



SISTEMAS DIGITALES II

PRIMERA EVALUACIÓN

SEGUNDO TÉRMINO 2011-2012

30 de noviembre del 2011

NOMBRE : _____

PARALELO : 1

PROBLEMA # 1 (20 p)

Diseñe una **MSS** modelo **MOORE** que controla una Ensambladora automática de mesas industriales.

Inicialmente la máquina espera que se presione y suelte el botón **Inicio**, con lo cual queda lista para arrancar. Si desde este estado de activación se vuelve a presionar y soltar el botón **Inicio**, la máquina regresará al estado inicial.

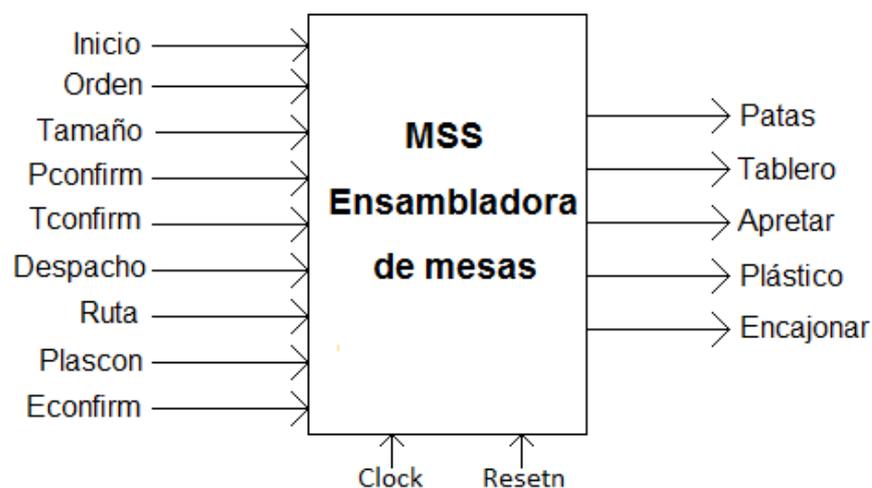
Desde el estado de activación, la máquina espera a que se active la señal **Orden** y de ser así al mismo tiempo pregunta por el valor de **Tamaño**: Si **Tamaño** = 0 se trata de una mesa pequeña, si **Tamaño** = 1 se trata de una mesa grande. Si **Orden** no está activa debe permanecer en el estado de activación.

- Si la mesa es pequeña el sistema activa la señal **Patatas**, la misma que debe seguir activa hasta que se reciba la señal **Pconfirm** que indica que las patatas se han terminado de instalar, Ahora el sistema debe activar la salida **Tablero** que debe seguir activa hasta que se reciba la señal **Tconfirm** que indica que una pieza de tablero ha sido instalada, luego de esto el sistema debe activar por un período de reloj la señal **Apretar** que ajustará todas las uniones.
- Si la mesa es grande se instalarán dos módulos de patatas y tableros, para esto el sistema debe activar las señales **Patatas** y **Tablero** igual que antes (esperando su respectiva confirmación) pero ahora cada una debe ser activada y desactivada dos veces en su orden (primero dos módulos de patatas y luego dos módulos de tableros). Luego de esto el sistema deberá activar por dos períodos consecutivos de reloj la señal **Apretar**.

Durante las dos últimas acciones no es necesario preguntar por las señales **Orden** y **Tamaño**.

Finalmente y sin importar el tamaño de la mesa, el sistema queda en espera de la señal **Despacho**. Cuando **Despacho** sea verdadera, el sistema al mismo tiempo pregunta por la señal **Ruta**. Si **Ruta** = 1 el sistema activa la señal **Plástico** que debe permanecer activa hasta que se active la señal de confirmación **Plascon**. Si **Ruta** = 0 el sistema activa la señal **Encajonar** que debe continuar activada hasta que se reciba la señal **Econfirm**.

Independientemente de la ruta escogida, el sistema regresa al estado de activación.



Presente únicamente el Diagrama ASM de la MSS

PROBLEMA # 2 (20 p)

Dado la siguiente descripción en **VHDL** de una **MSS**:

1. Grafique el **Diagrama ASM** que corresponde a este código.
2. Grafique el **Diagrama de Tiempo** adjunto para las condiciones de entrada dadas. Indique claramente los **nombres de cada estado (y)** y **tiempo de su duración**.

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

entity tema2 is
    port ( Resetn, Clock           : in std_logic;
          Start,P1,P2,Cont,Sensor,Upre : in std_logic;
          EnA,EnB,Luz1,Luz2,S1,S2,Fin : out std_logic);
end tema2;

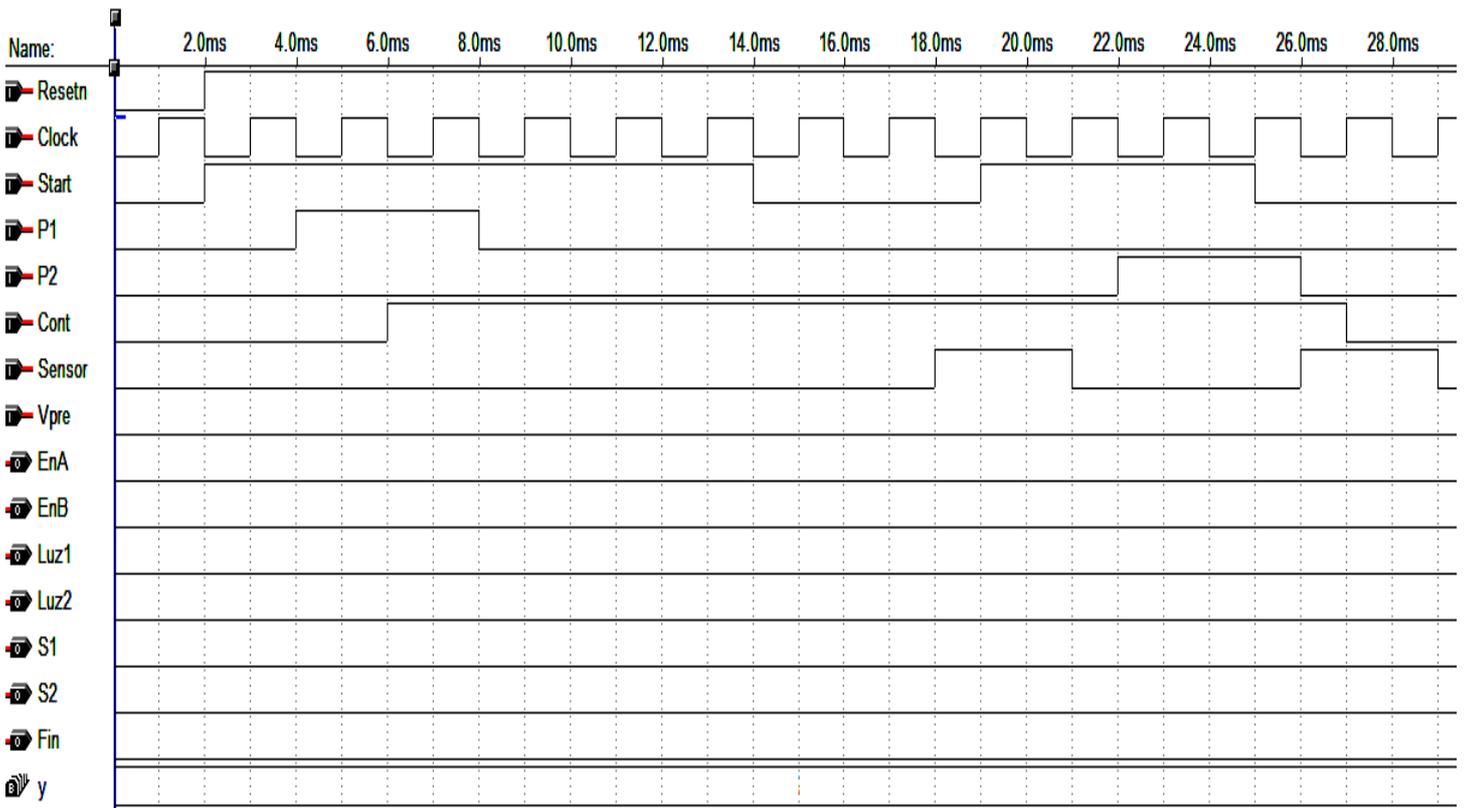
architecture comportamiento of tema2 is
    type estado is (Ta,Tb,Tc,Td,Te,Tf,Tg,Th,Ti,Tj);
    signal y: estado;
begin
    process (Resetn, Clock)
    begin
        if Resetn='0' then y<= Ta;
        elsif Clock'event and Clock = '1' then
            case y is
                when Ta => if Start = '0' then y <= Ta; else y <= Tb; end if;
                when Tb => if P1 = '1' then y <= Tc;
                           elsif P2 = '0' then y <= Td; else y <= Te; end if;
                when Tc => if Cont = '0' then y <= Tc; else y <= Tf; end if;
                when Td => if Cont = '0' then y <= Td; else y <= Tf; end if;
                when Te => if Sensor = '0' then y <= Te; else y <= Tg; end if;
                when Tf => if Upre = '1' then y <= Tf;
                           elsif Sensor = '0' then y <= Ti; else y <= Tj; end if;
                when Tg => if Cont = '0' then y <= Tg; else y <= Th; end if;
                when Th => if Sensor = '0' then y <= Ti; else y <= Tj; end if;
                when Ti => if Cont = '0' then y <= Ti; else y <= Ta; end if;
                when Tj => if Cont = '0' then y <= Tj; else y <= Ta; end if;
            end case;
        end if;
    end process;
```

```

process (y,Start,P1,P2,Cont,Sensor,Upre)
begin
  EnA<='0'; EnB<='0'; Luz1<='0'; Luz2<='0'; S1<='0'; S2<='0'; Fin<='0';
  case y is
    when Ta => EnA <= '1'; EnB <= '1';
    when Tb => if P1 = '1' then Luz1 <= '1';
               elsif P2 = '1' then Luz2 <= '1'; end if;
    when Tc => Luz1 <= '1';
               if Cont = '1' then EnA <= '1'; end if;
    when Td => if Cont = '1' then EnA <= '1'; end if;
    when Te => Luz2 <= '1';
    when Tf => EnA <= '1'; S1 <= '1';
               if Upre = '0' then
                 If Sensor = '0' then Luz2 <='1'; else Luz1 <= '1';
               end if; end if;
    when Tg => EnA <= '1'; EnB <= '1';
               if Cont = '1' then S2 <= '1'; end if;
    when Th => EnB <= '1'; S2 <= '1';
               if Sensor = '0' then Luz2 <= '1'; else Luz1 <= '1'; end if;
    when Ti => Fin <= '1'; Luz2 <= '1';
    when Tj => Fin <= '1'; Luz1 <= '1';
  end case;
end process;

end comportamiento;

```



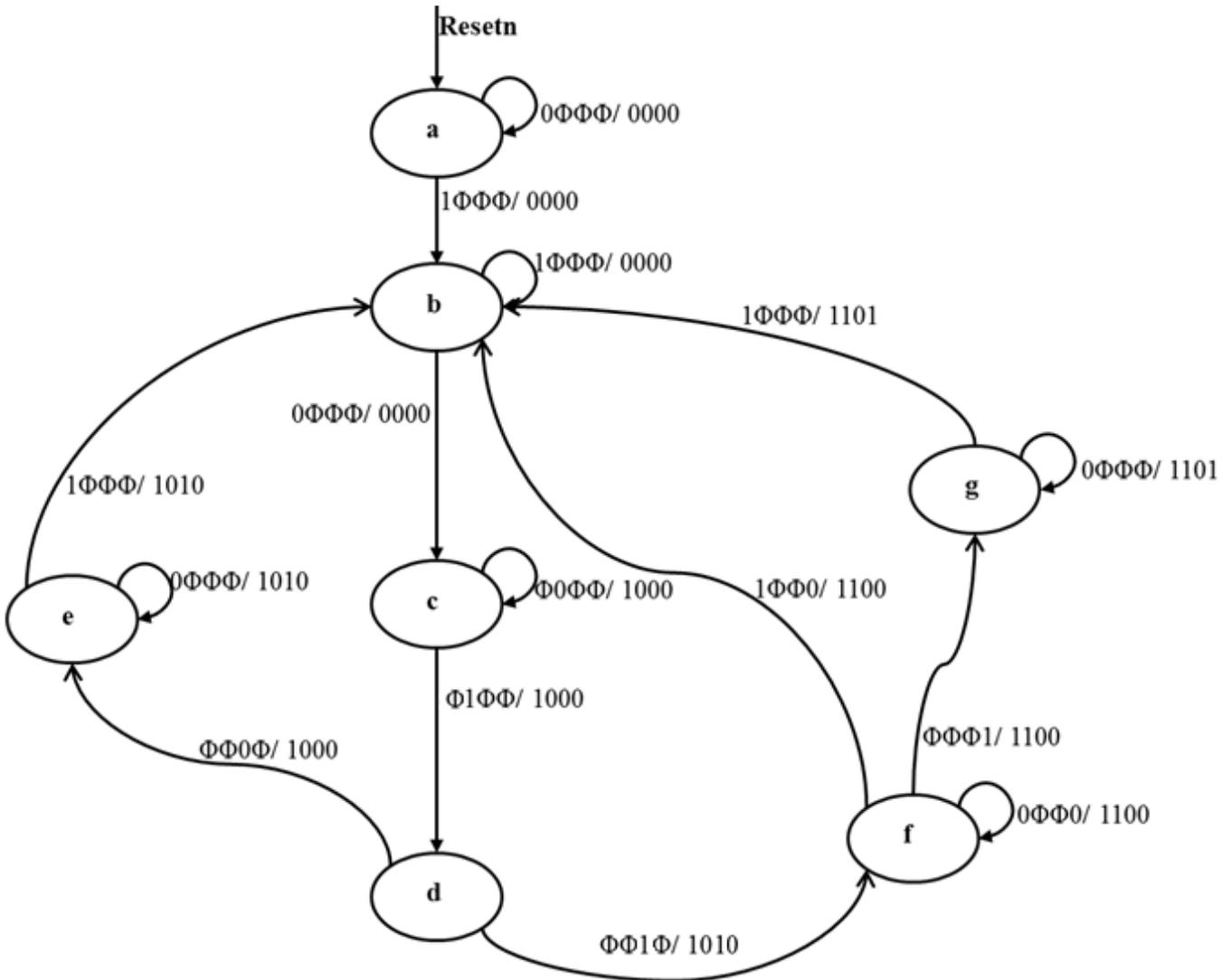
PROBLEMA # 3 (20 p)

Dado el **Diagrama de Estados** de una **MSS**:

1. Transfórmelo en un **Diagrama ASM**.
2. Basándose en **Diagrama ASM** grafique los mapas para el Decodificador de Estado Siguiendo asumiendo la asignación de **Código de Estados** dado. Simplifique las expresiones en las celdas de los mapas usando los Teoremas de álgebra de Boole.

Presente las ecuaciones para las salidas **X1 BCD Error Enable**.

Formato: **Start Final OK Mostrar / X1 BCD Error Enable**



y_2y_1	00	01	11	10
0	a	b	c	Φ
1	g	f	d	e