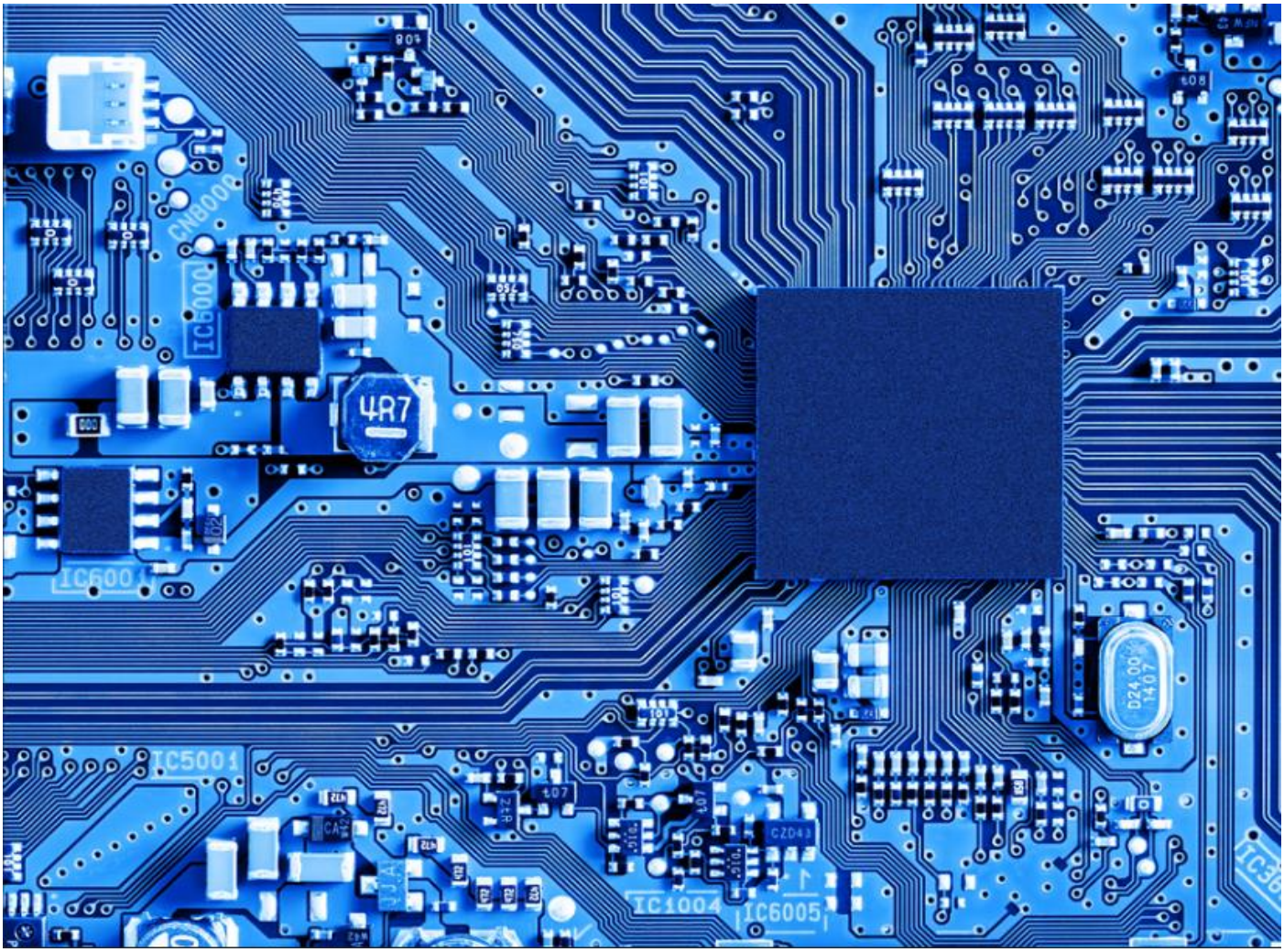
ITI

Lab ONE

VHDL



Waleed Emad Yahyaa Zakarya

```vhdl
LIBRARY ieee;
LIBRARY std;
USE ieee.numeric_bit.All;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.all;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE std.textio.all;

ENTITY adder_subtractor IS
  PORT(
    A, B: IN  std_logic_vector (3 DOWNTO 0);
    M:    IN  std_logic;
    S:    OUT std_logic_vector (3 DOWNTO 0);
    C:    OUT std_logic
  );
END adder_subtractor;

ARCHITECTURE behavioral OF adder_subtractor IS
BEGIN
P1:  PROCESS(A, B, M)
    variable TEMP : std_logic_vector (4 DOWNTO 0);
    BEGIN
        if M = '0' then
            TEMP := ('0' & A) + ('0' & B);
        else
            TEMP := ('0' & A) - ('0' & B);
        end if;
        S <= TEMP(3 DOWNTO 0);
        C <= TEMP(4);
    END PROCESS P1;
END ARCHITECTURE behavioral;
```

```
LIBRARY ieee;
LIBRARY std;
USE ieee.numeric_bit.All;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.all;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE std.textio.all;

ENTITY adder_subtractor IS
  PORT(
    A, B: IN  std_logic_vector (3 DOWNTO 0);
    M:    IN  std_logic;
    S:    OUT std_logic_vector (3 DOWNTO 0);
    C:    OUT std_logic
  );
END adder_subtractor;

ARCHITECTURE data_flow OF adder_subtractor IS
SIGNAL BXORED : std_logic_vector (3 DOWNTO 0);
SIGNAL MBAR : std_logic;
BEGIN
MBAR <= NOT M;
BXORED(3 DOWNTO 0) <= (B(3 DOWNTO 0) xor (MBAR & MBAR & MBAR & MBAR))+ MBAR;
S <= A+B when M = '0' else A-B;
C <= '0' when A>BXORED else '1';

END ARCHITECTURE data_flow;
```

```
LIBRARY std;
LIBRARY ieee;
USE ieee.numeric_bit.All;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.all;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE std.textio.all;

ENTITY adder_subtractor_tb is
end ENTITY adder_subtractor_tb;

ARCHITECTURE tb of adder_subtractor_tb is

component adder_subtractor IS
  PORT(
    A, B: IN  std_logic_vector (3 DOWNTO 0);
    M:    IN  std_logic;
    S:    OUT std_logic_vector (3 DOWNTO 0);
    C:    OUT std_logic
  );
END component;

FOR UT :adder_subtractor USE ENTITY WORK.adder_subtractor (behavioral);

    signal A, B:   std_logic_vector (3 DOWNTO 0);
    signal M:      std_logic;
    signal S:     std_logic_vector (3 DOWNTO 0);
    signal C:     std_logic;

begin
    UT : adder_subtractor
    port map (A,B,M,S,C);

    p2 : PROCESS
    file in_file : text open read_mode is "D:\in.txt";
    file out_file : text open write_mode is "D:\out.txt";
    variable input_line : line;
    variable output_line : line ;
    variable A_f, B_f:   std_logic_vector (3 DOWNTO 0);
    variable M_f:      std_logic;
    variable S_f:     std_logic_vector (3 DOWNTO 0);
    variable C_f:     std_logic;

    begin
        while (not endfile(in_file)) loop
            wait for 5 ns;
            readline(in_file , input_line);
            read(input_line , A_f);
            read(input_line , B_f);
            read(input_line , M_f);

            A <= A_f;
            B <= B_f;
```

```
            M <= M_f;

            wait for 15 ns ;

            S_f := S;
            C_f := C;


            write (output_line , string'("time = "));
            write (output_line , now);
            write (output_line , string'("  A = "));
            write (output_line ,A_f);
            write (output_line ,string'("   B = "));
            write (output_line ,B_f);
            write (output_line ,string'("   M = "));
            write (output_line ,M_f);
            write (output_line , string'("  carry_out = "));
            write (output_line , C_f);
            write (output_line , string'("  sum = "));
            write (output_line ,S_f);
            writeline (out_file , output_line);
            wait for 5 ns;

            end loop;

            wait;
    end PROCESS p2;
end ARCHITECTURE tb;
```

```
LIBRARY ieee;
LIBRARY std;
USE ieee.numeric_bit.All;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.all;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE std.textio.all;

ENTITY full_adder IS
  PORT(
    A,B,Cin:    IN  std_logic;
    S,Cout:    OUT std_logic
  );
END full_adder;

ARCHITECTURE full_adder OF full_adder IS
BEGIN
S <= A xor B xor Cin;
Cout <= (A and B) or (Cin and (A xor B));
END ARCHITECTURE full_adder;
```

```
LIBRARY ieee;
LIBRARY std;
USE ieee.numeric_bit.All;
USE ieee.std_logic_arith.all;
USE ieee.numeric_std.all;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.ALL;
USE std.textio.all;

ENTITY adder_subtractor IS
  PORT(
    A, B: IN  std_logic_vector (3 DOWNTO 0);
    M:    IN  std_logic;
    S:    OUT std_logic_vector (3 DOWNTO 0);
    C:    OUT std_logic
  );
END adder_subtractor;

ARCHITECTURE arch OF adder_subtractor IS

component full_adder IS
  PORT(
    A,B,Cin:    IN  std_logic;
    S,Cout:    OUT std_logic
  );
END component full_adder;

signal C0 , C1 , C2 : std_logic;
signal BXORED : std_logic_vector (3 DOWNTO 0);
BEGIN
    BXORED(3 DOWNTO 0) <= B(3 DOWNTO 0) xor (M & M & M & M);
    FULLADD0:full_adder PORT MAP (A=>A(0),B => BXORED(0),Cin => M  , S => S(0) , Cout =>
C0);
    FULLADD1:full_adder PORT MAP (A=>A(1),B => BXORED(1),Cin => C0 , S => S(1) , Cout =>
C1);
    FULLADD2:full_adder PORT MAP (A=>A(2),B => BXORED(2),Cin => C1 , S => S(2) , Cout =>
C2);
    FULLADD3:full_adder PORT MAP (A=>A(3),B => BXORED(3),Cin => C2 , S => S(3) , Cout => C
);

END ARCHITECTURE arch;
```

time = 20 ns A = 0000 B = 0000 M = 0 carry_out = U sum = 0000

time = 45 ns A = 0000 B = 0000 M = 1 carry_out = U sum = 0000

time = 70 ns A = 0000 B = 0001 M = 0 carry_out = U sum = 0001

time = 95 ns A = 0000 B = 0001 M = 1 carry_out = U sum = 1111

time = 120 ns A = 1010 B = 0101 M = 0 carry_out = U sum = 1111

time = 145 ns A = 1111 B = 0000 M = 1 carry_out = U sum = 1111

time = 170 ns A = 0000 B = 0001 M = 0 carry_out = U sum = 0001

time = 195 ns A = 0000 B = 0001 M = 1 carry_out = U sum = 1111

time = 220 ns A = 1001 B = 1100 M = 0 carry_out = U sum = 0101

time = 245 ns A = 0000 B = 1111 M = 1 carry_out = U sum = 0001

time = 270 ns A = 0000 B = 0001 M = 0 carry_out = U sum = 0001

time = 295 ns A = 1011 B = 0001 M = 1 carry_out = U sum = 1010

time = 320 ns A = 0101 B = 1111 M = 0 carry_out = U sum = 0100

time = 345 ns A = 0000 B = 0000 M = 1 carry_out = U sum = 0000

time = 370 ns A = 1001 B = 0001 M = 0 carry_out = U sum = 1010

time = 395 ns A = 1101 B = 0011 M = 1 carry_out = U sum = 1010