ALU :-

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL; entity ALU is

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

sel : in STD\_LOGIC\_VECTOR (3 downto 0); y : out STD\_LOGIC\_VECTOR(3 downto 0));

end ALU; architecture dataflow of ALU is begin

y<=a when sel = "0000" else a+1 when sel = "0001" else a-1 when sel = "0010" else b

when sel = "0011" else b+1 when sel = "0100" else b-1 when sel = "0101" else a+b when sel = "0110" else a-b when sel = "0111" else NOT a when sel = "1000" else NOT b when sel = "1001" else a AND b when sel = "1010" else a OR b

when sel = "1011" else a NAND b when sel = "1100" else a NOR b

when sel = "1101" else a XOR b when sel = "1110" else a XNOR b when sel = "1111"

; end dataflow;

**ALU test bench** :- library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

entity ALU\_tb is

-- Port ( ); end ALU\_tb;

architecture beh of ALU\_tb is

COMPONENT ALU is

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

sel : in STD\_LOGIC\_VECTOR (3 downto 0); y : out STD\_LOGIC\_VECTOR(3 downto 0)); end COMPONENT;

signal a,b,y: std\_logic\_vector(3 downto 0);

signal sel:std\_logic\_vector(3 downto 0):="0000"; begin

U1: ALU port map(a,b,sel,y); process

begin

a<="1100"; b<="1010";

wait for 100 ns;--sel<="0000"; sel<=sel+1;

--wait for 100 ns; end process;

end beh; Output:-

