# ISLAMIC UNIVERSITY OF TECHNOLOGY

# Organization of Islamic Cooperation

# Board Bazar, Gazipur

# Lab 05

# EEE 4484

# Safayat Bin Hakim

**Full-Adder Circuit**

Source Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
  
ENTITY full\_adder IS  
 PORT (  
 Num1, Num2, CarryIn : IN std\_logic;  
 Sum, CarryOut : OUT std\_logic  
 );  
END full\_adder;  
  
ARCHITECTURE behavorial OF full\_adder IS  
BEGIN  
 Sum <= Num1 XOR Num2 XOR CarryIn;  
 CarryOut <= (Num1 AND Num2) OR (CarryIn AND Num1) OR (CarryIn AND Num2);  
END behavorial;

VHDL

Testbench Code:

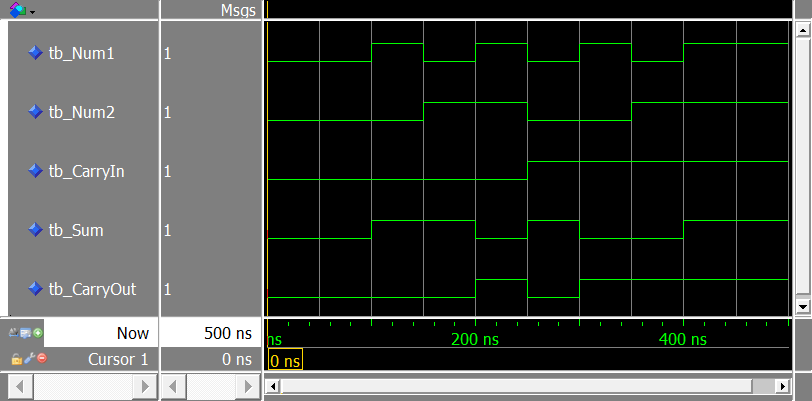
LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
  
ENTITY tb\_full\_adder IS  
END tb\_full\_adder;  
  
ARCHITECTURE behavior OF tb\_full\_adder IS  
 COMPONENT full\_adder  
 PORT (  
 Num1, Num2, CarryIn : IN std\_logic;  
 Sum, CarryOut : OUT std\_logic  
 );  
 END COMPONENT;  
  
 SIGNAL tb\_Num1, tb\_Num2, tb\_CarryIn : std\_logic := '0';  
 SIGNAL tb\_Sum, tb\_CarryOut : std\_logic;

BEGIN  
 uut : full\_adder  
 PORT MAP(  
 Num1 => tb\_Num1,   
 Num2 => tb\_Num2,   
 CarryIn => tb\_CarryIn,   
 Sum => tb\_Sum,   
 CarryOut => tb\_CarryOut  
 );  
  
 stim\_proc : PROCESS  
 BEGIN  
 WAIT FOR 100 ns;  
  
 tb\_Num1 <= '1';  
 tb\_Num2 <= '0';  
 tb\_CarryIn <= '0';  
 WAIT FOR 50 ns;  
  
 tb\_Num1 <= '0';  
 tb\_Num2 <= '1';  
 tb\_CarryIn <= '0';  
 WAIT FOR 50 ns;  
  
 tb\_Num1 <= '1';  
 tb\_Num2 <= '1';  
 tb\_CarryIn <= '0';  
 WAIT FOR 50 ns;  
  
 tb\_Num1 <= '0';  
 tb\_Num2 <= '0';  
 tb\_CarryIn <= '1';  
 WAIT FOR 50 ns;  
  
 tb\_Num1 <= '1';  
 tb\_Num2 <= '0';  
 tb\_CarryIn <= '1';  
 WAIT FOR 50 ns;  
  
 tb\_Num1 <= '0';  
 tb\_Num2 <= '1';  
 tb\_CarryIn <= '1';  
 WAIT FOR 50 ns;

tb\_Num1 <= '1';  
 tb\_Num2 <= '1';  
 tb\_CarryIn <= '1';  
 WAIT FOR 50 ns;  
  
 END PROCESS;  
END;

VHDL

Waveshape:



**1-to-8 Demultiplexer**

Source Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
  
ENTITY demux\_1\_8 IS  
 PORT (  
 input : IN std\_logic;  
 sel : IN std\_logic\_vector(2 DOWNTO 0);  
 output : OUT std\_logic\_vector(7 DOWNTO 0)  
 );  
END ENTITY;  
  
ARCHITECTURE behavioral OF demux\_1\_8 IS  
  
BEGIN  
 PROCESS (input, sel)  
 BEGIN  
 output <= "00000000";  
   
 CASE sel IS  
 WHEN "000" => output(0) <= input;  
 WHEN "001" => output(1) <= input;  
 WHEN "010" => output(2) <= input;  
 WHEN "011" => output(3) <= input;  
 WHEN "100" => output(4) <= input;  
 WHEN "101" => output(5) <= input;  
 WHEN "110" => output(6) <= input;  
 WHEN "111" => output(7) <= input;  
 WHEN OTHERS => output <= "00000000";  
 END CASE;  
 END PROCESS;  
END ARCHITECTURE behavioral;

VHDL

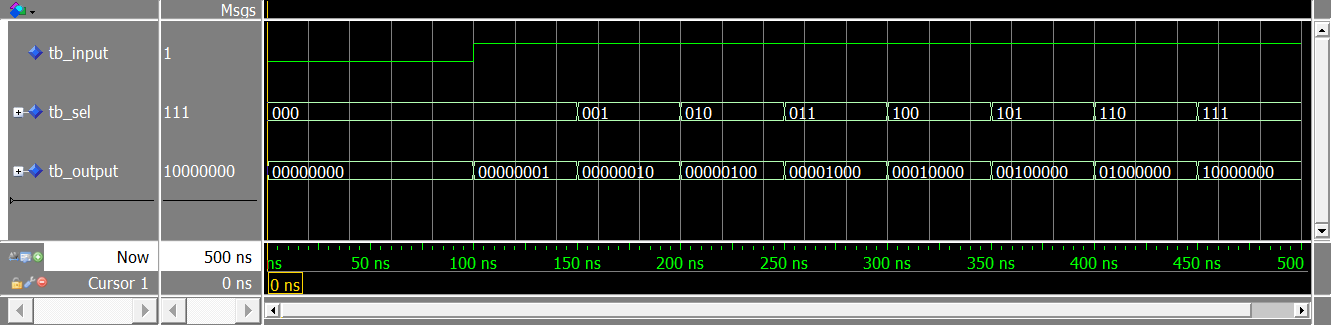
Testbench Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
  
ENTITY tb\_demux\_1\_8 IS  
END tb\_demux\_1\_8;  
  
ARCHITECTURE behavior OF tb\_demux\_1\_8 IS  
  
 COMPONENT demux\_1\_8  
 PORT (  
 input : IN std\_logic;  
 sel : IN std\_logic\_vector(2 DOWNTO 0);  
 output : OUT std\_logic\_vector(7 DOWNTO 0)  
 );  
 END COMPONENT;  
  
 SIGNAL tb\_input : std\_logic := '0';  
 SIGNAL tb\_sel : std\_logic\_vector (2 DOWNTO 0) := (OTHERS => '0');  
 SIGNAL tb\_output : std\_logic\_vector (7 DOWNTO 0);  
  
BEGIN  
 uut : demux\_1\_8  
 PORT MAP(  
 input => tb\_input,   
 sel => tb\_sel,   
 output => tb\_output  
 );  
  
 stim\_proc : PROCESS  
 BEGIN  
 WAIT FOR 100 ns;  
  
 tb\_input <= '1';  
 tb\_sel <= "000";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "001";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "010";  
 WAIT FOR 50 ns;

tb\_sel <= "011";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "100";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "101";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "110";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "111";  
 WAIT FOR 50 ns;  
   
 WAIT;  
 END PROCESS;  
END;

VHDL

Waveshape:



**4-Bit ALU**

Source Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
USE ieee.numeric\_std.ALL;  
  
ENTITY alu\_4\_bit IS  
 PORT (  
 num1, num2 : IN signed(3 DOWNTO 0);  
 sel : IN std\_logic\_vector (1 DOWNTO 0);  
 result : OUT signed(3 DOWNTO 0)  
 );  
END alu\_4\_bit;  
  
ARCHITECTURE behavioral OF alu\_4\_bit IS  
BEGIN  
 PROCESS (num1, num2, sel)  
 BEGIN  
   
 CASE sel IS  
 WHEN "00" => result <= num1 + num2;  
 WHEN "01" => result <= num1 - num2;  
 WHEN "10" => result <= num1 AND num2;  
 WHEN "11" => result <= num1 OR num2;  
 WHEN OTHERS => NULL;  
 END CASE;  
   
 END PROCESS;  
  
END behavioral;

VHDL

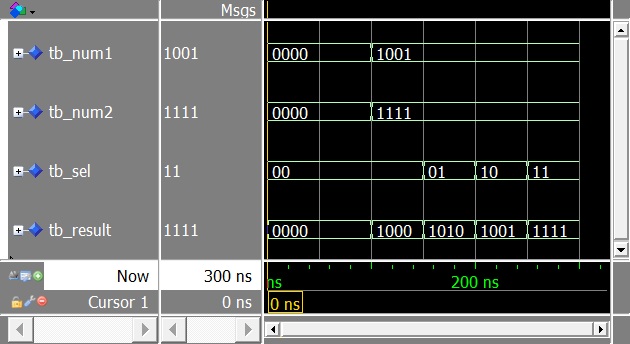
Testbench Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
USE ieee.numeric\_std.ALL;  
   
ENTITY tb\_alu\_4\_bit IS  
END tb\_alu\_4\_bit;  
   
ARCHITECTURE behavioral OF tb\_alu\_4\_bit IS  
   
 COMPONENT alu\_4\_bit  
 PORT (  
 num1, num2 : IN signed(3 DOWNTO 0);  
 sel : IN std\_logic\_vector(1 DOWNTO 0);  
 result: OUT signed(3 DOWNTO 0)  
 );  
 END COMPONENT;  
  
 SIGNAL tb\_num1, tb\_num2 : signed(3 DOWNTO 0) := (OTHERS => '0');  
 SIGNAL tb\_sel : std\_logic\_vector(1 DOWNTO 0) := (OTHERS => '0');  
 SIGNAL tb\_result : signed(3 DOWNTO 0);  
   
BEGIN  
 uut : alu\_4\_bit  
 PORT MAP(  
 num1 => tb\_num1,   
 num2 => tb\_num2,   
 sel => tb\_sel,   
 result => tb\_result  
 );  
  
 stim\_proc : PROCESS  
 BEGIN  
 WAIT FOR 100 ns;  
  
 tb\_num1 <= "1001";  
 tb\_num2 <= "1111";  
 tb\_sel <= "00";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "01";  
 WAIT FOR 50 ns;

tb\_sel <= "10";  
 WAIT FOR 50 ns;  
  
 tb\_sel <= "11";  
 END PROCESS;  
END behavioral;

VHDL

Waveshape:



**Synchronous 5-Bit Shift Register**

Source Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
  
ENTITY shift\_reg\_5\_bit IS  
 PORT (  
 input : IN STD\_LOGIC\_VECTOR (4 DOWNTO 0);  
 clock : IN STD\_LOGIC;  
 output : OUT STD\_LOGIC\_VECTOR (4 DOWNTO 0)  
 );  
END ENTITY shift\_reg\_5\_bit;  
  
ARCHITECTURE behavioral OF shift\_reg\_5\_bit IS  
BEGIN  
 PROCESS (clock, input)  
 BEGIN  
 IF (clock'EVENT AND clock = '1') THEN  
 output <= input;  
 END IF;  
 END PROCESS;  
END behavioral;

VHDL

Testbench Code:

LIBRARY ieee;  
USE ieee.std\_logic\_1164.ALL;  
USE ieee.numeric\_std.ALL;  
  
ENTITY tb\_shift\_reg\_5\_bit IS  
END ENTITY tb\_shift\_reg\_5\_bit;

ARCHITECTURE behavioral OF tb\_shift\_reg\_5\_bit IS  
 COMPONENT shift\_reg\_5\_bit  
 PORT (  
 input : IN STD\_LOGIC\_VECTOR (4 DOWNTO 0);  
 clock : IN STD\_LOGIC;  
 output : OUT STD\_LOGIC\_VECTOR (4 DOWNTO 0)  
 );  
 END COMPONENT shift\_reg\_5\_bit;  
   
 SIGNAL tb\_input : STD\_LOGIC\_VECTOR (4 DOWNTO 0) := (OTHERS => '0');  
 SIGNAL tb\_clock : STD\_LOGIC := '0';  
 SIGNAL tb\_output : STD\_LOGIC\_VECTOR (4 DOWNTO 0);  
 CONSTANT clock\_period : TIME := 100 ns;  
  
BEGIN  
 uut : shift\_reg\_5\_bit  
 PORT MAP(  
 input => tb\_input,   
 clock => tb\_clock,   
 output => tb\_output  
 );  
   
 clk\_process : PROCESS  
 BEGIN  
 tb\_clock <= '0';  
 WAIT FOR clock\_period/2;  
 tb\_clock <= '1';  
 WAIT FOR clock\_period/2;  
 END PROCESS;  
   
 stim\_process : PROCESS  
 BEGIN  
 WAIT FOR 100 ns;  
  
 tb\_input <= "00001";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "00010";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "00100";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "01000";  
 WAIT FOR 100 ns;  
  
  
 tb\_input <= "10000";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "10001";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "10010";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "10100";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "11000";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "11010";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "11110";  
 WAIT FOR 100 ns;  
  
 tb\_input <= "11111";  
 END PROCESS;  
END behavioral;

VHDL

Waveshape:

