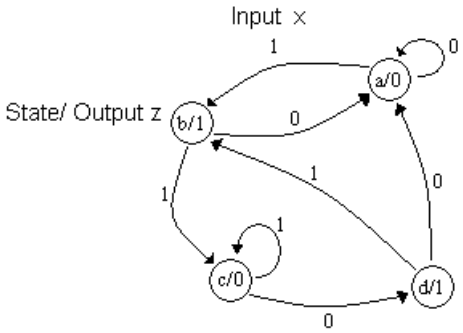
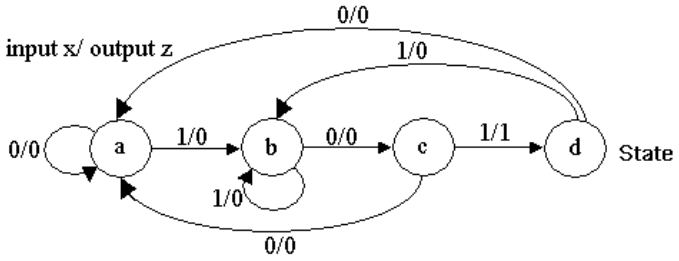
การทดลองที่ 7 State Machine

Moore Type




```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.ALL;
entity moore is
    port (x, clk : in std_logic;
            z : out std_logic);
end moore;
architecture moore_beh of moore 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 =>      if x = '1' then
                                state <= b;
                           else
                                state <= a;
                           end if;
            when b =>      if x = '1' then
                                state <= c;
                           else
                                state <= a;
                           end if;
            when c =>      if x = '1' then
                                state <=c;
                           else
                                state <= d;
                           end if;
            when d =>      if x = '1' then
                                state <= b;
                           else
                                state <= a;
                           end if;
       end case;
    end if;
end process;
process (state)
   begin
      case state is
             when a => z <= '0';
             when b => z <= '1';
             when c => z <= '0';
             when d => z <= '1';
      end case;
   end process;
end moore_beh;
```

Mealy Type




```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.ALL;
entity mealy is
    port (x, clk : in std_logic;
            z : out std_logic);
end mealy;
architecture mealy_beh of mealy 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 =>      if x = '1' then   state <= b;
                         else  state <= a; end if;
               when b =>      if x = '1' then   state <= b;
                         else  state <= c; end if;
               when c =>      if x = '1' then   state <=d;
                              else  state <= a;       end if;
               when d =>      if x = '1' then   state <= b;
                           else     state <= a;       end if;
      end case;
     end if;
end process;
process (state)
   begin
      case state is
            when a =>   if x = '1' then z <= '0';
                        else  z <= '0';   end if;
            when b =>   if x = '1' then z <= '0';
                        else  z <= '0';   end if;
            when c =>   if x = '1' then z <= '1';
                        else  z <= '0';   end if;
            when d =>   if x = '1' then z <= '0';
                        else  z <= '0';   end if;
      end case;
end process;
end mealy_beh;
```

1. จงออกแบบวงจรนับเลขฐานสองขนาด 2 บิตแบบนับขึ้น โดยใช้ State Machine มีเพียงสัญญาณนาฬิกาสำหรับควบคุมการนับเท่านั้น

2. จากข้อ 1 ให้เพิ่มสัญญาณควบคุมการนับ CE