## **Chapter-5**

## **Programming the FPGA**

In previous chapters, several digital circuits are implemented by using VHDL. In chapter 4, functional tests of the implemented designs are done. In this manner, next step is to upload our program to FPGA. FPGA coding and programming tools differs in between FPGA families of Xilinx, Altera, Actel, etc. In this lecture, programming via Xilinx's ISE Design Suite is going to be demonstrated for the explanation of the process.



## Figure 5-1

Figure 5-1 shows the Nexys 4 DDR programming kit and focuses on the FPGA chip. This IC belongs to the Xilinx's Artix 7 family with device code XC7A100T. This FPGA chip is going to be used during rest of the chapter.

**Example:** VHDL code of Boolean function f(x, y, z) = x'y' + y'z is given below PS 5.1. Open a new project. Create its bit file. Upload it to Nexys 4 DDR development kit and see the results.

```
library ieee;
use ieee.std_logic_1164.all;
entity f_xyz is
port( x, y, z: in std_logic;
    f: out std_logic );
end entity;
architecture logic_flow of f_xyz is
begin
    f<= (x nor y) or (not y and z);
end architecture;
```

PS 5.1

**Solution:** Step-1) As the first step to achive our task, we need to open a new project. After setting file our project file, ISE Design Suite asks for the FPGA chip family, device, package and speed properties of the chosen FPGA. These properties can be achived simply by reading the top side of the FPGA chip. In our case, it can be read from the Figure 5-1. Figure 5-2 shows the overall assignment.

elect the device and design flow for the p	roject	
Property Name	Value	
Evaluation Development Board	None Specified	~
Product Category	All	~
Family	Artix7	~
Device	XC7A100T	~
Package	CSG324	~
Speed	-1	¥
Top-Level Source Type	HDL	V
Synthesis Tool	XST (VHDL/Verilog)	~
Simulator	ISim (VHDL/Verilog)	~
Preferred Language	VHDL	~
Property Specification in Project File	Store all values	~
Manual Compile Order		
VHDL Source Analysis Standard	VHDL-200X	v
Frankla Marrie - Filtraine		
Enable Message Filtering		

Figure 5-2

Step-2) After configuration, click next. An empty project with XC7A100T is created. VHDL file and its pin assignment should be added to the project. By right clicking the FPGA chip on the project screen open a VHDL Module and name it "f\_xyz".

New So	urce Wizard ×
Select Source Type Select source type, file name and its location.	
<ul> <li>IP (CORE Generator &amp; Architecture Wizard)</li> <li>Schematic</li> <li>User Document</li> <li>Verilog Test Fixture</li> <li>VHDL Module</li> <li>VHDL Module</li> <li>VHDL Vackage</li> <li>VHDL Fast Bench</li> <li>Embedded Processor</li> </ul>	Elle name: f_xyz Logation: ::\Users\cagri_000\Desktop\\HDL Examples\f_xyz
More Info	Next > Cancel

Figure 5-3

After typing project name click next. Enter **architecture** name and I/O port names as seen in the Figure 5-4. Click next.

Specify ports	s for module.						
Entity name	f_xyz						
Architecture name	logic_flow						
	Port Name	Direct	ion	Bus	MSB	LSB	^
x		in	~				
У		in	~				
z		in	~				
f		out	~				
		in	~				
		in	~				
		in	~				
		in	~				
		in	~				
		in	~				
		in	~				~

Figure 5-4

Step-3) Our function is implemented. Now we have to tie FPGA pins to our variables x, y, z, f. Nexys 4 DDR development kit has basic electronic components to realize input signal possibilities and observe results. Among these components, we are going to use switches as inputs and an LED as output. As a result, we are going to use three switches (SW0, SW1, SW2) and an LED (LED0). Pin information about Nexys 4 DDR FPGA board can be found from the internet easily. After choosing our peripherals from Nexys 4 DDR FPGA board reference manual, now we can define them in our ISE project.

New Sc	ource Wizard ×
Select Source Type Select source type, file name and its location.	
ChipScope Definition and Connection File     ChipScope Definition and Connection File     If (CORE Generator & Architecture Wizard)     MEM File     Schematic     User Document     Verilog Module     Verilog Test Fixture     VHDL Nodule     VHDL Library     VHDL Package     VHDL Package     Embedded Processor	Elle name: [yyz_pins Logation: C:\Users\cagri_000\Desktop\\HDL Examples\f_xyz
More Info	Add to project Next > Cancel

Figure 5-5

Step-4) By right clicking the FPGA chip on the project screen open a "Implementation Constraints File" and name it "f\_xyz\_pins". Click Next.

Step-5) An empty file with the extension of "\*.ucf". Connect your one bit I/O variables to SW0, SW1, SW2 and LED0 as shown below in Figure 5-6. It is also important to state that "f\_xyz\_pins.ucf" file is in the sub directory of "f\_xyz.vhd" file on the Hierarchy window of the design. It means constraints file is created correctly.

Eile Edit View Project Source Process	<u>T</u> ools <u>W</u> indow La <u>v</u> out <u>H</u> elp	. <i>8</i> ×
o or 🗙 🗊 🕼 🐇 📲 📲 📢 🗋	» /P /P /B /P 🗟 💫 😤 🗄 🗖 📼 🥬 餐 🕨 🗵 📌	
Design ↔ □ ₽ × 4	1 Net "x" LOC = J15   IOSTANDARD = LVCMOS33;	
View: 🖲 🄯 Implementation 🗌 🔝 Simulation 🕞	2 Net "y" LOC = L16   IOSTANDARD = LVCMOS33;	
Je Hierarchy	3 Net "z" LOC = M13   IOSTANDARD = LVCMOS33;	
👔 🖳 😇 f_xyz		
📥 🖾 xc7a100t-1csg324	5 Net "I" LOC - HI7   IOSTANDARD - LVCMOS33;	
in the first		
2		
2	۶	
7	۶	
···· · > >	5	
No Processes Running		
Processes: f xyz pins.ucf		
User Constraints	—	
<u></u>		
	4	>
> Start 🖳 Design 🖺 Files 🚺 I 4 🕨	f_xyz.vhd 🗵   🧏 Design Summary 🗵 📄 f_xyz_pins.ucf* 🔯	

Figure 5-6

Step-6) We are ready to generate our "\*.bit" file. Select your main file "f\_xyz.vhd" and double click to *Generate Programming File* button as seen in Figure 5-7. By clicking this button

translating, mapping and routing stages start to work. If editor find no mistakes then green check mark emerges (Figure 5-8) and it means our "\*.bit" file is generated inside the project file.



Figure 5-7

Figure 5-8

Step-7) In this step, turn is to upload our bit file into the FPGA. Connect your FPGA development kit to your computer. If it is the first time that your board is connected to your computer, be sure its USB driver is installed. Generally, operating systems install it automatically. Open *Manage Configuration Project (iMPACT)* from the *Configure Target Device* menu as given in Figure 5-9.



Figure 5-9

Step-8) An empty project screen will open as it seen in Figure 5-10. Double click *Boundary Scan* to add connected FPGA devices to configuration project.

<b>B</b>					ISE iN	MPACT (P	.28xd)			-		×
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	Operations	<u>O</u> utput	Debug	Window	<u>H</u> elp					
	1	**	6 🖬 🖌	₽ №?								
iMPAC	T Flows	;		⇔□₽	×							
	Bou Syst Crea Web	ndary S emACE ate PRO Talk Da	can M File (PROM ta	File Forma	t							
iMPAC	T Proce	esses		⇔□₽	x							
								,,,,				
Conso	le										↔	⊐ ₽ ×
												0
<												>
	Console	2	Errors	Warnings								
										Γ	ΓΓ	ΓΓ

Figure 5-10

After Boundary Scan right side of the project window becomes ready to add FPGA chip.

😵 ISE iMPACT (P.28xd) - [Boundary Scan] – 🗖 💌						
🛞 File Edit View Operations Output	Debug Window Help	_ & ×				
🗋 🆻 🖥 🛍 🛱 🗟 🖬	₩ K?					
iMPACT Flows ↔ □ 문 ×						
Boundary Scan     SystemACE     A Create PROM File (PROM File Format     WebTalk Data						
	Add Xilinx Device Ctrl+D					
MDACT Processes 슈미국 X	Right clie Add Non-Xilinx Device Ctrl+K					
Available Operations are:	Initialize Chain Ctrl+I					
	Cable Auto Connect Cable Setup					
	Output File Type	•				
	Boundary Scan					
Console		⇔⊡∂×				
		^				
<		>				
Console 😰 Errors 📣 Warnings						
	No Cable Connection No File O	pen 🗍 🗍 🔡				

Figure 5-11

Step-9) Right click on the new-opened area and choose *Add Xilinx Device* (Figure 5-11). When clicked, window in the Figure 5-12 appears.

₽	Assign New Con	figuration File	×
🔄 🏵 🗉 🕇 🌗	▹ VHDL Examples → f_xyz →	✓ C Ara: f_xyz	م ر
Düzenle 🔻 Yer	ni klasör		= 🖬 🔞
CneDrive Crubu Ev Grubu Bu bilgisayar Belgeler Downloads Masaüstü Müzikler Resimler Videolar Lownlows (C:)	Ad Ad Ingo Ingo Indextored in the image of the image	Değiştirme tarihi 29.10.2017 11:08 29.10.2017 11:09 29.10.2017 10:58 29.10.2017 11:00 29.10.2017 11:08 29.10.2017 11:08 29.10.2017 11:10	Tür Dosya klasörü Dosya klasörü Dosya klasörü Dosya klasörü Dosya klasörü Dosya klasörü BIT Dosyası
	v < D <u>o</u> sya adı: f_xyz.bit	✓ All Design Files (* Aç	i.jed *.bit *.rbt ∨

Figure 5-12

Choose the bit file that we created previously during the *Generate Programming File* process and click Open. When it's opened FPGA chip is found; check its name and you can see that it's XC7A100T as we set before.

🚱 ISE iMI	PACT (P.28xd) - [Bou	undary Scan]	_ 🗆 🗙
File Edit View Operations Output	Debug Window H	Help	- 8 ×
Image: Second		₽ <b>\?</b>	
SystemACE     Greate PROM File (PROM File Format     WebTalk Data		) Program	
MPACT Processes ↔	xc7a1 f_xyz	Access eFUSE Registers Get Device ID Get Device Signature/Usercode	BPT L Examples/f_xyz
Available Operations are:		One Step SVF One Step XSVF	
		Add SPI/BPI Flash Assign New Configuration File Set Programming Properties	
	Boundar	Set Erase Properties	_
Console		Launch File Assignment Wizard Set Target Device	↔ □ ₽ ×
Console 🔞 Errors 📣 Warnings			>
		No Cable Connection No File	Open   [   [   [  :

Figure 5-13

Step-10) Right click to the green FPGA chip and **Program** it (Figure 5-13). When it's done the indicator "Program Succeeded" should be seen, otherwise check your USB driver status.

🚱 ISE iMP.	ACT (P.28xd) - [Boundary Scan] – C	X
🛞 <u>F</u> ile <u>E</u> dit <u>V</u> iew O <u>p</u> erations <u>O</u> utput	Debug <u>W</u> indow <u>H</u> elp	- 8 ×
🗋 🌶 🗐 🐰 🛍 🛍 🕹 🖬 🖬	# 🔁 II 🥕 🛠	
iMPACT Flows ↔ □ 문 ×		
Boundary Scan     SystemACE     Greate PROM File (PROM File Format     WebTalk Data		
iMPACT Processes ↔ □	f_xyz.bit	
Available Operations are: Program Get Device ID Get Device Signature/Usercode Read Device Status One Step SVF One Step XSVF	Program Succeeded	
	🛞 Boundary Scan	
Console		⇔⊡₽×
'1': Programmed successfully.		^ ~
Console C Errors A Warnings		,
	Configuration Digilent Nexys4DDR 1000000	

Figure 5-14

Step-11) This is the last step of the process. We successfully upload our code to FPGA. Now we should check the results whether they are true or not. In this case, we need to look at truth table of the given function at first. Truth table of the f(x, y, z) given below in Table 5-1.

X	У	Z	f		
0	0	0	1		
0	0	1	1		
0	1	0	0		
0	1	1	0		
1	0	0	0		
1	0	1	1		
1	1	0	0		
1	1	1	0		
Tabla 5 1					

Figure 5-15 shows the result of one possible combination of inputs, x = 1', y = 0', z = 1'. As it's expected the result equals f equals logic '1' such that, LED is on. It's also important to state that sliding switches in + y direction means input is logic '1', sliding switches in -y direction makes any input logic '0'.



Figure 5-15

**Example:** Implement the 2-to-1 multiplexer which is depicted in Figure 5-16. Inputs and output are should be two bit wide. Only the select bit of the multiplexer is one bit. Create "\*.bit" file, upload it to Nexys 4 DDR development kit and observe the results.



Figure 5-16

Solution: Below program segment PS 5.2 is the VHDL solution of this example.

library ieee; use ieee.std\_logic\_1164.all; entity multiplexer\_2x1 is port(I0, I1: in std\_logic\_vector (1 downto 0); S0: in std\_logic; F: out std\_logic\_vector(1 downto 0)); end entity; architecture logic\_flow of multiplexer\_2x1 is begin f<=x when s0='0' else y; end architecture;

PS 5.2

Follow the steps (1-3) that are shown in the previous example. At this time, constraints file a bit different since, some of our inputs and our single output are two bits wide. Constraints should be written as shown in Figure 5-17.



Figure 5-17

Constraints file is set, so step 4 and 5 are completed. When rest of the steps (6-11) are done, bit file of the multiplexer is created and uploaded into the development kit. In Figure 5-18, select bit of the multiplexer is set to '0' so, "01" is observed at the output which is equal to  $I_0$ .



Figure 5-18

Figure 5-19 shows the results if  $S_0$  becomes logic '1'. In this time, since  $I_1$  is set as "11" output *F* equals "11". As a result, we covered all the states of the truth table.



Figure 5-19

**Exercise:** Design a simple 3-to-8 Decoder whose structure and truth table are given below in Figure 5-20. Create "\*.bit" file, upload it to Nexys 4 DDR development kit and observe the results.



Figure 5-20