```
----------------------------------------------------------------------------------
-- Engineer: Taggart Bonham and Will Chisholm
--
-- Create Date: 05/23/2017 09:05:13 PM
-- Design Name:
-- Module Name: CWController- Architecture
-- Project Name: Etch-a-Sketch final project
-- Target Devices: Digilent Basys3 Board
-- Tool Versions: Vivado 2016.1
-- Description: Architecture of the Controller. Bundles outputs of debounced
--rotary encoders inputted into a knob-decoder. This is the controller for the
--system.

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


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity cwController is
    Port ( clk : in STD_LOGIC;
           a1 : in STD_LOGIC;
           a2 : in STD_LOGIC;
           b1 : in STD_LOGIC;
           b2 : in STD_LOGIC;
           rst : in STD_LOGIC;
           UP : out STD_LOGIC;
           DOWN : out STD_LOGIC;
           LEFT : out STD_LOGIC;
           RIGHT : out STD_LOGIC;
           CLR : out STD_LOGIC);
end cwController;

architecture Behavioral of cwController is


Component debouncer is
PORT (   clk, button    : in STD_LOGIC;
         result         : out std_logic );
  end component;


Component knobdecoder is
PORT (    clk : in STD_LOGIC;
          a   : in STD_LOGIC;
          b   : in STD_LOGIC;
          cw  : out STD_LOGIC;
          ccw : out STD_LOGIC);
  end component;

signal a1_db, a2_db, b1_db, b2_db : std_logic; --wires to go from debouncers' output to
knobdecoders' input

begin
```

```
clearScreen: process (clk) --synchronizes clear signal
begin
if rising_edge(clk) then
    CLR <= '0';
    if rst = '1' then
        CLR <= '1';
    end if;
end if;
end process;

--debouncers debounce the a and b signals from each of the knobs
dba1: debouncer port map (
    clk => clk,
    button => a1,
    result => a1_db);

dba2: debouncer port map (
    clk => clk,
    button => a2,
    result => a2_db);

dbb1: debouncer port map (
    clk => clk,
    button => b1,
    result => b1_db);

dbb2: debouncer port map (
    clk => clk,
    button => b2,
    result => b2_db);

--knob decoders receive the debounced signals and output their clockwise and counterclockwise
for each knob
--to the controller's 4 directions
horizontal: knobdecoder port map(
    clk => clk,
    a => a1_db,
    b => a2_db,
    cw => RIGHT,
    ccw => LEFT);

vertical: knobdecoder port map(
    clk => clk,
    a => b1_db,
    b => b2_db,
    cw => UP,
    ccw => DOWN);

end Behavioral;
```