|  |
| --- |
| clk\_div.vhd  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity clk\_div is  Port ( clk\_50 : in STD\_LOGIC; --50Mhz giriş clk  reset: in std\_logic; --reset  clk\_out : out STD\_LOGIC --çıkış clk  );  end clk\_div;  architecture Behavioral of clk\_div is  signal control: std\_logic:='0';  signal counter: integer range 0 to 5 :=0;  begin    process (clk\_50,reset)  begin  if (reset='1') then  counter <= 0;  control <= '0';  elsif (rising\_edge(clk\_50)) then  if (counter=5) then  control <= not(control);  counter <= 0;  else  counter <=counter + 1;  end if;  end if;  end process;  clk\_out <= control;  end Behavioral; |
| counter.vhd  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity counter is  Port ( enable : in STD\_LOGIC;  updown : in STD\_LOGIC;  reset : in STD\_LOGIC;  clk : in STD\_LOGIC;  bcd1 : out STD\_LOGIC\_VECTOR (3 downto 0);  bcd10 : out STD\_LOGIC\_VECTOR (3 downto 0)  );  end counter;  architecture Behavioral of counter is  signal count:std\_logic\_vector(7 downto 0):="00000000";  begin    process(clk,reset)  begin  if (reset='1') then  count<="00000000";  elsif(rising\_edge(clk)) then  if(enable='1')then  if(updown='1')then  count<=count+"00000001";  else  count<=count-"00000001";  end if;  end if;  end if;  end process;    bcd1<=conv\_std\_logic\_vector((conv\_integer(count)/10),4);  bcd10<=conv\_std\_logic\_vector((conv\_integer(count) mod 10),4);  end Behavioral; |
| decoder7segment.vhd  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity decoder7segment is  port ( bcd : in std\_logic\_vector(3 downto 0);  segment7 : out std\_logic\_vector(6 downto 0) );  end decoder7segment;  architecture Behavioral of decoder7segment is  begin    process (bcd)  BEGIN  case bcd is  when "0000"=> segment7 <="0000001"; -- '0'  when "0001"=> segment7 <="1001111"; -- '1'  when "0010"=> segment7 <="0010010"; -- '2'  when "0011"=> segment7 <="0000110"; -- '3'  when "0100"=> segment7 <="1001100"; -- '4'  when "0101"=> segment7 <="0100100"; -- '5'  when "0110"=> segment7 <="0100000"; -- '6'  when "0111"=> segment7 <="0001111"; -- '7'  when "1000"=> segment7 <="0000000"; -- '8'  when "1001"=> segment7 <="0000100"; -- '9'    when "1010"=> segment7 <="0001000"; -- 'A'  when "1011"=> segment7 <="1100000"; -- 'B'  when "1100"=> segment7 <="0110001"; -- 'C'  when "1101"=> segment7 <="1000010"; -- 'D'  when "1110"=> segment7 <="0110000"; -- 'E'  when "1111"=> segment7 <="0111000"; -- 'F'    when others=> segment7 <="1111111";  end case;  end process;  end Behavioral; |
| counter\_decimal.vhd  library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity counter\_decimal is  port ( enable : in STD\_LOGIC;  updown : in STD\_LOGIC;  reset : in STD\_LOGIC;  clk : in STD\_LOGIC;  segment1 : out STD\_LOGIC\_VECTOR (6 downto 0);  segment10 : out STD\_LOGIC\_VECTOR (6 downto 0) );  end counter\_decimal;  architecture Behavioral of counter\_decimal is  component clk\_div  Port ( clk\_50 : in STD\_LOGIC;  reset: in std\_logic;  clk\_out : out STD\_LOGIC );  end component;  component counter  Port ( enable : in STD\_LOGIC;  updown : in STD\_LOGIC;  reset : in STD\_LOGIC;  clk : in STD\_LOGIC;  bcd1 : out STD\_LOGIC\_VECTOR (3 downto 0);  bcd10 : out STD\_LOGIC\_VECTOR (3 downto 0) );  end component;  component decoder7segment  port ( bcd : in std\_logic\_vector(3 downto 0);  segment7 : out std\_logic\_vector(6 downto 0) );  end component;  signal ara1,ara10:std\_logic\_vector(3 downto 0);  signal clk\_ara:std\_logic;  begin  inst\_clk\_div: clk\_div  Port map (clk\_50=>clk,  reset=>reset,  clk\_out=>clk\_ara);  inst\_counter: counter  Port map(enable=>enable,  updown=>updown,  reset=>reset,  clk=>clk\_ara,  bcd1=>ara1,  bcd10=>ara10 );  inst\_decoder1: decoder7segment  port map (bcd=>ara1,  segment7=>segment1);  inst\_decoder10: decoder7segment  port map (bcd=>ara10,  segment7=>segment10 );  end Behavioral; |