

--------------------------------------------------------------------

-- Name: Prof Jeff Falkinburg

-- Date: Jan 12, 2017

-- File: lec5.vhdl

-- HW: Lecture 5

-- Crs: CSCE 436

-- Purp: This demonstrates some basic I/O as well as gated and ungated

-- outputs.

-- Documentation: No help, I based this off the class notes and readings.

-------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity Lec5 is

 Port ( clk : in STD\_LOGIC;

 reset : in STD\_LOGIC;

 btn : in STD\_LOGIC\_VECTOR (4 downto 0);

 JB : out STD\_LOGIC\_VECTOR (7 downto 0));

end Lec5;

architecture Behavioral of Lec5 is

 signal processQ: unsigned (7 downto 0);

 constant L0: unsigned(7 downto 0) := "00101111";

 constant L1: unsigned(7 downto 0) := "01011111";

 constant L2: unsigned(7 downto 0) := "10111111";

begin

 ------------------------------------------------------------------------------

 -- The buttons are all nominally 0 and equal to 1 when pressed.

 -- btn(3) = '1' Right

 -- btn(1) = '1' Left

 -- btn(2) = '1' Down

 -- btn(0) = '1' Up

 -- btn(4) = '1' Center

 ------------------------------------------------------------------------------

 process(clk)

 begin

 if (rising\_edge(clk)) then

 if (reset = '0') then

 processQ <= (others => '0');

 elsif (btn(4) = '0') then

 processQ <= processQ + 1;

 end if;

 end if;

 end process;

 ------------------------------------------------------------------------------------

 -- The upper nibble of JB will tell us about the range of processQ using combo logic

 ------------------------------------------------------------------------------------

 process(clk)

 begin

 if (rising\_edge(clk)) then

 if (reset = '0') then

 JB(6 downto 4) <= "000";

 elsif ((processQ >= 0) and (processQ < L0)) then

 JB(6 downto 4) <= "001";

 elsif ((processQ >= L0) and (processQ < L1)) then

 JB(6 downto 4) <= "010";

 elsif ((processQ >= L1) and (processQ < L2)) then

 JB(6 downto 4) <= "100";

 elsif (processQ >= L2) then

 JB(6 downto 4) <= "111";

 end if;

 end if;

 end process;

 JB(7) <= clk;

 ------------------------------------------------------------------------------------

 -- The lower nibble of JB will tell us about the range of processQ using gated logic

 ------------------------------------------------------------------------------------

 JB(3 downto 0) <= "0001" when ((processQ >= 0) and (processQ < L0)) else

 "0010" when ((processQ >= L0) and (processQ < L1)) else

 "0100" when ((processQ >= L1) and (processQ < L2)) else

 "1000";

end Behavioral;