL salidaDeco : STD_LOGIC_VECTOR (6 downto 0) := "0000000";
SIGNAL op : STD_LOGIC := '0';
SIGNAL salidaMux : STD_LOGIC_VECTOR (6 downto 0) := "0000000";
begin
u5 : mux
   Port map(
        salidaDeco => salidaDeco,
        op => op,
        salidaMux => salidaMux
   );


 stimulus: process
 begin

 op <= '0';
 wait for 70 ns;
 salidaDeco <= "0010101";
 op <= '1';
 wait for 70 ns;
 salidaDeco <= "1111110";
 op <= '0';
 wait for 30 ns;
 op <= '1';
 wait;
 end process;
end Behavioral;
```

Diagramas RTL:

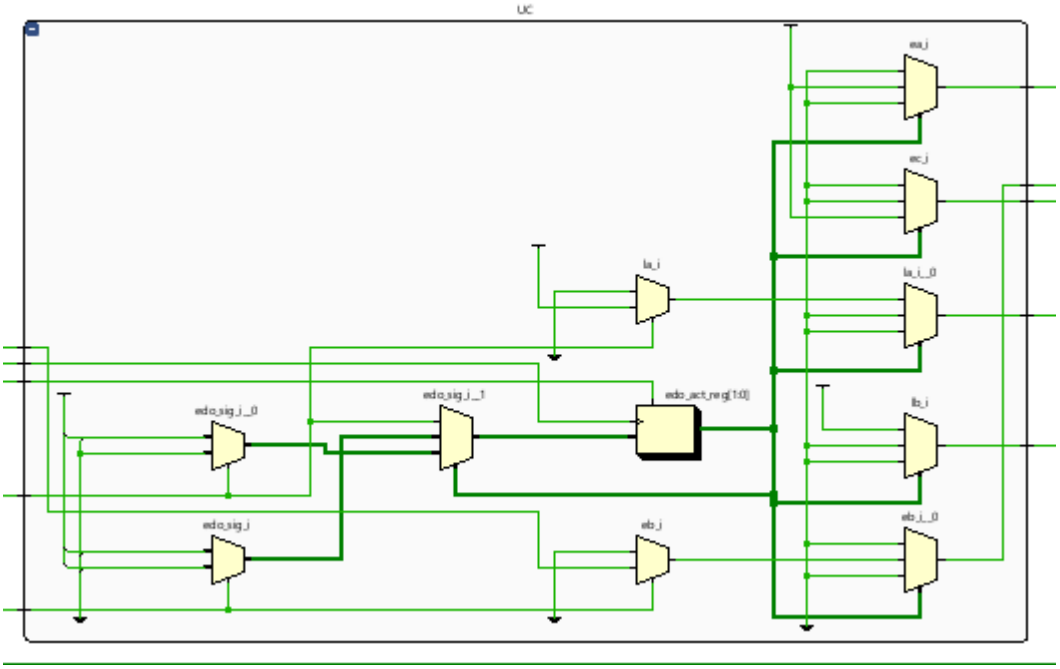
RTL Completa:



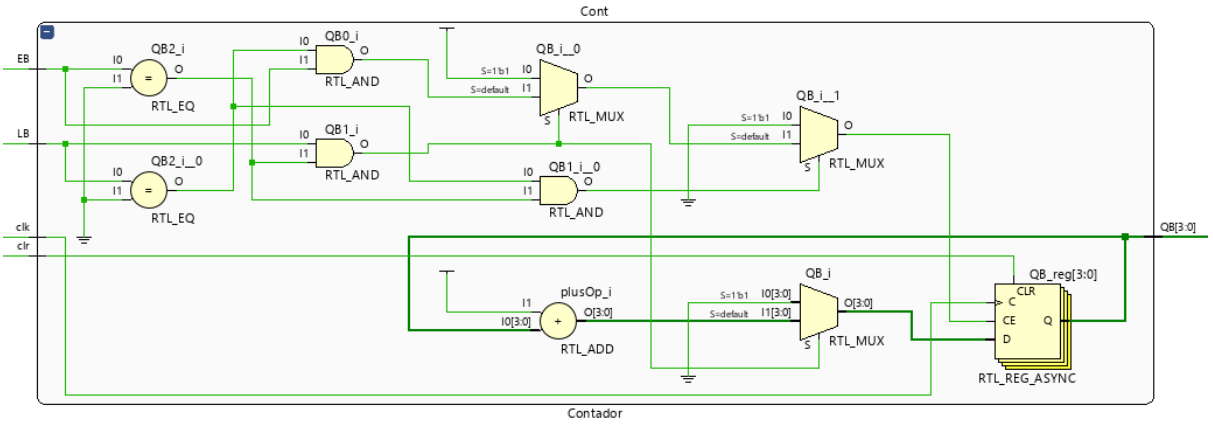
RTL divisor:



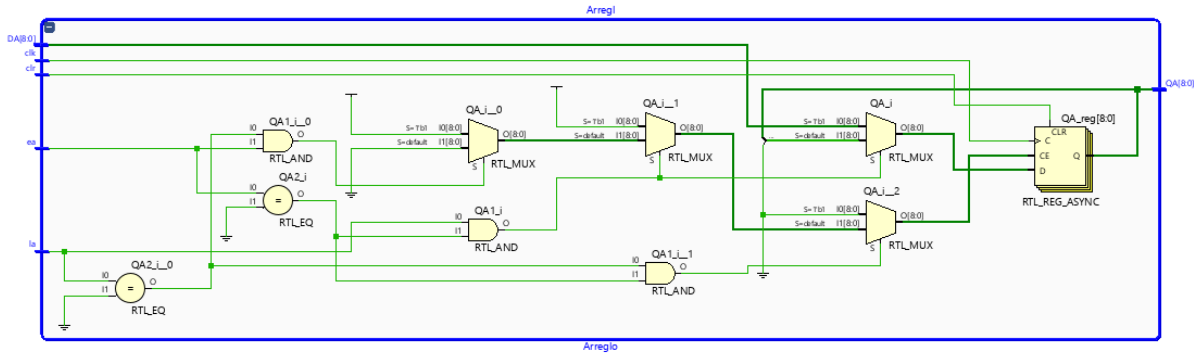
RTL UC:



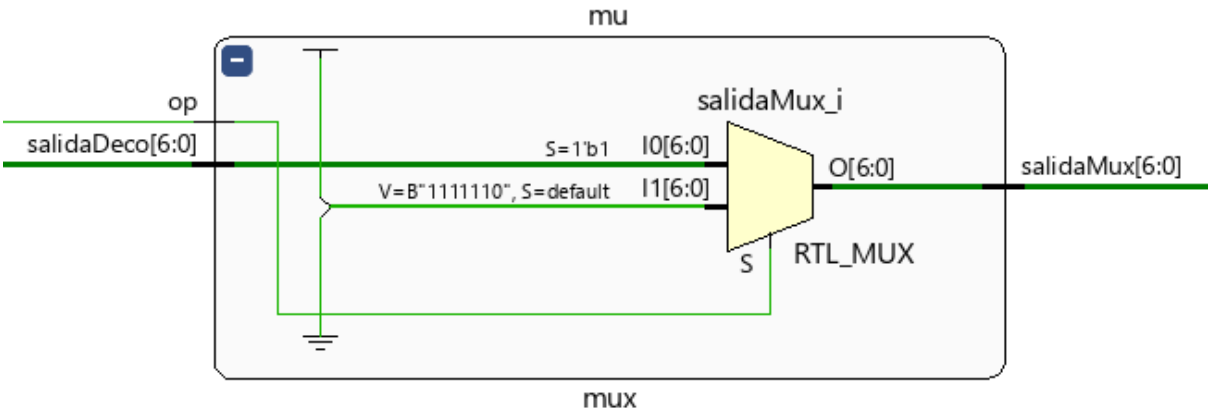
RTL Contador:



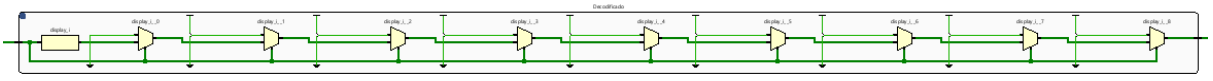
RTL Arreglo:



RTL Mux:



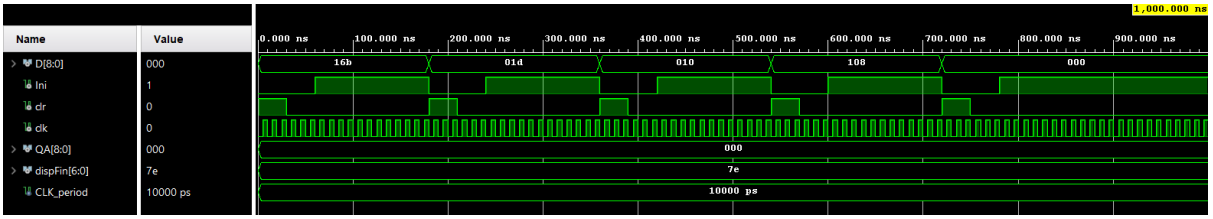
RTL Decodificador:



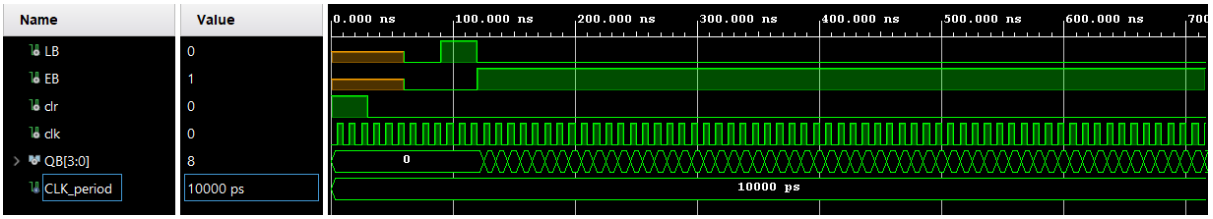
Sebastián Cipriano Damián P9

Simulaciones:

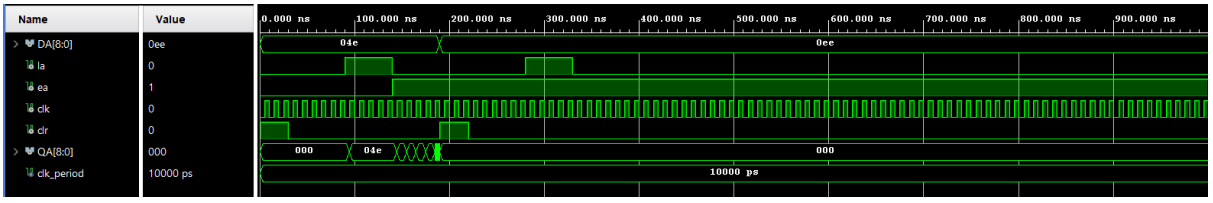
Simulación Completa:



Simulación Contador:



Simulación Arreglo:



Simulación Decodificador:

Simulación Mux:

Simulación UCS: