

DAISY Inputs DAISY Outputs

|  |  |  |
| --- | --- | --- |
| RFID Scanner = r | Cow Present = c | Timer Status = t |
| 1 - Cow checked in | 1 - cow present | 1 - timer up |
| 0 - Cow not processed | 0 - no cow | 0 - timer running |

|  |  |  |  |
| --- | --- | --- | --- |
| Gate 1 | Gate 2 | Timer | Air Valve |
| 0 – gate closed | 0 – gate closed | 00 Stop timer | 0 closed |
| 1 - gate open | 1 – gate open | 01 Set to 30 sec | 1 open |
|  |  | 10 Set to 3 sec |  |
|  |  | 11 Run timer |  |



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

-- Name: Chris Coulston

-- Date: Jan 28, 2015

-- File: lec10.vhdl

-- Event: Lecture 10

-- Crs: CSCE 436

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

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity lec09 is

 Port( clk: in STD\_LOGIC;

 reset : in STD\_LOGIC;

 sw: in STD\_LOGIC\_VECTOR(2 downto 0);

 cw: out STD\_LOGIC\_VECTOR(4 downto 0));

end lec09;

architecture behavior of lec09 is

 type state\_type is (WaitEnter, WaitRead, Set30, WaitLeave, Set3, Goose);

 signal state: state\_type;

 constant rfid: integer := 2; -- helps keep status bits straight

 constant cow: integer := 1;

 constant timer: integer := 0;

begin

 state\_process: process(clk,reset)

begin

if (rising\_edge(clk)) then

 if (reset = '0') then

 state <= WaitEnter;

 else

 case state is

 when WaitEnter =>

 if (sw(cow) = '1') then state <= WaitRead; end if;

 when WaitRead =>

 if (sw(rfid) = '1') then state <= Set30; end if;

 when Set30 =>

 state <= WaitLeave;

 when WaitLeave =>

 if (sw(cow) = '0') then state <= WaitEnter;

 elsif (sw(timer) = '1' and sw(cow) = '1') then state <= Set3; end if;

 when Set3 =>

 state <= Goose;

 when Goose =>

 if (sw(timer) = '1') then state <= Set30; end if;

 end case;

 end if;

 end if;

 end process;

 cw <= "10000" when state = WaitEnter else

 "00000" when state = WaitRead else

 "01010" when state = Set30 else

 "01110" when state = WaitLeave else

 "01100" when state = Set3 else

 "01111"; -- when state = Goose;

end behavior;