EXPERIMENT 4



OBJECTIVES:

- Examining the characteristics of XOR and XNOR gates.
- Demonstrate applications of XOR and XNOR gates
- Learn to use the VHDL approach to combinational logic design.

MATERIALS:

- Xilinx Vivado software, student or professional edition V2018.2 or higher.
- IBM or compatible computer with Pentium III or higher, 128 M-byte RAM or more, and 8 G-byte Or larger hard drive.
- BASYS 3 Board.

DISCUSSION:

So far we have studied five basic types of gates: AND, OR, NAND, NOR and NOT. In some applications, it is convenient to use two other types of gates: XOR and XNOR. The XOR and XNOR gates have their own symbols and unique characteristics. Common applications for XOR and XNOR gates are: comparators, switchable inverter/buffers, parity generator/checkers and adder/subtractor. They can also be used to simplify Boolean equations.

We will first discuss the properties of XOR and XNOR having two inputs.

Gate Characteristics:

1. The XOR Gate

Page 1 | 22

Symbol	Boolean Equation		Truth Table	
		In	put	Output
A	X=A'B+AB'	A	B	X
	$\mathbf{X} = \mathbf{A} \oplus \mathbf{B}$	0	0	0
		0	1	1
		1	0	1
		1	1	0

For a 2-input XOR gate, the output is High when the inputs are unequal. The output is Low when the inputs are equal. The Boolean equation for a 2-input XOR gate can be abbreviated as:

 $\mathbf{X} = \mathbf{A} \oplus \mathbf{B}$

However, the function definition remains the same.

Page 2 | 22

2. The XNOR Gate

Symbol	Boolean Equation		Truth Table	
		In	put	Output
	$\mathbf{Y} = \mathbf{A'B'} + \mathbf{AB}$	A	В	Y
в		0	0	1
		0	1	0
		1	0	0
		1	1	1

The output of an XNOR gate is the complement of that of a XOR. For a 2-input XNOR gate, the output is Low when the inputs are unequal but High when the inputs are equal. The Boolean equation for a 2-input XNOR gate can be written as:

$$Y = A \oplus B$$

The number of inputs for the XOR and XNOR gates can be two or more. The characteristics of XOR and XNOR gates can be extended to three or more inputs. We will examine the characteristics of 3-input XOR and XNOR gates.

PROCEDURE:

Section 1 XOR and XNOR characteristics:

Page 3 | 22

1. Open Xilinix Vivado.

HLx Editions		
Quick Start Create Project > Open Project > Open Scample Project >	Riccit Payers PCRIst Criticity reject.,1	
Tasks Manage IP > Open Hardware Marager > Xillink Td Store >		
Learning Center Documentation and Tutorisis > Quick Tabe Vdeos Release Notes Guide >		

- 3. Name the project.
- 4. Choose "RTL Project" and check the "Do not specify sources at this time" as we will configure all the settings manually through the navigator from inside the project.

of project to create.					
of project to create.					
ject					
be able to add sources, create l entation, design planning and a	block designs in IP Integra nalysis.	ator, generate IP,	run RTL analysis, s	synthesis,	
ot specify sources at this time					
thesis Project: You will be able entation.	to add sources, view devi	ice resources, ru	n design analysis, j	planning and	
ot specify sources at this time					
ning Project pecify design sources. You will	be able to view part/packa	age resources.			
l Project Vivado project from a Synplify, 3	XST or ISE Project File.				
Project					
new vivado project from a prec	defined template.				
	<	Back	Next >	Finish	Cancel
			Р	age 4 2	2
				0	
	be able to add sources, create antation, design planning and a ot specify sources at this time ithesis Project: You will be able entation. of specify sources at this time ning Project pecify design sources. You will d Project a Vivado project from a Synplify, Project a new Vivado project from a prec	be able to add sources, create block designs in IP Integrentation, design planning and analysis. ot specify sources at this time thesis Project: You will be able to add sources, view devi- entation. ot specify sources at this time ning Project pecify design sources. You will be able to view part/packa d Project a Vivado project from a Synplify, XST or ISE Project File. Project a new Vivado project from a predefined template.	be able to add sources, create block designs in IP Integrator, generate IP, antation, design planning and analysis. of specify sources at this time tithesis Project: You will be able to add sources, view device resources, run antation. of specify sources at this time ning Project pecify design sources. You will be able to view part/package resources. d Project a Vivado project from a Synplify, XST or ISE Project File. Project a new Vivado project from a predefined template. < <u>e Back</u>	be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, sentation, design planning and analysis. of specify sources at this time thesis Project. You will be able to add sources, view device resources, run design analysis, restantation. of specify sources at this time ning Project a Vivado project from a Synplify, XST or ISE Project File. Project a new Vivado project from a predefined template. (<back next=""></back>	be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, antation, design planning and analysis. of specify sources at this time thesis Project. You will be able to add sources, view device resources, run design analysis, planning and antation. of specify sources at this time ning Project specify design sources. You will be able to view part/package resources. d Project a Vivado project from a Synplify, XST or ISE Project File. > Project a new Vivado project from a predefined template. < <u>Back</u> Next> Einish P a g e 4 2

5. Select New Source... and the New window appears. In the New window, choose Schematic, type your file name (such as *source_1*) in the File Name editor box, click on OK, and then click on the Next button.

d Sources ecify HDL, netlist, BI on disk and add it to	ock Design, and If o your project. You	P files, or directo I can also add a	ories contair Ind create so	ning those files ources later.	, to add to your pro	oject. Create a new	source 🇼
+, - +	+						
	U	Jse Add Files, A	dd Directori	es or Create Fil	e buttons below		
		Add Files	Add Di	irectories	Create File	7	
Scan and add R	LL. FL <u>i</u> nclude files int	to project					
Copy <u>s</u> ources inf	o project n subdirectories						
Target language:	VHDL 🗸	Simulator lang	guage: VH	IDL 🗸			
0				< Back	Next >	Finish	Cancel
Create Source	e File			×			
Create Sourc Create a new s project.	e File ource file and	add it to you	r	×			
Create Sourc Create a new s project.	e File ource file and	add it to you	r	×			
Create Sourc Create a new s project. <u>F</u> ile type:	e File ource file and • VHDL	add it to you	r	×			
Create Sourc Create a new s project. <u>Fi</u> le type: F <u>i</u> le name:	e File ource file and VHDL source_1	add it to you	r	×			
Create Source Create a new s project. <u>Fi</u> le type: F <u>i</u> le name: Fil <u>e</u> location:	e File ource file and VHDL source_1 = <local td="" to<=""><td>add it to you Project></td><td></td><td>×</td><td></td><td></td><td></td></local>	add it to you Project>		×			
Create Source Create a new s project. <u>Fi</u> le type: F <u>i</u> le name: Fil <u>e</u> location:	e File ource file and VHDL source_1 < <local td="" to<=""><td>add it to you Project></td><td>r Cancel</td><td>×</td><td></td><td></td><td></td></local>	add it to you Project>	r Cancel	×			
Create Sourc Create a new s project. <u>File type:</u> F <u>ile name:</u> Fil <u>e</u> location: ?	e File ource file and VHDL source_1 << Local to	add it to you	r Cancel	×			
Create Source Create a new s project. <u>File type:</u> F <u>ile name:</u> Fil <u>e</u> location:	e File ource file and VHDL source_1 CH	add it to you	Cancel	×			
Create Source Create a new s project. File type: File name: Fil <u>e</u> location:	e File ource file and VHDL source_1 = <local td="" to<=""><td>add it to you</td><td>r Cancel</td><td></td><td></td><td></td><td></td></local>	add it to you	r Cancel				
Create Sourc Create a new s project. File type: File name: Fil <u>e</u> location:	e File ource file and VHDL source_1 <local p="" to<=""></local>	add it to you	r Cancel				

Add Source Specify HDL, n file on disk and	s etlist, Block Design, and I add it to your project. Yo	IP files, or directories			
file on disk and	add it to your project. Yo	IF mes, or directories	containing these files to a	dd te yeur preiest. Creete e p	
+, -		u can also add and c	reate sources later.	uu to your project. Create a m	wsource
•					
	Index Name 1 source	Library 1.vhd xil_defaultlib	HDL Source For Synthesis & Simulation	Location <local project="" to=""></local> 	
Scan and Copy sou	d add RTL include files in urces into project	Add Files	Add Directories	reate File	
Target lange	uage: VHDL ~	Simulator languag	e: VHDL 🗸		
			< Back	Next> Finish	
(?)			- Dack	<u> </u>	
(3)					
New Project Add Constr: Specify or crea +, =	aints (optional) te constraint files for phy	sical and timing const	traints.		
New Project Add Constr: Specify or crea + -	aints (optional) te constraint files for phy	sical and timing const	traints. s or Create File buttons bel	ow	
New Project Add Constru Specify or crea +, =	aints (optional) te constraint files for phy	sical and timing const Use Add File: Add F	raints. s or Create File buttons bel	ow	
Rew Project Add Constra Specify or creating +,	aints (optional) te constraint files for phy	sical and timing const Use Add File: Add F	s or Create File buttons bel	ow	
 Rew Project Add Constra Specify or creating +, Copy cor Copy cor 	aints (optional) te constraint files for phy	sical and timing const Use Add Files	s or Create File buttons bel	ow Next>	

Then Choose Finish.

Parts	Boards							
Reset All	Filters							
Category:	General Pu	rpose	~	Package:	cpg236	✓ Ter	mperature: I	~
Family:	Artix-7		~	Speed:	-1L	~		
Search:	Q∽ xc7a35t	icpg	8 ~	(1 match)				
Part		I/O Pin Count	Available IOBs	LUT Elemen	ts FlipFlops	Block RA	Ms Ultra RAMs	DSPs Gt
<								>
	ct			< <u>B</u> a	ick	lext >	Einish	Cancel
		 A new F 1 source No con: The def Default Product Family: Packag Speed 	RTL project name file will be added straints files will b ault part and prod Part: xc7a35ticpg : Artix-7 Artix-7 e: cpg236 Grade: -1L Srade: -1L	d'project will I. e added. Use uct family for 1 236-1L	be created. Add Sources to	add them	later.	
XIL	INX.	To create th						
XIL D	INX.	To create ti		< [ack	<u>N</u> ext >	<u>F</u> inish	Cancel

22

7. The Define Module Window that will appear, we will choose the input and output labels for the gates under investigation in this experiment.

each port spe SB and LSB v orts with blant	and specif cified: alues will c names w	y I/O Port be ignore ill not be	s to add i d unless written.	to your s a its Bus	le. n is checked.
lule Definitio	1				
<u>E</u> ntity name:	ex	_4			8
Architecture n	ame: Be	havioral			8
/O Port Defin	itions				
	Directio	Bus	MSB	LSB	
Port Name		×	0	0	
Port Name A	in				
Port Name A B	in in	~	0	0	
Port Name A B C	in in in	 <	0	0	
Port Name A B C X	in in in out	 <	0	0 0 0	

8. In the "source_1.vhd" created file, type the gates equivalent VHDL code for the XOR and XNOR gates between the "begin" and "end Behavioral" as follows and then save the file.

```
22
     library IEEE;
23
     use IEEE.STD LOGIC 1164.ALL;
24
25 \ominus -- Uncomment the following library declaration if using
26
     -- arithmetic functions with Signed or Unsigned values
27
     --use IEEE.NUMERIC_STD.ALL;
28
29
     -- Uncomment the following library declaration if instantiating
30
     -- any Xilinx leaf cells in this code.
31
     --library UNISIM;
32 -- use UNISIM.VComponents.all;
33
34 🖯 entity ex_4 is
      Port ( A : in STD_LOGIC;
35
36
                B : in STD LOGIC;
37
                C : in STD_LOGIC;
38
                X : out STD LOGIC;
                Y : out STD_LOGIC);
39
40 🔶 end ex_4;
41
42 😓 architecture Behavioral of ex_4 is
43
44
     begin
45
    X<= A xor B xor C;
46
     Y<= not (A XOR B XOR C);
47
48 😑 end Behavioral;
49
      <
                                                              Page 8 | 22
```

- 9. Next, we need to add To add a constraint file with the".xdc" extension, as following: Go to "Flow Navigator" and from "Project Manager" select "Add Sources" then "Add or create constraints". Next, choose "Create File" and enter the file name "lab_2" then "OK" followed by "Finish".
- 10. Then, we need to get a template xdc file that is going to be edited according to the different experiments. Google "basys 3 xdc file" and choose the "xilinix" link that appears (<u>https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/Basys3/Supporting%20Material/Basys3_Master.xdc</u>). Copy the whole file and paste it into the "lab_2.xdc" that you have just created in the last step. Then uncomment and edit the input Switches and the output LEDs as in the next step.



11. From the tool tab choose the play button and then "Run Implementation". Select "Number of jobs" =1 and then press OK.

12. The implementation errors window will appear if any or the successfully completed window. From this window select "Generate Bitstream" and then OK. This will make the software generate ".bin" file to be used in programing the hardware BAYAS 3.

Implementation Completed	\times
Implementation successfully completed.	
Open Implemented Design	
Generate Bitstream	
◯ <u>V</u> iew Reports	
Don't show this dialog again	
OK Cancel	

13. The next window will appear in which choose "Open Hardware Manger", connect the Hardware Kit to the USB port and then press OK.



- 14. A green tab will appear in the top of the Vivado window, from which choose "open target" to program the hardware.
- 15. From the window appears, select the ".bin" file from the Project you create by browsing for the generated ".bit file" under the ".runs" folder and program the board then press OK.
- 16. Fill in the following truth tables for all the gates by observing the inputs/outputs on the programmed board.

A. XOR Gate

Truth Table (1)

А	В	С	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Symbol

Boolean Equation

Page 11 | 22

B. XNOR Gate

Truth Table (2)

Α	В	С	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Symbol

Boolean Equation

17. Verify that the experimental results are consistent with the Discussion.

Checked by_____ Date _____

Page 12 | 22

Section 2 XOR gates used in a comparator:

- 1. Repeat section 1 from step 1 to 6.
- 2. The Define Module Window that will appear, we will choose the input and output labels for the gates under investigation in this experiment.

Define Module Define a module and specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.													
Mod	ule Definition												
Į	Entity name: ex												
,	A <u>r</u> chitecture name: Behavioral												
IJ	I/O Port Definitions												
	+ -	↑ ↓											
	Port Name	Direction	Bus	MSB	LSB								
	Α	in 💊	-	3	0								
	B in → ✓ 3 0												
	Х	out 🗠		0	0								
?								ОК	Cancel				

3. In the "source_1.vhd" created file, type the gates equivalent VHDL code for the XOR and XNOR gates between the "begin" and "end Behavioral" as follows and then save the file.

```
library IEEE:
  use IEEE.STD LOGIC 1164.ALL;

    -- Uncomment the following library declaration if using

    arithmetic functions with Signed or Unsigned values

   --use IEEE.NUMERIC STD.ALL;
   -- Uncomment the following library declaration if instantiating
   -- any Xilinx leaf cells in this code.
   --library UNISIM:

— --use UNISIM.VComponents.all;

😑 entity ex is
       Port ( A : in STD LOGIC VECTOR (3 downto 0);
              B : in STD_LOGIC_VECTOR (3 downto 0);
              X : out STD_LOGIC);
end ex;
architecture Behavioral of ex is
  begin
   X <= NOT ((A(0) XOR B(0)) OR (A(1) XOR B(1)) OR (A(2) XOR B(2))OR (A(3) XOR B(3)) );
end Behavioral;
```

4. Next, we need to add To add a constraint file with the".xdc" extension, as following: Go to "Flow Navigator" and from "Project Manager" select "Add Sources" then "Add or create constraints". Next, choose "Create File" and enter the file name "lab_2" then "OK" followed by "Finish".

```
11 ## Switches
12 set property PACKAGE_PIN V17 [get_ports {A[0]}]
      set property IOSTANDARD LVCMOS33 [get ports {A[0]}]
13
14 set property PACKAGE PIN V16 [get ports {A[1]}]
15
      set property IOSTANDARD LVCMOS33 [get ports {A[1]}]
16 set property PACKAGE_PIN W16 [get ports {A[2]}]
      set property IOSTANDARD LVCMOS33 [get ports {A[2]}]
17
18 set property PACKAGE_PIN W17 [get ports {A[3]}]
19
      set property IOSTANDARD LVCMOS33 [get ports {A[3]}]
20 set property PACKAGE_PIN W15 [get ports {B[0]}]
      set property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
21
22 set property PACKAGE_PIN V15 [get ports {B[1]}]
23
      set property IOSTANDARD LVCMOS33 [get ports {B[1]}]
24 set property PACKAGE_PIN W14 [get ports {B[2]}]
      set property IOSTANDARD LVCMOS33 [get ports {B[2]}]
25
26 set property PACKAGE_PIN W13 [get ports {B[3]}]
      set property IOSTANDARD LVCMOS33 [get ports {B[3]}]
27
```

```
46 ## LEDs
47 set_property PACKAGE_PIN U16 [get_ports {X}]
48 set_property IOSTANDARD LVCMOS33 [get_ports {X}]
```

```
Page 14 | 22
```

- 5. Then, we need to get a template xdc file that is going to be edited according to the different experiments. Google "basys 3 xdc file" and choose the "xilinix" link that appears (https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/Basys3/Supporting%20Material/Basys3_Master.xdc). Copy the whole file and paste it into the "lab_2.xdc" that you have just created in the last step. Then uncomment and edit the input Switches and the output LEDs as in the next step.
- 6. From the tool tab choose the play button and then "Run Implementation". Select "Number of jobs" =1 and then press OK.

À Launch Runs X	4
Launch the selected synthesis or implementation runs.	
Options	
 <u>L</u>aunch runs on local host: Number of jobs: 1 <u>G</u>enerate scripts only 	
Don't show this dialog again	
OK Cancel	

7. The implementation errors window will appear if any or the successfully completed window. From this window select "Generate Bitstream" and then OK. This will make the software generate ".bin" file to be used in programing the hardware BAYAS 3.

plementation Completed								
Implementation successfully completed.								
Open Implemented Design								
 <u>Generate Bitstream</u> <u>V</u>iew Reports 								
Don't show this dialog again								
OK Cancel								

8. The next window will appear in which choose "Open Hardware Manger", connect the Hardware Kit to the USB port and then press OK.

Bitstream Generation Completed									
Bitstream Generation successfully completed.									
O Open Implemented Design									
◯ <u>V</u> iew Reports									
• Open <u>H</u> ardware Manager									
O Generate Memory Configuration File									
Don't show this dialog again									
OK Cancel									

9. A green tab will appear in the top of the Vivado window, from which choose "open target" to program the hardware.

- 10. From the window appears, select the ".bin" file from the Project you create by browsing for the generated ".bit file" under the ".runs" folder and program the board then press OK.
- 11. Fill in the following truth tables for all the gates by observing the inputs/outputs on the programmed board.

	Wo	rd A			Output							
A0	Al	A2	A3	B0	B1	B2	B3	X				
0	1	1	0	1	0	0	1					
1	0	1	0	1	0	1	0					
0	0	1	1	1	1	0	0					
0	1	0	0	0	1	0	1					
1	1	0	1	1	1	0	1					
0	0	0	0	0	0	0	0					
0	1	1	1	0	1	1	1					
1	1	1	1	1	1	1	0					
1	0	1	1	0	0	1	1					
	Раде 17 2 ′											

Truth Table

12. Summarize the results on your own words.

Checked by_____ Date _____ Date ______ Date ______ Date _____ Date

If you examine the truth table of an XNOR gate carefully, you will notice an interesting fact: when input A is held Low, the output is the complement of input B. When input A is kept High, the output follows input B. This effect means that the XNOR gate can be used to construct a buffer/inverter circuit. What we have to do is use one of the inputs as the control signal and the other input as the data signal. The XNOR will act like a buffer when the control signal is high, but as an inverter when the control signal is pulled Low. Here we will build a 4-bit buffer/inverter circuit and then run a simulation to verify the result.

1. Repeat section 1 from step 1 to 6.

Page 18 | 22

2. The Define Module Window that will appear, we will choose the input and output labels for the gates under investigation in this experiment.

Define Module And Specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.											
Module Definition											
Entity name: sec_3											
Architecture name: Behavioral											
I/O Port Defini	itions										
+ -	t	+									
Port Name	Dire	ction	Bus	MSB	LSB						
D	in	~	\checkmark	3	0						
Control	in	~		0	0						
X	out	~	\checkmark	3	0						
?						OK Cancel					

3. In the "source_1.vhd" created file, type the gates equivalent VHDL code for the gates between the "begin" and "end Behavioral" as follows and then save the file.

```
22 library IEEE;
23
    use IEEE.STD LOGIC 1164.ALL;
24
25 Θ -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27
    --use IEEE.NUMERIC STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 !
     -- any Xilinx leaf cells in this code.
31 -- library UNISIM;
32 -- use UNISIM.VComponents.all;
33
34 🖯 entity sec_3 is
35
         Port ( D : in STD LOGIC VECTOR (3 downto 0);
36
                Control : in STD LOGIC;
37
                X : out STD LOGIC VECTOR (3 downto 0));
38 - end sec 3;
39
40 - architecture Behavioral of sec_3 is
41
42 !
    begin
43 X(0) <= Control XNOR D(0);</p>
44 ; X(1) <= Control XNOR D(1);
45 X(2) <= Control XNOR D(2);</p>
46 : X(3) <= Control XNOR D(3);</pre>
47 \bigcirc end Behavioral;
48
```

- 4. Next, we need to add To add a constraint file with the".xdc" extension, as following: Go to "Flow Navigator" and from "Project Manager" select "Add Sources" then "Add or create constraints". Next, choose "Create File" and enter the file name "lab_2" then "OK" followed by "Finish".
- 5. Repeat section 1 from step 10 to 15.
- 6. Fill in the following truth tables for all the gates by observing the inputs/outputs on the programmed board.

Truth Table

	T.e.e			Outputs								
	mp	outs			Cont	rol=0		Control=1				
D0	D1	D2	D3	X 0	X1	X2	X3	X0	X1	X2	X3	

7. Summarize the results on your own words.

Checked by_____ Date _____

Questions:

1.) A 3-input XOR gate is equivalent to the circuit shown below: ABCX



The Boolean equation can be written as:

 $\mathbf{X} = (\mathbf{A}' \mathbf{B} + \mathbf{A}\mathbf{B}')'\mathbf{C} + (\mathbf{A}'\mathbf{B} + \mathbf{A}\mathbf{B}')\mathbf{C}'$

Or it simply denoted as:

$$X = A \oplus B \oplus C$$

Using only AND, OR and inverter gates to implement the above Boolean equation, how many gates are needed? Draw the logic diagram. Compare the savings of a single XOR gate implementation with the circuit you just drew.

2.) How can you use a 2-input XOR gate to function as a 1-bit buffer/inverter? Draw the logic diagram. Show the logic connections for the control and data input lines.