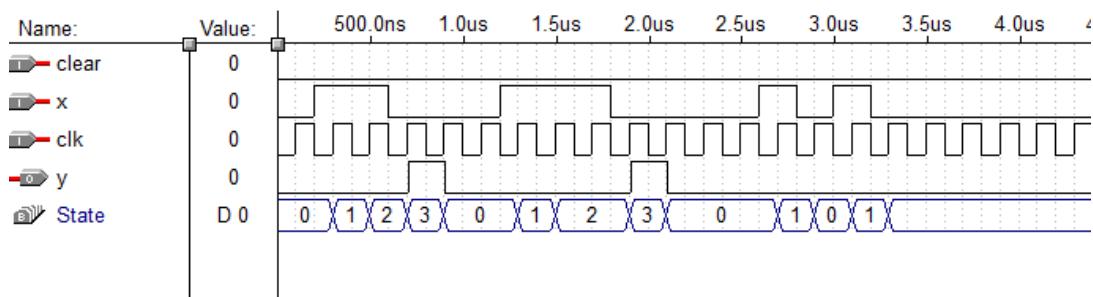
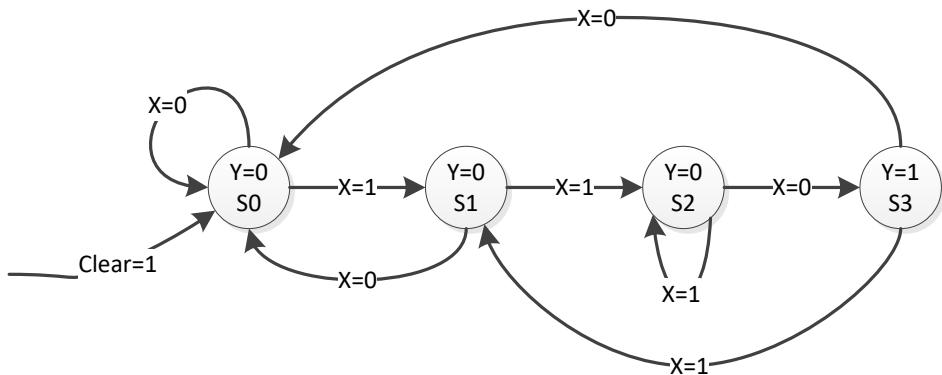


## מוכנת MOORE המגלה סדרה 110



```

entity moor110 is
  port ( clk,clear,x: in bit;
         y: out bit);
end ;

architecture moor of moor110 is
  type State_type is (S0,S1, S2, S3);
  signal State: State_type;

begin
  process (clk)
  begin

    if clk'event and clk = '1' then
      if clear='1' then      State <= S0;
      else
        case State is
          when S0 => if x='1' then      State <= S1;      end if;

          when S1 => if x='1' then State <= S2;
                        else      State <= S0;      end if;

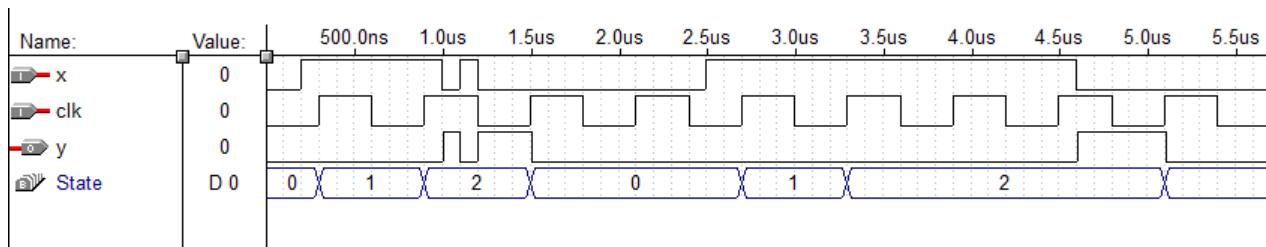
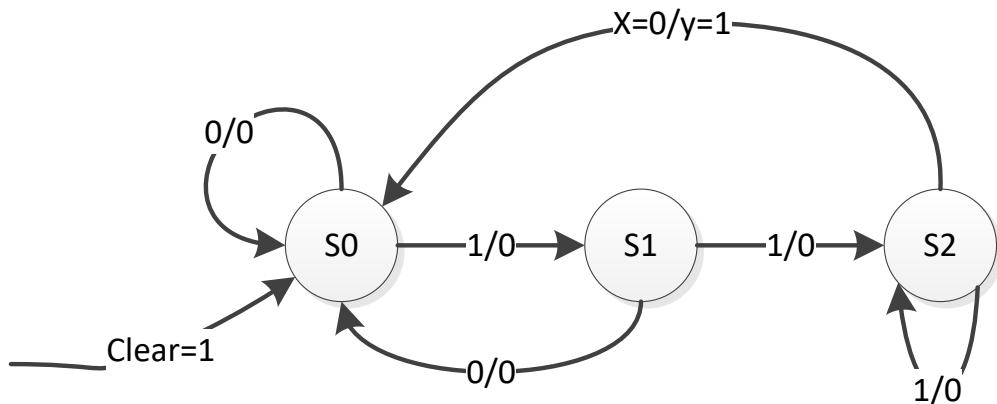
          when S2 =>      if x='0' then State <= S3;      end if;

          when S3 =>      if x='0' then State <= S0;
                        else      State <= S1;      end if;

        end case;
      end if;
    end process;
  -----
  y<='1' when state=s3 else '0';
  
```

```
end ;
```

## מכונת mealy המגלה סדרה 110



```

entity AsynMealy110 is
    port (      clk,clear,x: in bit;
                y: out bit);
end ;

architecture mealy of AsynMealy110 is
    type State_type is (S0,S1, S2);
    signal State: State_type;

begin
    process (clk)
    begin

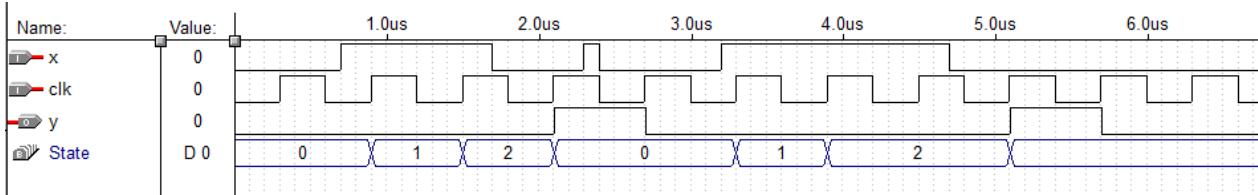
        if clk'event and clk = '1' then
            if clear='1' then      State <= S0;
            else
                case State is
                    when S0 => if x='1' then State <= S1; end if;

                    when S1 => if x='1' then State <= S2;
                    else State <= S0; end if;

                    when S2 =>      if x='0' then State <= S0; end if;
                end case;
            end if;
        end process;
        -----
        y<='1' when state=s2 and x='0' else '0';
    end ;

```

**מכונת mealy עם מוצא סינכרוני המגלה סדרה 110**



```

entity SynMealy110 is
    port ( clk,clear,x: in bit;
                    y: out bit);
end;

architecture mealy of SynMealy110 is
    type State_type is (S0,S1, S2);
    signal State: State_type;

begin
    process (clk)
begin

    if clk'event and clk = '1' then
        if clear='1' then      State <= S0;
        else
            case State is
                when S0 =>
                    if x='1' then State <= S1; end if;

                when S1 =>
                    if x='1' then State <= S2;
                    else State <= S0; end if;

                when S2 =>
                    if x='0' then      State <= S0; end if;
            end case;
        end if;
    end if;
end process;

process (clk)
begin
    if clk'event and clk = '1' then
        if clear='1' then      y <= '0';
        elsif state=S2 and x='0' then      y <= '1';
        else y <= '0';
        end if;
    end if;
end process;

end ;

```

ב- Mealy סינכרוני רואים שהמוצא י' משתנה במעבר מ מצב 2 ל- 0 (בעלית שעון) בסיום גילו הקוד.

אם ב- Moore המוצא תלוי אך ורק במצב, רואים ב- Mealy עברו אותו מצב 0 נוכל לקבל 1 או 0 בהתאם לבבבינה  $x$ .

## מוכנת PresentState, NextState: סדרה 110 MOORE בעזרת

```

entity moor110 is
    port ( clk,clear,x: in bit;
           y: out bit);
end ;

architecture moor of moor110 is
    type State_type is (S0,S1, S2, S3);
    signal PresentState, NextState: State_type;

begin
    process (clk)
    begin

        if clk'event and clk = '1' then
            if clear='1' then      PresentState <= S0;
            else      PresentState <= NextState;
            end if;
        end if;
    end process;

    process(PresentState , x)
    begin
        y<='0'; --default output
        case PresentState is
            when S0 => if x='1' then NextState <= S1;
            else      NextState <= S0; end if;

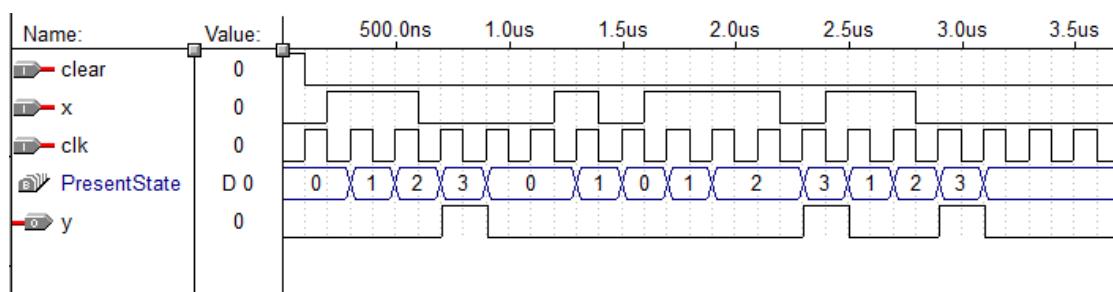
            when S1 => if x='1' then NextState <= S2;
            else      NextState <= S0; end if;

            when S2 => if x='0' then NextState <= S3;
            else      NextState <= S2; end if;

            when S3 => y<='1';
            if x='0' then      NextState <= S0;
            else      NextState <= S1; end if;

        end case;
    end process;
end ;

```



המוצא י נרשם מוחז לתנאי של הכניסה X ולכן תלוי אך ורק במצב של PresentState

## מכונת Mealy המגלת סדרה 110 בעזרת PresentState, NextState:

```

entity mealy110 is
    port ( clk,clear,x: in bit;
           y: out bit);
end ;

architecture mealy of mealy110 is
    type State_type is (S0,S1, S2);
    signal PresentState, NextState: State_type;

begin
    process (clk)
    begin

        if clk'event and clk = '1' then
            if clear='1' then      PresentState <= S0;
            else      PresentState <= NextState;
            end if;
        end if;
    end process;

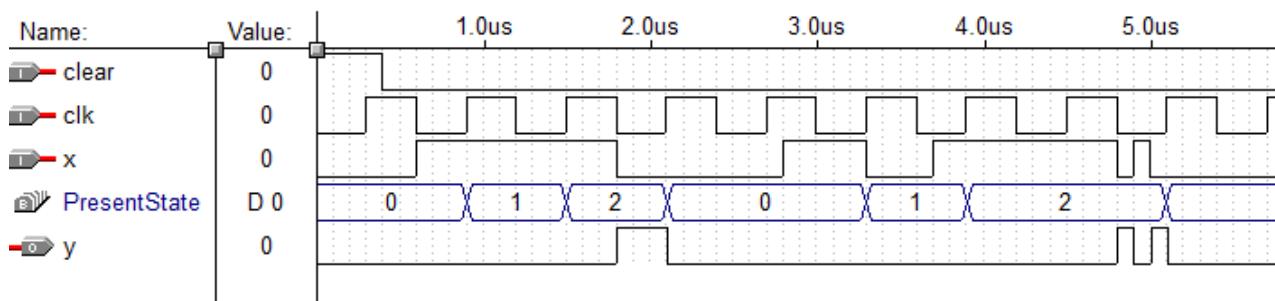
    process(PresentState , x)
    begin
        y<='0'; --default output
        case PresentState is
            when S0 => if x='1' then NextState <= S1;
            else      NextState <= S0;   end if;

            when S1 => if x='1' then NextState <= S2;
            else      NextState <= S0;   end if;

            when S2 => if x='0' then y<='1' ; NextState <= S0;
            else      NextState <= S2;   end if;

        end case;
    end process;
end ;

```



המוצא י נרשם בהתאם של הכניסה X ולכן תלוי במצב של PresentState ובכניסה X