

ECE-2206 Digital Design Lab

Experiment # 7

Decoders and Multiplexers in HDL

Objective: 1) Gain experience in creating designs in a Hardware Description Language.

- 2) Implementing functions using decoders.
- 3) Designing complex multiplexers.

VHDL is a description language for digital systems. Existing systems can be modeled or new systems described for hardware mapping. VHDL describes hardware and the Xilinx compiler targets the designs to the FPGA.

A VHDL description has 2 parts: an entity and an architecture. The entity describes the interface to the hardware, that is the Input/Output of the design. The architecture describes the logic functions performed on the inputs to produce the outputs.

Prelab: Please turn in the following parts at the **beginning of lab hour**.

Parts: 1A, 2A, 2C, and 3A

- 1) **A)** Use the equations already derived for each segment from Lab 5, and implement your design in VHDL. Using the Xilinx software will facilitate creating your VHDL file.

Turn in a copy of your VHDL file at the beginning of lab hour.

The following example demonstrates the steps for creating a VHDL module.

- I. Click on **Project=> New Source**, choose **VHDL Module**, and type in your filename in the **File Name** field. Click on next and the following window will open up for you.

The screenshot shows the 'New Source Wizard' window at step 2, 'Define Module'. The title bar includes a back arrow icon, the VHDL logo, and the text 'New Source Wizard'. A close button (X) is in the top right corner.

Define Module

Specify ports for module.

Entity name: examp

Architecture name: Behavioral

Port Name	Direction	Bus	MSB	LSB
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

At the bottom are three buttons: 'More Info', 'Next', and 'Cancel'.

- II. On each row, under **Port Name** type in your inputs and outputs, and under **Direction** choose **in** or **out**. The following is an example of a 4 input and 2 output circuit.

Define Module

Specify ports for module.

Entity name

Architecture name

Port Name	Direction	Bus	MSB	LSB
A	in	<input type="checkbox"/>		
B	in	<input type="checkbox"/>		
C	in	<input type="checkbox"/>		
D	in	<input type="checkbox"/>		
Y1	out	<input type="checkbox"/>		
Y2	out	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

More Info Next Cancel

- III. Click on **Next**, then **Finish**. The following window will open up for you. A VHDL file with the required libraries and an **entity** with the listed input and output signals, is created. The design of the circuit can be placed between the **begin** and **end Behavioral** statements of the **architecture** section.

```

12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  -----
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity examp is
33      Port ( A : in  STD_LOGIC;
34            B : in  STD_LOGIC;
35            C : in  STD_LOGIC;
36            D : in  STD_LOGIC;
37            Y1 : out STD_LOGIC;
38            Y2 : out STD_LOGIC);
39  end examp;
40
41  architecture Behavioral of examp is
42
43  begin
44
45
46  end Behavioral;
47
48

```

- B)** High light your source file, compile your VHDL module by clicking on **Synthesize** in the **Processes for Source** window. Correct errors, if there are any. Simulate your VHDL design for all the input combinations. Make sure your design is working accurately.

(For simulation: Use clock and assign 16ns for MSB (bit3)
 8ns for (bit2)
 4ns for (bit1)
 2ns for (bit0)
 Simulate for 20 ns.)

- C)** Download your design on the board and test it for all possible input combinations.

- 2) A)** Write a VHDL description for a 4 to 16 decoder with enable. Turn in a hard copy as prelab.

- B)** Compile and simulate the code for all possible combinations of inputs.

(For simulation : Use clock and assign 16ns for MSB (bit3)
 8ns for (bit2)
 4ns for (bit1)
 2ns for (bit0)
 Simulate for 20 ns.)

- C)** Use the 4 to 16 decoder of problem 2 **as a component** and write a VHDL code to implement the following functions. Turn in a hard copy as prelab.

$$F_0 (A, B, C, D) = \sum m(0 , 14 , 15)$$

$$F_1 (A, B, C, D) = \sum m(1 , 13 , 15)$$

$$F_2 (A, B, C, D) = \sum m(2 , 12 , 15)$$

$$F_3 (A, B, C, D) = \sum m(3 , 11 , 15)$$

$$F_4 (A, B, C, D) = \sum m(4 , 10 , 15)$$

$$F_5 (A, B, C, D) = \sum m(5 , 9 , 15)$$

$$F_6 (A, B, C, D) = \sum m(6 , 8 , 15)$$

$$F_7 (A, B, C, D) = \sum m(7 , 15)$$

NOTE: You need to create a **new** VHDL source for your design.

To make a component:

- Highlight your 4 to 16 decoder file, expand **Design Utilities** then double click on **View VHDL Instantiation Template**. This step creates the VHDL Instantiation Template of your 4 to 16 decoder file.

To use the template in your new file:

- From the instantiation template copy the **component** section and paste it between the **Architecture** and **Begin** statements of your **new** file.
- From the instantiation template copy the **instant** section and paste it between the **Begin** and **End Behavioral** statements of your **new** file.

Add the necessary code to complete the source code.

D) Compile and simulate the code for all possible combinations of inputs.

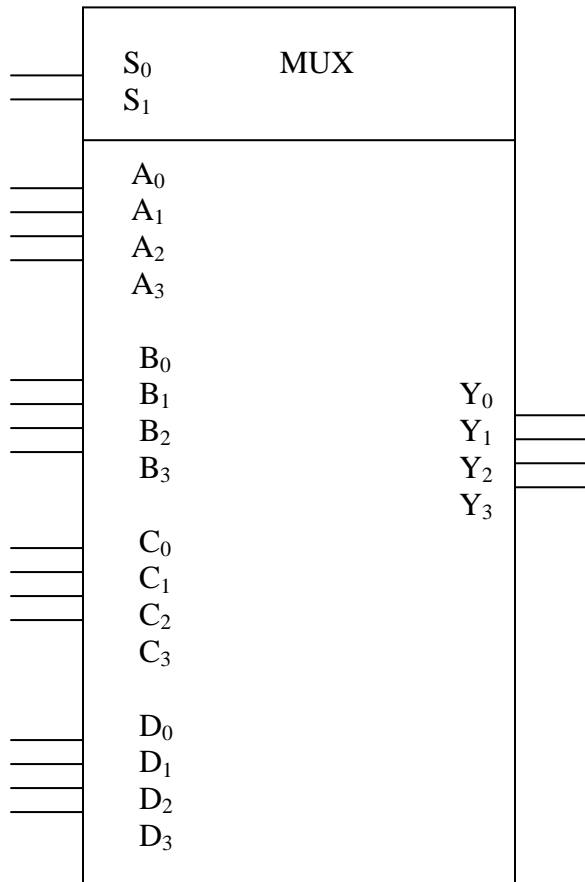
(For simulation : Use clock and assign 16ns for MSB (bit3)
8ns for (bit2)
4ns for (bit1)
2ns for (bit0)
Simulate for 20 ns.)

E) Download and test it on the board.

Note: Assign F_0 to LD0, F_1 to LD1, ..., F_7 to LD7

3) A) Write a VHDL code to implement a 4set:1set multiplexer that multiplexes 4 bits at a time. Turn in a hard copy as prelab.

(i.e.: If $S_1S_0 = 00$, then $A_3A_2A_1A_0 \Rightarrow Y_3Y_2Y_1Y_0$)



B) Compile and simulate the code for selected combinations of inputs.

(For simulation : Force the inputs and simulate for 5ns.)

$S(1:0) = 00$ $A(3:0) = 0000$ $B(3:0) = 0001$ $C(3:0) = 0010$ $D(3:0) = 0011$

$S(1:0) = 00$ $A(3:0) = 0110$ $B(3:0) = 0001$ $C(3:0) = 0010$ $D(3:0) = 0011$

$S(1:0) = 00$ $A(3:0) = 1100$ $B(3:0) = 0001$ $C(3:0) = 0010$ $D(3:0) = 0011$

$S(1:0) = 01$ $A(3:0) = 0001$ $B(3:0) = 0101$ $C(3:0) = 0010$ $D(3:0) = 0011$

$S(1:0) = 01$ $A(3:0) = 0001$ $B(3:0) = 1101$ $C(3:0) = 0010$ $D(3:0) = 0011$

S(1:0) = 01 A(3:0) = 0001 B(3:0) = 1111 C(3:0) = 0010 D(3:0) = 0011

S(1:0) = 10 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 0100 D(3:0) = 0011

S(1:0) = 10 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 0110 D(3:0) = 0011

S(1:0) = 10 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 1001 D(3:0) = 0011

S(1:0) = 11 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 0010 D(3:0) = 1010

S(1:0) = 11 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 0010 D(3:0) = 1100

S(1:0) = 11 A(3:0) = 0001 B(3:0) = 0001 C(3:0) = 0010 D(3:0) = 0011

All final VHDL files and simulation results for each part should be included in the report.