FOX2 LabView GUI

Yongxin Zhang

June 27, 2015

Contents

1 Start a new project

When we start the LabView, we can get this window.

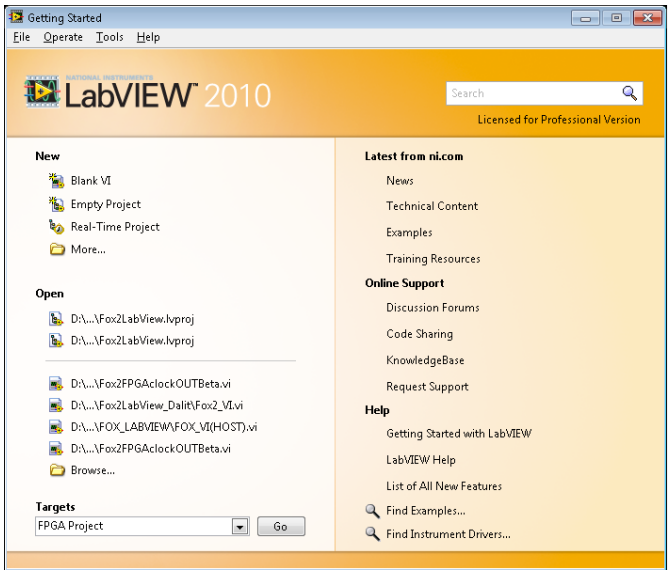


Figure 1: LabView start window

Then click more and choose create new “LabVIEW FPGA Project” like in the following window.

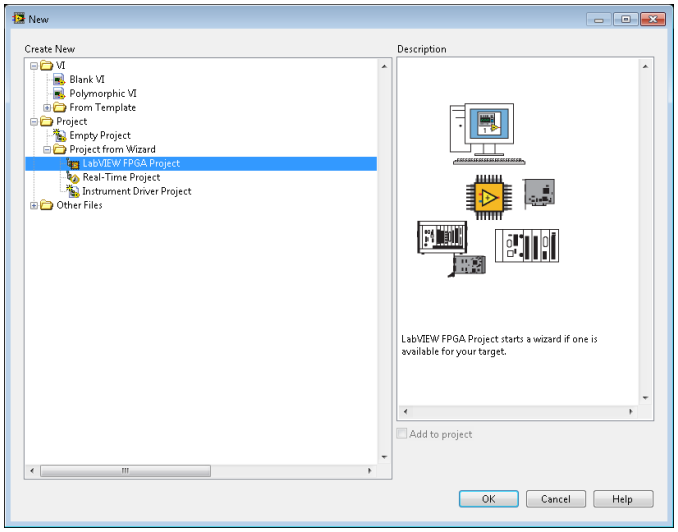


Figure 2: Choose project

Then choose the “Single-Board RIO Embedded System” and then click next.

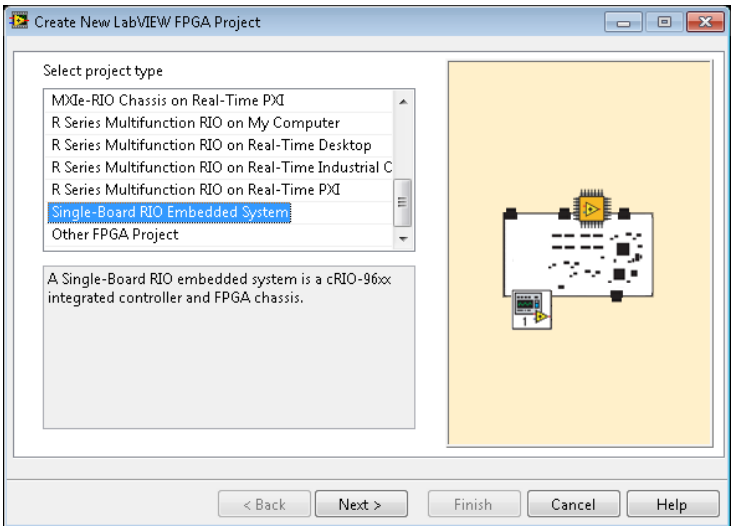


Figure 3: Choose Hardware type

Here we use the Discover existing system option to connect the board with our desktop.

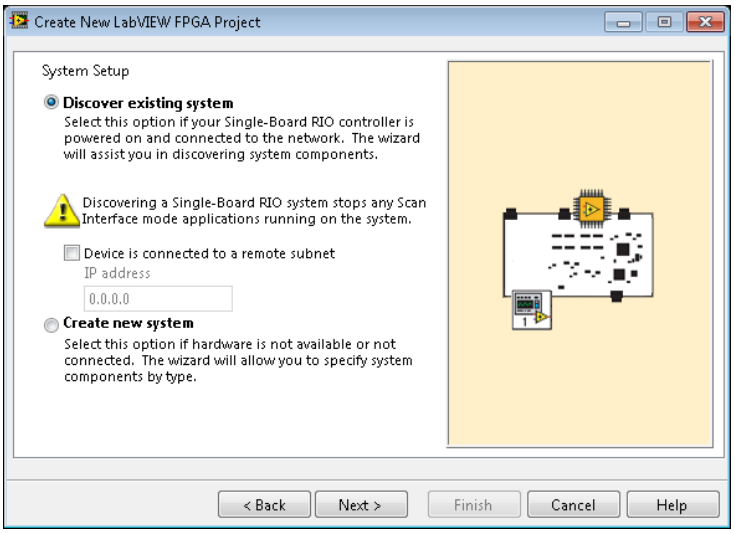


Figure 4: System Setup

If the connection and the hardware as well as the software are fine, we will see the hardware.

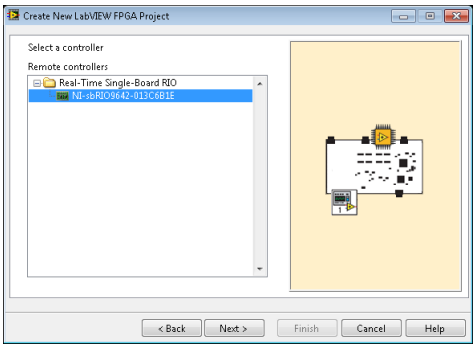


Figure 5: Find hardware

When the discovering process is done, we can get the Chassis in the Project Reivew.

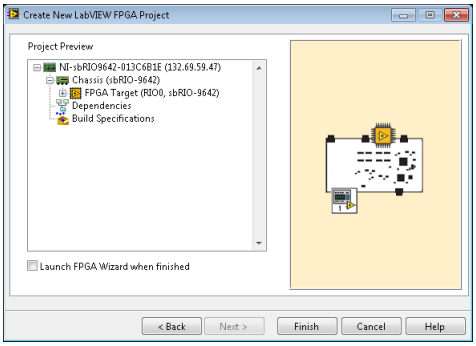


Figure 6: Project Review

Then we can click "Finish" to begin working in the LabView environment.

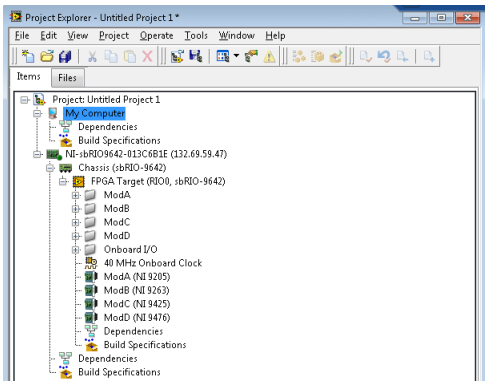


Figure 7: LabView environment

2 Config the IP address of the device

If you work in the board for the first time or it is the new board that you buy for the first time. Config the IP of the Chassis that is in the same network domain with the desktop is required.

2.1 Config the Ip of the desktop

Local desktop has 2 network domains, one for the global interconnect (or in the LAN) and another is used to connection the desktop with the device using cable.

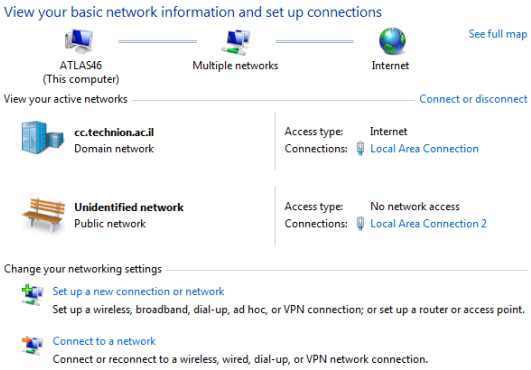
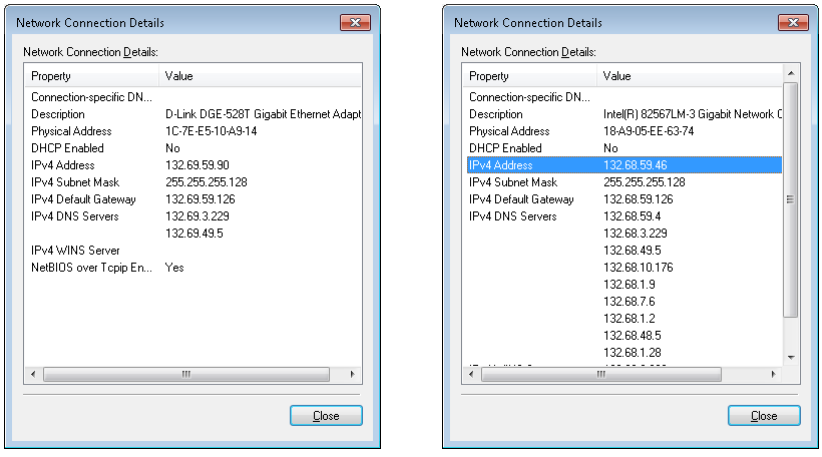


Figure 8: Desktop Network

First we need to config the IP of the second network.



(a) First IP (b) Second IP

Figure 9: 2 IPs after configuration

Note that it's better to config the IP address of the second network to be not in the same domain as the first IP address in case there is another computer in the first network domain that has the same IP address of the device and influence the communication between the desktop and the chassis. Here is the IP of the first network for this desktop. And we can see that the 2 IPs are not in the same domain.

2.2 Config the IP of the Chassis

We can use the Measurement & Automation Explorer to set the IP address of the chassis. To start this tool, we need to go to the LabView –> Tools.

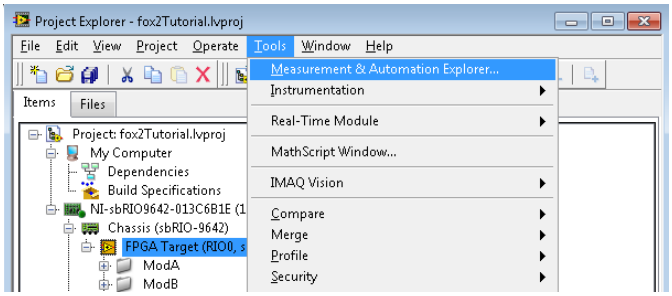


Figure 10: Start the LabView Tool to config the IP

Refer to the "Remote Systems" part to do the configuration.

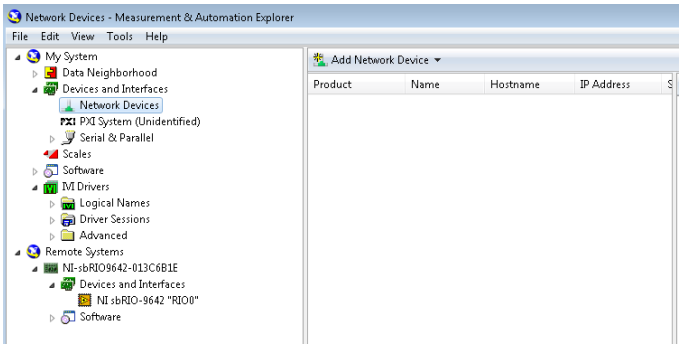


Figure 11: Measurement & Automation Explorer

And in this case, when the configuration is finished, you can go back to the first section to start the project.

3 FOX2 GUI

In this section, I will show how to make the GUI for our FOX2 application. I assume you have prepared the VHDL file for the FPGA in the board. The functionality for this digital controller is simpler than the digital controller in the FOX2 chip, which mainly focuses on writing data and reading data to the PCB board/FOX2 chip and some indicators, for example, the transmission has finished, etc.

3.1 Create new vi for the chassis

In order to get the GUI, we need to have a VI for the FPGA target (note that in this case, the vi doesn't belong to My computer).

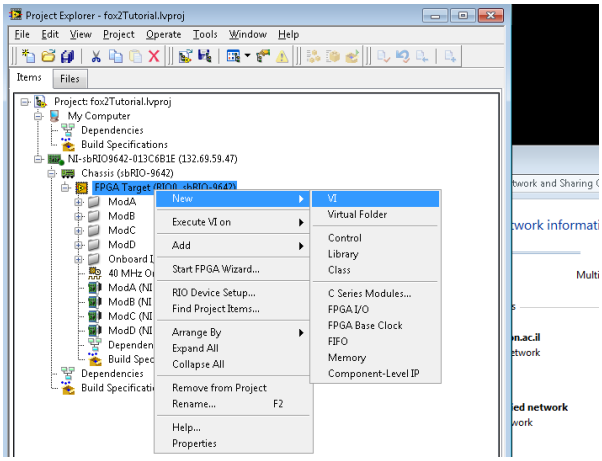


Figure 12: Create vi in LabView chassis

Then in the backend panel, add the IP Integration Node.

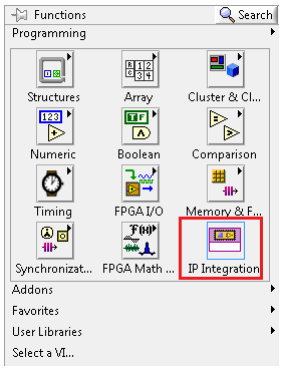


Figure 13: Add IP node in the backend panel

3.2 Config the IP node

We will add the digital controller of the FPGA in this IP block. After adding the Node to the backend panel, we need to double click on it.

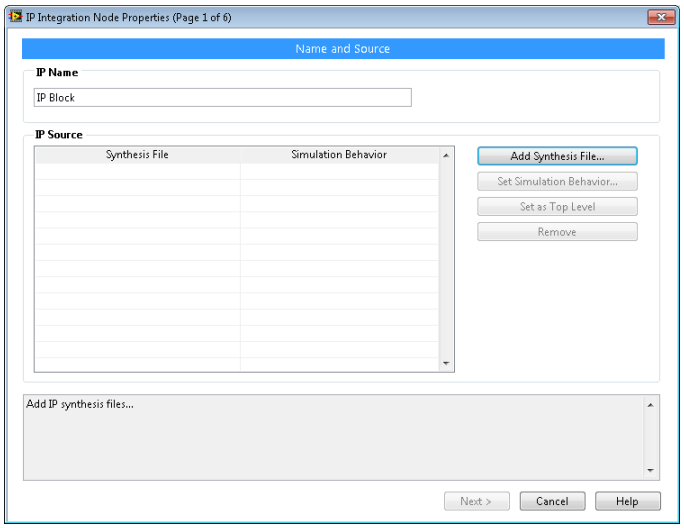


Figure 14: IP node property

Then we add the VHDL file by clicking "Add Synthesis File..."

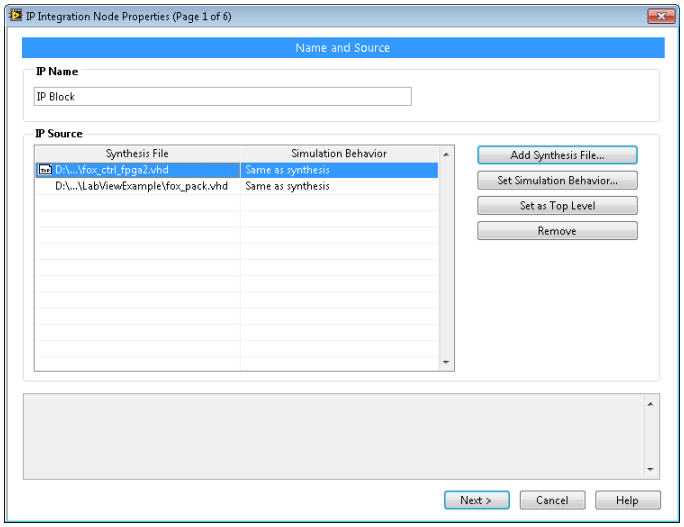


Figure 15: Add synthesis file

Here is the 2 VHDL files we had and I used fox_ctrl_fpga2.vhd as the top level

netlist and then click “next”.

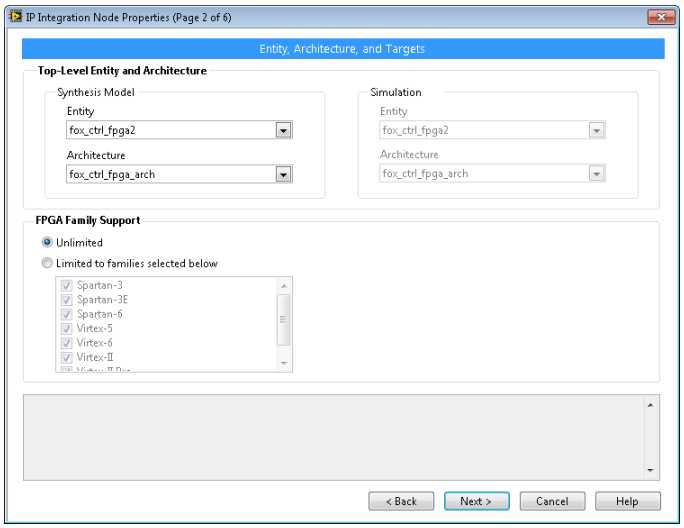


Figure 16: IP node property

Choose the correct “Entity” and “Architecture” to continue the configuration, the FPGA family for out chassis is Spartan-3, but we can set the FPGA Family Support to be “Unlimited” and then “Next” to check the VHDL file Syntax and then Generate Simulation model.

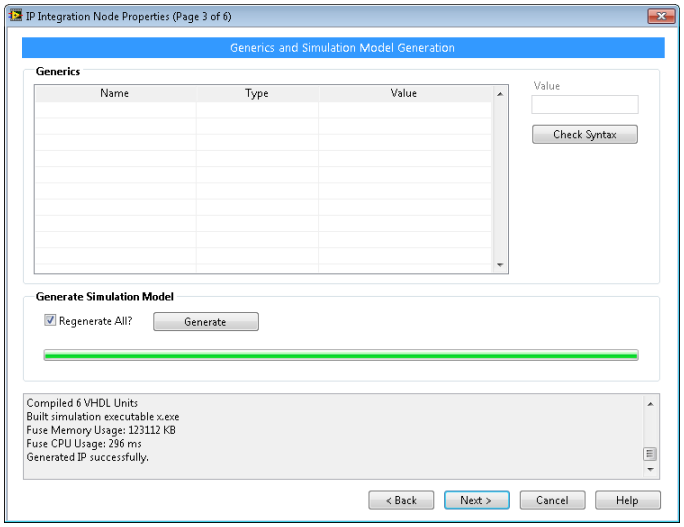


Figure 17: IP node property

After generating the IP successfully, we can continue by clicking “Next” and

set the clock signal. But no need to config the Derived multiple of single-cycle Timed Loop clock and the IP Enable signals.

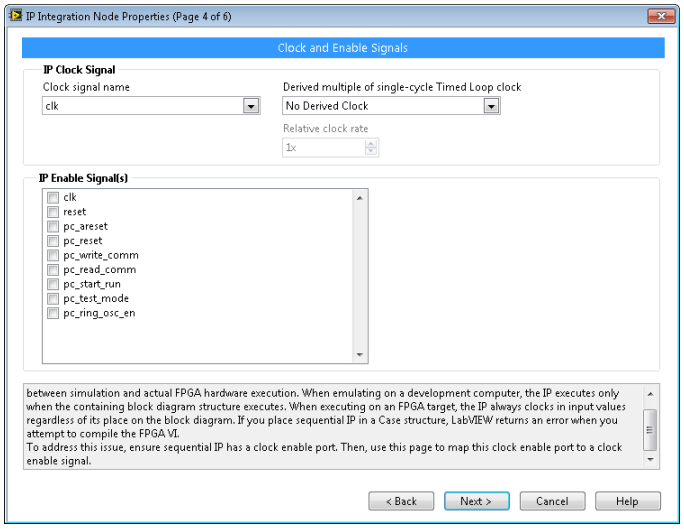


Figure 18: IP node property

Here I prefer not to config the reset signal.

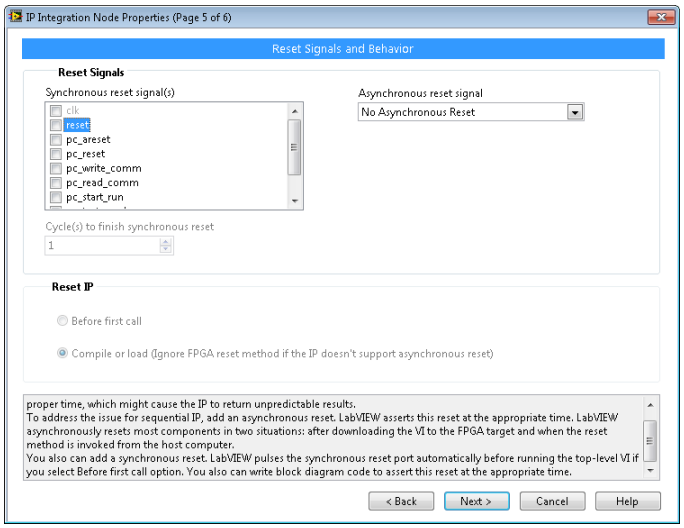


Figure 19: IP node property

Then we can see the terminal properties and can finish the config.

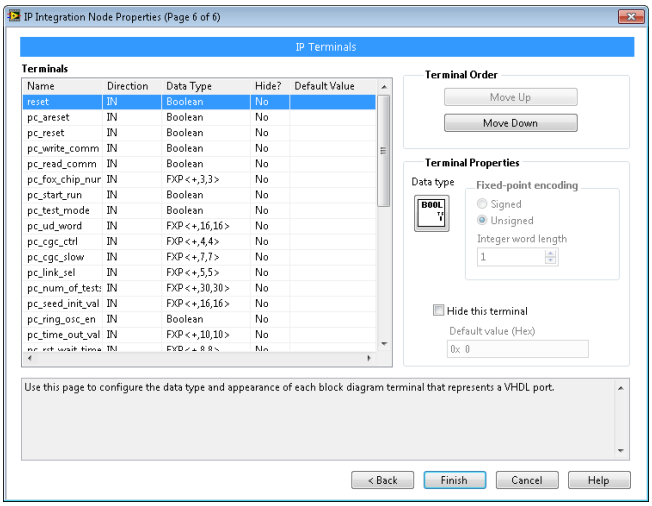


Figure 20: IP node property

We can get this block in the backend panel after the configuration.

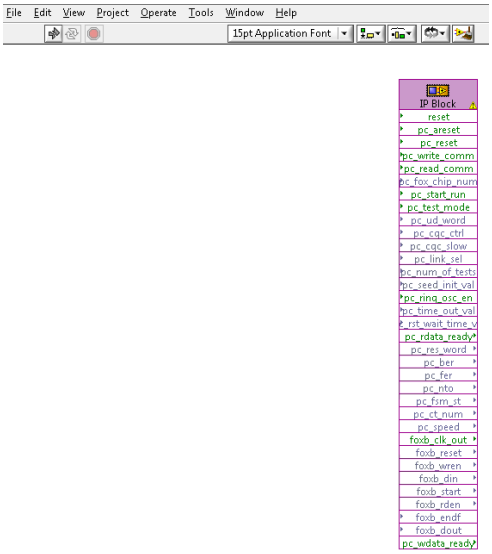


Figure 21: IP node in backend panel

3.3 Add add the control signals to the IP

Here we can see that even we finished the IP block configuration, we are not able to run the simulation yet since there is no input signal and output signals to the GUI.

The IP block has the digital netlist, and the Single-Cycle Timed Loop can support this simulation, so I will first add the SCTL structure at the backend panel.

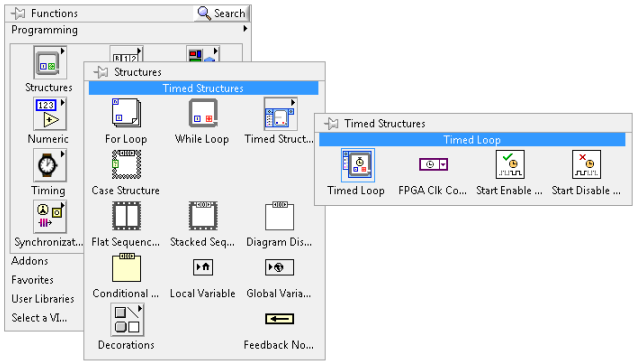


Figure 22: Add SCTL structure at backend panel

Then we are able to add the control signals to the backend panel and the corresponding buttons/text zone will appear at the front panel.

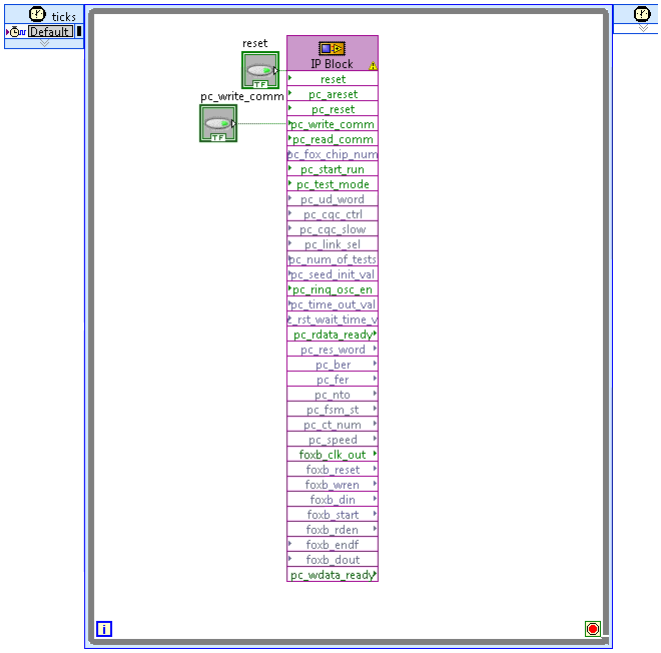


Figure 23: Add control signals to the IP node

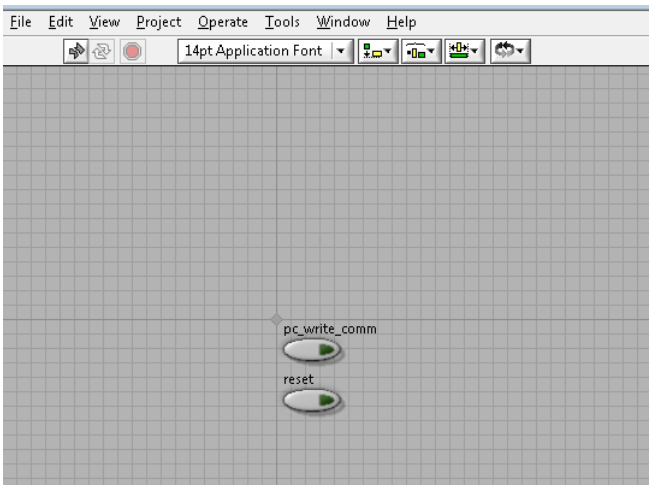


Figure 24: Control buttons at the front-end

When we finish adding the control signals and IOs to the IP block, we will be able to run the simulation. Here is the ready backend panel and front end panel.

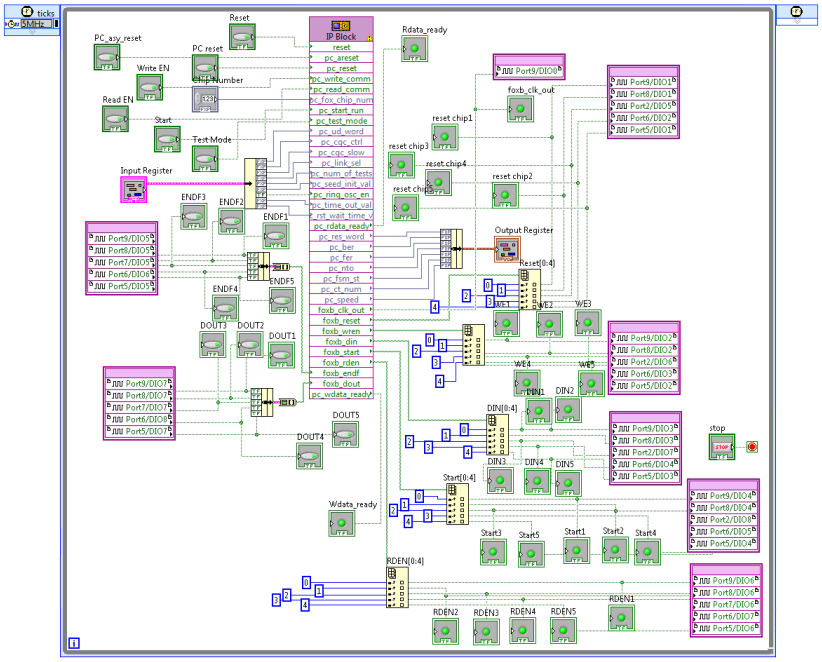


Figure 25: Ready backend panel

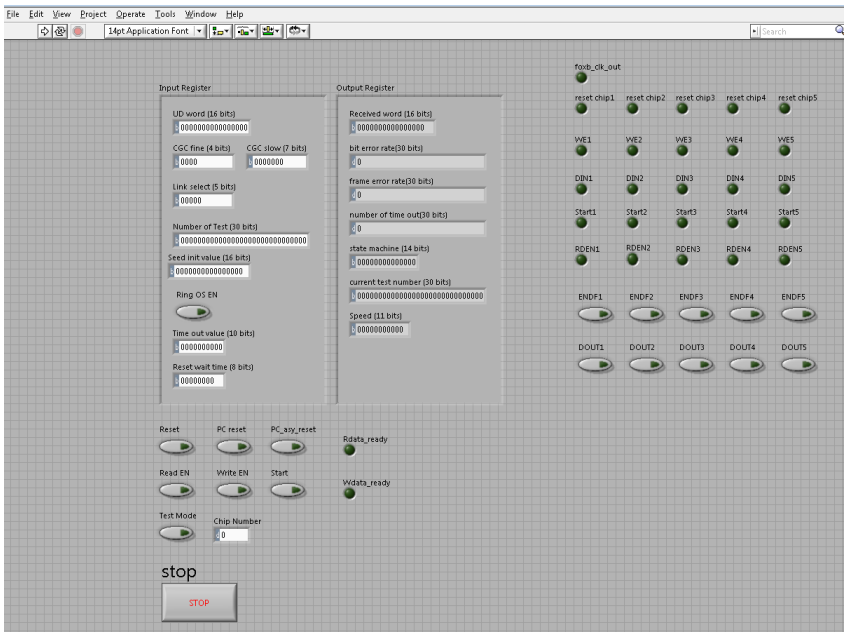


Figure 26: Ready front-end panel

Actually I cannot cover all the details of adding the control signals to the IN node. However, you can refer to my work when you need to create new. For example, how to bound the data, and use cluster, etc. But I think this manual will be enough for you to get started.