```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity encoder is
    Port (
          hex_in : in STD_LOGIC_VECTOR(3 DOWNTO 0);
          a : out STD_LOGIC;
          b : out STD_LOGIC;
          c : out STD_LOGIC;
          d : out STD_LOGIC;
          e : out STD_LOGIC;
          f : out STD_LOGIC;
          g : out STD_LOGIC
    );
end encoder;

architecture Behavioral of encoder is

    -- temporary signal to make our assignment of values (a through g) simpler
    signal seven_seg    : std_logic_vector(6 downto 0);

begin

    --seven_seg <= -- Enter your code to assign values for the entire 7 bits (a through
    g) based on the hex input
    process(hex_in)
    begin
        case hex_in is
            when "0000" => seven_seg <= "1111110";
            when "0001" => seven_seg <= "0110000";
            when "0010" => seven_seg <= "1101101";
            when "0011" => seven_seg <= "1111001";

            when "0100" => seven_seg <= "0110011";
            when "0101" => seven_seg <= "1011011";
            when "0110" => seven_seg <= "1011111";
            when "0111" => seven_seg <= "1110000";

            when "1000" => seven_seg <= "1111111";
            when "1001" => seven_seg <= "1111011";
            when "1010" => seven_seg <= "1110111";
            when "1011" => seven_seg <= "0011111";

            when "1100" => seven_seg <= "1001110";
            when "1101" => seven_seg <= "0111101";
            when "1110" => seven_seg <= "1001111";
            when "1111" => seven_seg <= "1000111";

            when others => seven_seg <= "0000000";

        end case;
    end process;
    -- Extract each individual bit and assign it to the 7 outputs
    a <= seven_seg(6);
    b <= seven_seg(5);
    c <= seven_seg(4);
    d <= seven_seg(3);
    e <= seven_seg(2);
    f <= seven_seg(1);
    g <= seven_seg(0);

end Behavioral;

```