<u>NEXYS4DRR board tutorial</u>

(VHDL Decoder design using Vivado 2015.1)

Note: you will need the Xilinx Vivado Webpack version installed on your computer (or you can use the department systems).

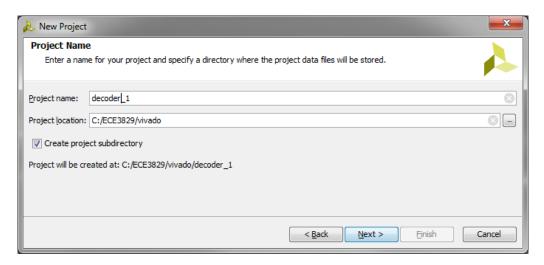
This tutorial shows how to create a simple combinational design (a 3 to 8 decoder using the slider switches and LEDs) that can be implemented on the Nexys4DDR board.

Start Vivado Design Suite:



Select Create New Project.

Click Next and then enter a Project name and location for your project:



1

Click Next and select the RTL project type:

🚴 New Project	x
Project Type Specify the type of project to create.	
 RTL Project You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis. Do not specify sources at this time Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation Do not specify sources at this time I/O Planning Project 	
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel	

Check the "Do not specify courses at this time" box and click Next:

Select the corect Xilinx FPGA that is on the Nexys4DDR board (XC7A100T-1CSG324C):

Choose a default Xil	inx part or bo	oard for your pro	ject. This can be	e changed later.					
Select: 💊 Parts 📓	Boards								
Product category:	All		•	Package:	csg324		-		
Eamily:	Artix-7		-	Spee <u>d</u> grade:	-1		-		
Sub-Family:	Artix-7		+	Temp grade:	С		-		
				Si Revision:	All Remain	ning	•		
			R	eset All Filters					
				Loci An Thiters					
<u>S</u> earch: Q,-									
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	DSPs	Gb Transceivers	GTPE2 Transceive	
i ui c		210	10400	20800	25	45	0	0	
xc7a15tcsg324-1	324	210					0	0	
	324 324	210	20800	41600	50	90	0	-	
xc7a15tcsg324-1			20800 32600	41600 65200	50 75	90 120	0	0	
xc7a15tcsg324-1 xc7a35tcsg324-1	324	210							
 xc7a15tcsg324-1 xc7a35tcsg324-1 xc7a50tcsg324-1 	324 324	210 210	32600	65200	75	120	0	0	
 xc7a15tcsg324-1 xc7a35tcsg324-1 xc7a50tcsg324-1 xc7a7a50tcsg324-1 xc7a75tcsg324-1 	324 324 324	210 210 210	32600 47200	65200 94400	75 105	120 180	0 0	0 0	

Click Next, and then Finish:

🚴 New Project	X
	New Project Summary
VIVADO.	(i) A new RTL project named 'decoder_1' will be created.
	 The default part and product family for the new project: Default Part: xc7a100tcsg324-1 Product: Artix-7 Family: Artix-7 Package: csg324 Speed Grade: -1
E XILINX	To create the project, dick Finish
	< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel

The Project window opens:

& decoder_1 - [C:/ECE3829/vivado/d	ecoler_L/decoder_Lxpi - Vivsdo 2015.1
File Edit Flow Tools Window I	Agout Vew Help
🏄 😂 in 🕫 🏬 🐘 🗙 🔈	🕨 🚵 🚳 🛞 🔼 🕼 📴 Default Layout 🔷 💘 🚸 🍾 🕼
Flow Navigator	Project Manager - decoder_1 X
🔍 🖾 🖨	Sources L × ∑ Project Summary × L ×
Project Manager	🔍 🛣 🚔 📸 📓 🚺 🔛 🔀 Project Settings Edit
Project Settings Add Sources Language Templates F: P Catalo	Project name: decoder_1 Project location: C/CES323/Wrado/decoder_1 Project location: C/CES323/Wrado/decoder_1 Project part: Artis-7 Project part: xr2/35/too226-1
▲ IP Integrator	Top module name: <u>Not defined</u>
📅 Create Block Design 📑 Open Block Design 🍓 Generate Block Design	Function Synthesis Implementation \$ Hierarchy Libraries Comple Order Status: Not started Status: Not started Hierarchy Libraries Comple Order Part: xc7a3Stop226-1 Part: xc7a3Stop226-1
 Simulation Simulation Settings Run Simulation 	A Sources () Templates Strategy: <u>Wusdo Synthesis Defaults</u> Properties X A Sources () Templates X
RTI Analysis G Elaboration Settings B Open Elaborated Design	DRC Violations Timing A Bun Imdementation to see DRC results Bun Imdementation to see timing results Bun Imdementation to see timing results
 Synthesis Synthesis Settings Run Synthesis 	Select an object to see properties Utilization Power Powe
Open Synthesized Design	
Implementation Implementation Settings Run Implementation	Designature Constraints Status Progress WINS THS WHS THS THS Faled Routes LUT % LUTs FF % FFs BRAM % DSP % DSP % Start Constraints Status Progress UINS THS WHS THS THUS Faled Routes LUT % LUTs FF % FF a BRAM % DSP % DSP % Start
Program and Debug Program and Debug Bistream Settings Generate Bitstream P	Impl_1 constrs_1 Not started

We now need to add a VHDLg design source to describe our decoder operation.

Click on Add Sources in the left Project Manager window (or select the menu item File => Add Sources:

Add Sources	×	J
VIVADO.	Add Sources This guides you through the process of adding and creating sources for your project	
	 Add or greate constraints Add or create design sources Add or create gimulation sources Add or create DSP sources Add existing block design sources 	
	○ Add existing IP To continue, dick Next < Back Next > Einish Cancel	

Select Next,

And then select Create File (click on the + symbol) and enter decoder for the file name:

Make sure the file type is set to VHDL

🚴 Create So	urce File	x
Create a new	source file and add it to your project.	~
<u>F</u> ile type:	I VHDL	-
File name:	decoder	Θ
File location:	🛜 <local project="" to=""></local>	-
	ОК	ancel

Then click OK and Finish.

We can now specify the inputs and outputs to create our 3 to 8 decoder (we will use three switches and eight LEDs):

De Fo	efine a n or each p MSB an	oort specified: d LSB values w	cify I/O Ports to add t ill be ignored unless its s will not be written.			cked.		×
Mo	odule De	finition						
	<u>E</u> ntity	name:	decoder					0
	A <u>r</u> chit	tecture name:	Behavioral					8
	I/O P	ort Definitions						
	+	Port Name	Direction	Bus	MSB	LSB		
	-	SW	in ·	-	2	0		
	1	LED	out	-	7	0		
	+							
							OK	Cancel

Click OK.

Back in the Project Manager Sources window double-click the new decoder.vhd file and you will then see the VHDL file appear in the window on the right:

& decoder_1 - [C:/ECE574/decoder_1	1/decoder 1.xpr] - Vivado 2015.1	X
File Edit Flow Tools Window I	Layout Yew Help Q- Search com	mands
🏄 😂 in 🕫 🗎 🐂 🗙 🔈	🕨 🔪 输 🛞 🔀 📴 DefaultLayout 🚽 🗶 🔖 🐛	Ready
Flow Navigator «	Project Manager - decoder_1	×
९ 🔀 🛱	Sources	۵¢×
	🔍 🕱 🖨 🖻 🐮	
Project Manager Project Settings	Design Sources (1)	*
Add Sources	l → a filecoder - Behavioral (lecoder / hd) 22 library IEEE;	
Canquage Templates	Ginulation Sources (1) 23 use IEEE.STD_LOGIC_1164.ALL;	
IP Catalog	⊕ @ sm_1(1) 24 25 Uncomment the following library declaration if using	
	26 arithmetic functions with Signed or Unsigned values	
IP Integrator Create Block Design	28 Z27 use IEEE. NUMERIC_SID. ALL;	
Open Block Design	29 Uncomment the following library declaration if instantiating	
Generate Block Design	30 any Xilinx leaf cells in this code.	
	Hierarchy Libraries Comple Order 31 Library UNISIN; 32 use UNISIN, VComponents, all;	
 Simulation 	<u>& Sources</u> ♥ Templates ♥ 33	
6 Simulation Settings	Source File Properties 2 × 🖤 34 entity decoder is	
🔍 Run Simulation	4 → 35 Port (sw : in STD LOGIC VECTOR (2 downto 0); 36 led : out STD LOGIC VECTOR (7 downto 0));	
A RTL Analysis	(i) decoder.vhd 37 end decoder;	
🔞 Elaboration Settings	Location: C:/ECE574/decoder_1/decoder_1.srcs/sources_1/mew 38 39 architecture Behavioral of decoder is	E
Open Elaborated Design	Type: VHDL = 40	
4 Synthesis	Ubrary: xi_defaultib m 42	
🚳 Synthesis Settings	Size: 1.0KB 43	
Run Synthesis	Modified: Today at 13:32:01PM 44 end Behavioral;	
Open Synthesized Design	Copied to: C:/ECES74/decoder 1/decoder 1.srcs/sources 1/new + 45	-
4 Implementation	General Properties 4	F
Maintain Settings	Design Runs	- 🗆 🖻 ×
Run Implementation	Q Name Constraints Status Progress WNS TNS WHS THS TPWS Faled Routes LUT % LUTs FF % FFs BRAM %	BRAMs DSP %
Open Implemented Design	□→ synth_1 constrs_1 Not started 0% □→ synth_1 constrs_1 Not started 0%	
Program and Debug		
Bitstream Settings	*	
Cenerate Bitstream		
Den Hardware Manager		
	IT Console O Messages D Log C Reports D Design Runs	F
	Carlose Correctione Correction Co	

You should add your name and a description of this file to the header description.

We can now add the verilog statements to design our 3 to 8 decoder.

There are a number of ways to design the decoder in VHDL, below is an example using a conditional signal assignment statement:

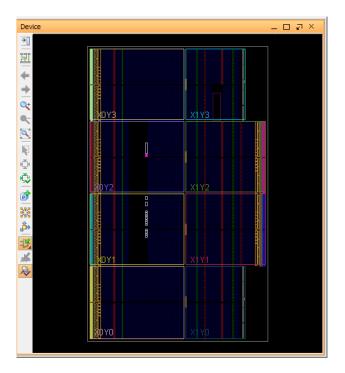
```
decoder.vhd *
                                                              _ 🗆 🖉 ×
C:/ECE574/decoder_1/decoder_1.srcs/sources_1/new/decoder.vhd
   34 entity decoder is
                                                                        ÷
6
   35
         Port ( SW : in STD LOGIC VECTOR (2 downto 0);
C 36
                 LED : out STD LOGIC VECTOR (7 downto 0));
37 end decoder;
   38
Ð
   39 architecture Behavioral of decoder is
40
🗙 41 begin
   42
//
   43
         -- using conditional signal assignment statement
44
         -- also consider using a case statement
   45
         LED <= "00000001" when SW = "000" else
æ
             "00000010" when SW = "001" else
   46
o
              "00000100" when SW = "010" else
   47
              "00001000" when SW = "011" else
   48
1 Ste
   49
              "00010000" when SW = "100" else
                                                                        Ξ
   50
              "00100000" when SW = "101" else
   51
              "01000000" when SW = "110" else
   52
              "10000000";
   53
   54 end Behavioral;
      .
                               Ш
```

Now we can synthesize the design.

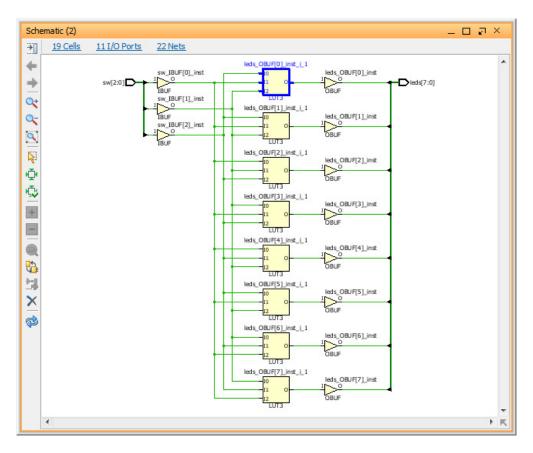
Click Run Synthesis in the Project Manager window.

After synthesis is complete there should be no errors or warnings reported (look at the Porject Summary window).

If you click on Open Synthesized Design you can see a device level representation (this is mostly empty since we just have a very simple design that only uses a small fraction of the available FPGA resources):



But you can also look at a schematic representation to see the input and output buffers and the LUTs used (click on Schematic under the Synthesized Design):



Before we can implement the design we need to specify the FPGA pins that will be used for the SW inputs and LED outputs.

Look at the Nexys4DDR manual to determine the FPGA pins.

Next click Add Sources and select 'Add or create constraints':

Add Sources	
VIVADO.	Add Sources This guides you through the process of adding and creating sources for your project
	 Add or greate constraints Add or create design sources Add or create gimulation sources Add or create DSP sources
E XILINX All programmable.	 Add existing <u>b</u>lock design sources Add <u>e</u>xisting IP To continue, dick Next
	< <u>B</u> ack <u>Next</u> Finish Cancel

We can name the constraints file decoder:

🚴 Create Co	nstraints File	×
Create a new project	constraints file and add it to your	4
<u>F</u> ile type:	TXDC	•
File name:	decoder	8
Fil <u>e</u> location:	🛜 <local project="" to=""></local>	-
	ОК	Cancel

In the Souces window select the constraints file by the hierarchy Constraints => constrs_1 => decoder.xdc. We can then add location constraints for all the inputs and outputs (you can download a copy of the Nexys4DDR XDC constraints from the Digilent website – just copy the pins you are using for the design):

These constraints specify the pins to use for each signal and what type of interface.

	oder.xdc * 📃 🗆 🖉	×
	C: JECE574/decoder_1/scs/constrs_1/new/decoder.xdc	
	1 #Switches 2	^
% ⊡	3 set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sv[0] 4 set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sv[1] 5 set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sv[2] 6	
	7 # LEDS 8	
	9 set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0] 10 set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]	
	11 set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #I0_L17N_T2_A25_15 Sch=led[2] 12 set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #I0_L8P_T1_D11_14 Sch=led[3] 13 set property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get ports { LED[4] }]; #I0 L7P T1 D09 14 Sch=led[4]	
	13 set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4] 14 set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5] 15 set property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L17P_T2_A14_D30 14 Sch=led[6]	
1000	16 set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7] 17	
	4	*

Now we have specified the correct pins to use for the design (so it matches the Nexys4DDR board layout) we can implement the design.

NOTE: although VHDL is not case sensitive the XDC constraints file is case sensitive and the port names need to match the port names in the VHDL entity.

Click Run Implementation under Implementation in the Flow Navigator window.

You will find there are two warnings after implementation. This is because we have not specified any timing constraints. For this simple combinational circuit we can ignore these warnings.

🚴 decoder_1 - [C:/ECE3829/vivado/d	ecoder_1/decoder_1.xpr] - Vivado 2015.1		
File Edit Flow Tools Window L	ayout View Help		Q,- Search commands
🯄 📸 i in 🕫 🐘 🐘 🗙 i 🐎	🕨 🐮 🍪 🐝 ∑ 🧔 🔚 Default Layout 🛛 👻 🎉	(E)	Implementation Complete
Flow Navigator «	Project Manager - decoder_1		×
🔍 🛣 🖶	Sources _ C ×	∑ Project Summary X @ decoder.v X ⓑ decoder.xdc X	0 C ×
4 Project Manager	오 🛣 🚔 📾 🐮 📓 🔠	Project Settings	Edit 🛠
Image: Control of the set	□ □	Project name: decoder_1 Project location: c:,ECE3829/wwado/decoder_1 Product family: Artix-7 Project part: xc2a35km226-1 Top module name: decoder.	
 IP Integrator 		Synthesis &	Implementation *
 Create Block Design Open Block Design Generate Block Design Generate Block Design Simulation Simulation Settings Run Simulation 		Status: Complete Messages: No errors or warnings Part: xx-AStopg23-1 Strategy: <u>Wyado Synthesis Defaults</u>	Status: ✓ Complete Messages: Ø 2.mennos Part: xr2n35trog256-1 Strategy: Whado Indementation Defaults Inorematia complex Itans Inorematia complex Itans Summary Route Status
A RTL Analysis		DRC Violations *	Timing *
 Baboration Settings Den Elaborated Design Synthesis Synthesis Settings Run Synthesis Den Synthesis Open Synthesized Design 	Hierarchy Libraris CompleOrder ▲ Sources ?? Templates Source File Properties → ★ ● ★ ● decoder.v Location: Cr;ECE329/rvado/decoder_1/decoder_1.srcs/ources_1/ne* Troe:	Summary: (1) I warning	Worst Negative Slack (WNS): NA Total Negative Slack (TNS): NA Number of Finipe Endpoints: NA Total Namber of Finipe Name Indemented Timing Report Setup Hold Pulse Width
4 Implementation	Library: xil_defaultib	Utilization - Post-Implementation	Power *
Implementation Implementation Settings Run Implementation P more Implemented Design Program and Debug Bitstream Settings Cenerate Bitstream	Loray: W_Jesuard () Ster: 1.0/88 Modifiet: Today at 11:40:54 MM Copred to: C_SEC3829/wade/decoder_1/decoder_1.arcs/sources_1/ne Read-onty: No Encrypted: No Gobal incide	UUT 1% 1/0 25 50 75 100 Utilization (%) Graph Table	Total On-Chip Power: 3.996 W Junction Temperature: 4.50 °C Thermal Margin: 40.0 °C (8.0 W) Effective BA: 5.0 °C/W Power supplied to off-chip devices: 0 W Confidence level: Low
👂 🔐 Open Hardware Manager	· · · · · · · · · · · · · · · · · · ·	Post-Synthesis Post-Implementation	Summary On-Chip
	Generative Properties Mesages Implementation (2 merrings) Implementation (2 merrings) <th>. Timing constraints are needed for proper timing analysis.</th> <th>_ = e ×</th>	. Timing constraints are needed for proper timing analysis.	_ = e ×
			9:5 Insert XDC .:
		\	513 Insert XUC .;

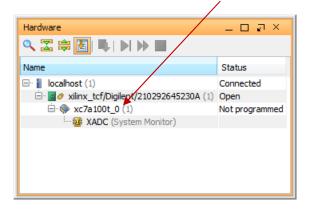
We can now generate the bitstream. Click on Generate Bitstream in the Flow Navigator window.

Next step is to program the FPGA with the bitstream (you will notice another warning message regarding configuration voltage – ignore this for now)

On the Nexys4DDR board make sure that JP1 Mode Jumper is set to the JTAG mode and connect the USB cable to the board and turn on the power.

Select the Hardware Manager in the Flow Navigator (under Program and Devug) and select Open Target and then Auto Connect.

You should now see the Xilinx xc7a100t_0 in the Hardware Window:



Click on Program Device and select the decoder.bit bitstream (automatically filled in):

🚴 Program Device	X
	ogramming file and download it to your hardware device. You can optionally select a debug esponds to the debug cores contained in the bitstream programming file.
Bitstre <u>a</u> m file: Debug probes file:	C:/ECE3829/vivado/decoder_1/decoder_1.runs/impl_1/decoder.bit
Enable end of s	tartup check
	Program Cancel

Then select Program (ignore the warning about the missing debug core and the rule violation).

Programming should just take a few seconds and then the DONE led on the Nexys4DDR board will turn on.

You should now find that your decoder is implemented on the FPGA. Change the three slider switches (SW2, SW1, SW0) through all eight combinations and verify the correct corresponding LED turns on.

Congratulations!

You have entered a design and then Synthesized, Implemented, and programmed the FPGA with the generated bit file. This was a simple design example but the same steps are just repeated for any design.

Close the Hardware manager to go back to the Project Manager window.

Programming the Serial Flash

The FPGA is a volatile device and so the bit file will not be present after power cycling the board. We can load the QSPI serial flash on the Nexys4DDR board so it loads the bit file from the flash on power up.

First we need to create a bin file to be able to program the serial flash.

Click on the Bitstream Settings in the Program and Debug section of the Flow Navigator.

Select the bin_file option:

	Bitstream	
30	(i) Note: Additional bitstream settings will be available	e once you open an implemented design
General	Write Bitstream (write_bitstream)	
	td.pre	
Simulation	td.post	
	-raw_bitfile	
	-mask_file	
Elaboration	-no_binary_bitfile	
	-bin_file	✓
>	-readback_file	
Synthesis	-logic_location_file	
	-verbose	
	More Options	
plementation		
1010		
Bitstream		
=	-bin_file	
3	Write a binary bit file without header (.bin).	
Īb		

Click OK.

Click on Generate Bitstream.

After generation, if you look in the decoder_1 => decoder_1.runs => impl_1 directory you will see a decoder.bit and a decoder.bin file.

Use the Hardware Manager to connect to the Nexys4DDR board then right-mouse click on the FPGA and select Add Configuration Memory Device:

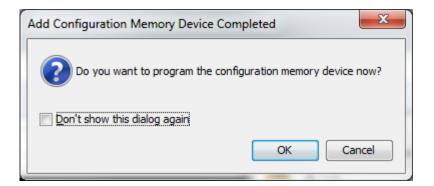
Hardware			-02	×
옥 🛣 🖨 🛃 🔍 🕨 🖿				
Name			Status	
🖃 🚪 localhost (1)			Connected	
😑 📓 🤌 xilinx_tcf/Digilent/210292	Open			
🗏 🧇 xc7a100t_0 (1)			Drogrammed	
🛄 🔯 👬 XADC (System Monil	9	Hardwa	are Device Pro	pertie
	٠	Progra	m Device	
	Ø	Refres	h Device	
	0	Add Co	onfiguration M	emory
		Boot fr	om Configurat	tion Me

We now need to select the serial flash device that is on theNexys4DDR board.

Select Spansion as the Manufacturer, then select the 128Mb device (S25FL128S):

Choose a configuration memory part. This can be changed later.	Chaose a conf	n Memory Device		anged later						<u> </u>
er Manufacturer Spansion Density (Mb) 128 Ceset All Filters Reset All Filters Cert Configuration Memory Part Search: Q Name Part Manufacturer Alias Family Type Density (Mb) Width Search: Search: S		igurador memory	part. This can be cr	langeu later.						
Density (Mb) 128 Width All Image: Configuration Memory Part Search: Q.* Name Part Manufacturer Alias Family Type Density (Mb) Width is 25f11285xxxxxx0-spi-x1_x2_x4 \$25f11285xxxxxx0 \$pansion \$25f11275 \$25f1xxxs \$pi 128 x1_x2_x4 is 25f11285xxxxxx1-spi-x1_x2_x4 \$25f1285xxxxxx1 \$pansion \$25f1275 \$25f1xxs \$pi 128 x1_x2_x4 is 25g128b-bpi-x16 \$29g128p \$pansion \$29g12xp bpi 128 x1.6 is \$29g128p-bpi-x8 \$29g128p \$pansion \$29g12xxp bpi 128 x8		:_0								
Reset All Filters Reset All Filters Search: Q.* Name Part Manufacturer Alias Family Type Density (Mb) Width Image: s25ff128sxxxxxx0-spi=x1_x2_x4 s25ff128sxxxxxx0 Spansion s25ff127s s25ffxxxs spi 128 x1_x2_x4 Image: s25ff128sxxxxxx1-spi=x1_x2_x4 s25ff128sxxxxxx1 Spansion s25ffxxxs spi 128 x1_x2_x4 Image: s29gl128p-bpi=x16 s29gl128p Spansion s29glxxxp bpi 128 x16 Image: s29gl128p-bpi=x8 s29gl128p Spansion s29glxxxp bpi 128 x8	Manufacturer	Spansion		-			Туре	All		*
Let Configuration Memory Part Search: Q.* Name Part Manufacturer Alias Family Type Density (Mb) Width Image: s25f1128sxxxxxx0-spi-x1_x2_x4 s25f1128sxxxxxx1 Spansion s25f1127s s25f1xxxs spi 128 x1_x2_x4 Image: s25f1128sxxxxxx1-spi-x1_x2_x4 s25f1128sxxxxxx1 Spansion s25f1xxxs spi 128 x1_x2_x4 Image: s25g1128p-bpi-x16 s29g128p Spansion s29g1xxp bpi 128 x1_6 Image: s25g1128p-bpi-x8 s29g128p Spansion s29g1xxxp bpi 128 x8	Density (Mb)	128		+			Width	All		+
Lett Configuration Memory Part Spearch: Q Name Part Manufacturer Alias Family Type Density (Mb) Width Image: s25f1128sxxxxxxx0-spi-x1_x2_x4 s25f1128sxxxxxx0 Spansion s25f1127s s25f1xxxs spi 128 x1_x2_x4 Image: s25f1128sxxxxxxx1-spi-x1_x2_x4 s25f1128sxxxxxx1 Spansion s25f1xxxs spi 128 x1_x2_x4 Image: s29g1128p-bpi-x16 s29g128p Spansion s29g1xxxp bpi 128 x16 Image: s29g1128p-bpi-x8 s29g128p Spansion s29g1xxxp bpi 128 x8					a strate					
Search: Q. Name Part Manufacturer Alias Family Type Density (Mb) Width s25f1128sxxxxxx0-spi-x1_x2_x4 s25f1128sxxxxxx0-spi-x1_x2_x4 s25f1127s s25f1xxxs spi 128 x1_x2_x4 s25f1128sxxxxxx1-spi-x1_x2_x4 s25f1128sxxxxxx1 Spansion s25f1xxxs spi 128 x1_x2_x4 s25g1128p-bpi-x16 s29g128p Spansion s29glxxxp bpi 128 x16 s29g1128p-bpi-x8 s29g1128p Spansion s29glxxxp bpi 128 x8				E	Reset All Filters	;				
Search: Q. Name Part Manufacturer Alias Family Type Density (Mb) Width s25f1128sxxxxxx0-spi-x1_x2_x4 s25f1128sxxxxxx0-spi-x1_x2_x4 s25f1127s s25f1xxxs spi 128 x1_x2_x4 s25f1128sxxxxxx1-spi-x1_x2_x4 s25f1128sxxxxxx1 Spansion s25f1xxxs spi 128 x1_x2_x4 s25g1128p-bpi-x16 s29g128p Spansion s29glxxxp bpi 128 x16 s29g1128p-bpi-x8 s29g1128p Spansion s29glxxxp bpi 128 x8										
Name Part Manufacturer Alias Family Type Density (Mb) Width \$\$25f1128\$xxxxxx0-spi-x1_x2_x4 \$\$25f1128\$xxxxxx0 \$\$pansion \$\$25f127\$ \$\$25f1xxs \$\$pi 128 \$\$x1_x2_x4 \$\$\$25f1128\$xxxxxx1-spi-x1_x2_x4 \$\$25f1128\$xxxxxx1 \$\$pansion \$\$25f1xxs \$\$pi 128 \$\$x1_x2_x4 \$	ct Configuration Me	emory Part								
s25f128sxxxxxx1-spi-x1_x2_x4 s25f128sxxxxx1-spi-x1_x2_x4 s25f128sxxxxx1-spi-x1_x2_x4 s25f128sxxxxx1-spi-x1_x2_x4 s25f128sxxxxx1-spi-x1_x2_x4 s25f128sxxxxx1-spi-x1_x2_x4 s25f128sxxx1-spi-x1_x2_x4 s25f128sxxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sxx1-spi-x1_x2_x4 s25f128sx1-spi-x1_x2_x4 s25f128sx1-spi-x2_x4 s25f128sx1-										
Image: s25fl128sxxxxxx1-spi-x1_x2_x4 s25fl128sxxxxxx1 Spansion s25flxxxs spi 128 x1_x2_x4 Image: s29gl128p-bpi-x16 s29gl128p Spansion s29glxxxp bpi 128 x16 Image: s29gl128p-bpi-x16 s29gl128p Spansion s29glxxxp bpi 128 x16 Image: s29gl128p-bpi-x8 s29gl128p Spansion s29glxxxp bpi 128 x8	Search: Q-									
\$\$ s29gl128p-bpi-x16 s29gl128p Spansion s29glxxxp bpi 128 x16 \$\$ s29gl128p-bpi-x8 s29gl128p Spansion s29glxxxp bpi 128 x8			Part	Manufacturer	Alias	Family	Type	Density (Mb)	Width	1
\$29gl128p-bpi-x8 \$29gl128p \$pansion \$29glxxxp bpi \$128 x8	Name	xx0-spi-x1_x2_x4								
	Name		s25fl128sxxxxxx0	Spansion		s25flxxxs	spi	128	x1_x2_x4	
\$\$\$ \$29gl128s-bpi-x16 \$\$29gl128s \$\$\$ \$pansion \$\$29glxxxs \$bpi 128 \$x16	Name \$25f1128xxxxxx \$25f1128xxxxxxx	xx1-spi-x1_x2_x4	s25fl128sxxxxx0 s25fl128sxxxxxx1	Spansion Spansion		s25flxxxs s25flxxxs	spi spi	128 128	x1_x2_x4 x1_x2_x4	
	Name \$25f11285xxxxx \$25f1285xxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxxx1 s29gl128p	Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi	128 128 128	x1_x2_x4 x1_x2_x4 x16	
	Name \$25fi128sxxxxx \$25fi128sxxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxx1 s29gl128p s29gl128p	Spansion Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi bpi	128 128 128 128 128	x1_x2_x4 x1_x2_x4 x16 x8	
	Name \$25fi128sxxxxx \$25fi128sxxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxx1 s29gl128p s29gl128p	Spansion Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi bpi	128 128 128 128 128	x1_x2_x4 x1_x2_x4 x16 x8	
	Name \$25f11285xxxxx \$25f11285xxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxx1 s29gl128p s29gl128p	Spansion Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi bpi	128 128 128 128 128	x1_x2_x4 x1_x2_x4 x16 x8	
	Name \$25f11285xxxxx \$25f11285xxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxx1 s29gl128p s29gl128p	Spansion Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi bpi	128 128 128 128 128	x1_x2_x4 x1_x2_x4 x16 x8	
OK Canc	Name \$25f11285xxxxx \$25f11285xxxxx \$29g1128p-bpi-: \$29g1128p-bpi-:	xx1-spi-x1_x2_x4 x16 x8	s25fl128sxxxxx0 s25fl128sxxxxx1 s29gl128p s29gl128p	Spansion Spansion Spansion Spansion		s25flxxxs s25flxxxs s29glxxxp s29glxxxp	spi spi bpi bpi	128 128 128 128 128	x1_x2_x4 x1_x2_x4 x16 x8 x16	Cancel

Click OK



Click OK

Select the Configuration File (decoder.bin) in the impl_1 directory.

Select a configuration fil	e and set pro	gramming options.
Memory Device: <u>C</u> onfiguration file: <u>S</u> tate of non-config m Program Operations	em I/O pins:	s25fl032p-spi-x1_x2_x4 i29/vivado/decoder_1/decoder_1.runs/impl_1/decoder.bin Pull-none
Address Range: Erase Blank Check Program <u>Verify</u>	Configuratio	on File Only 🔹

Click OK.

Note: this will erase any existing design in the QSPI flash (including the configuration file shipped with the Nexys4DDR board).

The QSPI Flash will now be erased and then programmed with the decoder.bin file.

Once programmed, you can power off the Nexys4DDR board and change the JP1 jumper mode from JTAG to QSPI.

Power back on the board and after a few seconds the DONE LED will turn on indicating that your decoder design has been automatically loaded into the FPGA from the serial flash. You can verify the decoder design by moving the slider switches and observing the leds as before.