

פתרון 1

.א.

```

library ieee;
use ieee.std_logic_1164.all;

entity buffer1 is
    port(A,S:in std_logic;
         F:out std_logic);
end;

architecture buff of buffer1 is
begin
    F<= A when S='1' else 'Z';
end;

```

.ב.

```

library ieee;
use ieee.std_logic_1164.all;

entity buftobuf is
    port(x,y,s1,s2:in std_logic;
         fout:out std_logic);
end;

architecture buf of buftobuf is
    component buffer1
        port(A,S:in std_logic;
             F:out std_logic);
    end component ;

begin
    u1:buffer1
        port map(A=>x,S=>s1,F=>fout);
    u2:buffer1
        port map(A=>y,S=>s2,F=>fout);
end;

```

.ג.

S1	S2	x	y	fout
0	0	0	0	Z
0	0	0	1	Z
0	0	1	0	Z
0	0	1	1	Z
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	X
1	1	1	0	X
1	1	1	1	1

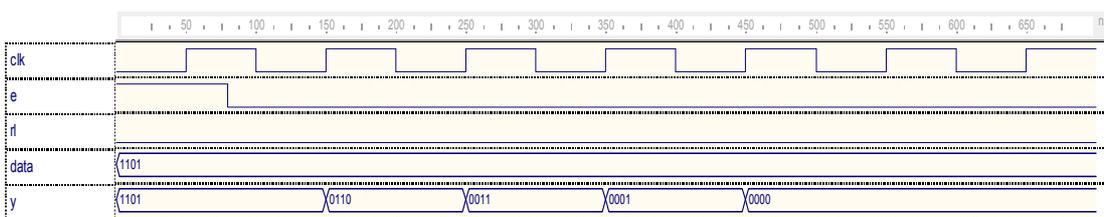
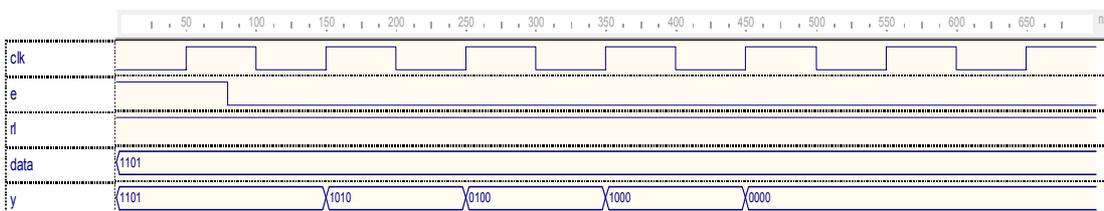
פתרון 2

F=0101

V= 1110

W=1010

פתרון 3

א. אוגר הזזה ימינה כאשר $rl=0$ ושמאלה כאשר $rl=1$, טעינת המידע כאשר $e=1$ ב. $Y=1101$ ג. $rl=0$ ד. $rl=1$ 

פתרון 4

.א.

DFF1

```

entity dff1 is
  port(
    ck,d:in bit;
    q:out bit
  );
end;

architecture dff of dff1 is
begin
  process(ck)
  begin
    if ck'event and ck='1' then
      q<=d;
    end if;
  end process;
end dff;

```

שער AND3

```

entity and3 is
  port(
    i0,i1,i2:in bit;
    o:out bit
  );
end;

architecture and3 of and3 is
begin
  o<=i0 and i1 and i2;
end;

```

ב. מעגל ראשי

הערה: יש לשים לב שאפשר לרשום בצורה מקוצרת את הביטוי

```
port map( ck=>ck, d=>din, q=>net1);
```

לצורה :

```
port map( ck,din,net1);
```

אבל חייבים לשמור על סדר החיבור.

```
entity shift3 is
  port(ck,din : in bit;
        out1: out bit);
end;

architecture shif3 of shift3 is

  signal net1,net2,net3: bit;
  component and3
    port(
      i0,i1,i2:in bit;
      o:out bit
    );
  end component ;

  component dff1
    port(
      ck,d:in bit;
      q:out bit
    );
  end component;

begin
  u1:and3
  port map(net1,net2,net3,out1);

  u2:dff1
  port map( ck,din,net1);

  u3:dff1
  port map(ck,net1,net2);

  u4:dff1
  port map(ck,net2,net3);

end;
```

פתרון 5

.א.

```
entity mux4to1 is
  port(d: in bit_vector(3 downto 0);
        s: in integer range 0 to 3;
        output: out bit);
end;
```

```
architecture mux of mux4to1 is
begin
  process(d,s)
  begin
    output<=d(s);
  end process;
end;
```

.ב.

```
entity mux256to1 is
  port(d: in bit_vector(255 downto 0);
        s: in integer range 0 to 255;
        output: out bit);
end;
```

```
architecture mux of mux256to1 is
begin
  process(d,s)
  begin
    output<=d(s);
  end process;
end;
```

פתרון 6

א. מונה נעלה מ-0 עד 255 העובד בעליית השעון עם איפוס סינכרוני הפעיל בנמוך

ב. במקום הפקודה : `cnt := cnt + 1;`

נרשום :

```

if cnt < 99 then
  cnt := cnt + 1;
else
  cnt := 0;
end if;

```

אפשר להגדיר את המוצא q והמשתנה cnt - `INTEGER RANGE 0 TO 99;`

ג.

```

PROCESS (clk,clear)
  VARIABLE cnt : INTEGER RANGE 0 TO 255;
  BEGIN
    IF clear = '0' THEN
      cnt := 0;

      elsif (clk'EVENT AND clk = '1') THEN

        cnt := cnt + 1;
        END IF;
        END IF;
        q <= cnt;
    END PROCESS;

```

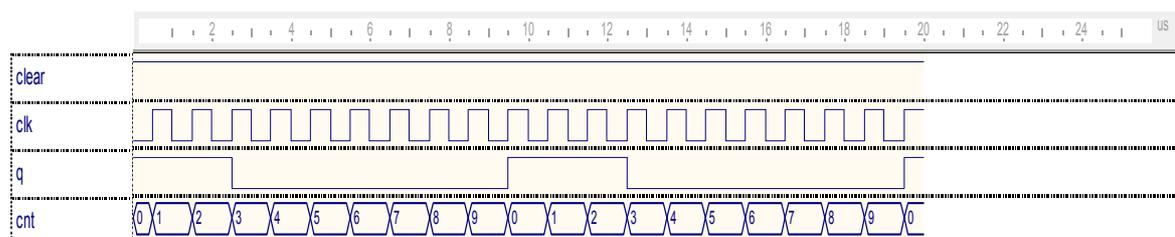
ד.

```

IF cnt < 50 AND cnt > 20 THEN
  f <= '1';
ELSE
  f <= '0';
END IF;

```

פתרון 7



```
ENTITY count_bcd IS
  PORT
    (clk,clr,u_d : IN BIT;
     q : OUT integer RANGE 0 TO 9);
END;

ARCHITECTURE count OF count_bcd IS
BEGIN

  PROCESS (clk)
    VARIABLE cnt : INTEGER RANGE 0 TO 9;
  BEGIN
    IF (clk'EVENT AND clk = '1') THEN
      IF clr = '1' THEN
        cnt := 0;
      ELSIF u_d='1' then
        IF (cnt<9) then
          cnt := cnt + 1;
        ELSE
          cnt := 0;
        END IF;
      ELSE
        IF (cnt>0) then
          cnt := cnt - 1;
        ELSE
          cnt:=9;
        END IF;
      END IF;
    END IF;
    q<=cnt;
  END PROCESS;
END ;
```

פתרון 9

א. לולאה של 100 פעמים בודקת את כל הסיביות מביט 0 עד 99

ב. אתחול המשתנה בכל פעם שיש שינוי ב-d

ג. `IF d(i) = '0' THEN`

ד.

```
FOR i IN 0 to 99 LOOP
    num_bits := num_bits xor d(i);
END LOOP;
```

ה. דרך א- שימוש ב while

```
ENTITY count_value_1 IS
    PORT(
        d : IN BIT_VECTOR (99 DOWNT0 0);
        q : OUT INTEGER RANGE 0 TO 100);
END count_value_1;

ARCHITECTURE maxpld OF count_value_1 IS
BEGIN
    PROCESS (d)
        VARIABLE i : INTEGER RANGE 0 TO 100;
    BEGIN
        i:=0;

        while d(i)='0' loop

            i:=i+1;
            exit when i=100;
        END LOOP;

        q <=i;
    END PROCESS;
END maxpld;
```

דרך ב – שימוש ב-for

```
ARCHITECTURE maxpld OF count_value_1 IS
BEGIN
    PROCESS (d)
        VARIABLE count : INTEGER RANGE 0 TO 100;
    BEGIN
        count:=0;
        for i in 0 to 99 loop
            exit when d(i)='1';
            count:=count+1;
        END LOOP;

        q <=count;
    END PROCESS;
END;
```

פתרון 10

בתוכנית הראשונה, המשתנה הוא variable לכן בתוך התהליך הוא יתקדם פעמיים ולכן כל פולס שעון המונה יתקדם ב-2

בתוכנית השנייה המשתנה הוא signal ורק ההשמה האחרונה מתבצעת בכל פולס שעון ולכן המונה יתקדם ב-1

פתרון 11

```
entity moor1 is
  port(reset,clk,x: in bit;
        q: out bit_vector(1 downto 0));
end;

architecture behave of moor1 is
  type state_type is(s1,s2,s3);
  signal state: state_type;
begin
  process (clk,reset)
  begin
    if reset='1' then state<=s1;
    elsif clk'event and clk='1' then
      case state is

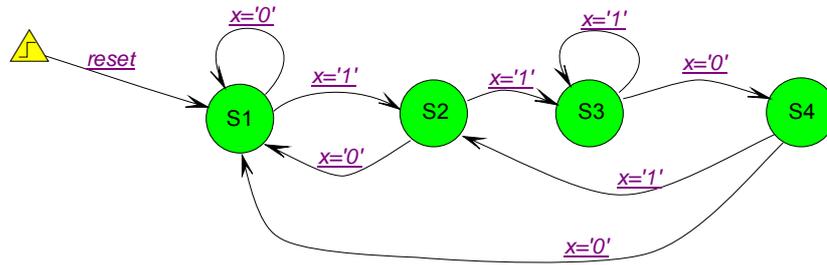
        when s1=> state<=s2;

        when s2=>
          if x='1' then state<=s3;
          end if;

        when s3=> state<=s1;

        end case;
      end if;
    end process;

  process(state)
  begin
    if state=s1 then
      q<="00";
    elsif state=s2 then
      q<="01";
    elsif state=s3 then
      q<="10";
    end if;
  end process;
end;
```



```

entity code110 is
  port(reset,clk,x: in bit;
        q: out bit);
end;

architecture behave of code110 is
  type state_type is(s1,s2,s3,s4);
  signal state: state_type;
begin
  process (clk)
  begin
    if reset='1' then state<=s1;
    elsif clk'event and clk='1' then
      case state is

        when s1=>
          if x='1' then state<=s2;
          end if;

        when s2=>
          if x='0' then state<=s1;
          else state<=s3;
          end if;

        when s3=>
          if x='0' then state<=s4;
          end if;

        when s4=>
          if x='0' then state<=s1;
          else state<=s2;
          end if;

      end case;
    end if;
  end process;

  process(state)
  begin
    if state=s4 then
      q<='1';
    else
      q<='0';
    end if;
  end process;
end;

```