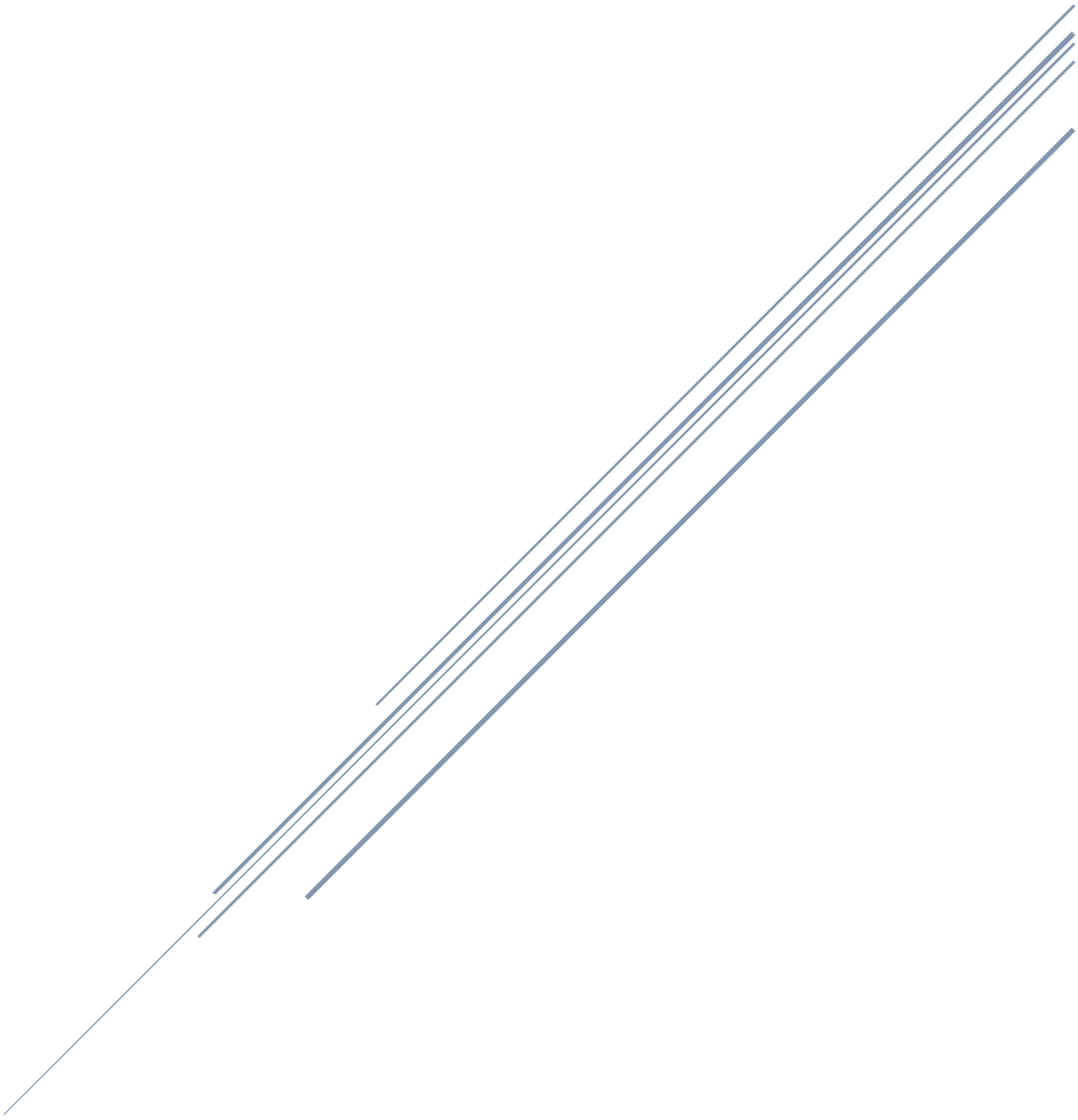
MOHAMED IBRAHIM GAD SAD

محمد ابراهيم جاد سعد 18011334



Second year communication lab 3

LAB 3 .VHDL

Components and packages

Problem 1 : 16-1 mux using 5 4-1 mux

Code of component :

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


entity mux4to1 is
port (x : in std_logic_vector(3 downto 0);
s : in std_logic_vector(1 downto 0 );
f:out std_logic);
end entity ;
architecture mux_arc of mux4to1 is
begin
with s select
f<= x(0)when "00",
        x(1)when"01",
        x(2)when"10",
        x(3)when others ;
end mux_arc ;
```

Code : of project

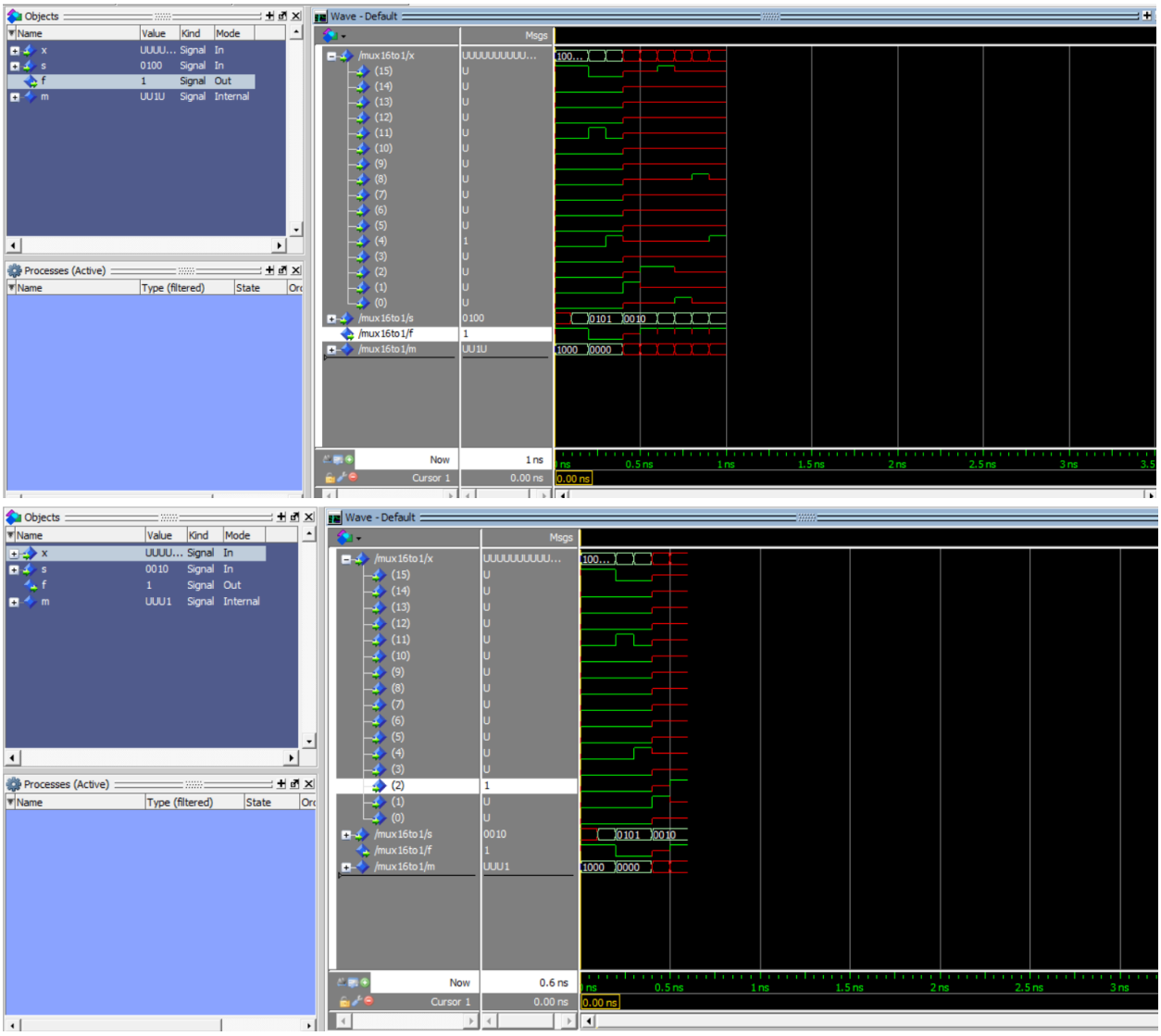
```
library ieee ;
use ieee.std_logic_1164.all;


entity mux16to1 is
port (x: in std_logic_vector(15 downto 0);
                    s:in std_logic_vector(3 downto 0);
                    f: out std_logic);


end entity ;
architecture mux_arc_ex of mux16to1 is
component mux4to1 is
port (x: in std_logic_vector(3 downto 0 );
                    s: in std_logic_vector(1 downto 0);
                    f : out std_logic);
end component ;
signal m :std_logic_vector(3 downto 0 );
begin
mux0:mux4to1 port map (x(3 downto 0  ),s(1 downto 0),m(0));
mux1:mux4to1 port map (x(7 downto 4  ),s(1 downto 0),m(1));
mux2:mux4to1 port map (x(11 downto 8  ),s(1 downto 0),m(2));
mux3:mux4to1 port map (x(15 downto 12 ),s(1 downto 0),m(3));
mux4:mux4to1 port map (m(3 downto 0),s(3 downto 2),f);
end mux_arc_ex ;
```

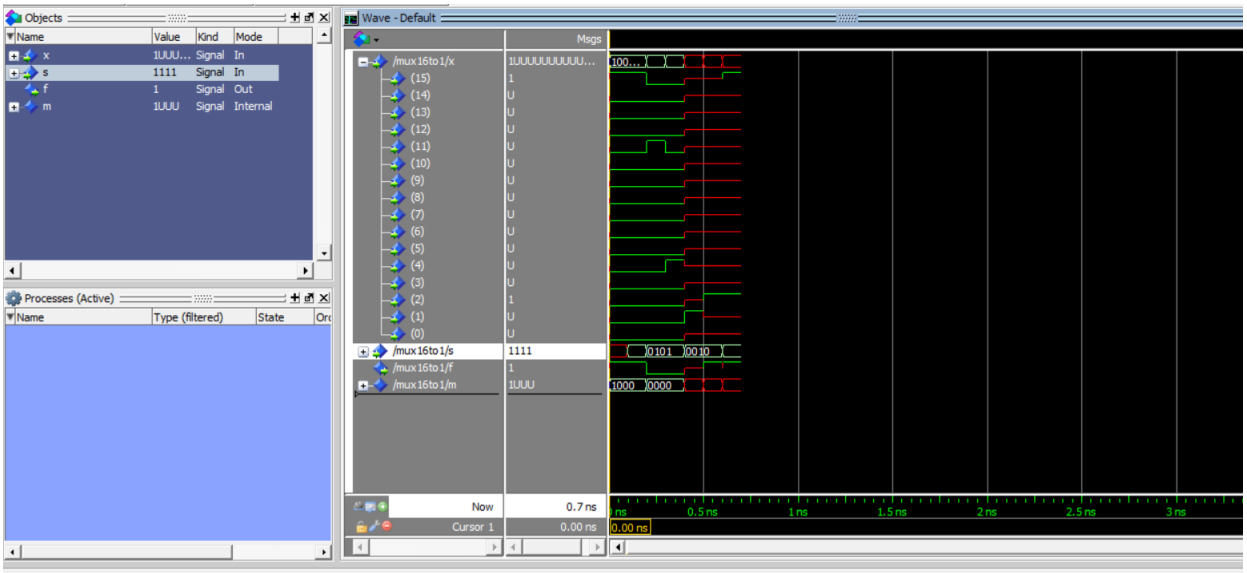
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity mux4to1 is
port (x : in std_logic_vector(3 downto 0);
 s : in std_logic_vector(1 downto 0 );
f:out std_logic);
end entity ;
architecture mux_arc of mux4to1 is
begin
with s select
f<= x(0)when "00",
   x(1)when"01",
   x(2)when"10",
   x(3)when others ;
end mux_arc ;
```

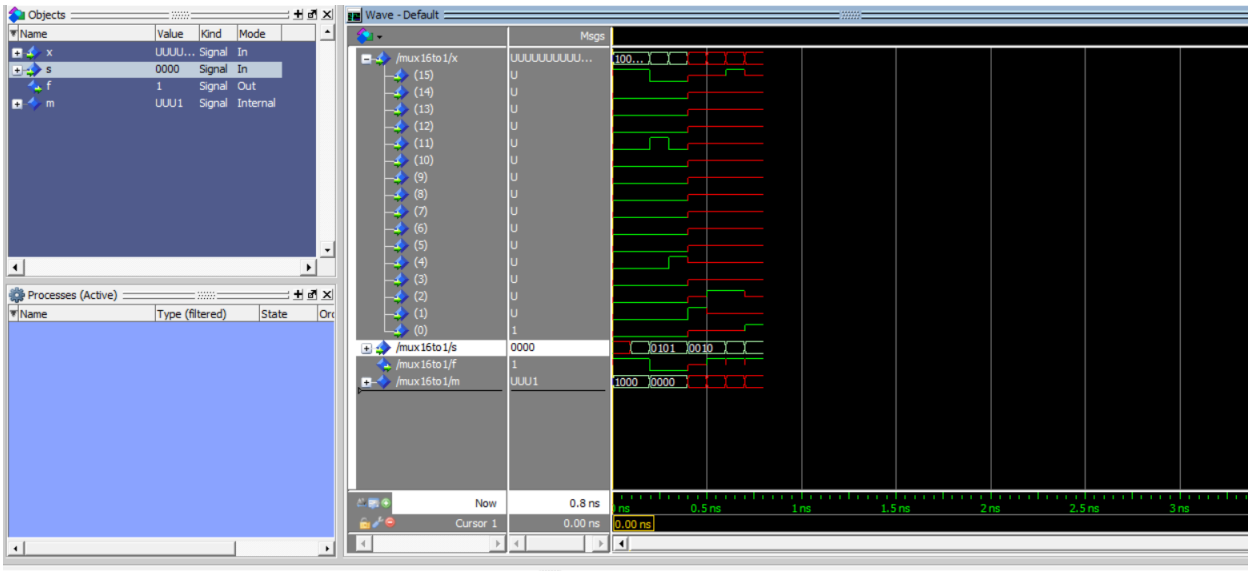
```vhdl
library ieee ;
use ieee.std_logic_1164.all;

entity mux16to1 is
port (x: in std_logic_vector(15 downto 0);
      s:in std_logic_vector(3 downto 0);
      f: out std_logic);

end entity ;
architecture mux_arc_ex of mux16to1 is
component mux4to1 is
port (x: in std_logic_vector(3 downto 0 );
      s: in std_logic_vector(1 downto 0);
      f : out std_logic);
end component ;
signal m :std_logic_vector(3 downto 0 );
begin
mux0:mux4to1 port map (x(3 downto 0  ),s(1 downto 0),m(0));
mux1:mux4to1 port map (x(7 downto 4  ),s(1 downto 0),m(1));
mux2:mux4to1 port map (x(11 downto 8  ),s(1 downto 0),m(2));
mux3:mux4to1 port map (x(15 downto 12 ),s(1 downto 0),m(3));
mux4:mux4to1 port map (m(3 downto 0),s(3 downto 2),f);
end mux_arc_ex ;
```

snapshots of working :

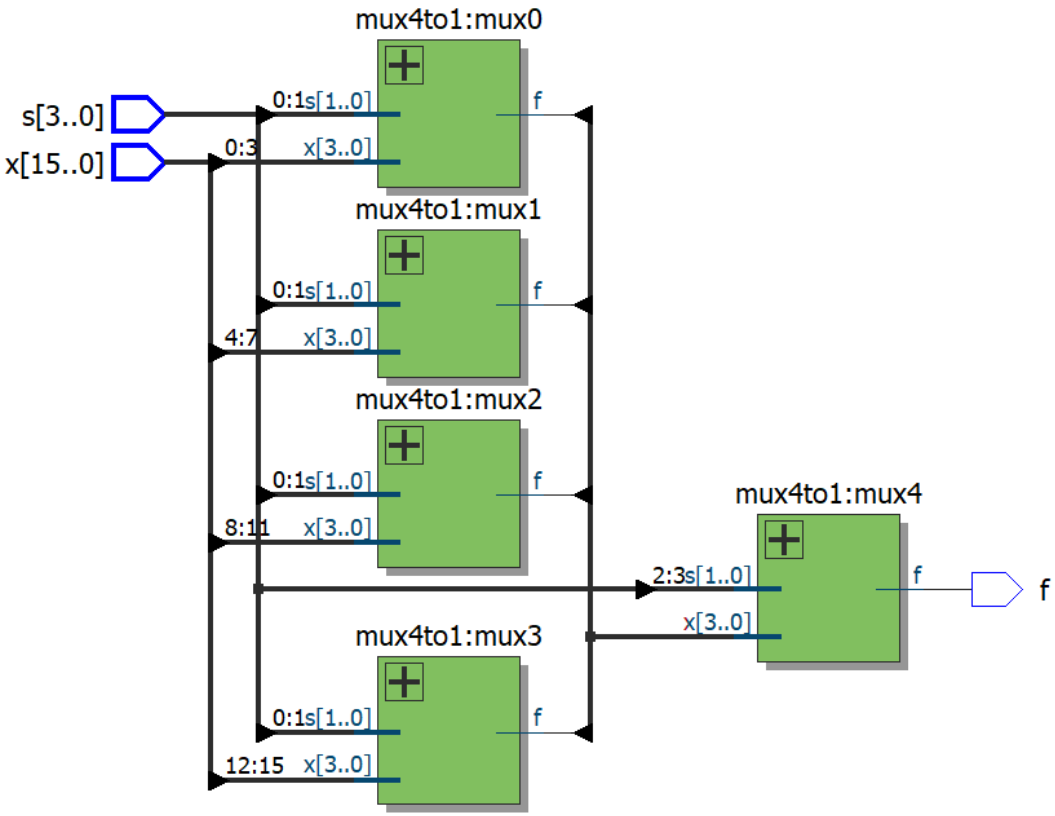
w → x




Objects
Name Value Kind Mode
x 1UUU... Signal In
s 1111 Signal In
f 1 Signal Out
m 1UUU Signal Internal
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
/mux16to1/x
/mux16to1/s
/mux16to1/f
/mux16to1/m
Now 0.7 ns
Cursor 1 0.00 ns
0.5 ns 1 ns 1.5 ns 2 ns 2.5 ns 3 ns


Objects
Name Value Kind Mode
x UUUU... Signal In
s 0000 Signal In
f 1 Signal Out
m UUU1 Signal Internal
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
/mux16to1/x
/mux16to1/s
/mux16to1/f
/mux16to1/m
Now 0.8 ns
Cursor 1 0.00 ns
0.5 ns 1 ns 1.5 ns 2 ns 2.5 ns 3 ns


Objects
Name Value Kind Mode
x UUUU... Signal In
s 1000 Signal In
f 1 Signal Out
m U1UU Signal Internal
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
/mux16to1/x
/mux16to1/s
/mux16to1/f
/mux16to1/m
Now 0.9 ns
Cursor 1 0.00 ns
0.5 ns 1 ns 1.5 ns 2 ns 2.5 ns 3 ns


mux4to1:mux0
s[3..0]
x[15..0]
0:1
s[1..0]
0:3
x[3..0]
f
mux4to1:mux1
0:1
s[1..0]
4:7
x[3..0]
f
mux4to1:mux2
0:1
s[1..0]
8:11
x[3..0]
f
mux4to1:mux4
2:3
s[1..0]
x[3..0]
f
f
mux4to1:mux3
0:1
s[1..0]
12:15
x[3..0]
f

Problem 2 : 1-bit - full adder

Code of component :

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

entity adder is
port (x,y,cin :in std_logic ;
                carry , sum : out std_logic );
end adder ;
architecture dataf of adder is
begin
        sum<= x xor y xor cin ;
        carry <= (x and y ) or (x and cin) or (y and cin );
end dataf;
```

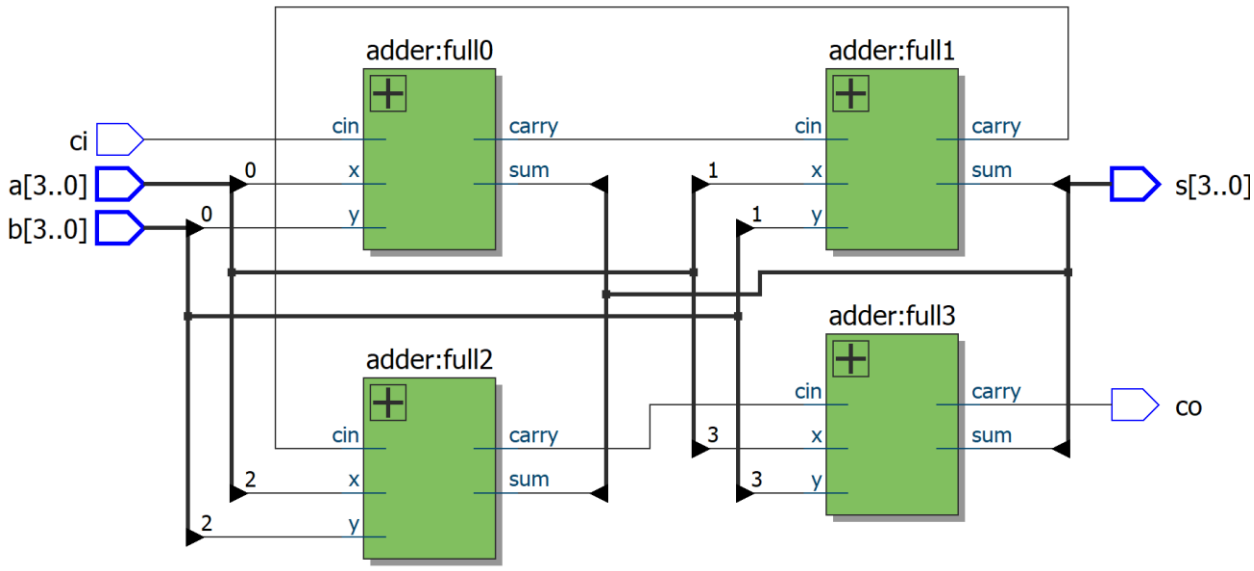
code of 4 bit binary full adder :

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

entity full_adder is
 port (a,b : in std_logic_vector(3 downto 0);
 ci : in std_logic ; co : out std_logic ;
 s : out std_logic_vector(3 downto 0 ));
end full_adder ;
```

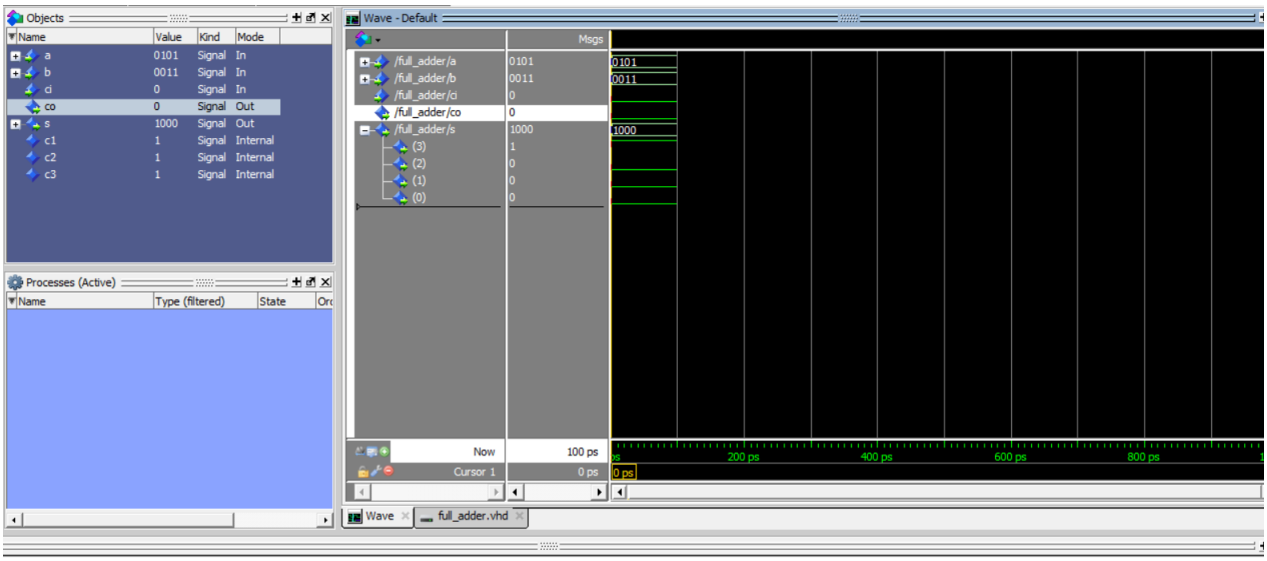
```
architecture fadder_arc of full_adder is
signal c1,c2,c3 :std_logic ;
component adder
        port(x,y,cin : in std_logic ; carry, sum :out std_logic);
end component ;
begin
 full0:adder port map (a(0) ,b(0),ci,c1,s(0));
 full1:adder port map (a(1) ,b(1),c1,c2,s(1));
 full2:adder port map (a(2) ,b(2),c2,c3,s(2));
 full3:adder port map (a(3) ,b(3),c3,co,s(3));
end fadder_arc ;
```

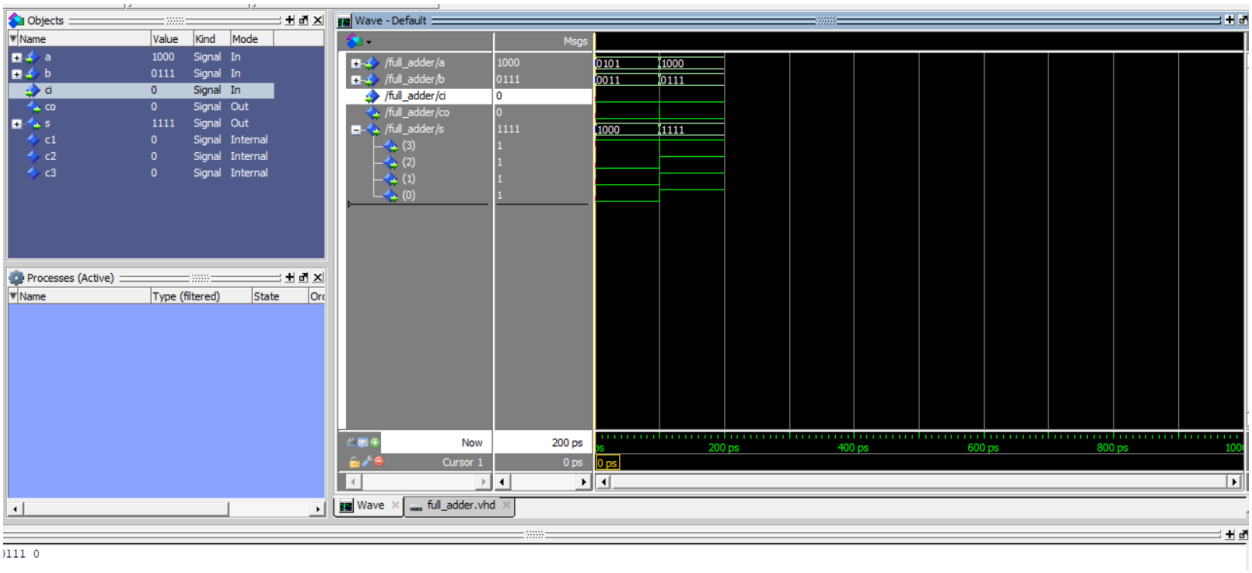
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity adder is
port (x,y,cin :in std_logic ;
      carry , sum : out std_logic );
end adder ;
architecture dataf of adder is
begin
   sum<= x xor y xor cin ;
   carry <= (x and y ) or (x and cin) or (y and cin );
end dataf;
```

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

entity full_adder is
 port (a,b : in std_logic_vector(3 downto 0);
  ci : in std_logic ; co : out std_logic ;
  s : out std_logic_vector(3 downto 0 ));
end full_adder ;
architecture fadder_arc of full_adder is
signal c1,c2,c3 :std_logic ;
component adder
   port(x,y,cin : in std_logic ; carry, sum :out std_logic);
end component ;
begin
 full0:adder port map (a(0) ,b(0),ci,c1,s(0));
 full1:adder port map (a(1) ,b(1),c1,c2,s(1));
 full2:adder port map (a(2) ,b(2),c2,c3,s(2));
 full3:adder port map (a(3) ,b(3),c3,co,s(3));
end fadder_arc ;
```



Working :


Objects
Name Value Kind Mode
a 0101 Signal In
b 0011 Signal In
ci 0 Signal In
co 0 Signal Out
s 1000 Signal Out
c1 1 Signal Internal
c2 1 Signal Internal
c3 1 Signal Internal
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
/full_adder/a 0101
/full_adder/b 0011
/full_adder/ci 0
/full_adder/co 0
/full_adder/s 1000
(3) 1
(2) 0
(1) 0
(0) 0
Now 100 ps
Cursor 1 0 ps
200 ps 400 ps 600 ps 800 ps
Wave
full_adder.vhd
0 0


Objects
Name Value Kind Mode
a 1000 Signal In
b 0111 Signal In
ci 0 Signal In
co 0 Signal Out
s 1111 Signal Out
c1 0 Signal Internal
c2 0 Signal Internal
c3 0 Signal Internal
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
/full_adder/a 1000
/full_adder/b 0111
/full_adder/ci 0
/full_adder/co 0
/full_adder/s 1111
(3) 1
(2) 1
(1) 1
(0) 1
Now 200 ps
Cursor 1 0 ps
200 ps 400 ps 600 ps 800 ps
Wave
full_adder.vhd
)111 0