



Creating a Base System for the Zynq in Vivado

by Jeff Johnson | Jul 31, 2014 | Vivado | 8 comments



5 Votes

Tutorial Overview

In the ISE/EDK tools, we'd use the Base System Builder to generate a base project for a particular hardware platform. Now with Vivado, the process is a little different but we have more control in how things are setup and we still benefit from some powerful automation features. In this tutorial we'll create a base design for the Zynq in Vivado and we'll use the MicroZed board as the hardware platform.



Requirements

Before following this tutorial, you'll need the following:

- Vivado 2014.2
- MicroZed
- Platform Cable USB II (or equivalent JTAG programmer)

Create a new Vivado project

Follow these steps to create a new project in Vivado:

1. Open Vivado. From the welcome screen, click "Create New Project".

🚴 Vivado 2014.2	
<u>File Flow Tools Window Help</u>	Q, Search commands
Productivity. Multiplied.	E XILINX ALL PROGRAMMABLE.
Quick Start	Recent Projects
	microzed_custom_ip E:/Github/fpgadeveloper/microzed_custom_ip zedboard_qgige E:/Github/fpgadeveloper/zedboard_qgige
Create New Project Open Project Open Example Project	microzed_base E:/Github/fpgadeveloper/microzed_base1
Tasks	zedboard-bsb E:/Github/fpgadeveloper/zedboard-bsb2
	zedboard-bsb E:/Github/fpgadeveloper/zedboard-bsb1
Manage IP Open Hardware Manager Xilinx Tcl Store	:
Information Center	
 Image: A state of the state of	
Documentation and Tutorials Quick Take Videos Release Notes Guide	
I Tcl Console	
New Project Wizard will guide you through the process of selecting design sources and a targe	et device for a new project.

2. Specify a folder for the project. I've created a folder named "microzed_custom_ip".

Click "Next".

🚴 New Project	
Project Name	
Enter a name	e for your project and specify a directory where the project data files will be stored
Project name:	microzed_custom_ip
Project location:	E:/Github/fpgadeveloper/microzed_custom_ip
Create proje	ct subdirectory
Project will be cr	reated at: E:/Github/fpgadeveloper/microzed_custom_ip
	\mathbf{G}
	< Back Mexics, Similar Cancel

3. For the Project Type window, choose "RTL Project". Click "Next".

🚴 New Project
Project Type
Specify the type of project to create.
 <u>BTL Project</u> You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis. <u>D</u> on ot specify sources at this time <u>Post-synthesis Project</u> You will be able to add sources, view device resources, run design analysis, planning and implementation. <u>D</u> on to specify sources at this time <u>J</u> O Planning Project Do not specify design sources. You will be able to view part/package resources. <u>Imported Project</u> Create a Vivado project from a Synplify, XST or ISE Project File.

4. For the Add Sources window, click "Next". We will add our multiplier source code later.

- 5. For the Add Existing IP window, click "Next".
- 6. For the Add Constraints window, click "Next".
- 7. For the Default Part window, select the "MicroZed Board" and click "Next".

🚴 New Project						X
Default Part						
Choose a default Xilinx part or board for your p	roject. This	can be changed	l later.			
Specify Filter						
Parts	Vendor	All		*		
Disr	nlav Name	All		*		
Boards	Board Rev	latest		*		
		Reset All F	ilters			
Search: Q-						
Display Name	Vendor	Board Rev	Part	I/O Pin Count	File Version	Availat IOBs
MicroZed Board	em.avnet.co	om e	🔷 xc7z010clg400-1	400	1.0	100
📓 ZedBoard Zynq Evaluation and Development Kit	em.avnet.co	om d	xc7z020clg484-1	484	1.0	200
Artix-7 AC701 Evaluation Platform	xilinx.com	1.0	xc7a200tfbg676-2	676	1.0	400
Kintex-7 KC705 Evaluation Platform	xilinx.com	1.1	xc7k325tffg900-2	900	1.0	500
Virtex-7 VC707 Evaluation Platform	xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.0	700
Virtex-7 VC709 Evaluation Platform	xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.0	850
ZYNQ-7 ZC702 Evaluation Board	xilinx.com	1.0	xc7z020clg484-1	484	1.0	200
ZYNQ-7 ZC706 Evaluation Board	xilinx.com	1.1	xc7z045ffg900-2	900	1.0	362
	\searrow					
III						
			< <u>B</u> ack	Next >>		ancél

8. Click "Finish" to complete the new project wizard.

Change the project's default language

By default, the project will be created using Verilog as the default language. As we'll be importing VHDL, let's change that to VHDL:

- 1. From the menu, select Tools->Options.
- 2. In the "General" tab select target language : VHDL.

Setup the Zynq PS

The new Vivado project starts off blank, so to create a functional base design, we need to at least add the Zynq PS (processor system) and make the minimal required connections. Follow these steps to add the PS to the project:

1. From the Vivado Flow Navigator, click "Create Block Design".

👃 microzed_custom_ip - [E:/Gith	ub/fpgadeveloper/microzed_custon	n_ip1/microze	ed_custo	m_ip.	xpr] - Viv	/ad 🔄	_ 0	X
<u>File Edit Flow Tools W</u> indow	La <u>y</u> out <u>V</u> iew <u>H</u> elp				Q- Se	arch con	nmands	
🧦 🖻 🕼 💷 🐘 🗙 🔈 🕨	📸 🍪 💥 ∑ 🎼 😬 Default Layo	ut 👻 🕽	K 🚸 K	(S				Ready
Flow Navigator «	Project Manager - microzed_custon	n_ip						×
Q 🔀 🖨	Sources		- 🗆 🖻	×	\Sigma Proj	ect Sun	mary	
Project Manager				_	🔺 Pr	oject Se	ettings	^
🏀 Project Settings	Constraints				🛱 Pro	oject nar	ne:	micro:
🔂 Add Sources	Simulation Sources				Pro	oduct far	nily:	Zynq-:
手 IP Catalog					Pro	oject par	t:	Microz
IP Integrator					То	p modul	e name:	<u>Not de</u>
A Create Block Resign		ir		:	Bo	ard Par	t	
Den Block D	Sources V Templates				Dis	splay nar	ne: I	MicroZe
Generate Blog	nd add an IP subsystem to the proje	ect.	_ 🗆 🖻	×	Bo	ard part	name: (em.avn
A Simulation					Re	pository	path: (C:/Xilin
Simulation Settings					UR	RL:	1	http://v 🗸
Run Simulation					•	HI		- F
A DTI Analysia	Design Runs						_ [⊐ ⊵ ×
 RTL Analysis Chan Elaborated Decian 	Name	Constraints	WNS	TNS	WHS	THS	TPWS	Failed F
	Synth_1	constrs_1						
 Synthesis 	impl_1	constrs_1						
Synthesis Settings	•							
Run Synthesis								
Open Synthesized Desig								
 Implementation 	*							
🚳 Implementation Settings								
Run Implementation		1						
🕨 🖻 Open Implemented Desi 🗖	Tcl Console De Messages	Log 🗋 Repo	rts 🐌 D	esign	Runs		-	
Create and add an IP subsystem to the	e project				180		107	er E

2. Specify a name for the block design. Let's go with the default "design_1" and leave it local to the project. Click "OK".

👃 Create Bloc	k Design
Please s	pecify name of block design
Design name:	design_1
Directory:	So <local project="" to=""></local>
	OKCancel

3. In the Block Design Diagram, you will see a message that says "This design is empty. To get started, Add IP from the catalog.". Follow this advice by clicking on the blue "Add IP" link, or by using the "Add IP" icon. Creating a Base System for the Zynq in Vivado | FPGA Developer



.stom ip E:/Github/fpgadeveloper/microzed custom ip1 -part xc7z010clg400-1

4. The IP catalog should appear. Go to the end of the list and double click on the

block named "ZYNQ7 Processing System" – it should be the second last on the list.

Vivado will now add the PS to the block diagram.

Search: Q-			
▲ 1 Name	VENV		
L SMPTE2022-5/6 Video ov	viliny com		
SMPTE 2022-5/0 Video ov	xilinx.com		^
SMPTE 2022-1/2 Video ov	xilinx.com		
SMPTE 2022-1/2 VIGEO 0V	xilinx.com		
SMPTE SD/HD/3G-SDI	xilinx.com		
System Cache	xilinx.com		
F Test Pattern Generator	xilinx.com		
Tri Mode Ethernet MAC	xilinx.com		
🞐 Utility Differential IO Buffer	xilinx.com		
🞐 Utility Reduced Logic	xilinx.com		
🞐 Utility Vector Logic	xilinx.com		
🞐 Video Deinterlacer	xilinx.com		
🞐 Video In to AXI4-Stream	xilinx.com		
🞐 Video On Screen Display	xilinx.com		
Video Scaler	xilinx.com		
Video Timing Controller	xilinx.com		
VIO (Virtual Input/Output)	xilinx.com		
👎 Viterbi Decoder	xilinx.com		_
XADC Wizard	xilinx.com		_
YCrCb to RGB Color-Space	xilinx.com		-
ZYNO7 Processing System	xilinx.com		-
F 7YN07 Processing System	xilinx.com	5	
Colort and proce ENTER or drag	and drove SC	Stellers a	

5. In the Block Design Diagram, you will see a message that says "Designer Assistance available. Run Block Automation". Click on the "Run Block Automation" link and select

Creating a Base System for the Zynq in Vivado | FPGA Developer

"processing_system7_0" from the drop-down menu. Block Automation makes connections and pin assignments to external hardware such as the DDR and fixed IO. It does this using the board definition of the hardware platform you specified when you created the project (MicroZed). We could make these connections ourselves if we were using a custom board, but for off-the-shelf boards, Block Automation makes the process a lot easier.

t Layout		Ready
		×
	🚰 Diagram 🗙 🔣 Address Editor 🗙	
	📲 📥 design_1	
ocessing Si	🔍 🗔 Designer Assistance available. <u>Run Block Automation</u>	
ocessing 5)	process / /processing_system7_0	~
► ■ Boar □ @ ×	DDR DDR FIXED_IO FIXED_IO M_AXI_GP0_ACLK M_AXI_GP0 FCLK_CLK0 FCLK_RESETO_N	
	ZYNQ7 Processing System	
	freatoral	<u>کہ دیت</u>

6. In the Block Automation window, make sure "Apply Board Preset" is ticked and click "OK".

👃 Run Block Autom	tion 🗾
Zynq7 block au FIXED_IO, Trig	mation applies current board preset and generate external connections for er and DDR interfaces
NOTE: Apply Bo wish to retain p	rd Preset will discard existing IP configuration - please uncheck this box, if you evious configuration
Instance: /proc	ssing_system7_0
Make Interface Externa	FIXED_IO, DDR
Apply Board Preset:	
<u>Cross Trigger In:</u>	Disable 💌
Cross <u>T</u> rigger Out:	Disable 🔻
	S

7. Now our block diagram has changed and we can see that the DDR and FIXED_IO are connected externally. Now the only remaining connection to make is the clock that we will use for the AXI buses. We must configure the Zynq to generate a clock and enable a general purpose AXI bus. To make these settings, double click on the





8. The Re-customize IP window will open. From the Page Navigator, select "Clock

Configuration" and open the "PL Fabric Clocks" tree.

Fre-customize IP					X
ZYNQ7 Processing	System (5.4)				4
🎁 Documentation 🏀 Pr	resets ៉ IP Location 🊳 Import XPS	5 Settings			
Page Navigator «	Clock Configuration				Summary Report
Zynq Block Design	Basic Clocking Advanced Clo	ocking			
PS-PL Configuration	Finput Frequency (MHz) 33.33	3333 🛞 C	PU Clock Ratio 6:2:1	•	
Peripheral I/O Pins	Search: Q-				
MIO Configuration	Component	Clock Source	Requested Frequ	Actual Frequency	Range(MHz)
Clock Configuration	Processor/Memory Clocks				
DDR Configuration	PL Fabric Clocks	-	1	1	
SMC Timing Calculation	FCLK_CLK0	IO PLL 🔻	100	100.000000	0.100000 : 250.000000
Interrupts	FCLK_CLK1	IO PLL	100	100.000000	0.100000 : 250.000000
	FCLK_CLK2	IO PLL	33.333333	33.333336	0.100000 : 250.000000
	FCLK_CLK3	IO PLL	50	50.000000	0.100000 : 250.000000
	System Debug Clocks				
	timers				
				2	
	L			ប្រែព្រះ	OK Cancel

9. Make sure that the FCLK_CLK0 is enabled (ticked) and that it is set for a frequency of 100MHz. This will be our AXI clock.

10. Now from the Page Navigator, select "PS-PL Configuration" and open the "GP Master AXI Interface" tree.

11. Tick the "M AXI GP0 interface" checkbox to enable it.

🖵 Re-customize IP				X
ZYNQ7 Processing	Syster	m (5.4)		4
🎁 Documentation 🊳 Pr	resets 這] IP Location 🍈 Import XPS Setting	S	
Page Navigator «	PS-PL C	Configuration		Summary Report
Zynq Block Design	🗲 50	earch: Q-		
PS-PL Configuration			Colort	Description
Peripheral I/O Pins		General	Select	Description
MIO Configuration		DMA Controller GP Master AXI Interface		
Clock Configuration		🗄 M AXI GP0 interface		Enables General purpose AXI master interface 0
DDR Configuration	(H AXI GP1 interface		Enables General purpose AXI master interface 1
SMC Timing Calculation		GP Slave AXI Interface HP Slave AXI Interface ACP Slave AXI Interface		
Interrupts	÷1	PS-PL Cross Trigger interface		Enables PL cross trigger signals to PS and vice-versa
	•		III	
				Cancel

12. Now click "OK" to close the Re-customize IP window.

13. You should now see a new input port on the left side of the Zynq PS block. This is the AXI clock input.



We must now connect the FCLK_CLK0 output to the AXI clock input. To do this, click on the FCLK_CLK0 output and then click on the M_AXI_GP0_ACLK input. This will trace a wire between the pins and make the connection.



Create the HDL wrapper

Now the Zynq is setup and all we need to do to create a functional project is to create a HDL wrapper for the design.

1. Open the "Sources" tab from the Block Design window.

Sources	_ 🗆	$\mathbb{R}^{2}\times\mathbb{R}$	3-	Diagr
🔍 🛣 😂 🖬 🔂	Ł		[€]	🙏 de
Design Sources (1)			0+	
🗄 🏯 design_1 (design	n_1.bd) (1)		0-	
Constraints	(1)			
initiation Sources	(1)		ie	
			R	
			2	
			\$	
			9	
			₽	
			₽ _K	
			-	
			6	
Hierarchy IP Sources	Libraries Com	∢ ▶ 🗉	ø	
& Sour 🗄 Desig	🧾 Signals 📲	Boar	<u>گ</u>	
System Net Pr Sources	_ 🗆	Ŀ"×		
← → 🖏 R				

2. Right-click on "design_1" and select "Create HDL wrapper" from the drop-down menu.

BIOCK Design - des	sign_	1 -		
Sources		_ D @ ×	🗄 Diagram 🗙 🛛	
🔍 🔀 😂 🔀	?	2	注 🙏 design_1	
🖃 ն Design Source	es (1 desi) an 1.bd)(1)	Q+	
🕀 🗁 Constraints	6	Source Node Properties	Ctrl+E	
Simulation So	3	Open File	Alt+O	
		Create HDL Wrapper		
		View Instantiation Templat	e	
		Generate Output Products		
		Reset Output Products		
		Package Block Design		

3. From the "Create HDL wrapper" window, select "Let Vivado manage wrapper and auto-update". Click "OK".

🚴 Create HDL Wrapper
You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.
Options
Copy generated wrapper to allow user edits
<u> <u> </u>Let Vivado manage wrapper and auto-update </u>
GOK

From this point, we have a base design containing the Zynq PS from which we could generate a bitstream and test on the MicroZed. We haven't exploited any of the FPGA fabric, but the Zynq PS is already connected to the Gigabit Ethernet PHY, the USB PHY, the SD card, the UART port and the GPIO, all thanks to the Block Automation feature. So there is already quite a lot we could do with the design at this point, such as running Linux on the PS or running a bare metal application on it.

Generate the bitstream

To generate the bitstream, click "Generate Bitstream" in the Flow Navigator.



Once the bitstream is generated, the following window appears. Select "Open Implemented Design" and click "OK". Creating a Base System for the Zynq in Vivado | FPGA Developer

Bitstream Generation Completed 🔓 🛛 🔀
Bitstream Generation successfully completed.
Next
Open Implemented Design
Open <u>H</u> ardware Manager
Don't show this dialog again
OK Cancel

The implemented design will open in Vivado showing you a map of the Zynq device and how the design has been placed. In our case, we haven't used any of the FPGA fabric (only the PS), so the map is empty for the most part.



Export the hardware to SDK

Once the bitstream has been generated, the hardware design is done and we're ready to develop the code to run on the processor. This part of the design process is done in Xilinx Software Development Kit (SDK), so from Vivado we must first export the project to SDK.