Name: Clyde E. Aparente

Course: BSCpE-3A

**20. STEPPER MOTOR INTERFACE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity STEPPER\_MOTOR\_INTERFACE is

Port (

clk : in STD\_LOGIC;

reset\_n : in STD\_LOGIC;

enable\_n : in STD\_LOGIC;

dir\_n : in STD\_LOGIC;

coil\_n : out STD\_LOGIC\_VECTOR(3 downto 0)

);

end STEPPER\_MOTOR\_INTERFACE;

architecture Behavioral of STEPPER\_MOTOR\_INTERFACE is

signal step\_counter : unsigned(1 downto 0) := "00";

signal clk\_div : unsigned(20 downto 0) := (others => '0');

signal step\_clk : STD\_LOGIC := '0';

signal enabled : STD\_LOGIC := '0';

signal direction : STD\_LOGIC := '0';

type step\_sequence is array (0 to 3) of std\_logic\_vector(3 downto 0);

constant full\_step : step\_sequence := (

"1100",

"0110",

"0011",

"1001"

);

begin

process(clk)

begin

if rising\_edge(clk) then

clk\_div <= clk\_div + 1;

step\_clk <= clk\_div(20);

end if;

end process;

process(step\_clk, reset\_n)

begin

if reset\_n = '0' then

step\_counter <= "00";

coil\_n <= "1111";

elsif rising\_edge(step\_clk) then

if enabled = '1' then

if direction = '1' then

step\_counter <= step\_counter + 1;

else

step\_counter <= step\_counter - 1;

end if;

coil\_n <= full\_step(to\_integer(step\_counter));

else

coil\_n <= "1111";

end if;

end if;

end process;

enabled <= not enable\_n;

direction <= not dir\_n;

end Behavioral;



