ECE 424 – INTRODUCTION TO VLSI DESIGN + LAB

LABORATORY WORK 6

In this laboratory work, we will try to understand multiplexers, decoders and encoders by writing their vhdl codes.

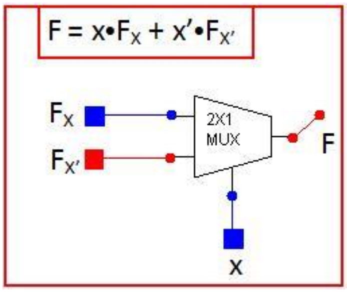
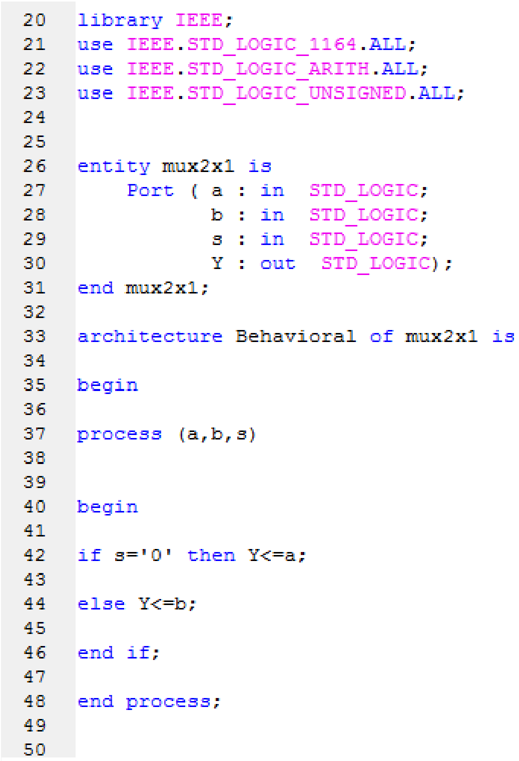


Figure 1

1) Figure 1 is explaining the working principle of a 2x1 multiplexer. The digital logic formula is also given in figure. We will write behavioral vhdl codes of this multiplexer together;



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


entity mux2x1 is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           s : in  STD_LOGIC;
           Y : out  STD_LOGIC);
end mux2x1;

architecture Behavioral of mux2x1 is

begin

process (a,b,s)


begin

if s='0' then Y<=a;

else Y<=b;

end if;

end process;


```

Figure 2

Write the codes in figure 2 by opening a new project on your computer. Simulate it and verify the results by using simulation result.

Then write a 4x1 multiplexer by using this information. Open a new project for this program. Don't use the same project folder.

2) If you want to use the programs that you had written before in a new project, you can do it by adding a "component" to your project. You wrote a 4x1 mux during first part, but you can also write this program by using 2x1 mux.

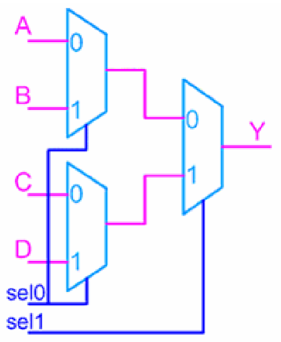
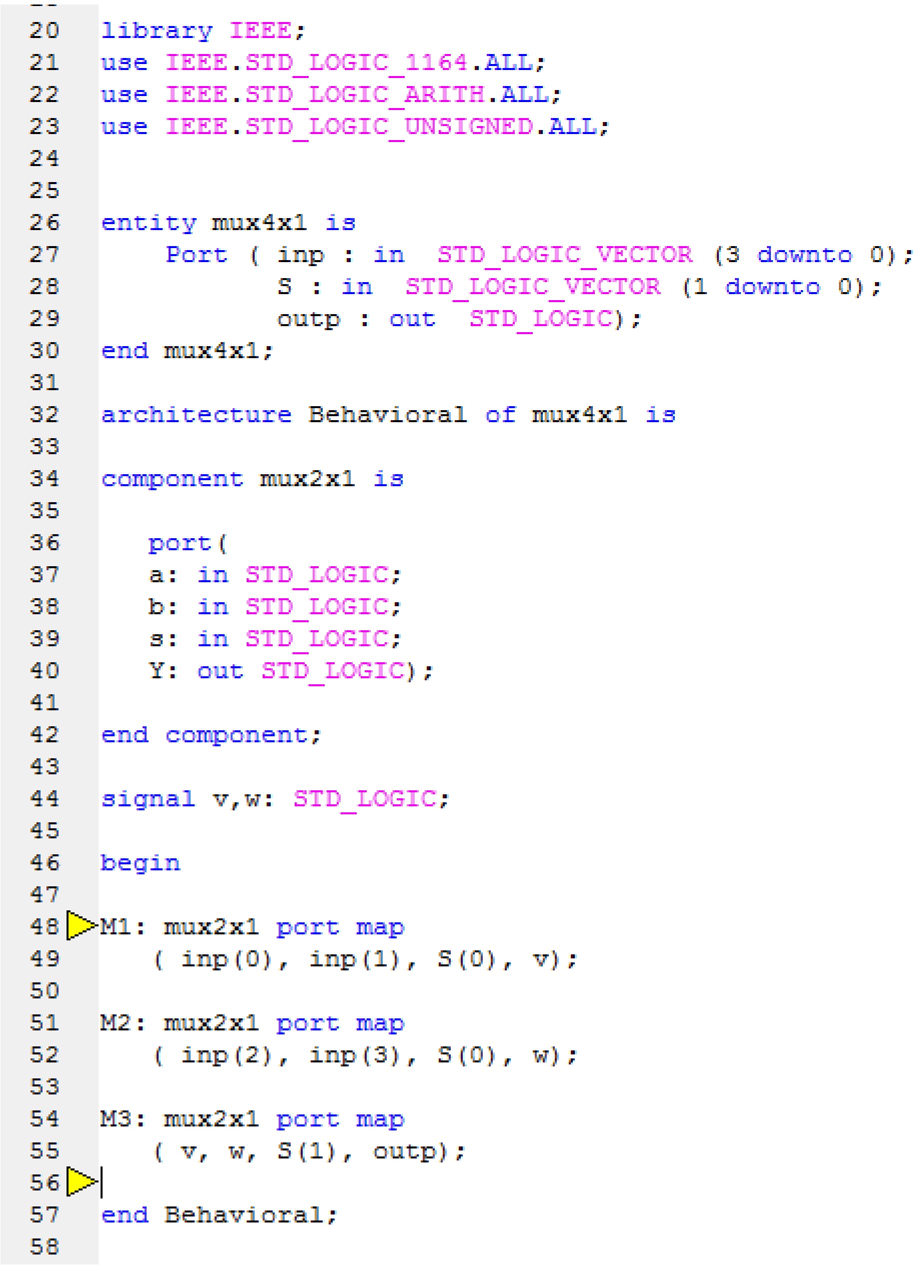


Figure 3

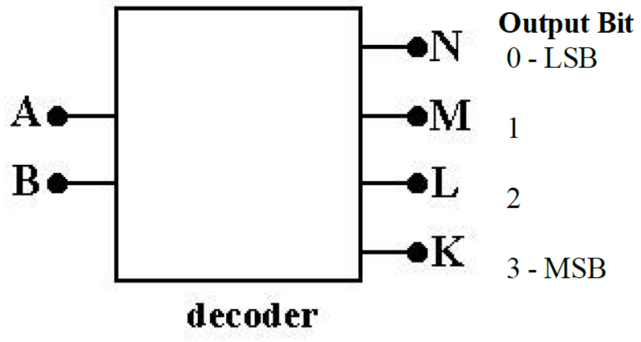
Figure 3 is explaining this condition. If you use three 2x1 mux, you can create a 4x1 mux. To do this work by using 2x1 mux, write the codes below in Figure 4 by opening a new project.



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


entity mux4x1 is
    Port ( inp : in  STD_LOGIC_VECTOR (3 downto 0);
           S : in  STD_LOGIC_VECTOR (1 downto 0);
           outp : out  STD_LOGIC);
end mux4x1;

architecture Behavioral of mux4x1 is

component mux2x1 is

   port(
   a: in STD_LOGIC;
   b: in STD_LOGIC;
   s: in STD_LOGIC;
   Y: out STD_LOGIC);

end component;

signal v,w: STD_LOGIC;

begin

M1: mux2x1 port map
   ( inp(0), inp(1), S(0), v);

M2: mux2x1 port map
   ( inp(2), inp(3), S(0), w);

M3: mux2x1 port map
   ( v, w, S(1), outp);

end Behavioral;

```

Figure 4

3) A 2 to 4 decoder generates an output signal on its output pins that is dependent on the binary code applied to its inputs. Only one of the output pins is high at any time. For 'n' inputs there will be **2^{n}** outputs. E.g. 2 to 4, 3 to 8, 4 to 16 etc.



inputs		outputs			
A	B	K	L	M	N
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

decoder Truth Table

Figure 5

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

entity Decoder_24 is
Port ( A : in STD_LOGIC;
       B : in STD_LOGIC;
       K : out STD_LOGIC;
       L : out STD_LOGIC;
       M : out STD_LOGIC;
       N : out STD_LOGIC );
end Decoder_24;

architecture Behavioral of Decoder_24 is

begin
-- assume that for input A and B that B
-- is Least Significant Bit (LSB)
-- and for outputs K,L,M, N that N is LSB

N <= '1' when A = '0' and B = '0' else '0';
M <= '1' when A = '0' and B = '1' else '0';
L <= '1' when A = '1' and B = '0' else '0';
K <= '1' when A = '1' and B = '1' else '0';

end Behavioral;
```

Figure 6

Write the codes in Figure 6 and simulate it to verify that this code is working as a 2x4 decoder.

Write also a 4x2 encoder and simulate it by using this information.

Homework 1: Write 16x1 multiplexer by using 4x1 mux as a component. After writing the codes simulate it and verify the truth table of 16x1 mux.

Homework 2: Make a Google search about writing for - loop in vhdl. Then write 3x8 decoder by using for loop. After writing the codes, simulate it.

Send me whole project folders. (Not screenshots or just codes)

Deadline: 17.12.2013 Tuesday, 21:00.

No late submission is going to be accepted!

Prepared By: **İbrahim BOZKURT**
e-mail: ibrahimbozkurt@cankaya.edu.tr