library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FA is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Co : out STD\_LOGIC);

end FA;

architecture Behavioral of FA is

begin

S <= A xor B xor C;

Co <= (A and B) OR (B AND C) OR (C AND A);

end Behavioral;

4\_BIT\_ADDER

----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 12:40:59 11/20/2023

-- Design Name:

-- Module Name: FourBitAdder - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FourBitAdder is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Co : out STD\_LOGIC);

end FourBitAdder;

architecture Behavioral of FourBitAdder is

COMPONENT FA is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Co : out STD\_LOGIC);

end COMPONENT;

signal C1 : std\_logic\_vector(2 DOWNTO 0);

begin

FA1 : FA PORT MAP (A(0), B(0), '0', S(0), C1(0));

FA2 : FA PORT MAP (A(1), B(1), C1(0), S(1), C1(1));

FA3 : FA PORT MAP (A(2), B(2), C1(1), S(2), C1(2));

FA4 : FA PORT MAP (A(3), B(3), C1(2), S(3), Co);

end Behavioral;

FS

----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 12:49:34 11/20/2023

-- Design Name:

-- Module Name: FS1Bit - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FS1Bit is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Bi : in STD\_LOGIC;

D : out STD\_LOGIC;

Bo : out STD\_LOGIC);

end FS1Bit;

architecture Behavioral of FS1Bit is

begin

D <= A XOR B XOR Bi;

Bo <= (NOT A AND Bi) OR (NOT A AND B) OR (B AND Bi);

end Behavioral;

4\_BIT\_SUB

----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 12:54:06 11/20/2023

-- Design Name:

-- Module Name: FourBitSubtractor - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FourBitSubtractor is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

D : out STD\_LOGIC\_VECTOR (3 downto 0);

Bo : out STD\_LOGIC);

end FourBitSubtractor;

architecture Behavioral of FourBitSubtractor is

COMPONENT FS1Bit is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Bi : in STD\_LOGIC;

D : out STD\_LOGIC;

Bo : out STD\_LOGIC);

end COMPONENT;

SIGNAL B1 : STD\_LOGIC\_VECTOR(2 DOWNTO 0);

SIGNAL D1 : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

begin

FS1 : FS1Bit PORT MAP (A(0), B(0), '0', D1(0), B1(0));

FS2 : FS1Bit PORT MAP (A(1), B(1), B1(0), D1(1), B1(1));

FS3 : FS1Bit PORT MAP (A(2), B(2), B1(1), D1(2), B1(2));

FS4 : FS1Bit PORT MAP (A(3), B(3), B1(2), D1(3), Bo);

D<= D1;

end Behavioral;

**ALU**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ALU is

Port ( OP : in STD\_LOGIC\_VECTOR (3 downto 0);

A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

CLK : in STD\_LOGIC;

RST : in STD\_LOGIC;

Co : out STD\_LOGIC;

Bo : out STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR (3 downto 0));

end ALU;

architecture Behavioral of ALU is

COMPONENT FourBitAdder is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Co : out STD\_LOGIC);

end COMPONENT;

COMPONENT FourBitSubtractor is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

D : out STD\_LOGIC\_VECTOR (3 downto 0);

Bo : out STD\_LOGIC);

end COMPONENT;

SIGNAL ADD, SUB,TEMP , COMP\_1: STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL CARRY , BORROW: STD\_LOGIC := '0';

begin

ADD1 : FourBitAdder PORT MAP(A,B, ADD, CARRY);

SUB1 : FourBitSubtractor PORT MAP ( A, B, SUB, BORROW);

PROCESS(OP, ADD, SUB, RST, A, B)

BEGIN

COMP\_1 <= (NOT A(3) ) & (NOT A(2) ) & (NOT A(1) ) & (NOT A(0) );

CASE RST IS

WHEN '1' =>

TEMP <= "0000";

WHEN '0' =>

Bo <= '0';

Co <= '0';

CASE OP IS

WHEN "0000" =>

TEMP <= "0000";

WHEN "0001" =>

TEMP <= ADD;

Co <= CARRY;

WHEN "0010" =>

TEMP <= SUB;

Bo <= BORROW;

WHEN "0011" =>

TEMP <= (A(3) AND B(3))&(A(2) AND B(2))&(A(1) AND B(1))&(A(0) AND B(0));

WHEN "0100" =>

TEMP <= (A(3) XOR B(3))&(A(2) XOR B(2))&(A(1) XOR B(1))&(A(0) XOR B(0));

WHEN "0101" =>

TEMP <= (A(3) OR B(3))&(A(2) OR B(2))&(A(1) OR B(1))&(A(0) OR B(0));

WHEN "0110" =>

TEMP <= COMP\_1 + '1';

WHEN "0111" =>

TEMP <= COMP\_1;

WHEN "1000" =>

CASE B IS

WHEN "0001" =>

TEMP <= A(2 DOWNTO 0)&'0';

WHEN "0010" =>

TEMP <= A(1 DOWNTO 0)&"00";

WHEN "0011" =>

TEMP <= A(0)&"000";

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

WHEN "1001" =>

CASE B IS

WHEN "0001" =>

TEMP <= '0' & A(3 DOWNTO 1);

WHEN "0010" =>

TEMP <= "00" & A(3 DOWNTO 2);

WHEN "0011" => TEMP <= "000" & A(3);

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

WHEN "1010" =>

CASE B IS

WHEN "0001" => TEMP <= A(2 DOWNTO 0)& A(3);

WHEN "0010" =>

TEMP <= A(1 DOWNTO 0)& A(3 DOWNTO 2);

WHEN "0011" =>

TEMP <= A(0)&A(3 DOWNTO 1);

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

WHEN "1011" =>

CASE B IS

WHEN "0001" =>

TEMP <= A(0) & A(3 DOWNTO 1);

WHEN "0010" =>

TEMP <= A(1 DOWNTO 0) & A(3 DOWNTO 2);

WHEN "0011" =>

TEMP <= A(2 DOWNTO 0)& A(3);

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

WHEN OTHERS =>

TEMP <= (OTHERS =>'0');

END CASE;

END PROCESS;

Q<= TEMP;

end Behavioral;

TEST BENCH

--------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 17:28:55 11/20/2023

-- Design Name:

-- Module Name: /home/ise/mini\_project/ALU\_Design/TEST\_ALU.vhd

-- Project Name: ALU\_Design

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: ALU

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

--------------------------------------------------------------------------------

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;

ENTITY TEST\_ALU IS

END TEST\_ALU;

ARCHITECTURE behavior OF TEST\_ALU IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT ALU

PORT(

OP : IN std\_logic\_vector(3 downto 0);

A : IN std\_logic\_vector(3 downto 0);

B : IN std\_logic\_vector(3 downto 0);

CLK : IN std\_logic;

RST : IN std\_logic;

Co : OUT std\_logic;

Bo : OUT std\_logic;

Q : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal OP : std\_logic\_vector(3 downto 0) := (others => '0');

signal A : std\_logic\_vector(3 downto 0) := (others => '0');

signal B : std\_logic\_vector(3 downto 0) := (others => '0');

signal CLK : std\_logic := '0';

signal RST : std\_logic := '0';

--Outputs

signal Co : std\_logic;

signal Bo : std\_logic;

signal Q : std\_logic\_vector(3 downto 0);

-- Clock period definitions

constant CLK\_period : time := 100 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ALU PORT MAP (

OP => OP,

A => A,

B => B,

CLK => CLK,

RST => RST,

Co => Co,

Bo => Bo,

Q => Q

);

-- Clock process definitions

CLK\_process :process

begin

CLK <= '0';

wait for CLK\_period/2;

CLK <= '1';

wait for CLK\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

RST <= '1';

wait for 100 ns;

RST <='0';

A<="1100";

B <="0011";

OP<= "0000";

WAIT FOR 100 NS;

OP<= "0001";

WAIT FOR 100 NS;

OP<= "0010";

WAIT FOR 100 NS;

OP<= "0011";

WAIT FOR 100 NS;

OP<= "0100";

WAIT FOR 100 NS;

OP<= "0101";

WAIT FOR 100 NS;

OP<= "0110";

WAIT FOR 100 NS;

OP<= "0111";

WAIT FOR 100 NS;

OP<= "1000";

WAIT FOR 100 NS;

OP<= "1001";

WAIT FOR 100 NS;

OP<= "1010";

WAIT FOR 100 NS;

OP<= "1011";

WAIT FOR 100 NS;

OP<= "1100";

WAIT FOR 100 NS;

OP<= "1101";

WAIT FOR 100 NS;

OP<= "1110";

WAIT FOR 100 NS;

OP<= "1111";

WAIT FOR 100 NS;

OP<= "0000";

WAIT FOR 100 NS;

A<="0100";

B <="0101";

OP<= "0001";

WAIT FOR 100 NS;

OP<= "0010";

WAIT FOR 100 NS;

A<="1101";

B <="0101";

OP<= "0001";

WAIT FOR 100 NS;

OP<= "0010";

WAIT FOR 100 NS;

WAIT FOR 100 NS;

end process;

END;