



## SISTEMAS DIGITALES II

PRIMERA EVALUACION (60/100) I SEMESTRE 2010/ 2011

julio 2010

Nombre: \_\_\_\_\_

Paralelo: \_\_\_\_\_

### Problema #1 (20/60)

Diseñe una **MSS** modelo **MOORE** que funcione como vendedor de boletos para un estadio de fútbol.

Inicialmente el sistema envía la salida **Listo** para indicar que está disponible para vender. Si el comprador presiona **Ingresar** se activa la secuencia de ingreso al estadio (descrita posteriormente).

Si en lugar de presionar ingresar, se presiona **Vender**, el sistema pasa a otro estado donde se pregunta por la localidad deseada, esto es Palco, Tribuna o Cancelar que regresa al estado inicial.

Si se recibe la señal **Palco**, el sistema pregunta por la señal **Pagot** que indica que ya no hay palcos y de ser así regresa para pedir otra localidad. Si no está agotado, activa la salida **DP** (Dinero Palco) para solicitar el valor de la entrada respectivo, y se mantiene activada hasta recibir **DPC** (Dinero de Palco Confirmado), luego activa la salida **EP** (Entregar Palco) que se mantiene activa hasta que se reciba la señal **PD** (Palco Despachado)

Si se recibe la señal **Tribuna**, el sistema pregunta por la señal **Tagot** que indica que ya no hay tribunas y de ser así regresa para pedir otra localidad. Si no está agotado, activa la salida **DT** (Dinero Tribuna) para solicitar el valor de la entrada respectivo, y se mantiene activada hasta recibir **DTC** (Dinero de Tribuna Confirmado), luego activa la salida **ET** (Entregar Tribuna) que se mantiene activa hasta que se reciba la señal **TD** (Tribuna Despachada)

Luego de vender cualquier localidad, el sistema regresa al estado inicial donde (como se indico anteriormente) puede presionar **Ingresar**. De ser así el sistema espera a que la máquina lectora de boletos externa envíe una de las dos señales de lectura **LP** o **LT** (Palco o tribuna respectivamente), luego de eso activa el respectivo acceso **AP** o **AT** y lo mantiene activo hasta que se confirme el acceso de la persona a la respectiva localidad mediante las señales **APC** o **ATC** respectivamente. Después debe regresar al estado inicial.



Presentar únicamente el Diagrama ASM de la MSS.

## Problema #2 (20/60)

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 problema2 is
    port(Resetn,Clock,Start      : in std_logic;
          Uno,Dos,Tres,Cuatro,LuzF,LuzP : in std_logic;
          Act_LuzF,Act_LuzP,Des_LuzF  : out std_logic;
          Des_LuzP,Off, Listo        : out std_logic);
end problema2;

architecture comportamiento of problema2 is
    type estado is (Ta,Tb,Tc,Td,Te,Tf,Tg,Th,Ti);
    signal y : estado;
begin
    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 Uno = '1' then
                    if LuzF = '0' then y <=Td;
                    elsif LuzP = '0' then y <=Te; else y <=Tc; end if;
                    elsif Uno = '0' and Dos = '1' then
                        if LuzP = '1' then y <=Tf;
                        elsif LuzF = '1' then y <=Tg; else y <=Tc; end if;
                    elsif Uno='0' and Dos = '0' and Tres = '1' then
                        if LuzF = '0' then y <=Th; else y <=Tc; end if;
                    elsif Uno = '0' and Dos = '0' and Tres = '0' and Cuatro = '1' then
                        if LuzF = '1' then y <=Ti; else y <=Tc; end if;
                    else y <=Tc; end if;
                when Td=> if LuzF = '0' then y <=Td;
                    elsif LuzP = '0' then y <=Te; else y <=Tc; end if;
                when Te=> if LuzP = '0' then y <=Te; else y <=Tc; end if;
                when Tf=> if LuzP = '1' then y <=Tf;
                    elsif LuzF = '1' then y <=Tg; else y <=Tc; end if;

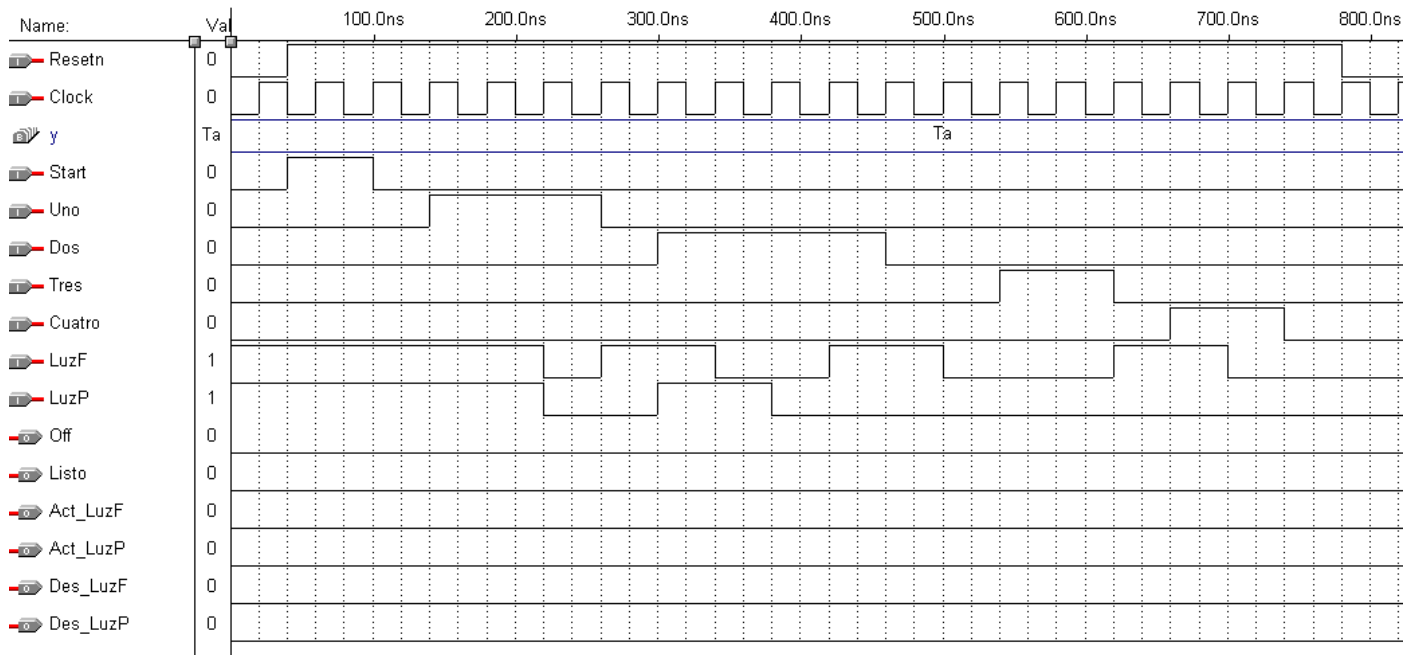
                when Tg=> if LuzF='1' then y <=Tg; else y <=Tc; end if;
                when Th=> if LuzF = '0' then y <=Th; else y <=Tc; end if;
                when Ti=> if LuzF = '1' then y <=Ti; else y <=Tc; end if;
            end case;
        end if;
    end process;
```

```

Off <='1' when (y =Ta and Start ='0') else '0';
Listo <='1' when ((y =Tb and Start ='0') or y =Tc) else '0';
Act_LuzF <='1' when (y =Td or y =Th) else '0';
Act_LuzP <='1' when y =Te else '0';
Des_LuzF <='1' when (y =Tg or y =Ti) else '0';
Des_LuzP <='1' when y =Tf else '0';

```

end comportamiento;

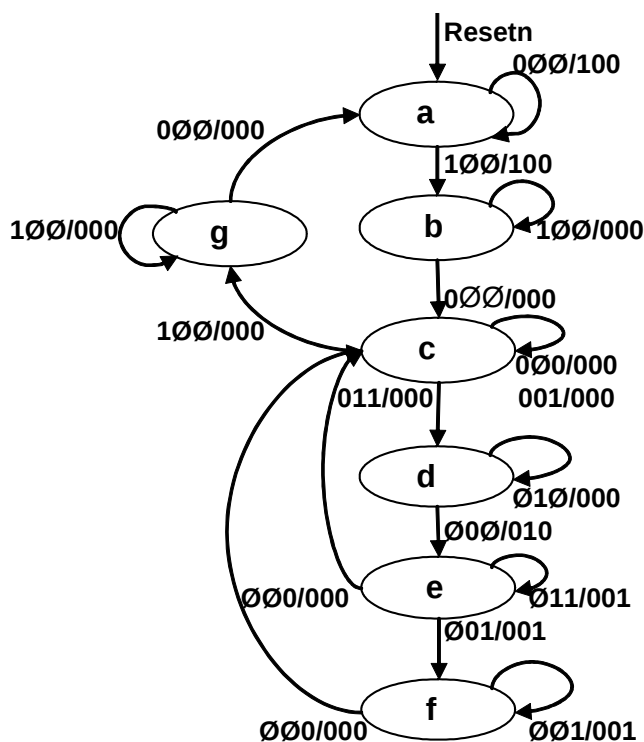


### Problema #3 (20/60)

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 Siguiente. Implemente la **MSS** con flip-flops, mux8a1 y puertas lógicas adicionales necesarias, asumiendo la asignación de **Código de Estados** dado.

Formato: **Ac Ing Vp / Ready Tk PI**



|    |      |      |      |      |
|----|------|------|------|------|
|    | y2   |      |      |      |
|    | 0. a | 2. g | 6. d | 4. f |
| y0 | 1. b | 3. c | 7. ∅ | 5. e |
|    | y1   |      |      |      |