# Laboratorio VHDL

## Hoja de respuestas del laboratorio “Máquinas de Estados Finitos”

Asignatura: DSED

Número de grupo:1

Nombres y apellidos de los miembros del grupo: Álvaro Escribano Vilar y Guillermo Pagés Scasso

**1-1: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity lab5\_1\_1 is

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

ain : in STD\_LOGIC;

yout : out STD\_LOGIC;

cont : out STD\_LOGIC\_VECTOR (3 downto 0));

end lab5\_1\_1;

architecture Behavioral of lab5\_1\_1 is

type state\_type is (S0,S1,S2);

signal state, next\_state : state\_type;

signal next\_cont : STD\_LOGIC\_VECTOR(3 downto 0) := "0000";

signal cont\_reg: STD\_LOGIC\_VECTOR(3 downto 0) := "0000";

signal yout\_reg: std\_logic;

begin

SYNC\_PROC : process (clk)

begin

if rising\_edge(clk) then

if (reset = '1') then

cont\_reg <= (others=>'0');

state <= S0;

else

state <= next\_state;

cont\_reg <= next\_cont;

end if;

end if;

end process;

OUTPUT\_DECODE : process (state,ain)

begin

yout\_reg <= '0';

case (state) is

when S0 =>

yout\_reg <= '1';

when S1 =>

yout\_reg <= '0';

when S2 =>

yout\_reg <= '0';

when others =>

yout\_reg <= '0';

end case;

end process;

NEXT\_STATE\_DECODE : process (state,ain,cont\_reg)

begin

next\_state <= S0;

next\_cont <= cont\_reg;

case (state) is

when S0 =>

if (ain = '1') then

next\_state <= S1;

next\_cont <= std\_logic\_vector(unsigned(cont\_reg)+1);

elsif(ain = '0') then

next\_state <= S0;

end if;

when S1 =>

if (ain = '1') then

next\_state <= S2;

next\_cont <= std\_logic\_vector(unsigned(cont\_reg)+1);

elsif(ain = '0') then

next\_state <= S1;

end if;

when S2 =>

if (ain = '1') then

next\_state <= S0;

next\_cont <= std\_logic\_vector(unsigned(cont\_reg)+1);

elsif(ain = '0') then

next\_state <= S2;

end if;

when others =>

next\_state <= S0;

end case;

end process;

yout <= yout\_reg;

cont <= cont\_reg;

end Behavioral;

**2-1: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity lab5\_2\_1 is

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

ain : in STD\_LOGIC\_VECTOR (1 downto 0);

yout : out STD\_LOGIC);

end lab5\_2\_1;

architecture Behavioral of lab5\_2\_1 is

type state\_type is (Sres,S00,S01,S10,S11,St0,St1);

signal state, next\_state : state\_type;

signal yout\_s : STD\_LOGIC := '0';

begin

SYNC\_PROC : process (clk)

begin

if rising\_edge(clk) then

if (reset = '1') then

state <= Sres;

else

state <= next\_state;

end if;

end if;

end process;

OUTPUT\_DECODE : process (state)

begin

case (state) is

when Sres =>

yout\_s <= '0';

when S00 =>

when S01 =>

yout\_s <= '0';

when S10 =>

when S11 =>

yout\_s <= '1';

when St0 =>

when St1 =>

yout\_s <= not(yout\_s);

when others =>

end case;

end process;

NEXT\_STATE\_DECODE : process (state, ain)

begin

if(ain = "01") then

next\_state <= S00;

elsif(ain = "10") then

next\_state <= St0;

elsif(ain = "11") then

next\_state <= S10;

end if;

case (state) is

when Sres =>

if (ain = "00") then

next\_state <= Sres;

end if;

when S00 =>

if (ain = "00") then

next\_state <= S01;

end if;

when S01 =>

if (ain = "00") then

next\_state <= S01;

end if;

when S10 =>

if (ain = "00") then

next\_state <= S11;

end if;

when S11 =>

if (ain = "00") then

next\_state <= S11;

end if;

when St0 =>

if (ain = "00") then

next\_state <= St1;

end if;

when St1 =>

if (ain = "00") then

next\_state <= St1;

end if;

end case;

end process;

yout <= yout\_s;

end Behavioral;

**3-1: Haz una demo del comportamiento de tu diseño al profesor. Copia y pega el contenido de los ficheros .vhd:**

entity lab5\_3\_1 is

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

count : out STD\_LOGIC\_VECTOR (2 downto 0));

end lab5\_3\_1;

architecture Behavioral of lab5\_3\_1 is

type state\_type is (S000,S001,S011,S101,S111,S010);

signal state, next\_state : state\_type;

signal cont\_reg, next\_cont: STD\_LOGIC\_VECTOR(2 downto 0) := "000";

begin

SYNC\_PROC : process (clk)

begin

if rising\_edge(clk) then

if (reset = '1') then

cont\_reg <= "000";

state <= S000;

else

state <= next\_state;

cont\_reg <= next\_cont;

end if;

end if;

end process;

NEXT\_STATE\_DECODE : process (state)

begin

next\_cont <= "000";

next\_state <= S000;

case(state) is

when S000 =>

next\_cont <= "001";

next\_state <= S001;

when S001 =>

next\_cont <= "011";

next\_state <= S011;

when S011 =>

next\_cont <= "101";

next\_state <= S101;

when S101 =>

next\_cont <= "111";

next\_state <= S111;

when S111 =>

next\_cont <= "010";

next\_state <= S010;

when S010 =>

next\_cont <= "000";

next\_state <= S000;

when others =>

next\_cont <= "000";

next\_state <= S000;

end case;

end process;

count <= cont\_reg;

end Behavio**ral;**