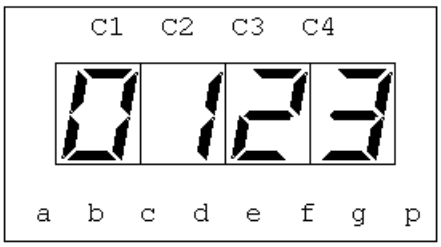
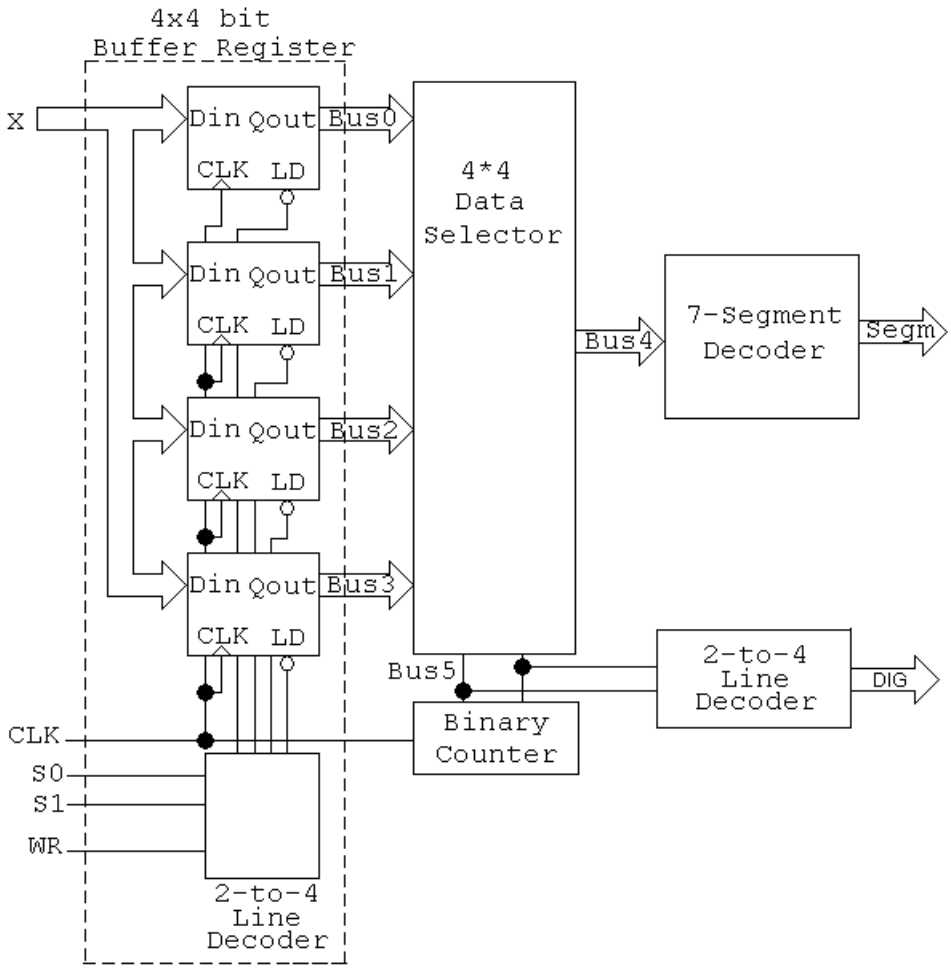
การทดลองที่ 9 วงจรสแกนดิสเพลย์ 7 ส่วน 4 หลัก






```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec2seg is
  port (bin : in std_logic_vector(3 downto 0);
          segm : out std_logic_vector(7 downto 0));
end dec2seg;

architecture Behavioral of dec2seg is
begin
```

```
      with bin select
            segm <= "00000011" when "0000",
                    "10011111" when "0001",
                    "00100101" when "0010",
                    "00001101" when "0011",
                    "10011001" when "0100",
                    "01001001" when "0101",
                    "01000001" when "0110",
                    "00011111" when "0111",
                    "00000001" when "1000",
                    "00001001" when others;
end Behavioral;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.ALL;
entity binco is
    port (clk : in std_logic;
            y : out std_logic_vector(1 downto 0));
end binco;
architecture binco_beh of binco is
type state_type is (a, b, c, d);
signal state: state_type;
begin
  process (clk)
  begin
    if clk = '1' and clk'event then
       case state is
               when a =>  state <= b;
               when b =>  state <= c;
               when c =>  state <= d;
               when d =>  state <= a;
       end case;
    end if;
end process;
process (state)
   begin
      case state is
      when a => y <= "00";
      when b => y <= "01";
      when c => y <= "10";
      when d => y <= "11";
      end case;
   end process;
end binco_beh;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity disp is
    Port (clk,wr : in std_logic;
             x : in std_logic_vector(3 downto 0);
```

```
             s : in std_logic_vector(1 downto 0);
             seg : out std_logic_vector(7 downto 0);
             dig : out std_logic_vector(3 downto 0));
end disp;

architecture Behavioral of disp is

component dec2seg
  port (bin : in std_logic_vector(3 downto 0);
          segm : out std_logic_vector(7 downto 0));
end component;
component binco
    port (clk : in std_logic;
            y : out std_logic_vector(1 downto 0));
end component;
signal bus0,bus1,bus2,bus3, bus4 : std_logic_vector(3 downto 0);
signal bus5 :  std_logic_vector(1 downto 0);
begin


process (CLK)
begin
    if CLK='1' and CLK'event then
       if wr = '0' then
          case s is
               when "00"  => bus0 <= x;
               when "01"  => bus1 <= x;
               when "10"  => bus2 <= x;
               when others  => bus3 <= x;
          end case;
       end if;
    end if;
end process;

process (bus5, bus0, bus1, bus2, bus3)
begin
    case bus5 is
               when "00"  =>
                  bus4 <= bus0;
                  dig <= "1110";
               when "01"  =>
                  bus4 <= bus1;
                  dig <= "1101";
               when "10"  =>
                  bus4 <= bus2;
                  dig <= "1011";
               when others  =>
                  bus4 <= bus3;
                  dig <= "0111";
    end case;
end process;
  c1: binco port map(clk, bus5);
  c2: dec2seg port map(bus4, seg);

end Behavioral;
```