WIA2009: Lab 02

Q1

Write VHDL code to implement the function $f(x1, x2, x3) = \sum m(0, 1, 3, 4, 5, 6)$.

Answer:

Expression using k-map: f = (x2)' + (x1)'(x3) + (x1)(x3)'

VHDL code

```
entity combination-circuit is
port(x1,x2,x3:in bit;
        f:out bit);

end combination-circuit;
architecture equations of combination-circuit is
begin
f = (not x2) or ((not x1) and x3) or (x1 and (not x3));
end equations;
```

Q2

a) Write VHDL code to describe the following functions

$f_1 = x_1x_3' + x_2x_3' + x_3'x_4' + x_1x_2 + x_1x_4'$
$f_2 = (x_1 + x_3') \cdot (x_1 + x_2 + x_4') \cdot (x_2 + x_3' + x_4')$

from the given statements

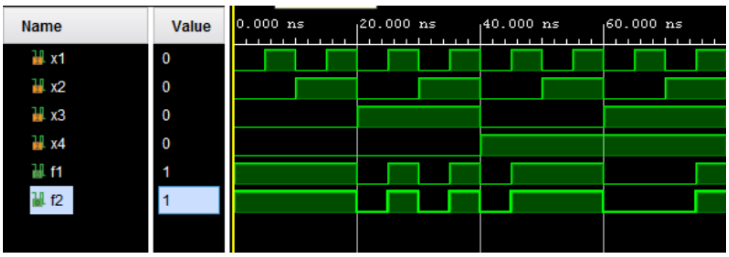
f1 = x1x3 + x1'x3' + x2x4 + x2'x4'
f2 = x1x2x3'x4' + x1'x2'x3x4 + x1x2'x3'x4 + x1'x2x3x4'

code:

```
entity code is
  port (x1,x2,x3,x4 : in std_logic;
        f1,f1: out std_logic);
  end code;

Architecture behavioural of code is
Begin
F1 <= (x1 and x3) or (not x1 and not x3) or
        (x2 and x4) or (not x2 and not x4);
F2 <= (x1 and x2 and (not x3) and (not x4)) or
        ((not x1) and (not x2) and x3 and x4) or
        (x1 and (not x2) and (not x3) and x4) or
        ((not x1) and x2 and x3 and (not x4));
End behavioural.
```

b) Use functional simulation to prove that $f_1 = f_2$.



Q3

Consider the following VHDL assignment statements

f_1 <= ((x1 AND x3) OR (NOT x1 AND NOT x3)) OR ((x2 AND x4) OR (NOT x2 AND NOT x4)) ;

f_2<= (x1 AND x2 AND NOT x3 AND NOT x4) OR (NOT x1 AND NOT x2 AND x3 AND x4) OR (x1 AND NOT x2 AND NOT x3 AND x4) OR (NOT x1 AND x2 AND x3 AND NOT x4) ;

a) Write complete VHDL code to implement f_1 and f_2.

```
library ieee;
use ieee.std_logic_1164.all;
entity functions is
port ( x1, x2, x3, x4 : in std_logic;
f1, f2 : out std_logic
);
end functions;
architecture arch of functions is
begin
f1 <= (x1 and not x3) or (x2 and not x3) or (not x3 and not x4) or (x1 and x2) or (x1 and not x4);
f2 <= (x1 or not x3) and (x1 or x2 or not x4) and (x2 or not x3 or not x4);
end arch;
```

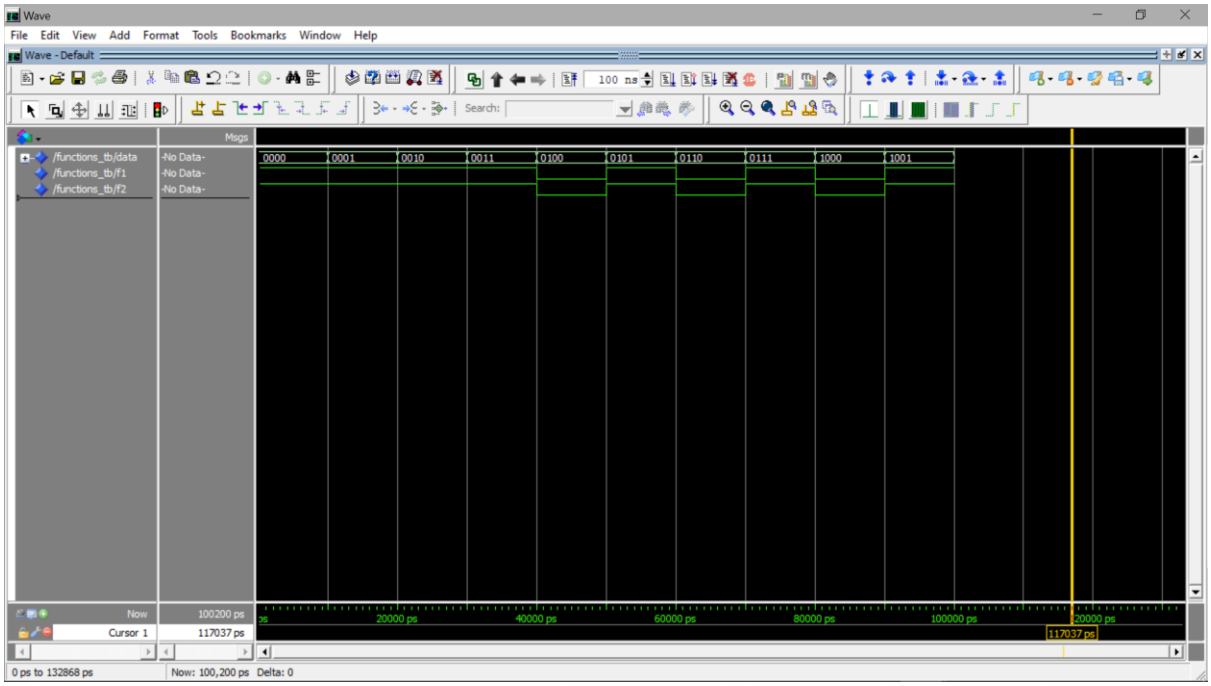
b) Use functional simulation to prove that $f_1 = f_2$'.

```
--Testbench
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;
entity functions_tb is
end;
architecture bench of functions_tb is
component functions
port ( x1, x2, x3, x4 : in std_logic;
f1, f2 : out std_logic
);
```

```
end component;
signal data : std_logic_vector(3 downto 0);
signal f1, f2: std_logic ;
begin
uut: functions port map ( x1 => data(0),
x2 => data(1),
x3 => data(2),
x4 => data(3),
f1 => f1,
f2 => f2 );
stimulus: process
begin

data <= "0000";
wait for 10 ns;
data <= "0001";
wait for 10 ns;
data <= "0010";
wait for 10 ns;
data <= "0011";
wait for 10 ns;
data <= "0100";
wait for 10 ns;
data <= "0101";
wait for 10 ns;
data <= "0110";
wait for 10 ns;
data <= "0111";
wait for 10 ns;
data <= "1000";
wait for 10 ns;
data <= "1001";
wait for 10 ns;
data <= "1010";
wait for 10 ns;
data <= "1011";
wait for 10 ns;
data <= "1100";
wait for 10 ns;
data <= "1101";
wait for 10 ns;
data <= "1110";
wait for 10 ns;
data <= "1111";
wait for 10 ns;
wait;
end process;

end;
```


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/functions_tb/data
/functions_tb/f1
/functions_tb/f2
-No Data-
-No Data-
-No Data-
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
Now
100200 ps
Cursor 1
117037 ps
20000 ps
40000 ps
60000 ps
80000 ps
100000 ps
120000 ps
0 ps to 132868 ps
Now: 100,200 ps Delta: 0