Exp1: Half Adder

Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity HA is
Port ( a : in STD_LOGIC; -- Input A
       b : in STD_LOGIC; -- Input B
       s : out STD_LOGIC; -- Sum output
       c : out STD_LOGIC); -- Carry output
       end HA;
architecture Behavioral of HA is
       begin
       s <= a xor b;
       c <= a and b;
end Behavioral;
```

TB Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity HA_TB is -- Port ( );
end HA_TB;
architecture Behavioral of HA_TB is
component HA is
Port ( A : in STD_LOGIC;
       B : in STD_LOGIC;
       S : out STD_LOGIC;
```

```
        C : out STD_LOGIC);
end component;
signal A,B,S,C:std_logic:='0';
begin
uut: HA port map (A=> A ,B=>B ,S=>S ,C=>C);
process
begin
A<='0';
B<='0';
wait for 10ns;
A<='0';
B<='1';
wait for 10ns;
A<='1';
B<='0';
wait for 10ns;
A<='1';
B<='1';
wait for 10ns;
WAIT;
end process;
end Behavioral;
```

Port Names	FPGA Pins
a	R2
b	T1
s	P1
c	L1

Full Adder

Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Entity declaration
entity Full_Adder is
    Port (
        a    : in  STD_LOGIC;
        b    : in  STD_LOGIC;
        cin  : in  STD_LOGIC;
        sum  : out STD_LOGIC;
        cout : out STD_LOGIC
    );
end Full_Adder;

-- Architecture
architecture Behavioral of Full_Adder is
begin
    sum  <= a XOR b XOR cin;
    cout <= (a AND b) OR (b AND cin) OR (a AND cin);
end Behavioral;
```

TB CODE

```
Library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FA_test is
end FA_test;

architecture Behavioral of FA_test is
   component Full_Adder
      Port (
         a    : in  STD_LOGIC;
         b    : in  STD_LOGIC;
         cin  : in  STD_LOGIC;
         sum  : out STD_LOGIC;
         cout : out STD_LOGIC
      );
   end component;

   signal a, b, cin, sum, cout : STD_LOGIC;

begin
   -- Instantiate the Full Adder
   uut: Full_Adder port map (
      a => a,
      b => b,
      cin => cin,
```

```
        sum => sum,
        cout => cout
    );

    -- Test Process
    process
    begin
        -- Test all 8 input combinations
        a <= '0'; b <= '0'; cin <= '0'; wait for 100 ns;
        a <= '0'; b <= '0'; cin <= '1'; wait for 100 ns;
        a <= '0'; b <= '1'; cin <= '0'; wait for 100 ns;
        a <= '0'; b <= '1'; cin <= '1'; wait for 100 ns;
        a <= '1'; b <= '0'; cin <= '0'; wait for 100 ns;
        a <= '1'; b <= '0'; cin <= '1'; wait for 100 ns;
        a <= '1'; b <= '1'; cin <= '0'; wait for 100 ns;
        a <= '1'; b <= '1'; cin <= '1'; wait for 100 ns;
        wait; -- stop simulation
    end process;
end Behavioral;
```

Port Names	FPGA Pins
a	R2
b	T1
s	P1
co	L1
ci	U1

Full Adder Using Half Adder

OR Gate:

```
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;
```

Half Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity HA is
Port ( a : in STD_LOGIC; -- Input A
       b : in STD_LOGIC; -- Input B
       s : out STD_LOGIC; -- Sum output
       c : out STD_LOGIC); -- Carry output
       end HA;
architecture Behavioral of HA is
       begin
       s <= a xor b;
```

```
        c <= a and b;
end Behavioral;
```

Full Adder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Entity declaration
entity Full_Adder is
   Port (
      a    : in  STD_LOGIC;
      b    : in  STD_LOGIC;
      cin  : in  STD_LOGIC;
      sum  : out STD_LOGIC;
      cout : out STD_LOGIC
   );
end Full_Adder;

-- Architecture
architecture Behavioral of Full_Adder is
begin
   sum  <= a XOR b XOR cin;
   cout <= (a AND b) OR (b AND cin) OR (a AND cin);
end Behavioral;
```

FA_TB

---code

Port Names	FPGA Pins
a	R2
b	T1
s	P1
co	L1
ci	U1

Ripple Carry Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity FA is
   Port ( a  : in  STD_LOGIC;
          b  : in  STD_LOGIC;
          ci : in  STD_LOGIC;
          s  : out STD_LOGIC;
          co : out STD_LOGIC);
end FA;

architecture Behavioral of FA is
begin
   s <= a xor b xor ci;
   co <= (a and b) or (ci and (a xor b));
end Behavioral;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity RCA is
   Port ( a  : in  STD_LOGIC_VECTOR (3 downto 0);
          b  : in  STD_LOGIC_VECTOR (3 downto 0);
          ci : in  STD_LOGIC;
```

```
        s  : out STD_LOGIC_VECTOR (3 downto 0);
        co : out STD_LOGIC);
end RCA;

architecture Structural of RCA is

   component FA is
      Port ( a  : in  STD_LOGIC;
             b  : in  STD_LOGIC;
             ci : in  STD_LOGIC;
             s  : out STD_LOGIC;
             co : out STD_LOGIC);
   end component;

   signal c1, c2, c3 : STD_LOGIC;

begin

   FA0: FA port map(
      a  => a(0),
      b  => b(0),
      ci => ci,
      s  => s(0),
      co => c1
   );
```

```
    FA1: FA port map(
        a  => a(1),
        b  => b(1),
        ci => c1,
        s  => s(1),
        co => c2
    );

    FA2: FA port map(
        a  => a(2),
        b  => b(2),
        ci => c2,
        s  => s(2),
        co => c3
    );

    FA3: FA port map(
        a  => a(3),
        b  => b(3),
        ci => c3,
        s  => s(3),
        co => co
    );
end Structural;
```

Port Names	FPGA Pins
a3	R2
a2	T1
a1	U1
a0	W2
b3	R3
b2	T2
b1	T3
b0	V2
ci	V17
co	U16
s3	L1
s2	P1
s1	N3
s0	P3