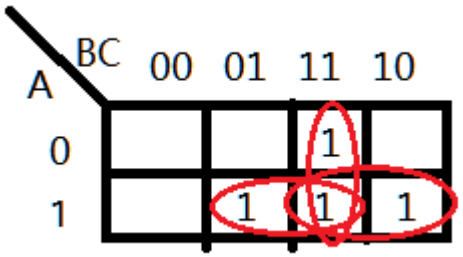
1. Majority circuit

a) Truce Table:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

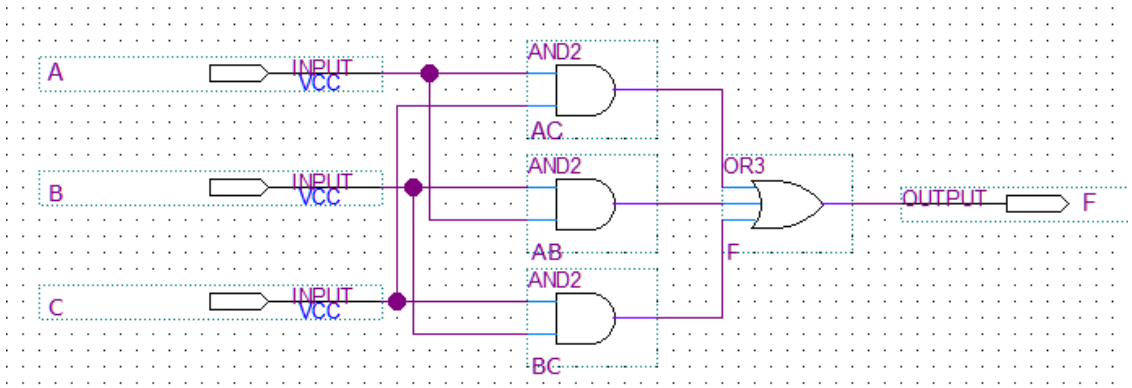
K-map:



Boolean equation:

F = AC + AB + BC

Logic Circuit:

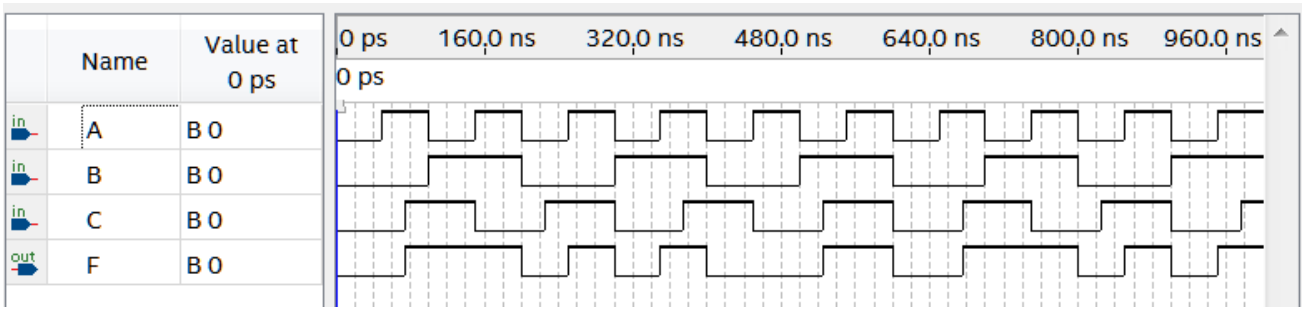


b) VHDL code:

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity and_2 is
5      port ( a1, a2  : in  std_logic;
6             F_and_2 : out std_logic );
7   end and_2;
8
9   architecture design_and_2 of and_2 is
10  begin
11     F_and_2 <= a1 and a2;
12  end design_and_2;
13
14  library ieee;
15  use ieee.std_logic_1164.all;
16
17  entity or_3 is
18     port ( o1, o2, o3 : in  std_logic;
19            F_or_3     : out std_logic );
20  end or_3;
21
22  architecture design_or_3 of or_3 is
23  begin
24     F_or_3 <= o1 or o2 or o3;
25  end design_or_3;
26
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity A1_Task1 is
    port( A, B, C  : in  std_logic;
          F        : out std_logic );
end A1_Task1;

architecture design of A1_Task1 is

component and_2
    port ( a1, a2  : in  std_logic;
           F_and_2 : out std_logic );
end component;

component or_3
    port ( o1, o2, o3 : in  std_logic;
           F_or_3     : out std_logic );
end component;

signal out1, out2, out3 :  std_logic;

begin
    A1: and_2 port map (A, C, out1);
    A2: and_2 port map (A, B, out2);
    A3: and_2 port map (B, C, out3);
    O1: or_3  port map (out1, out2, out3, F);
end design;
```

Test Bench:



2. Subtractor Circuit

a) Equation of full adders:

Sum = x ⊕ y ⊕ Cin

Cout = x*y + Cin*x + Cin*y

For subtractor, use full adders and let:

y = y'

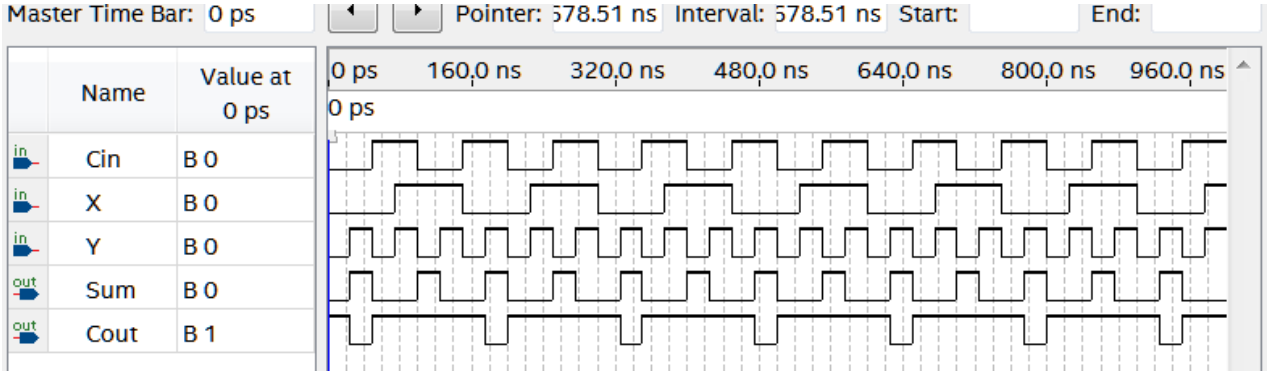
Cin = 1

VHDL code:

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity adder is
    port (X, Y, Cin   :  in    std_logic;
          Cout, Sum   :  out   std_logic);
end adder;

architecture eq of adder is
begin
    Sum   <= X xor Y xor Cin;
    Cout  <= (X and Y) or (X and Cin) or (Y and Cin);
end eq;
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Subtractor_1 is
   port (X, Y, Cin   :  in    std_logic;
         Cout, Sum   :  out   std_logic);
end Subtractor_1;

architecture sub of Subtractor_1 is
   component adder
      port (X, Y, Cin   :  in    std_logic;
            Cout, Sum   :  out   std_logic);
   end component;
begin
   sub1 : adder port map (X, not Y, '1', Cout, Sum);
end sub;
```

b) Test bench:

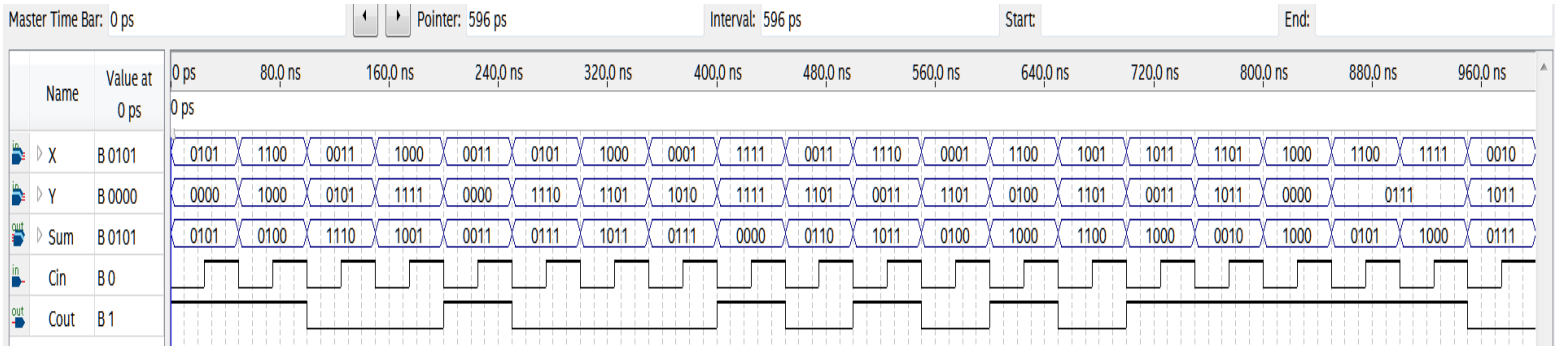


c) 4-bit subtractor:

VHDL code:

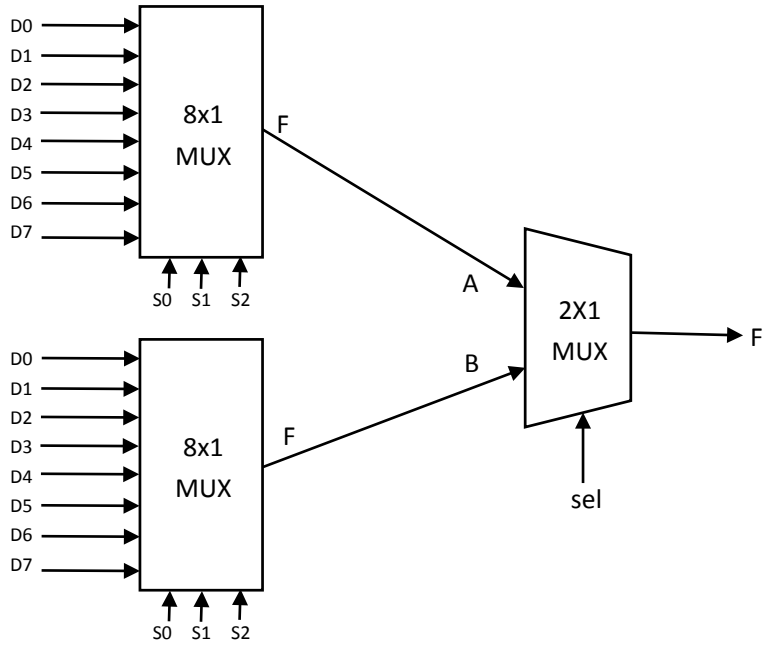
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity Subtractor is
   port (X, Y  :  in    std_logic_vector (3 downto 0);
         Cin   :  in    std_logic;
         Cout  :  out   std_logic;
         Sum   :  out   std_logic_vector (3 downto 0));
end Subtractor;

architecture sub4 of Subtractor is
   component adder
      port (X, Y, Cin   :  in    std_logic;
            Cout, Sum   :  out   std_logic);
   end component;

   signal c :  std_logic_vector  (3 downto 1);
begin
   sub1 : adder port map (X(0), not Y(0), '1',  c(1), Sum(0));
   sub2 : adder port map (X(1), not Y(1), c(1), c(2), Sum(1));
   sub3 : adder port map (X(2), not Y(2), c(2), c(3), Sum(2));
   sub4 : adder port map (X(3), not Y(3), c(3), Cout, Sum(3));
end sub4;
```

d)test bench:



3. 16x1 mux

a) Block diagram



b) VHDL code:

```
library ieee;
use ieee.std_logic_1164.all;

entity and_2 is
   port ( a1, a2  : in  std_logic;
          F_and_2 : out std_logic );
end and_2;

architecture design_and_2 of and_2 is
begin
    F_and_2 <= a1 and a2;
end design_and_2;

library ieee;
use ieee.std_logic_1164.all;

entity and_4 is
   port ( a1, a2, a3, a4   : in  std_logic;
          F_and_4          : out std_logic );
end and_4;

architecture design_and_4 of and_4 is
begin
    F_and_4 <= a1 and a2 and a3 and a4;
end design_and_4;
```

```
26
27    library ieee;
28    use ieee.std_logic_1164.all;
29
30    entity or_2 is
31       port ( o1, o2     : in  std_logic;
32              F_or_2     : out std_logic );
33    end or_2;
34
35    architecture design_or_2 of or_2 is
36    begin
37       F_or_2 <= o1 or o2;
38    end design_or_2;
39
40    library ieee;
41    use ieee.std_logic_1164.all;
42
43    entity or_8 is
44       port ( o1, o2, o3, o4, o5, o6, o7, o8  : in  std_logic;
45              F_or_8                          : out std_logic );
46    end or_8;
47
48    architecture design_or_8 of or_8 is
49    begin
50       F_or_8 <= o1 or o2 or o3 or o4 or o5 or o6 or o7 or o8;
51    end design_or_8;
52
53    library ieee;
54    use ieee.std_logic_1164.all;
55
56    entity inv is
57       port (i1    :  in    std_logic;
58             F_inv :  out   std_logic );
59    end inv;
60
61    architecture design_inv of inv is
62    begin
63       F_inv <= not i1;
64    end design_inv;
65
66    library ieee;
67    use ieee.std_logic_1164.all;
68
69    entity mux8 is
70       port (s0, s1, s2                        :  in  std_logic;
71             d0, d1, d2, d3, d4, d5, d6, d7    :  in  std_logic;
72             F_mux8                            :  out std_logic );
73    end mux8;
74
75    architecture design_mux8 of mux8 is
76
77    component and_4 is
78       port ( a1, a2, a3, a4   : in  std_logic;
79              F_and_4          : out std_logic );
80    end component;
81
82    component or_8 is
83       port ( o1, o2, o3, o4, o5, o6, o7, o8  : in  std_logic;
84              F_or_8                          : out std_logic );
85    end component;
86
87    component inv is
88       port (i1    :  in    std_logic;
89             F_inv :  out   std_logic );
90    end component;
91
92    signal s0_bar, s1_bar, s2_bar, out1, out2, out3, out4, out5, out6, out7, out8 : std_logic;
93
94    begin
95    inv0: inv port map (s0, s0_bar);
96    inv1: inv port map (s1, s1_bar);
97    inv2: inv port map (s2, s2_bar);
98
```

```
 99    A1:   and_4 port map (d0, s0_bar, s1_bar, s2_bar, out1);
100    A2:   and_4 port map (d1, s0_bar, s1_bar, s2, out2);
101    A3:   and_4 port map (d2, s0_bar, s1, s2_bar, out3);
102    A4:   and_4 port map (d3, s0_bar, s1, s2, out4);
103    A5:   and_4 port map (d4, s0, s1_bar, s2_bar, out5);
104    A6:   and_4 port map (d5, s0, s1_bar, s2, out6);
105    A7:   and_4 port map (d6, s0, s1, s2_bar, out7);
106    A8:   and_4 port map (d7, s0, s1, s2, out8);
107
108    O1:   or_8 port map (out1, out2, out3, out4, out5, out6, out7, out8, F_mux8);
109
110    end design_mux8;
111
112    library ieee;
113    use ieee.std_logic_1164.all;
114
115    entity mux2 is
116       port(sel    :  in    std_logic;
117            a, b   :  in    std_logic;
118            F_mux2:  out   std_logic );
119    end mux2;
120
121    architecture design_mux2 of mux2 is
122
123    component and_2 is
124       port ( a1, a2  : in  std_logic;
125              F_and_2 : out std_logic );
126    end component;
127
128    component or_2 is
129       port ( o1, o2     : in  std_logic;
130              F_or_2     : out std_logic );
131    end component;
132
133    component inv is
134       port (i1    :  in    std_logic;
135             F_inv :  out   std_logic );
136    end component;
137
138    signal sel_bar, out1, out2 : std_logic;
139
140    begin
141
142    inv0: inv port map (sel, sel_bar);
143
144    A1:   and_2 port map (a, sel_bar, out1);
145    A2:   and_2 port map (b, sel, out2);
146
147    O1:   or_2 port map (out1, out2, F_mux2);
148
149    end design_mux2;
150
151    library ieee;
152    use ieee.std_logic_1164.all;
153
154    entity mux16 is
155       port (s0, s1, s2, s3                          :  in  std_logic;
156             d0, d1, d2, d3, d4, d5, d6, d7          :  in  std_logic;
157             d8, d9, d10, d11, d12, d13, d14, d15    :  in  std_logic;
158             F_mux16                                 :  out std_logic );
159    end mux16;
160
161    architecture design_mux16 of mux16 is
162
163    component mux8 is
164       port (s0, s1, s2                          :  in  std_logic;
165             d0, d1, d2, d3, d4, d5, d6, d7      :  in  std_logic;
166             F_mux8                              :  out std_logic );
167    end component;
168
169    component mux2 is
170       port(sel    :  in    std_logic;
171            a, b   :  in    std_logic;
172            F_mux2:  out   std_logic );
173    end component;
174
```

```
175   signal F1, F2  :  std_logic;
176
177   begin
178   mux8_1   :  mux8 port map (s1, s2, s3, d0, d1, d2, d3, d4, d5, d6, d7, F1);
179   mux8_2   :  mux8 port map (s1, s2, s3, d8, d9, d10, d11, d12, d13, d14, d15, F2);
180
181   mux2_1   :  mux2 port map (s0, F1, F2, F_mux16);
182
183   end design_mux16;
184
185
186
```

c) Test bench