1.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY ASU IS
   PORT (Cin    : IN STD_LOGIC;
      X, Y    : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      S    : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
      Cout, Overflow, Sign : OUT STD_LOGIC);
END ASU;

ARCHITECTURE Behavior OF ASU IS
   SIGNAL Sum : STD_LOGIC_VECTOR(4 DOWNTO 0);
    SIGNAL Yp : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
   Yp(3) <= Y(3) XOR Cin;
    Yp(2) <= Y(2) XOR Cin;
    Yp(1) <= Y(1) XOR Cin;
    Yp(0) <= Y(0) XOR Cin;
   Sum <= ('0' & X) + ('0' & Y) + Cin;
   S <= Sum(3 DOWNTO 0);
   Cout <= Sum(4);
   Overflow <=  Sum(4) XOR X(3) XOR Y(3) XOR Sum(3);
    Sign <= Sum(3);
END BEHAVIOR;
```

2 + 3

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;


ENTITY sseg IS
PORT(bcd   : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
leds: OUT STD_LOGIC_VECTOR(1 TO 7);
ledss: OUT STD_LOGIC_VECTOR(1 TO 7);
Sign: IN STD_LOGIC);
END sseg;

ARCHITECTURE Behavior OF sseg IS
BEGIN
PROCESS(bcd, Sign)
```

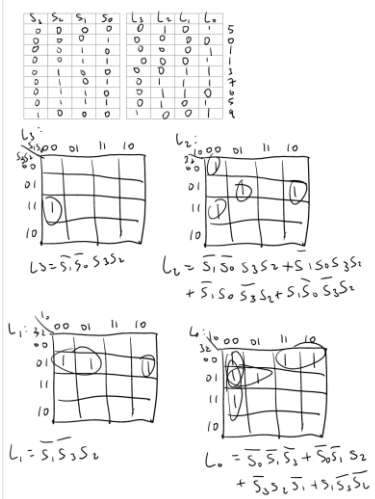
```
BEGIN

    CASE bcd IS
      WHEN "0000" => leds <= "1111110";
      WHEN "0001" => leds <= "0110000";
      WHEN "0010" => leds <= "1101101";
      WHEN "0011" => leds <= "1111001";
      WHEN "0100" => leds <= "0110011";
      WHEN "0101" => leds <= "1011011";
      WHEN "0110" => leds <= "1011111";
      WHEN "0111" => leds <= "1110000";
      WHEN "1000" => leds <= "1111111";
      WHEN "1001" => leds <= "1110011";
      WHEN OTHERS => leds <= "-------";
    END CASE;

IF(Sign = '1') then
ledss <= "0000001";
                                    else
ledss <= "0000000";
      END IF;

  END PROCESS;
END BEHAVIOR;
```

4.



5.

```
library ieee;
USE ieee.std_logic_1164.all;

ENTITY C IS
PORT(
    S : IN STD_logic_vector(3 DOWNTO 0);
L: OUT STD_logic_vector(3 DOWNTO 0)
     );
END C;

ARCHITECTURE Behavior OF C IS
BEGIN
L(3) ≤ (NOT S(1) AND NOT S(0) AND S(3) AND S(2));
L(2) <= ((NOT S(1) AND NOT S(0) AND S(3) AND S(2))+(NOT S(1) AND S(0) AND S(3)
AND S(2)));
L(1) ≤ (NOT S(1) AND NOT S(3) AND S(2));
L(0) <= ((NOT S(0) AND NOT S(1) AND NOT S(3))+(NOT S(0) AND NOT S(1) AND
S(2))+(NOT S(3) AND S(2) AND NOT S(1))+(S(1) AND NOT S(3) AND NOT (2) ));
END Behavior;
```