**LAB 2**

**VHDL modeling Realization of MUX, DMUX using VHDL.**

VHDL code for half adder using structural model.

library ieee;

use ieee.std\_logic\_1164.all;

entity half\_adder is -- Entity declaration for half adder

port (a, b: in std\_logic;

sum, carry\_out: out std\_logic);

end half\_adder;

architecture structure of half\_adder is -- Architecture body for half adder

component xor\_gate -- xor component declaration

port (i1, i2: in std\_logic;

o1: out std\_logic);

end component;

component and\_gate -- and component declaration

port (i1, i2: in std\_logic;

o1: out std\_logic);

end component;

begin

u1: xor\_gate port map (i1 => a, i2 => b, o1 => sum);

u2: and\_gate port map (i1 => a, i2 => b, o1 => carry\_out);

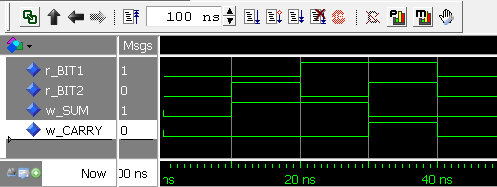
-- We can also use Positional Association

-- => u1: xor\_gate port map (a, b, sum);

-- => u2: and\_gate port map (a, b, carry\_out);

end structure;

simulation diagram.



**Logic gates realization using behavioural model**

**NAND gate**

library ieee;

use ieee.std\_logic\_1164.all;

entity nand2 is

port(

    a, b: in std\_logic;

    c: out std\_logic

);

end nand2;

architecture arch of nand2 is

    begin

    process(a, b)

    begin

    if a='1' and b='1' then

        c <= '0';

    else

c <= '1';

    end if;

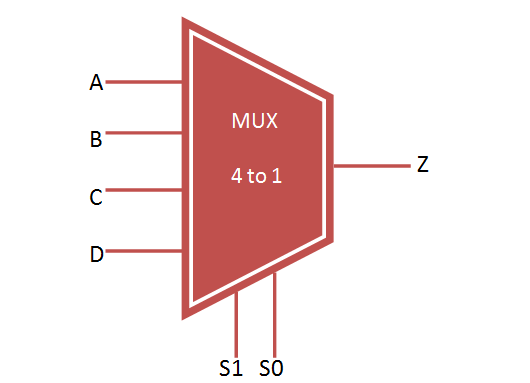
    end process;

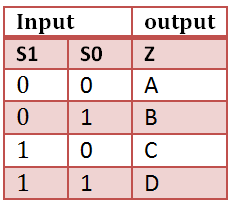
end arch;

**MUX.**

Multiplexer (MUX) select one input from the multiple inputs and forwarded to output line through selection line. It consist of 2 power n input and 1 output. The input data lines are controlled by n selection lines.

For Example, if n = 2 then the mux will be of 4 to 1 mux with 4 input, 2 selection line and 1 output as shown below.



****

VHDL code.

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity MUX4\_1 is

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

s : in STD\_LOGIC\_VECTOR (1 downto 0);

y : out STD\_LOGIC);

end MUX4\_1;

architecture dataflow of MUX4\_1 is

begin

with s select

y <= i(0) when "00",

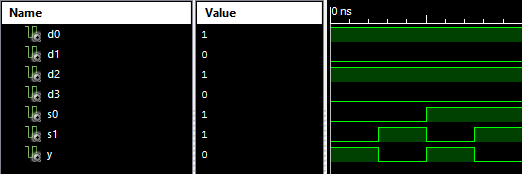
i(1) when "01",

i(2) when "10",

i(3) when others;

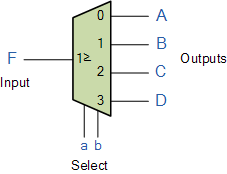
end dataflow;

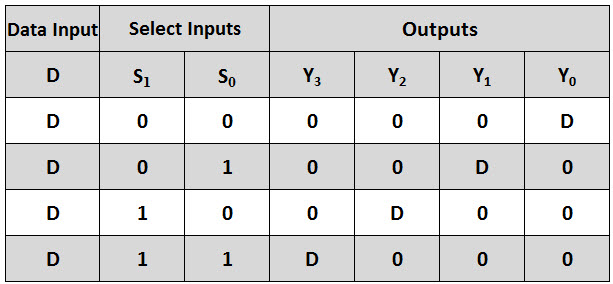
simulation waveform:



Demultiplexer:

Demultiplexer (DEMUX) select one output from the multiple output line and fetch the single input through selection line. It consist of  1 input and 2 power n output. The output data lines are controlled by n selection lines. For Example, if n = 2 then the demux will be of 1 to 4 mux with 1 input, 2 selection line and 4 output as shown below. Also VHDL Code for 1 to 4 Demux described below.





library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity demux\_1to4 is

port(

F : in STD\_LOGIC;

S0,S1: in STD\_LOGIC;

A,B,C,D: out STD\_LOGIC

);

end demux\_1to4;

architecture bhv of demux\_1to4 is

begin

process (F,S0,S1) is

begin

if (S0 ='0' and S1 = '0') then

A <= F;

elsif (S0 ='1' and S1 = '0') then

B <= F;

elsif (S0 ='0' and S1 = '1') then

C <= F;

else

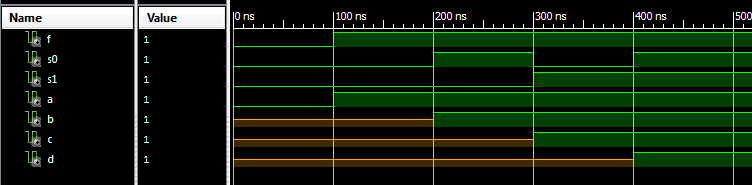
D <= F;

end if;

end process;

end bhv;

simulation waveform:



Discussion and conclusion:

In this lab we studied about structural and behavioural model. We used VHDL code to realize these models and viewed them using simulation environment.