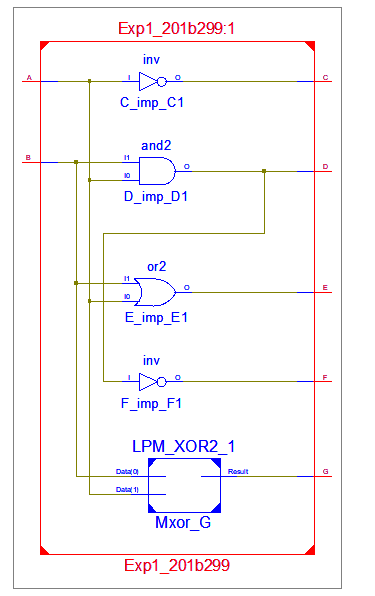
# Experiment 1

**Aim: Design of All-in-One logic gate circuits.**

**Exercise 1:** Design two inputs and five outputs All-in-One logic gate circuit shown in Fig.1. Write the VHDL code in data flow style of modeling.

**Design code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Exp1\_201b299 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC;

D : out STD\_LOGIC;

E : out STD\_LOGIC;

F : out STD\_LOGIC;

G : out STD\_LOGIC);

end Exp1\_201b299;

architecture Dataflow of Exp1\_201b299 is

begin

C <= not A;

D <= A and B;

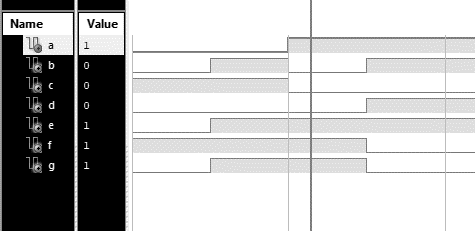
E <= A or B;

F <= A nand B;

G <= A xor B;

end Dataflow;

**Test Bench code:**



A <= '0'; B<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

wait for 100 ns;

A <= '1'; B<= '0';

wait for 100 ns;

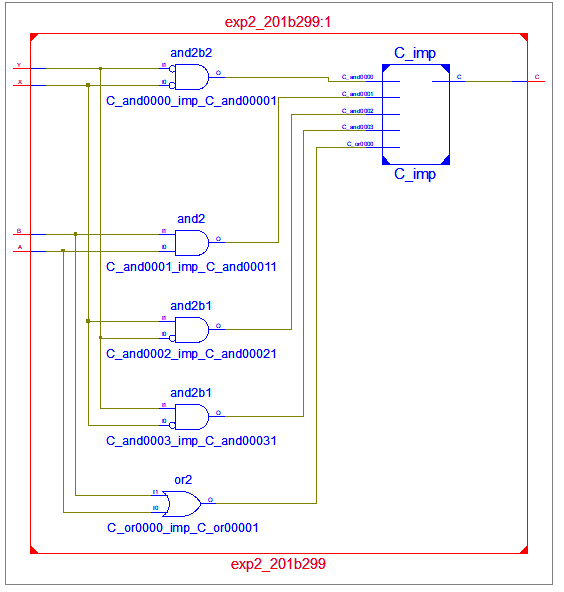
A <= '1'; B<= '1';

wait for 100 ns;

**Exercise#2:** Design two inputs and one output All-in-One logic gate diagram shown in Fig.2. Write theVHDL code in behavioral style of modeling using (i) if-then-else (ii) case-when.

**Design code (if –then-else):**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity exp2\_201b299 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC;

X : in STD\_LOGIC;

Y : in STD\_LOGIC);

end exp2\_201b299;

architecture Behavioral of exp2\_201b299 is

begin

process (A,B,X,Y)

begin

if(X = '0' and Y = '0') then

C <= A and B;

elsif (X = '1' and Y = '0') then

C <= A or B;

elsif (X = '0' and Y = '1') then

C <= A nor B;

else

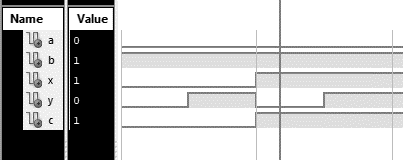
C <= A nand B;

end if;

end process;

end Behavioral;

**Test Bench code:**

A <= '0'; B<= '1';

X <= '0'; Y<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

X <= '0'; Y<= '1';

wait for 100 ns;

A <= '0'; B<= '1';

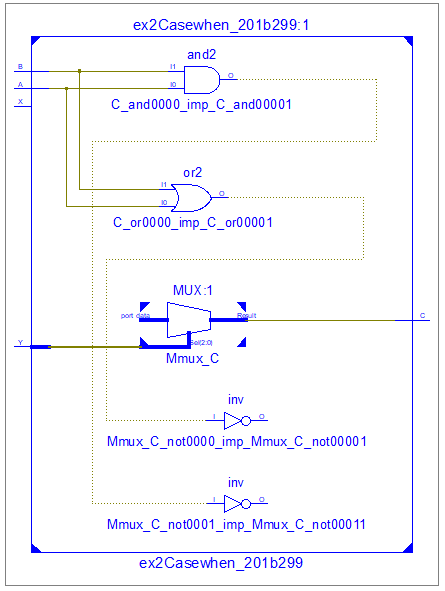
X <= '1'; Y<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

X <= '1'; Y<= '1';

wait for 100 ns;

**Design code (case-when):**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ex2Casewhen\_201b299 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC;

X : in STD\_LOGIC;

Y : in STD\_LOGIC);

end ex2Casewhen\_201b299;

architecture Behavioral of ex2Casewhen\_201b299 is

begin

process(X,Y,A,B)

variable SEL: std\_logic\_vector(1 downto 0);

begin

SEL := X & Y;

case SEL is

when "00" => C <= A and B;

when "01" => C <= A or B;

when "10" => C <= A nand B;

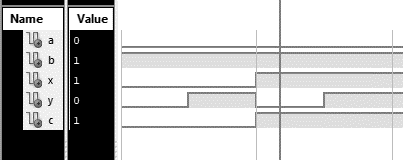
when others => C <= A nor B;

end case;

end process;

end Behavioral;

**Test Bench code:**

A <= '0'; B<= '1';

X <= '0'; Y<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

X <= '0'; Y<= '1';

wait for 100 ns;

A <= '0'; B<= '1';

X <= '1'; Y<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

X <= '1'; Y<= '1';

wait for 100 ns;