



SISTEMAS DIGITALES II

SEGUNDA EVALUACION (60/100) I SEMESTRE 2011/ 2012

septiembre 2011

Nombre: _____

Paralelo: _____

Problema #1 (20/60)

Diseñe un pequeño Sistema Digital **Clasificador de Peso**.

Inicialmente el sistema espera la activación de la señal **Start**. Cuando esto ocurre queda esperando a que se active la señal **Paquete**, que indica que hay un paquete presente, ahora el sistema debe leer (y almacenar) en la entrada **Peso** el valor en binario del peso en Kg del paquete:

Si el paquete pesa menos de 10Kg, se lo considera carga ligera, si el paquete pesa de 10Kg hasta 20 Kg, se lo considera carga mediana, y si pesa más de 20 Kg se lo considera pesado.

Luego de procesar la información y verificar que la señal paquete ya se desactivo, el sistema pregunta por si se han ingresado hasta 32 paquetes (lo máximo) o si se activa la señal **Final**. Si ambas condiciones son falsas, el sistema regresa a esperar por otro paquete.

Si **Final** es verdadera o si se completaron los 32 paquetes, el sistema mostrará en tres displays de siete segmentos (una información a la vez) la cantidad de paquetes ligeros ingresados, luego el total del Kg de carga ligera, luego la cantidad de paquetes medianos, luego el total en Kg de carga ligera, luego la cantidad de paquetes pesados y finalmente el total en Kg de carga pesada. Para dejar de mostrar una información y pasar a la siguiente se debe presionar y soltar la entrada **Continuar**.

Después de que se presiono y soltó **Continuar** por última vez (luego de mostrar los Kg de pesados) el sistema debe regresar al estado inicial.

Nota: Mientras se están ingresando y procesando los pesos, los tres displays deben estar desactivados.



Presente:

1. **Diagrama ASM** del circuito **Controlador** debidamente documentada. (indicar todos las entradas y salidas) (10/20)
2. **Partición Funcional** (10/20)

Problema #2 (25/60)

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 y la duración de cada estado (y).**

```
library ieee;
use ieee.std_logic_1164.all;

entity problema2_vhdl is
    port(Resetn, Clock, Inicio    : in std_logic;
         Dato                    : in std_logic_vector(3 downto 0);
         Fin, Agitar, Llenar     : out std_logic);
end problema2_vhdl;

architecture mixta of problema2_vhdl is
    type estado is (S1, S2, S3, S4, S5, S6);
    signal y : estado;

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

    component contador_down
        port(Clock, Ld, En    : in std_logic;
             Ent             : in std_logic_vector(3 downto 0);
             Q               : out std_logic_vector(3 downto 0));
    end component;

    component contador_up
        port(Clock, Ld, En    : in std_logic;
             Ent             : in std_logic_vector(1 downto 0);
             Q               : out std_logic_vector(1 downto 0));
    end component;

    signal En_A, En_B, En_C, En_D, LCM, ECM, LC, EC : std_logic;
    signal Cig3, CMig0, tiempo : std_logic;
    signal DtA, DtB, DtC, DtD, Dt, CM : std_logic_vector(3 downto 0);
    signal C, zero : std_logic_vector(1 downto 0);

begin

-- Circuito Controlador
    MSS_transiciones: process (Resetn, Clock)
    begin
        if Resetn = '0' then y <= S1;
        elsif (Clock'event and Clock = '1') then
            case y is
                when S1 => if Inicio = '0' then y <= S1; else y <= S2; end if;
                when S2 => if Cig3 = '0' then y <= S2; else y <= S3; end if;
                when S3 => y <= S4;
            end case;
        end if;
    end process;
end architecture mixta;
```

```

    when S4 => if CMig0 ='0' then y <= S4;
                elsif Cig3 ='1' then y <= S6; else y <= S5; end if;
    when S5 => y <= S4;
    when S6 => if Inicio ='1' then y <= S6; else y <= S1; end if;
end case;
end if;
end process;

```

```

MSS_salidas: process (y, Cig3, C(1), CMig0)

```

```

begin
    EC <='0'; LC <='0'; tiempo <='0'; ECM <='0';
    LCM <='0'; Agitar <='0'; Llenar <='0'; Fin <='0';
    case y is
        when S1 => EC <='1'; LC <='1';
        when S2 => tiempo <='1'; EC <='1';
        when S3 => ECM <='1'; LCM <='1';
        when S4 => ECM <='1';
            if C(1)='1' then Agitar <='1'; else Llenar <='1'; end if;
            if CMig0 ='1' and Cig3 ='0' then EC <='1'; end if;
        when S5 => ECM <='1'; LCM <='1';
        when S6 => Fin <='1';
    end case;
end process;

```

```

-- Procesador de Datos

```

```

reg0: registro_sost port map(Resetn, Clock, Dato, En_A, DtA);
reg1: registro_sost port map(Resetn, Clock, Dato, En_B, DtB);
reg2: registro_sost port map(Resetn, Clock, Dato, En_C, DtC);
reg3: registro_sost port map(Resetn, Clock, Dato, En_D, DtD);

```

```

zero <= "00";
contador1: contador_down port map(Clock, LCM, ECM, Dt, CM);
contador2: contador_up port map(Clock, LC, EC, zero, C);

```

```

with C select
    Dt <= DtA when "11",
           DtB when "10",
           DtC when "01",
           DtD when others;

```

```

Dec_2a4: process(C, tiempo)
begin
    if tiempo ='1' then
        case C is
            when "00"=> En_D <='1';
            when "01"=> En_C <='1';
            when "10"=> En_B <='1';
            when others=> En_A <='1';
        end case;
        else En_A <='0'; En_B <='0'; En_C <='0'; En_D <='0';
    end if;
end process;

```

```

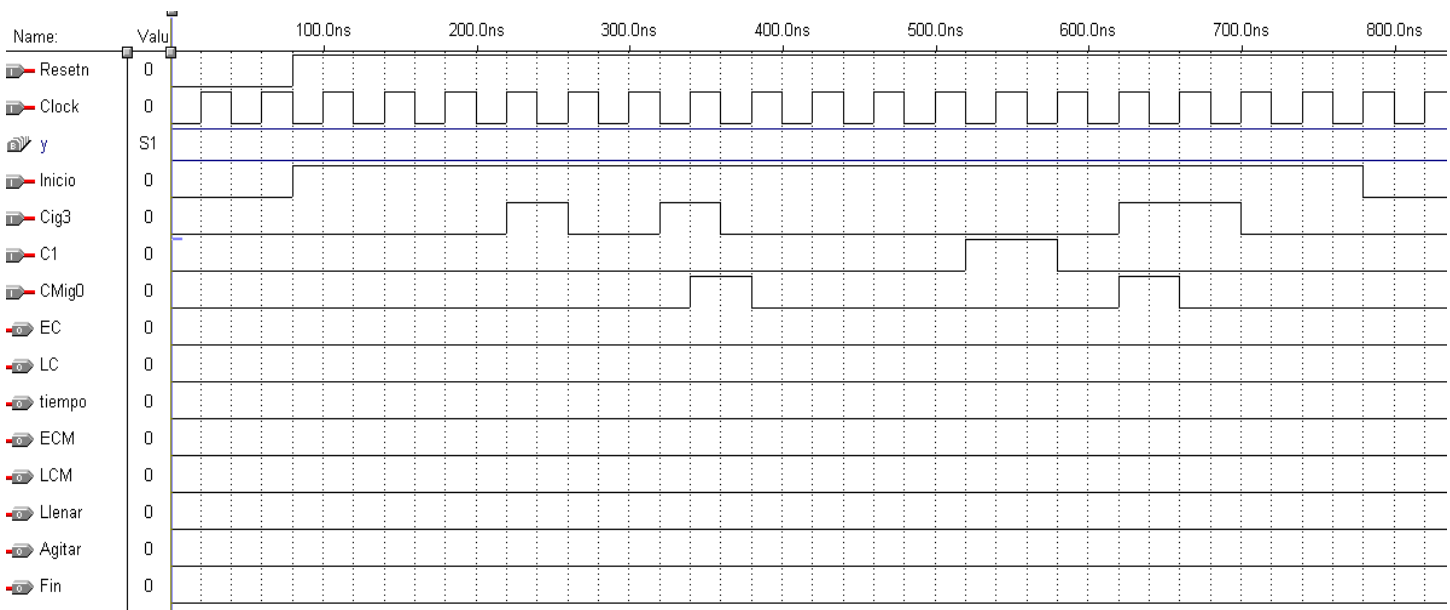
CMig0 <='1' when CM = "0000" else '0';
Cig3 <='1' when C = "11" else '0';

```

```

end mixta;

```



Problema #3 (15/60)

Diseñe una **MSA (Maquina Secuencial Asincrónica)**, en modo fundamental, que puede activar una compuerta electrónica.

La **MSA** tiene dos entradas **X1** y **X2** y una salida **Cmp**. Inicialmente las entradas **X1**, **X2** son iguales a 0 y la compuerta esta desactivada (**Cmp** = 0).

Para activar la compuerta solo se requiere presionar **X1** (con **X2** desactivada). Si se presiona primero **X2** o si se presionan ambas (primero **X2** y luego **X1**), la compuerta sigue desactivada.

Una vez que la compuerta esta activada, para desactivarla primero se debe soltar **X1** y luego se debe presionar únicamente **X2**, solo en ese momento se desactiva la compuerta.

Si luego de haber soltado **X1** se vuelve a presionar **X1** o si se presionan ambas (primero **X1** y luego **X2**), la compuerta permanece activada.



Presentar:

1. Diagrama de Estados Primitivo (Formato: **X1 X2 / Cmp**). Tabla de Estados Primitivo. Tabla de Implicantes. Diagrama de Equivalencia máxima.
2. Diagrama de Estados Reducida. Mapa de asignación de Código de Estados.
3. Mapa de Excitación. Mapas y las expresiones para **Y1** y **Y0** y para la salida **Cmp**.
4. Diagramas de tiempo para la salida **Cmp** asumiendo valores de las entradas **X1** y **X2** dados. Indique claramente los periodos de tiempo correspondiente a cada estado de su Diagrama de Estados Reducido.
5. Indica si su circuito corre riesgo de tener los **Hazard Estáticos** o no. ¿Como se puede evitar?

