



## SISTEMAS DIGITALES II

PRIMERA EVALUACION (60/100) | SEMESTRE 2011/ 2012

julio 2011

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

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

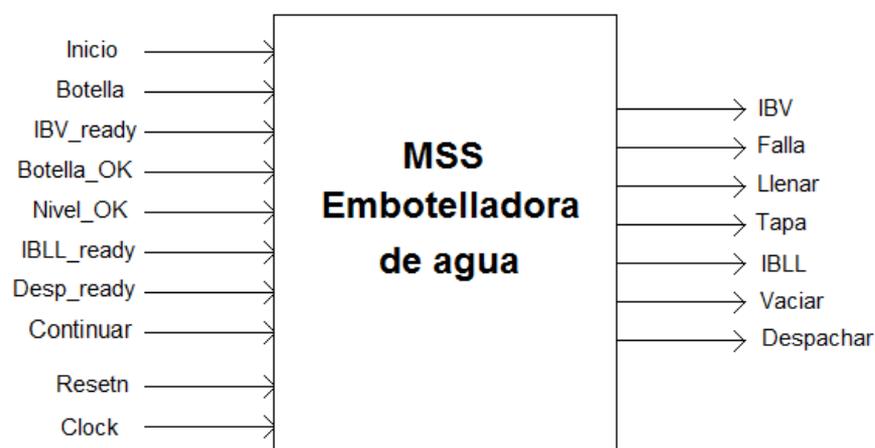
### Problema #1 (20/60)

Diseñe una MSS modelo MOORE que funcione como **controladora de una embotelladora de agua**.

Inicialmente el sistema espera a que se active la señal **Inicio**, luego de lo cual pasa a un estado de activación y espera a que la señal **Botella** se active indicando la presencia de una botella vacía. Si hay una botella se activa la salida **IBV** que pone a funcionar al inspector de botellas vacías, esta señal se sigue activando hasta que se reciba **IBV\_ready** que significa que el detector termino su análisis. Posteriormente se evalúa la señal **Botella\_OK**, si es falsa, significa que la botella no puede ser procesada por tanto se debe activar la salida **Falla** y debe regresar al estado de activación a esperar otra botella. Si **Botella\_OK** es verdadera, la botella si está apta para ser llenada y debe continuar el proceso.

En el siguiente paso del proceso, se debe activar la salida **Llenar** que debe permanecer activa hasta que reciba la señal **Nivel\_OK** que le indica que el agua llegó al nivel deseado. Luego se debe activar la salida **Tapa** exactamente durante dos períodos de reloj. Posteriormente se debe activar la salida **IBLL** que activa al inspector de botellas llenas, esta señal continuará activa hasta que se reciba la señal **IBLL\_ready** que significa que el inspector terminó su análisis. A continuación si la señal **Botella\_OK** es falsa significa que la botella no se lleno lo suficiente, por tanto se debe activar la señal **Vaciar** y luego regresar al estado donde se active nuevamente la salida **Llenar**. Si **Botella\_OK** es verdadera significa que la botella está lista, por tanto ahora se debe activar la salida **Despachar** que se mantiene activa hasta que se reciba la señal **Desp\_ready**.

Posteriormente se debe evaluar la entrada **Continuar**, si es verdadera el sistema debe regresar al estado de activación a pedir otra botella, caso contrario debe regresar al estado inicial de desactivación.



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,Ingresar,Vender,Cancelar : in std_logic;
         Palco,Tribuna,Pagot,Tagot,LP,LT      : in std_logic;
         DPC,DTC,APC,ATC,PDS,TDS            : in std_logic;
         Listo,DP,DT,EP,ET,AP,AT           : out std_logic);
end problema2;

architecture comportamiento of problema2 is
    type estado is (Ta,Tb,Tc,Td,Te,Tf,Tg,Th,Ti,Tj,Tk);
    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 Ingresar ='1' then y <=Ti;
                    elsif Vender ='1' then y <=Tb; else y <=Ta; end if;
                when Tb=> if Palco ='1' then y <=Tc;
                    elsif Tribuna ='1' then y <=Tf;
                        elsif Cancelar ='1' then y <=Ta; else y <=Tb; end if;
                when Tc=> if Pagot ='0' then y <=Td; else y <=Tb; end if;
                when Td=> if DPC ='0' then y <=Td; else y <=Te; end if;
                when Te=> if PDS ='0' then y <=Te; else y <=Ta; end if;
                when Tf=> if Tagot ='0' then y <=Tg; else y <=Tb; end if;
                when Tg=> if DTC ='0' then y <=Tg; else y <=Th; end if;
                when Th=> if TDS ='0' then y <=Th; else y <=Ta; end if;
                when Ti=> if LP ='1' then y <=Tj;
                    elsif LT ='1' then y <=Tk; else y <=Ti; end if;
                when Tj=> if APC ='0' then y <=Tj; else y <=Ta; end if;
                when Tk=> if ATC ='0' then y <=Tk; else y <=Ta; end if;
            end case;
        end if;
    end process;

MSS_salidas: process(y,Ingresar,Vender,DPC,DTC,LP,LT)
Begin

Listo <='0'; DP <='0'; DT <='0'; EP <='0'; ET <='0'; AP <='0'; AT <='0';
```

```

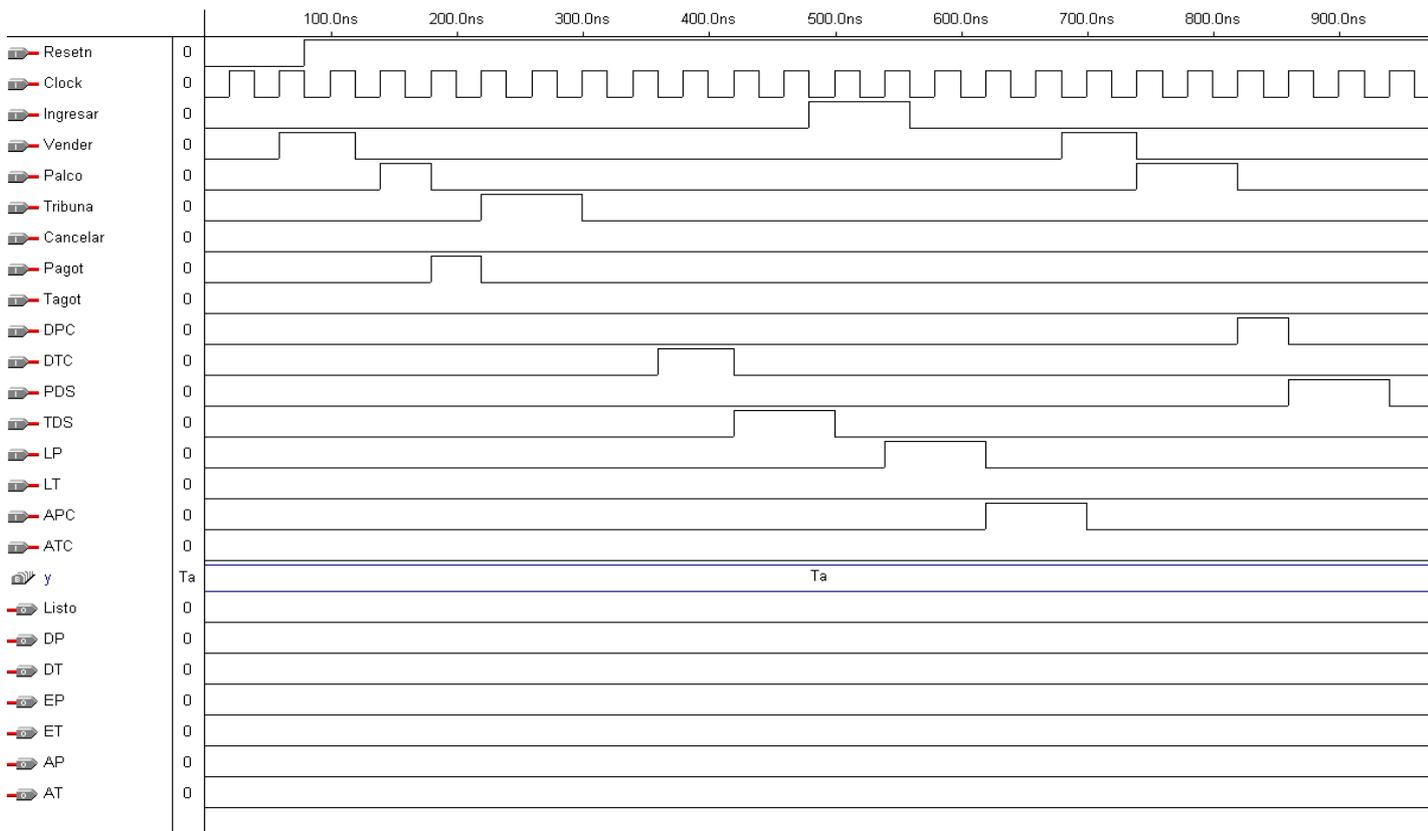
if y =Ta and Ingresar ='0' and Vender ='0' then Listo <='1'; end if;
if y =Td then DP <='1'; end if;
if y =Tg then DT <='1'; end if;
if (y =Td and DPC ='1') or y =Te then EP <='1'; end if;
if (y =Tg and DTC ='1') or y =Th then ET <='1'; end if;
if (y =Ti and LP ='1') or y =Tj then AP <='1'; end if;
if (y =Ti and LT ='1') or y =Tk then AT <='1'; end if;

```

```

end process;
end comportamiento;

```

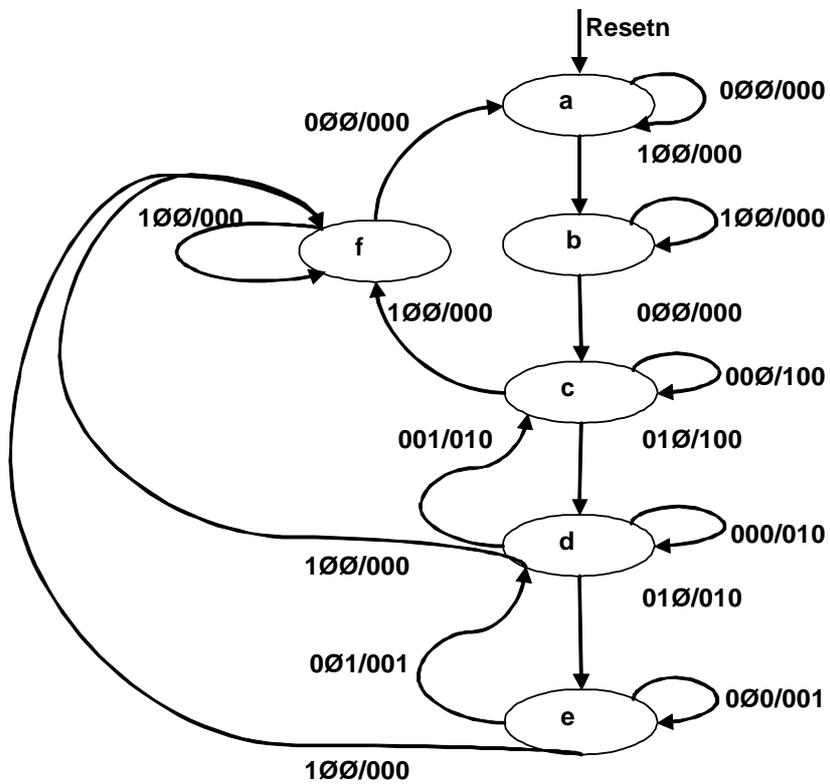


### Problema #3 (20/60)

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, decoder y puertas lógicas adicionales necesarias, asumiendo la asignación de **Código de Estados** dado.

Formato: Inicio  $S_1 S_0 / P_1 P_2 P_3$



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