```
----------------------------------------
--      Author:      Roger Grayson
--      Date: Mar, 2003
--      Purpose:Control Unit for my implementation
--                   of the Coulston computer
----------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity control is
    port(clk, reset: in std_logic;
         cw: out std_logic_vector(20 downto 0);
         sw: in std_logic_vector(6 downto 0));
end control;

architecture behavior of control is
    type state_type is (RST,F1,F2,F3,HFU,R,I,L1,L2,L3,S1,S2,S3,B1,B2);
    signal state: state_type;

begin

    state_process: process(clk, reset)
    begin
        if (reset'event and reset = '0') then
            state <= F1;
        elsif (clk'event and clk='1') then
            case state is
                when RST=> state <= F1;
                when F1 => state <= F2;
                when F2 => state <= F3;
                when F3 => state <= HFU;
                when HFU=>
                    if (sw(5 downto 0) = "000000") then state <= R;
                    elsif (sw(5 downto 0)="00" & X"4" or sw(5 downto 0)="00" & X"5") then
state <= B1;
                    elsif (sw(5 downto 0)="10" & X"3" or sw(5 downto 0)="10" & X"0" or
                        sw(5 downto 0)="10" & X"4") then state <= L1;
                    elsif (sw(5 downto 0)="10" & X"B" or sw(5 downto 0)="10" & X"8") then
state <= S1;
                    else state <= I;end if;
                when R  => state <= F1;
                when I  => state <= F1;
                when B1 =>
                    if (sw(6)='1') then state <= B2;
                    else state <= F1;end if;
                when B2 => state <= F1;
                when L1 => state <= L2;
                when L2 => state <= L3;
                when L3 => state <= F1;
                when S1 => state <= S2;
                when S2 => state <= S3;
                when S3 => state <= F1;
```

```
            end case;
        end if;
    end process;

    output_process: process(state)
    begin
    case state is
        when RST=> cw <= '0' & X"00000";
        when F1 => cw <= '0' & X"01004";
        when F2 => cw <= '0' & X"1C6D4";
        when F3 => cw <= '0' & X"02003";
        when HFU=> cw <= '0' & X"00000";
        when R  => cw <= '0' & X"A0000";
        when I  => cw <= '0' & X"70008";
        when L1 => cw <= '0' & X"19808";
        when L2 => cw <= '0' & X"006C0";
        when L3 => cw <= '0' & X"60003";
        when S1 => cw <= '0' & X"19808";
        when S2 => cw <= '1' & X"00400";
        when S3 => cw <= '0' & X"001A0";
        when B1 => cw <= '0' & X"88000";
        when B2 => cw <= '0' & X"1C00C";
    end case;
    end process;
end behavior;
```