Vivado FPGA-PS Project Steps

Jason Losh, revised September 18, 2023

This document provides steps for making a Xilinx FPGA Processing Subsystem (PS) project in Vivado. These steps can be used to build the gpio example. For other projects, change references to gpio, gpio_system_top, ports, and interrupts as needed.

Connect the Xilinx XUP Blackboard

1. Connect a USB A to USB microB cable from the PC to the PROG/UART jack.
2. Connect a high current USB microB power supply to EXTP jack. Make sure the jumper selects the EXTP power source.

Create Project

3. Open Vivado
4. Select Quick Start > Create Project, Click Next
5. Name your project, Enter the location for the project (gpio), Click Next
6. Select RTL Project, Check Do not specify sources at this time, Click Next
7. Search for and select part: xc7z007sclg400-1, Click Next
8. Click Finish

Add Constraints File to Project

9. Select Project Manager > Add Sources, Select Add or create constraints
10. Click +, Create File…, File name: blackboard.xdc, Click OK, Click Finish
    (file will be at <*full_project_path*>/<gpio.srcs>/constrs_1/new)
11. Add the constraints to the constraints file (provided blackboard.xdc is an example)

Create PS Block Design

12. Select IP Integrator > Create Block Design, Design Name: system, Click OK
13. Block Design > Diagram > Press +, Select Zynq 7 Processing System
14. Copy BlackBoard_ps_presets.tcl to <*full_project_path*>/<gpio.srcs>/sources_1
15. Double click on Zynq 7 Processing System, Preset, Apply Configuration,
    and add file above, Click OK
16. Block Design > Diagram > Run Block Automation to add the PS DDR and fixed I/O
17. Block Design > Diagram > Press +, Select Processor System Reset
18. Block Design > Diagram > Run Connection Automation (connects clock and reset)

Create Your Custom IP

19. Tools > Create and Package New IP, Next
20. Select Create AXI4 Peripheral, Next
21. Enter Peripheral Details: Name: gpio, Display name: gpio, Description GPIO, Next
22. Enter Interfaces: Name: AXI Interface Type: Lite, Data Width (bits): 32,
    Number of Registers: 8, Next
23. Select Add IP to the repository, click Finish
24. Goto IP catalog, Search for gpio, Open gpio_v1.0, Right click, Open in IP Packager

25. Add ports to the wrapper and call the implementation module
    Design Sources > Open gpio_v1_0.v wrapper
        Add this port: input [31:0] gpio_data_in,
        Add this port: output [31:0] gpio_data_out,
        Add this port: output [31:0] gpio_data_oe,
        Add to gpio_v1_0_AXI instantiation: .gpio_data_in(gpio_data_in)
        Add to gpio_v1_0_AXI instantiation: .gpio_data_out(gpio_data_out)
        Add to gpio_v1_0_AXI instantiation: .gpio_data_oe(gpio_data_oe)
        Remove from gpio_v1_0_AXI instantiation: parameter C_AXI_DATA_WIDTH = 32,
        (these ports will connect to gpio_system_top where they will be routed
        to pins on the SoC device)
26. Add ports to module implementing the AXI4-lite device and the IP functionality
    Design Sources > Open gpio_v_1_0_AXI.v
        Add this port: input [31:0] gpio_data_in,
        Add this port: output [31:0] gpio_data_out,
        Add this port: output [31:0] gpio_data_oe,
        (these ports will connect to the wrapper)
27. Add the user logic in gpio_v_1_0_AXI.v
    (for the gpio project, it is suggested that you completely replace this file with the version from class which added the GPIO implementation and refactors the AXI register code)
28. Package IP > File Groups > Merge changes from File Groups Wizard
29. Package IP > Ports and Interfaces > Merge changes from Ports and Interfaces Wizard
30. Package IP > Addressing and Memory, Set Base Address (0 for gpio)
31. Package IP > Review and Package > Click Re-Package IP to close the IP module project
32. Select Project Manager > Synthesis > Run Synthesis
    (this step verifies that at least syntactically, this IP module is OK)

Use the Custom IP Block in Your Project

33. IP Integrator > Open Block Design > Click +, Search for gpio, Select gpio_v1.0,
34. Run Connection Automation (adds AXI Interconnect, connects AXI bus, reset, clock)
35. IP Integrator > Open Block Design > Click +, Search for concat, Select concat,
    Double-click on concat, expand to 3 input ports
36. Right click on gpio module > IRQ port, Make Connection, connect to In2 of concat
37. Right click on concat > dout port, Make Connection, connect to IRQ_F2P
38. Verify Address is 0x43C0 0000 in Block Design > Address Editor
39. Block Design > Diagram, Right-click on gpio_data_out, gpio_data_in, and gpio_data_oe ports, Make external, to export (remove the _0 suffices on the labels)
40. Block Design > Diagram, Click Validate Design button
41. IP Integrator > Generate Block Design, Click Generate button
42. Sources > Design Sources, Select system.bd, Create HDL wrapper

Add a Top Level Module

43. Select Project Manager > Add Sources, Select Add or create design sources
44. Click +, Create File…, File type: SystemVerilog, File name: *gpio_system_top*.sv,
    Click OK, Click Finish
    (file will be at <*full_project_path*>/<*project_name* .srcs>/sources_1/new)
45. Add ports to the source file for the blackboard.xdc file and the passthrough ports from the system wrapper.

46. Instantiate the system wrapper as system and connect the ports from gpio_system_top to system.

Run Synthesis and Implementation

47. Select Project Manager > Synthesis > Run Synthesis
48. Select Project Manager > Implementation > Run Implementation

Generate the Bitstream and Program the Device

49. Select Project Manager > Program and Debug > Generate Bitstream
50. Select Project Manager > Program and Debug > Open Hardware Manager
51. Select Hardware Manager > Open target > Auto Connect
52. Click on Program Device

Note: You can select Tools>Run Tcl Script with the provided synth_to_prog.tcl file to automate steps 47-52.