A) OR GATE CODE:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or\_gate is

Port ( p : in STD\_LOGIC;

q : in STD\_LOGIC;

r : out STD\_LOGIC);

end or\_gate;

architecture Behavioral of or\_gate is

begin

r<= p or q;

end Behavioral;

B) HALF ADDER CODE:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity half\_adder is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

s : out STD\_LOGIC;

c : out STD\_LOGIC);

end half\_adder;

architecture Behavioral of half\_adder is

begin

s <= a xor b ;

c<= (a and b);

end Behavioral;

C) FULL ADDER Using Half Adders :-

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity full\_adderst is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

cin : in STD\_LOGIC;

s : out STD\_LOGIC;

cout : out STD\_LOGIC);

end full\_adderst;

architecture Structural of full\_adderst is

component half\_adder is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

s : out STD\_LOGIC;

c : out STD\_LOGIC);

end component;

component or\_gate is

Port ( p : in STD\_LOGIC;

q : in STD\_LOGIC;

r : out STD\_LOGIC);

end component;

signal temp1,temp2,temp3 : std\_logic;

begin

u1: half\_adder port map(a,b,temp1,temp2);

u2: half\_adder port map(temp1,cin,s,temp3);

u3: or\_gate port map(temp2,temp3,cout);

end Structural;

D) Test Bench-

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity full\_addertb is

-- Port ( );

end full\_addertb;

architecture Behavioral of full\_addertb is

component full\_adderst is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

cin : in STD\_LOGIC;

s : out STD\_LOGIC;

cout : out STD\_LOGIC);

end component;

signal a,b,cin,s,cout : std\_logic;

begin

u1 : full\_adderst port map(a,b,cin,s,cout);

process

begin

--stimuli generation

a<='0';

b<='0';

cin<='0';

wait for 100ns;

a<='0';

b<='0';

cin<='1';

wait for 100ns;

a<='0';

b<='1';

cin<='0';

wait for 100ns;

a<='0';

b<='1';

cin<='1';

wait for 100ns;

a<='1';

b<='0';

cin<='0';

wait for 100ns;

a<='1';

b<='0';

cin<='1';

wait for 100ns;

a<='1';

b<='1';

cin<='0';

wait for 100ns;

a<='1';

b<='1';

cin<='1';

wait for 100ns;

end process;

end Behavioral;