



SISTEMAS DIGITALES II

TERCERA EVALUACION

I SEMESTRE 2011/2012

Septiembre 2011

Nombre: _____

Paralelo: _____

Problema #1 (30/100)

Diseñar un pequeño **Sistema Digital** que funciona como un circuito **Detector de Ceros**.

El circuito tiene las entradas **Start**, **Load**, **Mostrar** y **Dato** de cuatro bits.

También tiene la salida **Fin** y la salida **Numero de Ceros** representada con un display que inicialmente muestra **0**.

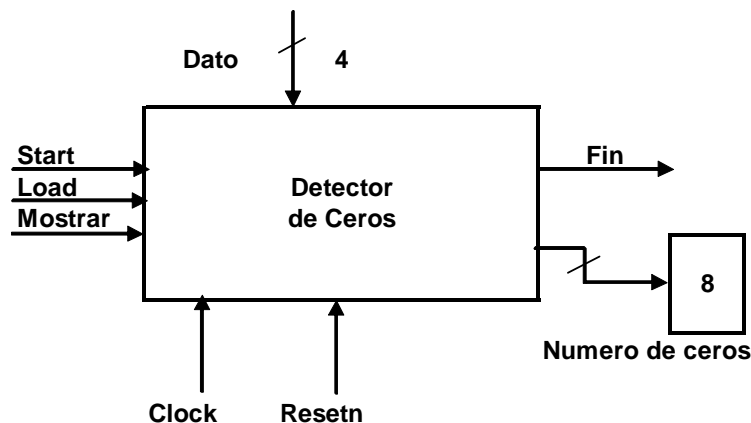
Primeramente se debe presionar y soltar la botonera de la entrada **Start**.

Luego se debe activar la entrada **Load**. Entonces el circuito recibe el número binario de cuatro bits de la entrada **Dato**. Circuito analiza el número ingresado bit por bit y genera la salida **Fin**.

Para ver el resultado se requiere presionar la botonera de la entrada **Mostrar**.

Entonces en el display aparece el número de ceros que tenía el número binario ingresado. El número debe mantenerse en el display mientras la botonera **Mostrar** sigue presionada.

Luego el circuito regresa al estado inicial.



Presente:

1. **Partición Funcional del Sistema Digital.**
2. **Diagrama ASM del circuito Controlador debidamente documentado** (indicar todas las entradas y salidas).

Problema #2 (35/100)

Dada la siguiente descripción en **VHDL** del funcionamiento de un **Sistema Digital**:

Presentar:

1. **Partición Funcional** del Sistema Digital.
2. **Diagrama ASM** del circuito Controlador del Sistema Digital, indicando claramente todas las salidas que deben ser generadas.
3. **Diagramas de Tiempo** del circuito Controlador asumiendo las condiciones de entrada dadas. Indique claramente los nombres de los estados (**y**) que corresponden a cada periodo de **Clock**.

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity tema1vhd is
    port(Resetn, Clock, Start      : in std_logic;
          Suma, Resta, Mostrar, Fin : in std_logic;
          Teclado                  : in std_logic_vector(0 to 9);
          Dig1, Dig0                : out std_logic_vector(1 to 7);
          Done, Err                 : out std_logic);
end tema1vhd;

architecture mixta of tema1vhd is

    component conv_dec_bcd
        port(dec      : in std_logic_vector(0 to 9);
              bcd     : out std_logic_vector(3 downto 0));
    end component;

    component registro_sost
        port(Resetn, Clock : in std_logic;
              En           : in std_logic;
              EntPar       : in std_logic_vector (3 downto 0);
              Q            : out std_logic_vector (3 downto 0));
    end component;

    component registro_sost_esp
        port(Resetn, Clock : in std_logic;
              En           : in std_logic;
              EntPar       : in std_logic_vector (4 downto 0);
              Q            : out std_logic_vector (4 downto 0));
    end component;

    component conv_5bin_2bcd
        port(bin      : in std_logic_vector(4 downto 0);
              bcd1, bcd0 : out Std_logic_vector(3 downto 0));
    end component;

    component dec_dr_en
        port(en      : in std_logic;
              bcd    : in std_logic_vector (3 downto 0);
              seg7   : out std_logic_vector (1 to 7));
    end component;

    type estado is (Ta, Tb, Tc, Td, Te, Tf, Tg, Th, Ti, Tj, Tk, Tl);
    signal y : estado;
    signal tecla, EntReg4, EnA, EnB, EnR, Cout : std_logic;
    signal Sel, Cin, Ssel, Cout_esp, EnDisp, zero : std_logic;
    signal Data, A, B, smux, D1, D0 : std_logic_vector(3 downto 0);
    signal Sum, EntReg, Sal : std_logic_vector(4 downto 0);
begin

```

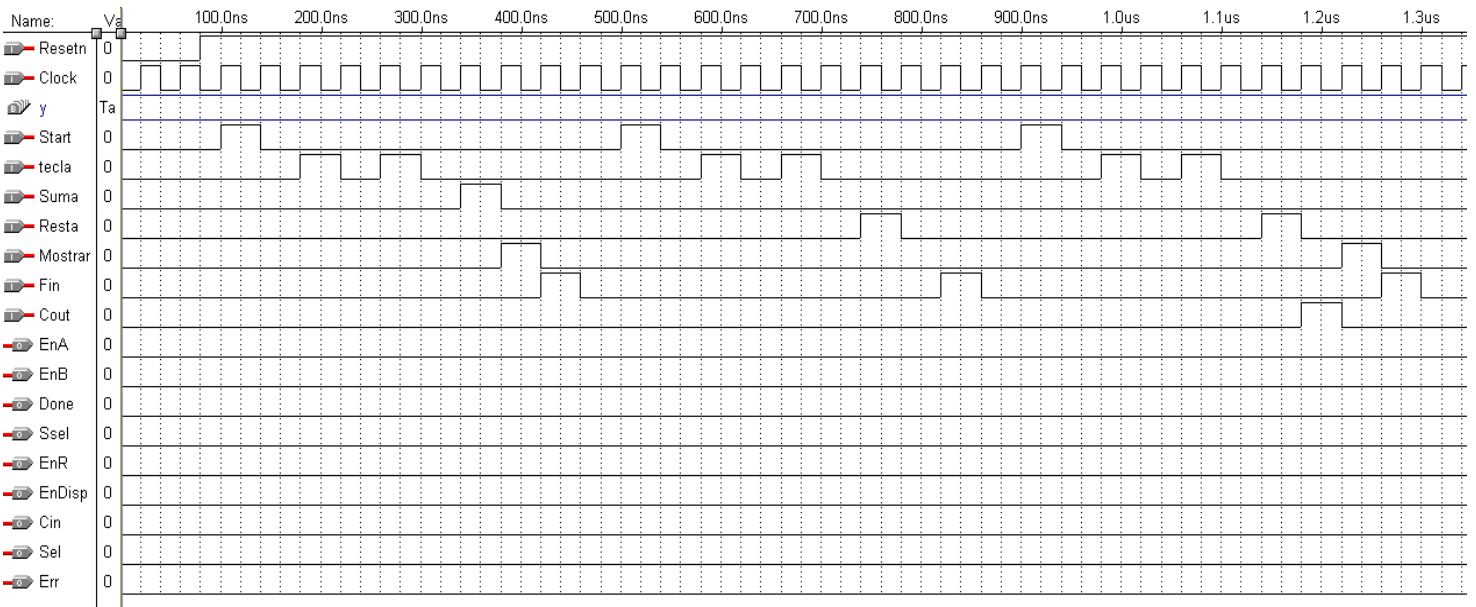
```

-- Controlador
MSS_transiciones: process(Resetn, Clock)
begin
  if Resetn = '0' then y <=Ta;
  elsif Clock'event and Clock = '1' then
    case y is
      when Ta=> if Start = '0' then y <=Ta; else y <=Tb; end if;
      when Tb=> if Start = '0' then y <=Tc; else y <=Tb; end if;
      when Tc=> if tecla = '0' then y <=Tc; else y <=Td; end if;
      when Td=> if tecla = '0' then y <=Te; else y <=Td; end if;
      when Te=> if tecla = '0' then y <=Te; else y <=Tf; end if;
      when Tf=> if tecla = '0' then y <=Tg; else y <=Tf; end if;
      when Tg=> if Suma = '1' then y <=Th;
                    elsif Resta = '1' then y <=Tj; else y <=Tg; end if;
      when Th=> if Mostrar = '0' then y <=Th; else y <=Ti; end if;
      when Ti=> if Fin = '0' then y <=Ti; else y <=Ta; end if;
      when Tj=> if Cout = '0' then y <=Tk; else y <=Tl; end if;
      when Tk=> if Fin = '0' then y <=Tk; else y <=Ta; end if;
      when Tl=> if Mostrar = '0' then y <=Tl; else y <=Ti; end if;
    end case;
  end if;
end process;
MSS_salidas: Process(y, tecla, Mostrar)
begin
  EnA <= '0'; EnB <= '0'; EnR <= '0'; EnDisp <= '0';
  Sel <= '0'; Cin <= '0'; Ssel <= '0'; Done <= '0'; Err <= '0';
  case y is
    when Ta=>
    when Tb=>
    when Tc=> if tecla = '1' then EnA <= '1'; end if;
    when Td=>
    when Te=> if tecla = '1' then EnB <= '1'; end if;
    when Tf=>
    when Tg=>
    when Th=> Done <= '1';
                    if Mostrar = '1' then EnR <= '1'; end if;
    when Ti=> EnDisp <= '1';
    when Tj=> Sel <= '1'; Cin <= '1';
    when Tk=> Err <= '1';
    when Tl=> Sel <= '1'; Cin <= '1'; Ssel <= '1'; Done <= '1';
                    if Mostrar = '1' then EnR <= '1'; end if;
  end case;
end process;
-- Procesador de datos
zero <= '0';
Cout <= Sum(4);
EntReg <= Cout_esp & Sum(3 downto 0);
tecla <= '0' when Teclado = "0000000000" else '1';
conv1: conv_dec_bcd port map(Teclado, Data);
conv2: conv_5bin_2bcd port map(Sal, D1, D0);
dec0: dec_dr_en port map(EnDisp, D0, Dig0);
dec1: dec_dr_en port map(EnDisp, D1, Dig1);
reg1: registro_sost port map(Resetn, Clock, EnA, Data, A);
reg2: registro_sost port map(Resetn, Clock, EnB, Data, B);
reg3: registro_sost_esp port map(Resetn, Clock, EnR, EntReg, Sal);

smux <= B when Sel = '0' else not B;
Cout_esp <= Cout when Ssel = '0' else zero;
Sum <= A + smux + Cin;

end mixta;

```



Problema #3 (35/100)

Dado el Mapa de Estados Primitivo de una MSA (Formato: w2 w1 / Z)

		w2 w1				z
		00	01	11	10	
a	a	e	∅	c	0	
b	∅	e	b	h	1	
c	a	∅	f	c	0	
d	a	d	b	∅	1	
e	a	e	b	∅	0	
f	∅	d	f	c	0	
h	a	∅	b	h	1	

Presente:

- **Tabla de Implicantes. Diagrama de Estados Reducido.**
- **Mapa de asignación de Código de Estados. Mapa de Excitación.**
- **Mapas para Y_1 y Y_0 y para la salida Z . Ecuaciones de Excitación Y_1 y Y_0 . Ecuación de salida Z .**
- **Indique si su circuito corre riesgo de tener Hazards Estáticos o no. ¿Porque?**
- **Diagrama de tiempo para la salida Z asumiendo los valores de las entradas w_1 y w_2 dados. Indique claramente los periodos de tiempo correspondiente a cada estado de su Diagrama de Estados Reducido.**

