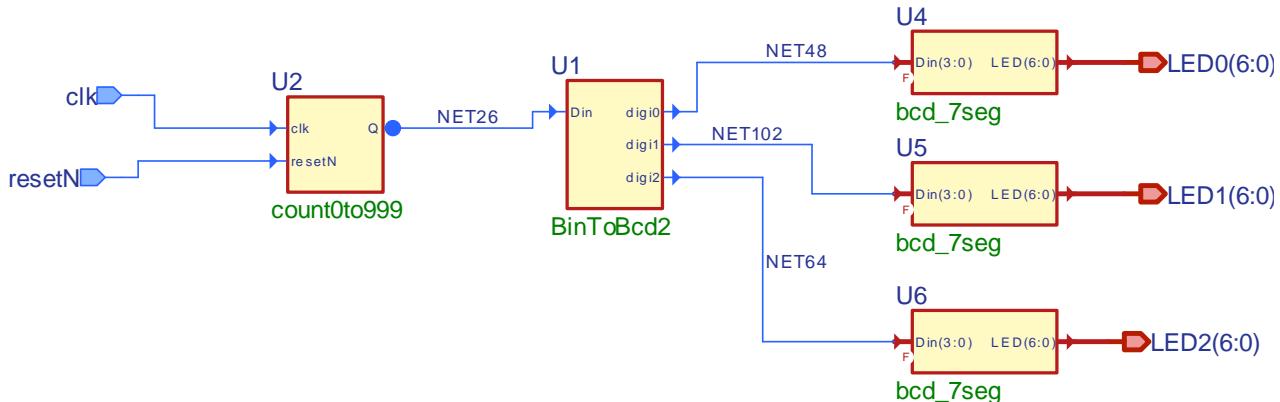


מונח מ-0 עד 999 עם 3 תצוגות 7segment



המונח מורכב מ- 3 חלקים :

1. מונח הסופר מ-0 עד .999
2. מפענח מערך 0 עד 999 ל-BCD .
3. מפענחים מ-BCD לתצוגת 7segment אנוודיה משותפת.

תוכנית מונח 0 עד 999

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity count0to999 is
    port(clk, resetN: in std_logic;
        Q: buffer integer range 0 to 999);
end;

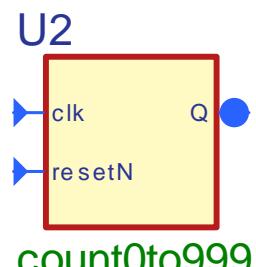
architecture count of count0to999 is
begin

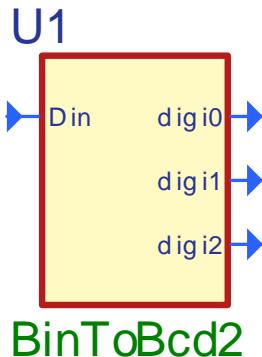
    process(clk,resetN)

    begin
        if resetN='0' then Q<=0;
        elsif rising_edge(clk) then
            if Q<999 then Q<=Q+1;
            else Q<=0;
            end if;
        end if;

    end process;
end;

```





```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity BinToBcd2 is
    port(Din: in integer range 0 to 999;
        digi0,digi1,digi2: out integer range 0 to 9);
end;

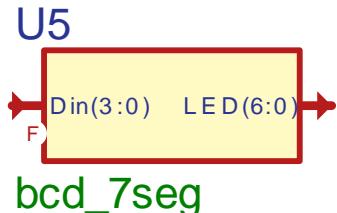
architecture arc_BinToBcd2 of BinToBcd2 is
begin

    process(Din)
    begin
        digi0<=Din mod 10;
        digi1<=(Din/10) mod 10;
        digi2<=(Din/100);

    end process;
end;

```

מפענה מ-BCD לתצוגת 7segment אנוודת משותפת



```

library IEEE;
use IEEE.std_logic_1164.all;

entity bcd_7seg is
    port(LED: out STD_LOGIC_VECTOR (6 downto 0);
         Din: in std_logic_vector(3 downto 0));
end;
architecture arc_7seg of bcd_7seg is

begin

    -- segment encoding
    --   0
    --   ---
    -- 5 |   | 1
    --   --- <- 6
    -- 4 |   | 2
    --   ---
    --   3

    with Din select
    LED <=
    "1111001" when "0001", --1
    "0100100" when "0010", --2
    "0110000" when "0011", --3
    "0011001" when "0100", --4
    "0010010" when "0101", --5
    "0000010" when "0110", --6
    "1111000" when "0111", --7
    "0000000" when "1000", --8
    "0010000" when "1001", --9
    "1000000" when others; --0

end;

```

תוכנית ראשית שהתוכנה יוצרה מהתצוגה הגרפית

```

-- Design unit header --
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_signed.all;
use IEEE.std_logic_unsigned.all;

entity top is
  port(
    clk : in STD_LOGIC;
    LED0 : out STD_LOGIC_VECTOR(6 downto 0);
    LED1 : out STD_LOGIC_VECTOR(6 downto 0);
    LED2 : out STD_LOGIC_VECTOR(6 downto 0);
    resetN : in STD_LOGIC
  );
end top;

architecture top of top is

function aldec_int2slv4 (val:integer) return std_logic_vector is
begin
  return conv_std_logic_vector(val,4);
end aldec_int2slv4;

----- Component declarations -----

component bcd_7seg
  port(
    LED : out STD_LOGIC_VECTOR(6 downto 0);
    Din : in STD_LOGIC_VECTOR(3 downto 0)
  );
end component;
component BinToBcd2
  port(
    Din : in INTEGER range 0 to 999;
    digi0 : out INTEGER range 0 to 9;
    digi1 : out INTEGER range 0 to 9;
    digi2 : out INTEGER range 0 to 9
  );
end component;
component count0to999
  port(
    clk : in STD_LOGIC;
    resetN : in STD_LOGIC;
    Q : buffer INTEGER range 0 to 999
  );
end component;

----- Signal declarations used on the diagram -----

signal NET102 : INTEGER range 0 to 9;
signal NET26 : INTEGER range 0 to 999;
signal NET48 : INTEGER range 0 to 9;
signal NET64 : INTEGER range 0 to 9;

begin

```

---- Component instantiations ----

```

U1 : BinToBcd2
  port map(
    Din => NET26,
    digi0 => NET48,
    digi1 => NET102,
    digi2 => NET64
  );

U2 : count0to999
  port map(
    clk => clk,
    resetN => resetN,
    Q => NET26
  );

U4 : bcd_7seg
  port map(
    LED => LED0,
    Din => aldec_int2slv4(NET48)
  );

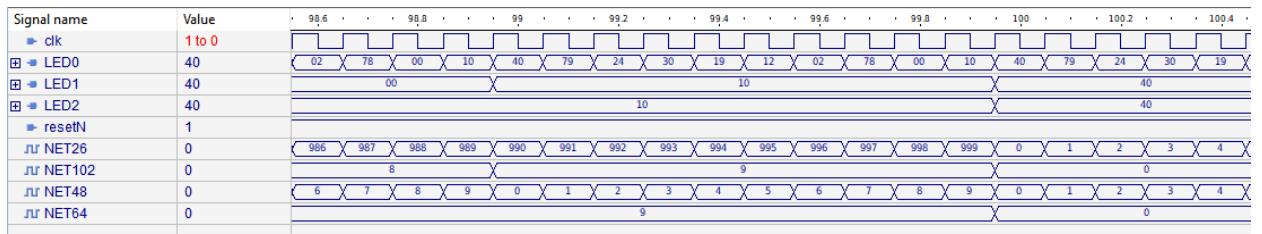
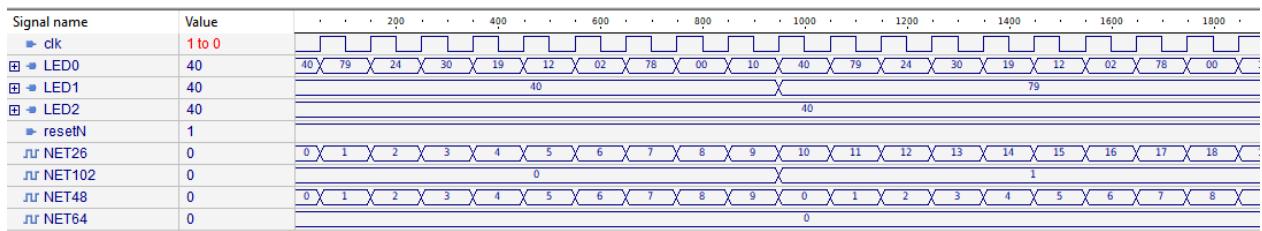
U5 : bcd_7seg
  port map(
    LED => LED1,
    Din => aldec_int2slv4(NET102)
  );

U6 : bcd_7seg
  port map(
    LED => LED2,
    Din => aldec_int2slv4(NET64)
  );

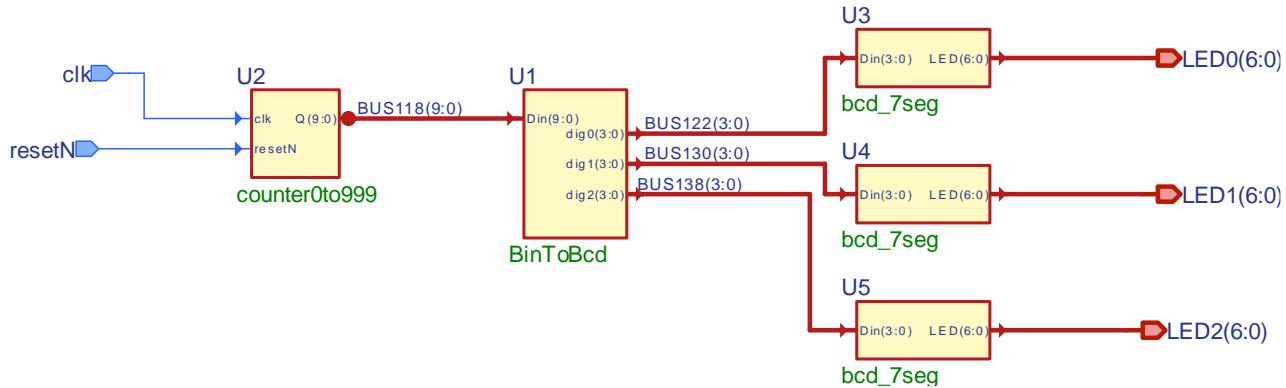
end top;

```

סימולציה

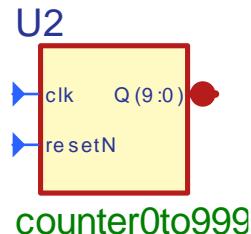


**מימוש המונה כאשר כל הכניסות והיציאות
מוגדרות כ- `std_logic` ונשתמש בספריות המרה.**



תוכנית מונה 0 עד 999

הספירה; `use IEEE.std_logic_unsigned.all;`



```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity counter0to999 is
    port(clk, resetN: in std_logic;
         Q: buffer std_logic_vector(9 downto 0));-- 0 to 999
end;

architecture counter of counter0to999 is
begin

process(clk,resetN)
begin
    if resetN='0' then Q<=(others => '0');
    elsif rising_edge(clk) then
        if Q < 999 then Q<=Q+1;
        else Q<=(others => '0');
        end if;
    end if;

end process;
end;

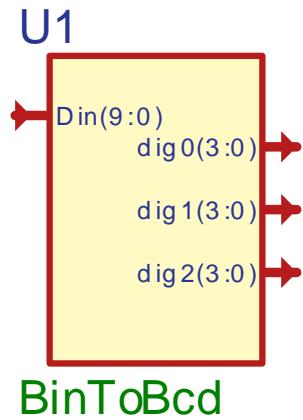
```

מפענה מערך 0 עד 999 ל-BCD

הספריה `use IEEE.std_logic_arith.all;` מאפשרת פעולות המרה באמצעות הפונקציות :

integer-ל std_logic – conv_integer

std_logic -ל integer – conv_std_logic_vector



```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use IEEE.std_logic_arith.all;

entity BinToBcd is
    port(Din: in std_logic_vector(9 downto 0);-- 0 to 999
          dig0,dig1,dig2: out std_logic_vector(3 downto 0));
end;

architecture arc_BinToBcd of BinToBcd is
begin

    process(Din)

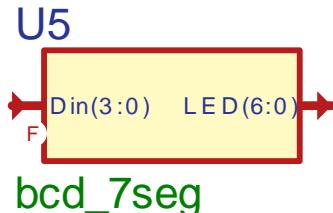
        variable temp : integer range 0 to 999;
        variable Unity, dozens, hundreds :integer range 0 to 9;
    begin
        temp:= conv_integer(Din);
        Unity:=temp mod 10;
        dozens:=(temp/10) mod 10;
        hundreds:= temp/100;

        dig0<= conv_std_logic_vector(Unity,4);
        dig1<= conv_std_logic_vector(dozens,4);
        dig2<= conv_std_logic_vector(hundreds,4);

    end process;
end;

```

מפענה מ-BCD לתצוגת 7segment אנוודת משותפת



```

library IEEE;
use IEEE.std_logic_1164.all;

entity bcd_7seg is
    port(LED: out STD_LOGIC_VECTOR (6 downto 0);
         Din: in std_logic_vector(3 downto 0));
end;
architecture arc_7seg of bcd_7seg is

begin

    -- segment encoding
    --   0
    --   ---
    -- 5 |   | 1
    --   --- <- 6
    -- 4 |   | 2
    --   ---
    --   3

    with Din select
    LED <=
    "1111001" when "0001", --1
    "0100100" when "0010", --2
    "0110000" when "0011", --3
    "0011001" when "0100", --4
    "0010010" when "0101", --5
    "0000010" when "0110", --6
    "1111000" when "0111", --7
    "0000000" when "1000", --8
    "0010000" when "1001", --9
    "1000000" when others; --0

end;

```

תוכנית ראשית שהתוכנה יוצרה מהתצוגה הגרפית

```

-- Design unit header --
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_signed.all;
use IEEE.std_logic_unsigned.all;

entity top2 is
  port(
    clk : in STD_LOGIC;
    resetN : in STD_LOGIC;
    LED0 : out STD_LOGIC_VECTOR(6 downto 0);
    LED1 : out STD_LOGIC_VECTOR(6 downto 0);
    LED2 : out STD_LOGIC_VECTOR(6 downto 0)
  );
end top2;

architecture top2 of top2 is

---- Component declarations ----

component bcd_7seg
  port(
    LED : out STD_LOGIC_VECTOR(6 downto 0);
    Din : in STD_LOGIC_VECTOR(3 downto 0)
  );
end component;
component BinToBcd
  port(
    Din : in STD_LOGIC_VECTOR(9 downto 0);
    dig0 : out STD_LOGIC_VECTOR(3 downto 0);
    dig1 : out STD_LOGIC_VECTOR(3 downto 0);
    dig2 : out STD_LOGIC_VECTOR(3 downto 0)
  );
end component;
component counter0to999
  port(
    clk : in STD_LOGIC;
    resetN : in STD_LOGIC;
    Q : buffer STD_LOGIC_VECTOR(9 downto 0)
  );
end component;

---- Signal declarations used on the diagram ----

signal BUS118 : STD_LOGIC_VECTOR(9 downto 0);
signal BUS122 : STD_LOGIC_VECTOR(3 downto 0);
signal BUS130 : STD_LOGIC_VECTOR(3 downto 0);
signal BUS138 : STD_LOGIC_VECTOR(3 downto 0);

begin

---- Component instantiations ----

U1 : BinToBcd
  port map(
    Din => BUS118,
    dig0 => BUS122,
    dig1 => BUS130,

```

```

dig2 => BUS138
);

U2 : counter0to999
port map(
    clk => clk,
    resetN => resetN,
    Q => BUS118
);

U3 : bcd_7seg
port map(
    LED => LED0,
    Din => BUS122
);

U4 : bcd_7seg
port map(
    LED => LED1,
    Din => BUS130
);

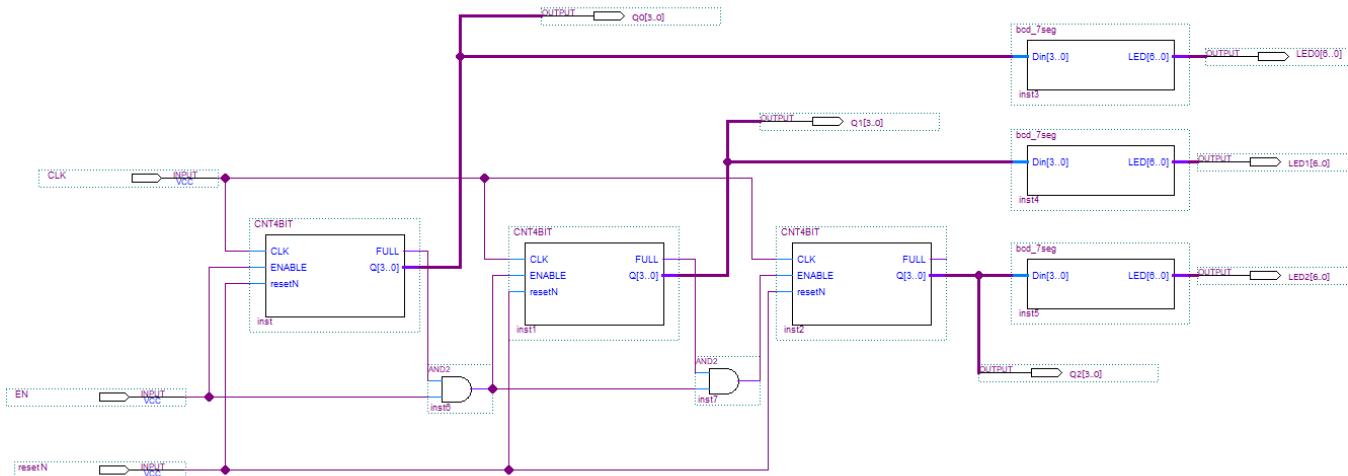
U5 : bcd_7seg
port map(
    LED => LED2,
    Din => BUS138
);

end top2;

```

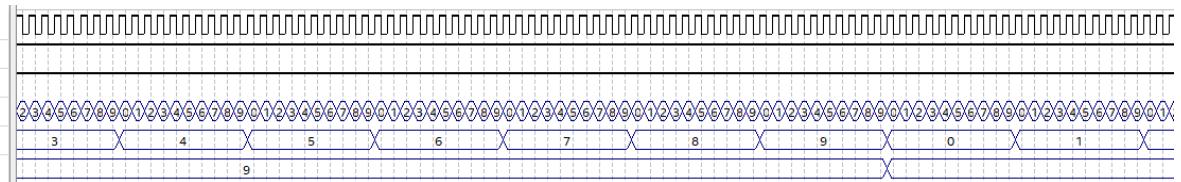
דרך נוספת

תכון באמצעות 3 מונחים משוררים עם אפשרות באמצעות שער AND

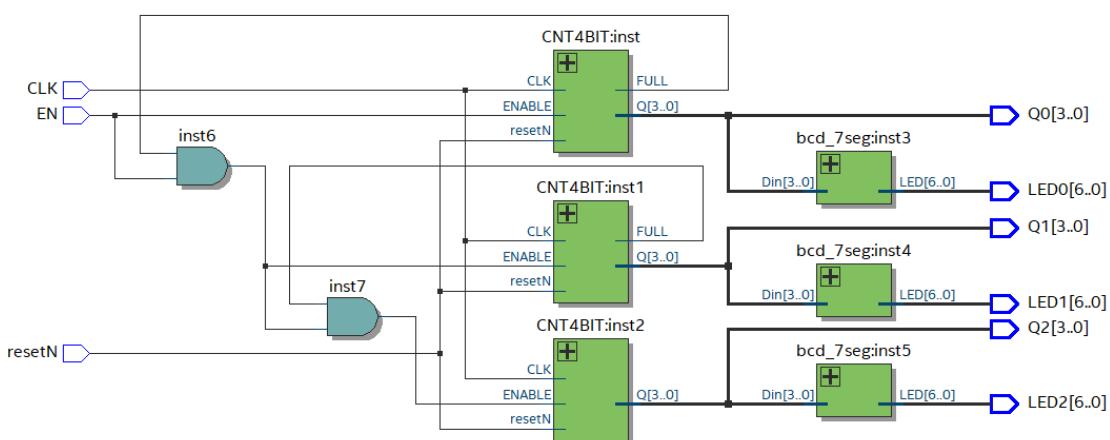


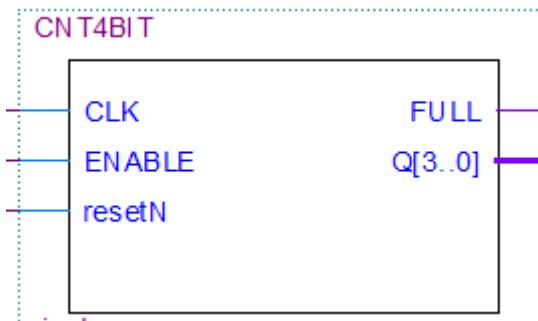
סימולציה

מונה אחדות סופר מ-0 עד 9 כל דופק שעון, מונה עשרות מתקדם כל 10 דופקי שעון, ומונה מאות כל 100 דופקי שעון



RTL Viewer





```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity CNT4BIT is
  port(
    CLK : in STD_LOGIC;
    FULL : out STD_LOGIC;
    ENABLE : in STD_LOGIC;
    Q : out STD_LOGIC_VECTOR(3 downto 0);
    resetN : in STD_LOGIC
  );
end CNT4BIT;

architecture CNT4BIT of CNT4BIT is
  signal Qsig: STD_LOGIC_VECTOR(3 downto 0);

begin
  process (CLK, resetN)
  begin
    if resetN = '0' then
      Qsig <= (others => '0');
    elsif CLK='1' and CLK'event then
      if ENABLE = '1' then
        if Qsig = 9 then
          Qsig <= (others => '0');
        else
          Qsig <= Qsig + 1;
        end if;
      end if;
    end if;
  end process;

  Q <= Qsig;

  FULL <= '1' when (Qsig = 9) else '0';

end CNT4BIT;

```