**TE2002B : Actividad 3.1 – MOORE**

**Equipo 3:**

**David Israel Vázquez Leal**

**Juan Carlos Hernández Ríos**

**José Emiliano Flores Martínez**

**Programa MOORE\_3.vhd (entidad principal):**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity MOORE\_3 is

port (clk\_3, rst\_3 : in std\_logic;

left\_3, right\_3, interm\_3 : in std\_logic;

lLights\_3, rLights\_3 : out std\_logic\_vector(2 downto 0));

end MOORE\_3;

architecture Behavior of MOORE\_3 is

-- State types: s0: both off | s1\_x: left directional | s2\_x: right directional | s3\_x: intermittent

type state\_type is (s0, s1\_0, s1\_1, s1\_2, s1\_3, s2\_0, s2\_1, s2\_2, s2\_3, s3\_0, s3\_1);

signal present\_state\_3, next\_state\_3 : state\_type;

-- Define constants

constant clk\_3\_FREQ: integer := 50000000; -- 50 MHz clock frequency

constant DELAY\_100ns\_3: integer := 5; -- A delay of 100ns for intermittents (The 2 state cycle lasts 200ns) for intermittent lights

constant DELAY\_40ns\_3: integer := 2; -- A delay of 40ns for directional lights (the 4 state cycles last 160ns) for right and left lights

-- Define signals

signal counter\_3: integer range 0 to 50000001 := 0;

signal delay\_done\_3: std\_logic := '1';

signal skip\_delay\_3: std\_logic := '0';

signal delay\_sel\_3: std\_logic := '0'; -- 0: delay of 100ns | 1: delay of 20ns

begin

process(clk\_3, rst\_3)

begin

if (rising\_edge(clk\_3)) then

-- State transitions

present\_state\_3 <= next\_state\_3;

end if;

end process;

-- Delay for each sequence to take some time

delay: process (clk\_3)

begin

if (rising\_edge(clk\_3)) then

-- Checks if delay is skipped, or if delay is done (either 100ns or 20ns according to delay\_sel\_3)

if ((counter\_3>=DELAY\_100ns\_3 and delay\_sel\_3 = '0') or (counter\_3>=DELAY\_40ns\_3 and delay\_sel\_3 = '1') or skip\_delay\_3='1') then

counter\_3 <= 0;

delay\_done\_3 <= '1';

else

counter\_3 <= counter\_3 + 1;

delay\_done\_3 <= '0';

end if;

end if;

end process;

-- Selecting next state

C1: process(delay\_done\_3, clk\_3, present\_state\_3, left\_3, right\_3, interm\_3)

begin

-- most actions will not require to skip delay

skip\_delay\_3 <= '0';

case present\_state\_3 is

--Only activated when lights are off (sequence of lights has completed its cycle)

--Allows for skipping delay if lights are off, in state s0 (to activate a sequence instantly)

when s0 =>

if (interm\_3 = '1') then

next\_state\_3 <= s3\_1;

delay\_sel\_3 <= '0';

skip\_delay\_3 <= '1';

elsif (right\_3 = '1') then

next\_state\_3 <= s2\_1;

delay\_sel\_3 <= '1';

skip\_delay\_3 <= '1';

elsif (left\_3 = '1') then

next\_state\_3 <= s1\_1;

delay\_sel\_3 <= '1';

skip\_delay\_3 <= '1';

else

next\_state\_3 <= s0;

delay\_sel\_3 <= '0';

end if;

when s1\_1 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s1\_2;

delay\_sel\_3 <= '1';

end if;

when s1\_2 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s1\_3;

delay\_sel\_3 <= '1';

end if;

when s1\_3 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s1\_0;

delay\_sel\_3 <= '1';

end if;

-- Only s1\_0, s2\_0, s3\_0 and s0 allow to change lights behavior, as the cycle has ended.

when s1\_0 =>

if (delay\_done\_3 = '1') then

if (interm\_3 = '1') then

next\_state\_3 <= s3\_1;

delay\_sel\_3 <= '0';

elsif (right\_3 = '1') then

next\_state\_3 <= s2\_1;

delay\_sel\_3 <= '1';

elsif (left\_3 = '1') then

next\_state\_3 <= s1\_1;

delay\_sel\_3 <= '1';

else

next\_state\_3 <= s0;

delay\_sel\_3 <= '0';

end if;

end if;

when s2\_1 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s2\_2;

delay\_sel\_3 <= '1';

end if;

when s2\_2 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s2\_3;

delay\_sel\_3 <= '1';

end if;

when s2\_3 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s2\_0;

delay\_sel\_3 <= '1';

end if;

when s2\_0 =>

if (delay\_done\_3 = '1') then

if (interm\_3 = '1') then

next\_state\_3 <= s3\_1;

delay\_sel\_3 <= '0';

elsif (right\_3 = '1') then

next\_state\_3 <= s2\_1;

delay\_sel\_3 <= '1';

elsif (left\_3 = '1') then

next\_state\_3 <= s1\_1;

delay\_sel\_3 <= '1';

else

next\_state\_3 <= s0;

delay\_sel\_3 <= '0';

end if;

end if;

when s3\_1 =>

if (delay\_done\_3 = '1') then

next\_state\_3 <= s3\_0;

delay\_sel\_3 <= '0';

end if;

when s3\_0 =>

if (delay\_done\_3 = '1') then

if (interm\_3 = '1') then

next\_state\_3 <= s3\_1;

delay\_sel\_3 <= '0';

elsif (right\_3 = '1') then

next\_state\_3 <= s2\_1;

delay\_sel\_3 <= '1';

elsif (left\_3 = '1') then

next\_state\_3 <= s1\_1;

delay\_sel\_3 <= '1';

else

next\_state\_3 <= s0;

delay\_sel\_3 <= '0';

end if;

end if;

end case;

end process;

-- State actions

C2 : process(present\_state\_3)

begin

case present\_state\_3 is

when s0 =>

rLights\_3 <= "000";

lLights\_3 <= "000";

when s1\_1 =>

rLights\_3 <= "000";

lLights\_3 <= "001";

when s1\_2 =>

rLights\_3 <= "000";

lLights\_3 <= "011";

when s1\_3 =>

rLights\_3 <= "000";

lLights\_3 <= "111";

when s1\_0 =>

rLights\_3 <= "000";

lLights\_3 <= "000";

when s2\_1 =>

lLights\_3 <= "000";

rLights\_3 <= "001";

when s2\_2 =>

lLights\_3 <= "000";

rLights\_3 <= "011";

when s2\_3 =>

lLights\_3 <= "000";

rLights\_3 <= "111";

when s2\_0 =>

lLights\_3 <= "000";

rLights\_3 <= "000";

when s3\_1 =>

lLights\_3 <= "111";

rLights\_3 <= "111";

when s3\_0 =>

lLights\_3 <= "000";

rLights\_3 <= "000";

when others =>

rLights\_3 <= "000";

lLights\_3 <= "000";

end case;

end process;

end Behavior;

**Programa tb\_MOORE\_3.vhd (Testbench):**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.std\_logic\_signed.ALL;

USE ieee.std\_logic\_textio.ALL;

USE std.textio.ALL;

--Entity: no port list!

ENTITY tb\_MOORE\_3 IS

END tb\_MOORE\_3;

--Architecture

ARCHITECTURE test\_architecture OF tb\_MOORE\_3 IS

COMPONENT MOORE\_3

PORT (clk\_3, rst\_3 : in std\_logic;

left\_3, right\_3, interm\_3 : in std\_logic;

lLights\_3, rLights\_3 : out std\_logic\_vector(2 downto 0)

);

END COMPONENT;

SIGNAL clk\_3\_tb : STD\_LOGIC := '0';

SIGNAL rst\_3\_tb, left\_3\_tb, right\_3\_tb, interm\_3\_tb : STD\_LOGIC := '0'; --INPUT

SIGNAL lLights\_3\_tb, rLights\_3\_tb : STD\_LOGIC\_VECTOR( 2 DOWNTO 0 ) := "000"; --INPUT

constant clk\_3\_period : time := 20 ns; --clock period of fpga de10

BEGIN

--DUT Instantiation

DUT : MOORE\_3 PORT MAP( clk\_3\_tb, rst\_3\_tb, left\_3\_tb, right\_3\_tb, interm\_3\_tb, lLights\_3\_tb, rLights\_3\_tb);

-- Process for generating the clock

clock\_process: PROCESS

BEGIN

clk\_3\_tb <= '0';

wait for clk\_3\_period/2;

clk\_3\_tb <= '1';

wait for clk\_3\_period/2;

END PROCESS;

--Stimulus by hand drawn waves, poor coverage

stim\_proc : PROCESS

BEGIN

WAIT FOR 0 ns;

rst\_3\_tb <= '0'; left\_3\_tb <= '0'; right\_3\_tb <= '0'; interm\_3\_tb <= '0';

WAIT FOR clk\_3\_period \* 40;

rst\_3\_tb <= '0'; left\_3\_tb <= '1'; right\_3\_tb <= '0'; interm\_3\_tb <= '0';

WAIT FOR clk\_3\_period \* 40;

rst\_3\_tb <= '0'; left\_3\_tb <= '0'; right\_3\_tb <= '0'; interm\_3\_tb <= '1';

WAIT FOR clk\_3\_period \* 40;

rst\_3\_tb <= '0'; left\_3\_tb <= '0'; right\_3\_tb <= '1'; interm\_3\_tb <= '0';

WAIT FOR clk\_3\_period \* 40;

rst\_3\_tb <= '0'; left\_3\_tb <= '0'; right\_3\_tb <= '0'; interm\_3\_tb <= '0';

WAIT;

END PROCESS;

--Monitor

txt\_out : PROCESS(lLights\_3\_tb, rLights\_3\_tb)

VARIABLE str\_o : LINE;

BEGIN

WRITE( str\_o, STRING'( " lLights\_3\_tb= " )); WRITE( str\_o, lLights\_3\_tb );

WRITE( str\_o, STRING'( " rLights\_3\_tb= " )); WRITE( str\_o, rLights\_3\_tb );

ASSERT false REPORT TIME'IMAGE( NOW ) & str\_o.ALL

SEVERITY note;

DEALLOCATE( str\_o );

END PROCESS;

END ARCHITECTURE;

**Graphical user interface, application

Description automatically generatedResultados de la Simulación:**

**A screenshot of a computer

Description automatically generated***Full testbench simulation, covering left lights, intermittent and right lights.*

*A screenshot of a computer

Description automatically generatedLeft lights behavior close-up, with the 3 LEDs turning on one by one, with a cycle lasting 160ns.*

*A screenshot of a computer

Description automatically generatedBlinking lights behavior, with all lights turning on and off on a cycle lasting 200ns*

*Right lights behavior close-up, with the 3 LEDs turning on one by one, with a cycle lasting 160ns.*