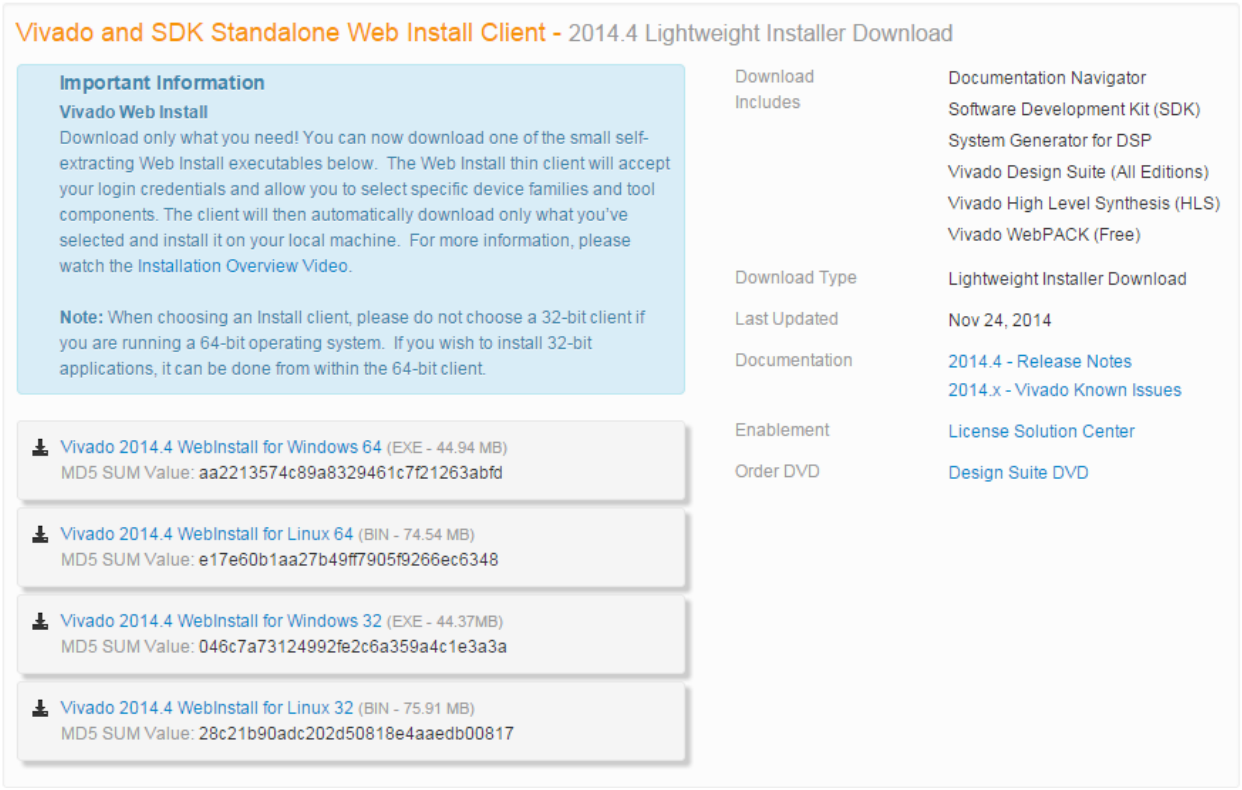
Step 1: Get and install Vivado webpack



1. Go to http://www.xilinx.com/support/download.html
2. Go to Vivado and sdk standalone web install client
3. Select your OS
4. Download the file
5. Install Vivado webpack and the SDK by following the instructions given

Step 2: Download the base system design for your board and use it with Vivado

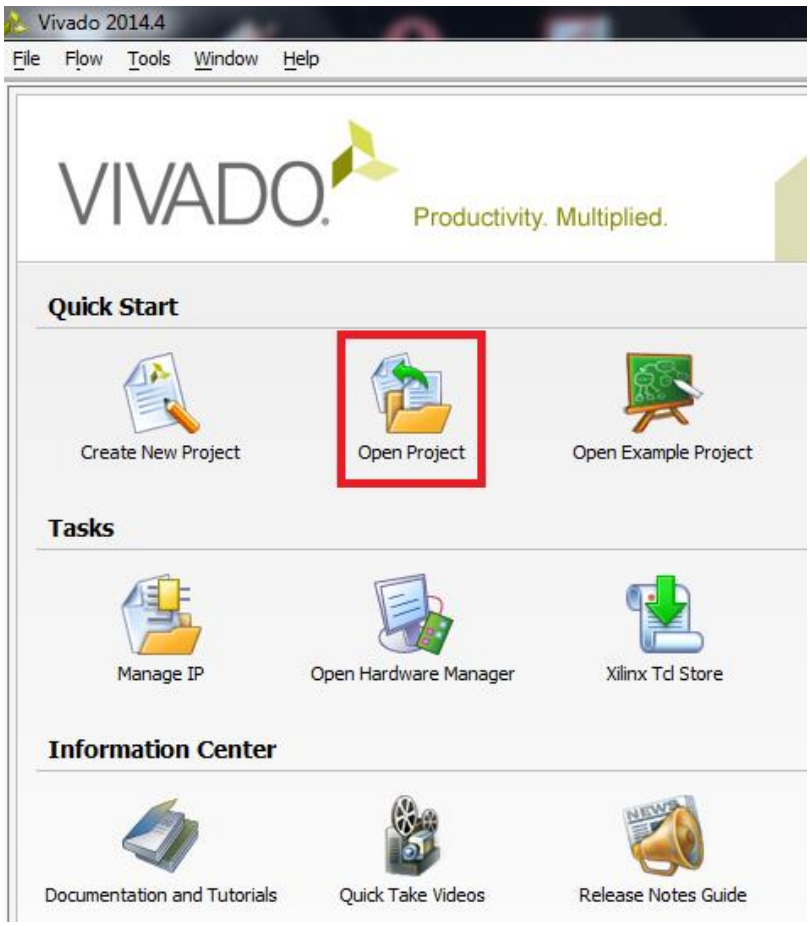
In this example, we will use the ZYBO from Digilent.

1. Go to https://www.digilentinc.com/Products/Detail.cfm?Prod=ZYBO
2. Download the ZYBO Base System Design:

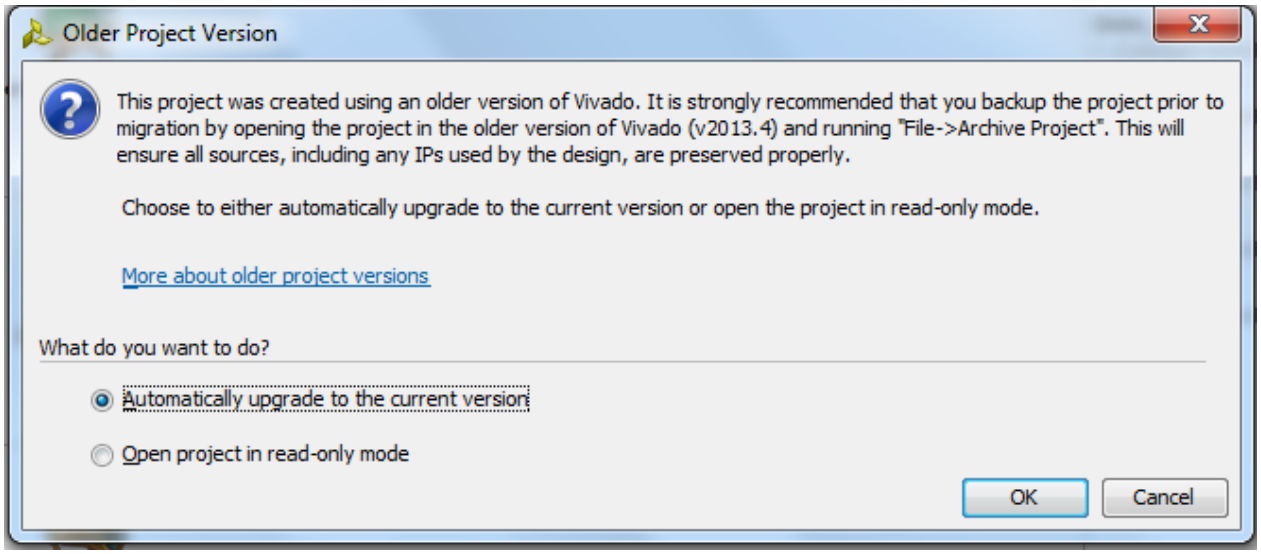
Support Documents:

Doc #	Date	Categories	Description	
500-279	2/13/14	PD	ZYBO™ schematics.	Download
502-279	2/14/14	PD	ZYBO™ reference manual.	Download
594-008	1/27/15	PD	Embedded Linux hands-on tutorial for the ZYBO. This document provides step-by-step instructions for customizing your hardware, compiling the Linux Kernel, and writing driver and user applications.	Download
DSD-0000444	2/18/14	RD	ZYBO Master UCF File for ISE designs.	Download
DSD-0000445	2/18/14	RD	ZYBO Board Definition File for configuring the Zynq Processing System core in Xilinx Platform Studio and Vivado IP Integrator.	Download
DSD-0000446	2/19/14	RD	ZYBO Master XDC File for Vivado designs.	Download
DSD-0000447	4/25/14	DP	ZYBO Base System Design. A good starting point for custom ZYBO designs, including custom IP cores for VGA output, HDMI output, and Audio codec input/output. Project files included for ISE 14.7 and Vivado 2013.4.	Download
DSD-0000464	8/06/14	RD	PetaLinux 2014.2 board support package (BSP) for the Zybo. This allows you to use your ZYBO with Xilinx's PetaLinux embedded Linux solution.	Download

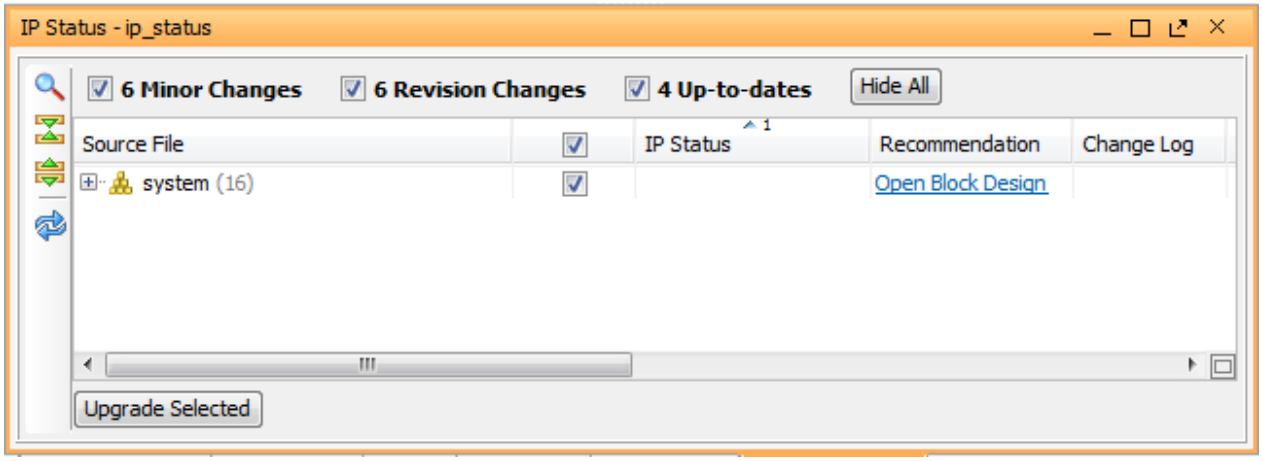
3. Extract the file
4. Launch Vivado
5. Click on open file



6. Open the file zybo_bsd.xpr which is in the folder:
   zybo_base_system\source\vivado\hw\zybo_bsd
7. Select Automatically upgrade to the current version (if offered)



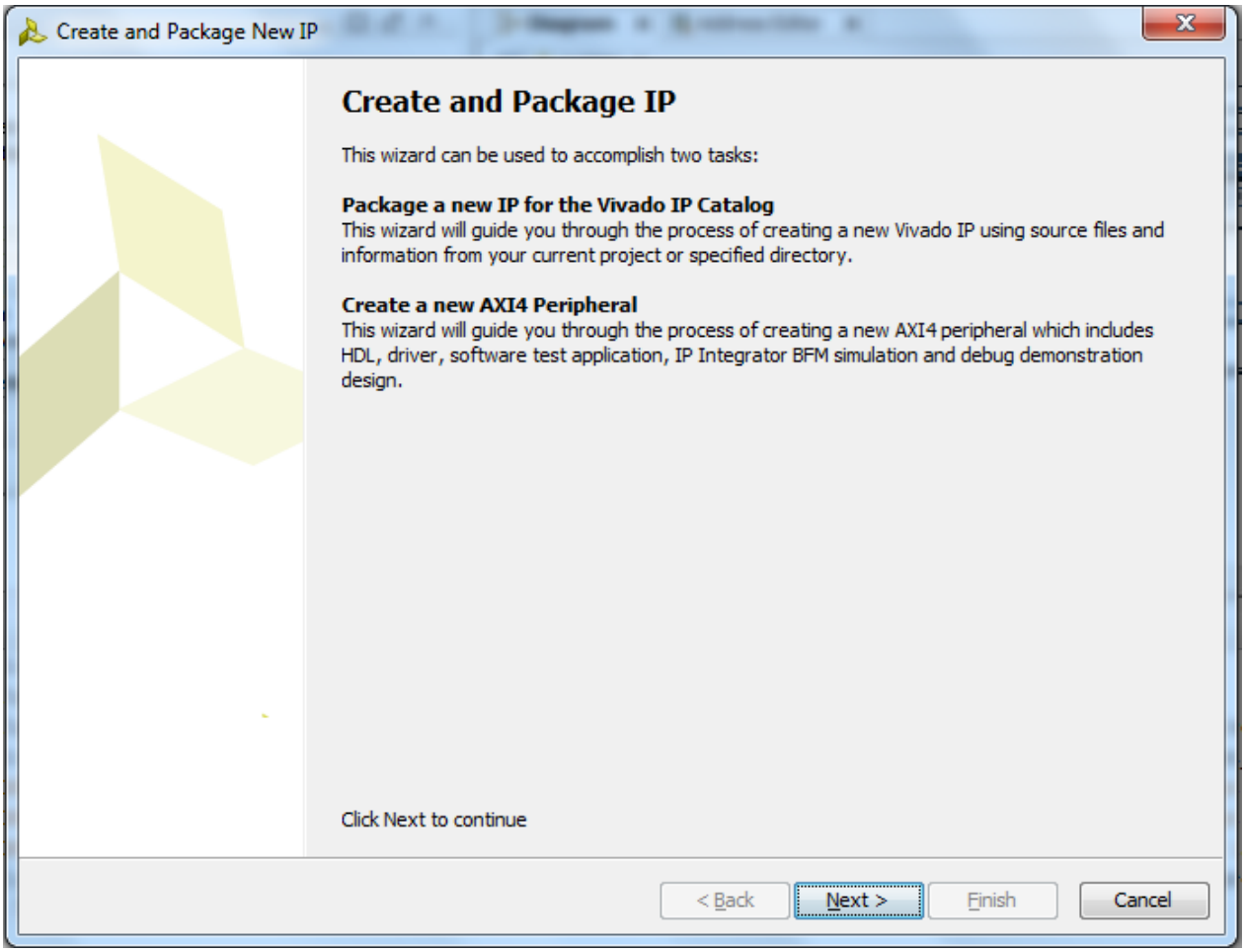
8. Then click on Report IP Status and choose Upgrade Selected and OK



9. Close the warning window (don't worry about it) and go to Tools/Create and Packages IP

Step 3: Create the IP for the globe

We will now use the IP packager of Vivado to insert the globe as an IP in the global system.



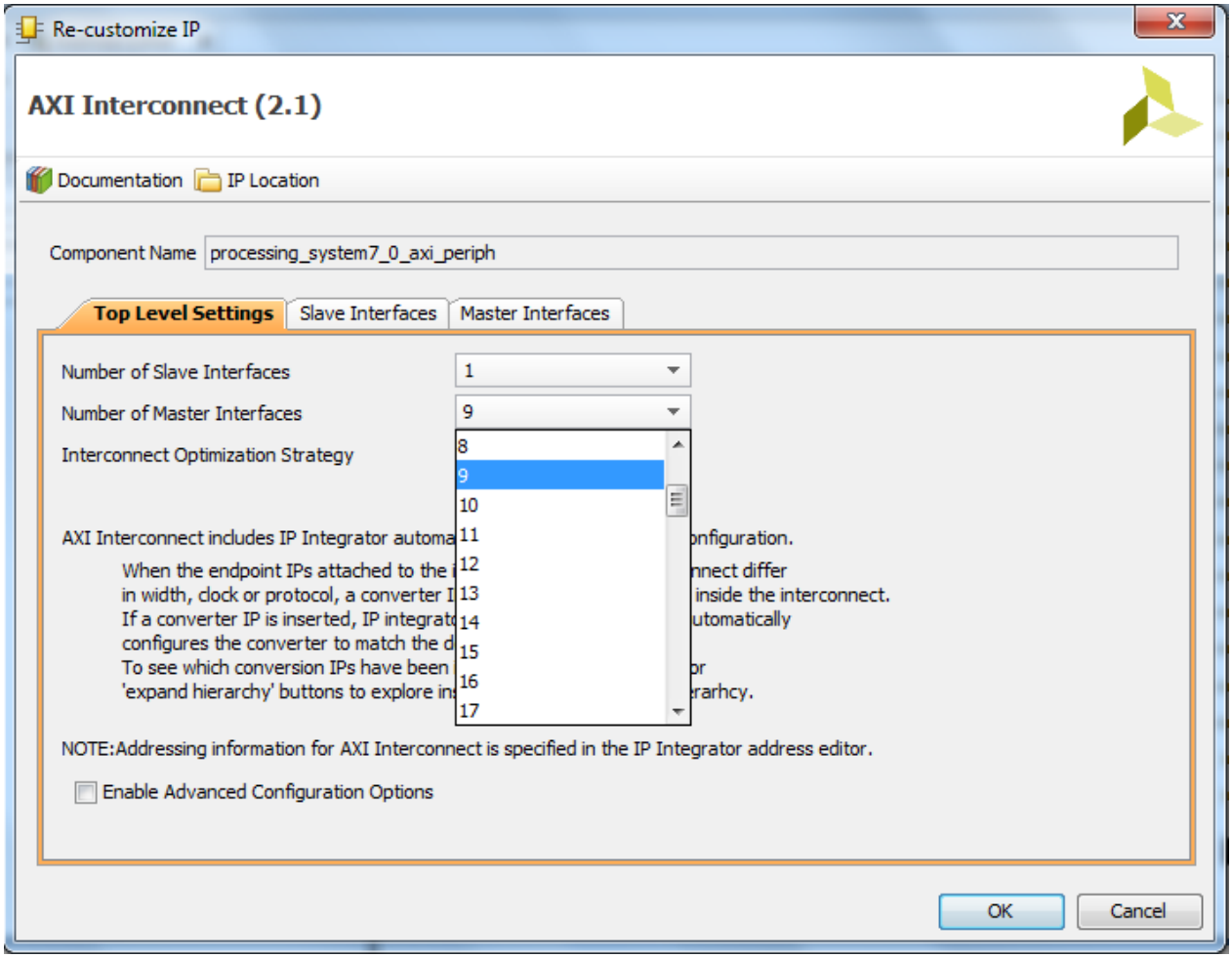
1. When you see the window above you have to click on next and then Create a new AXI4 peripheral
2. Then name the peripheral as you want (you can add a little description)
3. For the AXI interface you have to let the name S00_AXI and select the type *Lite* and the mode *Slave.* Data width should be 32 and the rest doesn't matter
4. Finally you can click on *next* and select *Edit IP* then *Finish*

Step 4: Integrate the source code in the IP

We have now the IP packager opened; we will insert the source code in this IP in order to the correct system.

1. First download the archive file from here
2. Delete the file present in the IP and import all the file from the archive folder (right click/add sources/create or add design source)

3. Then click on the tab *Package IP* and do everything asked for each part and finally click on *Re-Package IP*
4. You are now back to the main window of Vivado
5. Do a right click on the diagram and choose *Add IP* then select your IP (your IP is now added to the system you will have to connect your IP to it)
6. For all the external ports (LED0-5, PWM_OUT and INFRA_SENSOR) do a right click on it and choose *Make External* (you will have now all the external port of the globe created)
7. Then double click on the *AXI Interconnect* and add one Master interface



8. Now connect this new Master interface to your *S00_AXI*
9. Then connect *M08_ACLK* and *M08_ARESETN* to the other one
10. Then connect your *s00_axi_aclk* and *s00_axi_aresetn* to the previous one
11. in the tab *Address Editor* and right click on your IP and select *Auto Assign Address*
12. In the *Sources* tab right click on the *system_i* and select *Generate Output Products*
13. And then click again on it and select *Create HDL Wrapper*
14. Finally update the constraint file base.xdc (the following pictures is an example)

```
75 ## User constraints
76 ##Input signals
77 set_property PACKAGE_PIN T17 [get_ports INFRA_SENSOR]
78 set_property IOSTANDARD LVCMOS33 [get_ports INFRA_SENSOR]
79
80 ##Output signals
81 set_property PACKAGE_PIN V12 [get_ports LED0]
82 set_property IOSTANDARD LVCMOS33 [get_ports LED0]
83
84 set_property PACKAGE_PIN W16 [get_ports LED1]
85 set_property IOSTANDARD LVCMOS33 [get_ports LED1]
86
87 set_property PACKAGE_PIN J15 [get_ports LED2]
88 set_property IOSTANDARD LVCMOS33 [get_ports LED2]
89
90 set_property PACKAGE_PIN H15 [get_ports LED3]
91 set_property IOSTANDARD LVCMOS33 [get_ports LED3]
92
93 set_property PACKAGE_PIN V13 [get_ports LED4]
94 set_property IOSTANDARD LVCMOS33 [get_ports LED4]
95
96 set_property PACKAGE_PIN U17 [get_ports LED5]
97 set_property IOSTANDARD LVCMOS33 [get_ports LED5]
98
99 set_property PACKAGE_PIN Y17 [get_ports PWM_OUT]
100 set_property IOSTANDARD LVCMOS33 [get_ports PWM_OUT]
```

15. You can now generate the Bitstream

Step 5: Generate the FSBL

You have now the system_wrapper.bit generated. You will now generate the FSBL file.

1. At the end of the Bitstream generation a window appears: close it and go to the File/Export/Export hardware and select *include bitstream* then click on OK
2. Go to File/Launch SDK (and follow the different steps)
3. Once the SDK launches go to File/New/Board Support Package and click on Finish then select the library *xilffs, xilrsa and xilmfs* and then OK
4. Go to File/New/Application Project give it a name click on use existing (select the board package you just created) and Next
5. The FSBL is generated!