VIVEKANANDA COLLEGE
THAKURPUKUR
KOLKATA-700063

NAAC ACCREDITED 'A' GRADE

Topic: VHDL_2

Course Title: Digital Electronics and VHDL

Paper: CC-9

Unit:

Semester: 4

Name of the Teacher: Sanchari Guha

Name of the Department: Electronics

Adding Operators :-

+ ; −, & → Concatenation ('B'-'0' & '1'&'1' = 010).

Multiplying Operators :-

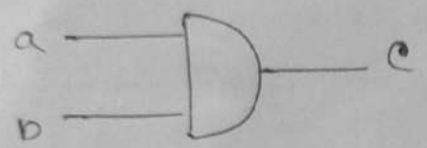
*, /, Mod, Rem

Miscellaneous Operators :-

abs : absolute

** : exponentiation

AND procen statement :



c = a · b

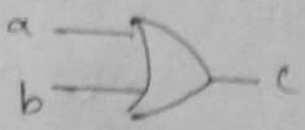
truth table

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

```
library IEEE ;
Use IEEE.STD_LOGIC_1164.ALL;

entity and_beh is
port (a: in STD_LOGIC;
      b: in STD_LOGIC;
      c: out STD_LOGIC);
end and_beh;
architecture Behavioral of and_beh is
begin
  process(a,b) => (statement)
  begin
    c <= a and b;
  end procen;
  end ~~and_beh~~; Behavioral;
```

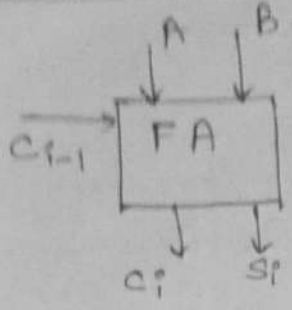
OR gate



C = a+b

```
library IEEE,
USE IEEE.STD_LOGIC_1164.ALL;
entity test_or_beh is
port ( a: in STD_LOGIC;
       b: in STD_LOGIC;
       c: out STD_LOGIC);
end test_or_beh;
architecture Behavioral of
test_or_beh is
begin
procen (a,b)
begin
if (a='0' and b='0') then
 c<='0';
elsif (a='0' and b='1') then
 c<='1';
elsif (a='1' and b='0') then
c<='1';
elseif (a='1'and b='1') then
 c<='1';
end if;
end procen;
end Behavioral.
```

Full adder



$S \leftarrow a \text{ xor } B \text{ xor } C_{in}$

$Cout \leftarrow cin \text{ and } (a \text{ xor } b)$ or $(a \text{ and } b)$

```
library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_ARITH.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity full adder is
  port (cin : in std_logic;
        Cout : out std_logic;
        a : in std_logic;
        b : in std_logic;
        s : out std_logic);
  end fulladder;

architecture Behavioral of fulladder is
begin process (a, b, cin)
begin
    if cin = '1' then
      if a = '1' then
          if b = '0' then
            S <= '0';
            Cout <= '1';
      else
            s <= '1';
            Cout <= '0';
      end if;
    else
      if b = '0' then
        s <= '1';
        Cout <= '0';
      else
        s <= '0';
        Cout <= '1';
    end if; end if;
```

Cin	a	b	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

```
else
    if a = '0' then
    if b = '0' then
        s <= '0';
        cout <= '0';
    else
        s <= '0';
        cout <= '0';
    end if;
else
    if b = '0' then
        s <= '1';
        cout <= '0';
    else
        s <= '0';
        cout <= 1;
    end if;
end if;
end if;
end process;
end Behavioral;
```

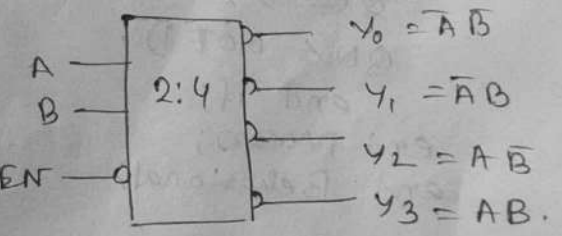
c=0 a=0 b=0

c=0 a=0 b=1

a=1 b=0

0 0 1

2 to 4 line decoder




```
library IEEE;
use IEEE_LOGIC_1164.all;
entity decoder 2_4 is
 port(
     a: in  STD_LOGIC
     b: in  STD_LOGIC
     w: OUT
     x: OUT
     y: OUT
     z: OUT
     );
 end decoder2_4;
architecture decoder 2_4_arc of decoder 2_4 is
 begin
     w <= (not a) and (not b);
     x <= (not a) and b;
     y <= a and (not b);
     z <= a and b;
 end decoder 2_4_arc;
```

D - latch :-

(D) I/p	Q(t)	Q(t+1)
0	0	0
0	1	0
1	0	1
1	1	1

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

entity DFF1 is
  port (D: in std_logic;
        CLK: in std_logic;
        Q: out std_logic;
        QN: out std_logic);
end DFF1;

architecture Behavioral of DFF1 is
begin
  process (CLK)
  begin
    if CLK = '1' then
      Q <= D;
      QN <= Not D;
    end if;
  end process;
end Behavioral;
```