Creating an Accelerator

Zynq
Vivado HLS 2014.2 Version

This material exempt per Department of Commerce license exception TSU

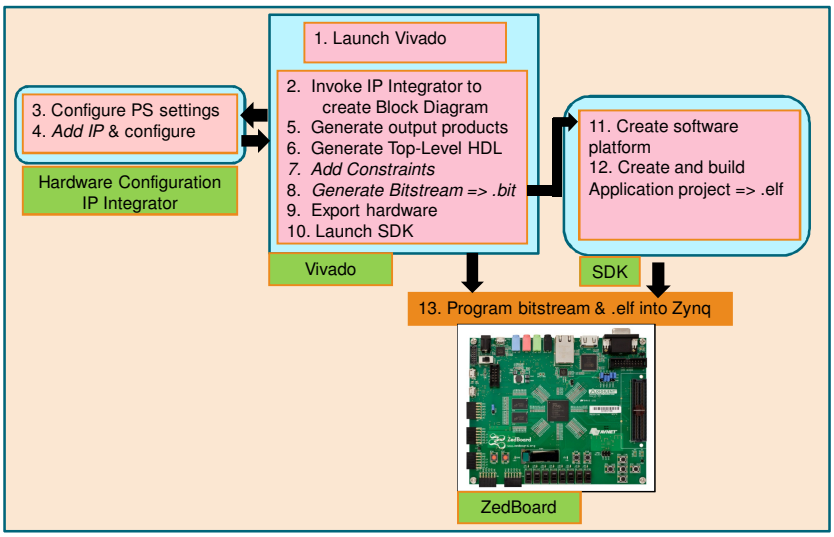
© Copyright 2014 Xilinx

Objectives

- **After completing this module, you will be able to:**
  - Describe embedded system development flow in Zynq using Vivado
  - List the steps involved in creating an hardware accelerator
  - State how an accelerator created in Vivado HLS is used in Vivado Design Suite

© Copyright 2014 Xilinx

Embedded System Design using Vivado



© Copyright 2014 Xilinx

Outline

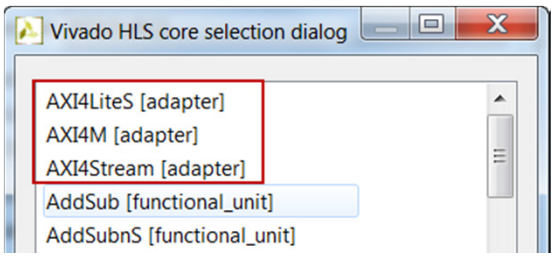
- ***Creating IP-XACT Adapters***
- **Integrating the IP-XACT Adapter in AXI System**
- **Summary**

© Copyright 2014 Xilinx

Adapter Cores

- **List of Adapter Cores for reference**

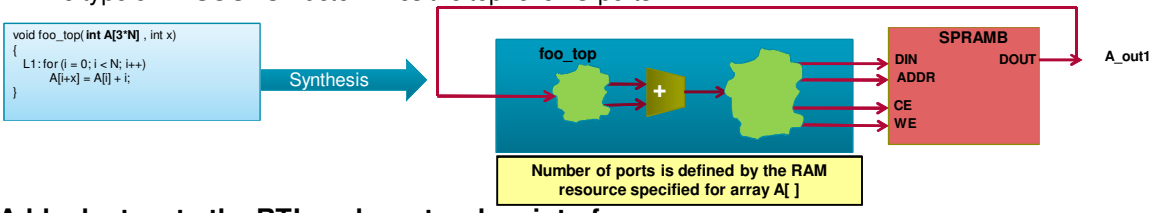
Type	Core Name	Description
AXI4	AXI4M	AXI4 Master interface
	AXI4Stream	AXI4 stream interface
	AXI4LiteS	AXI4Lite slave interface



© Copyright 2014 Xilinx

Using RESOURCE for external connections

- **When the array is an argument of the top-level function**
  - The array/RAM is “off-chip”
  - The type of RESOURCE determines the top-level IO ports

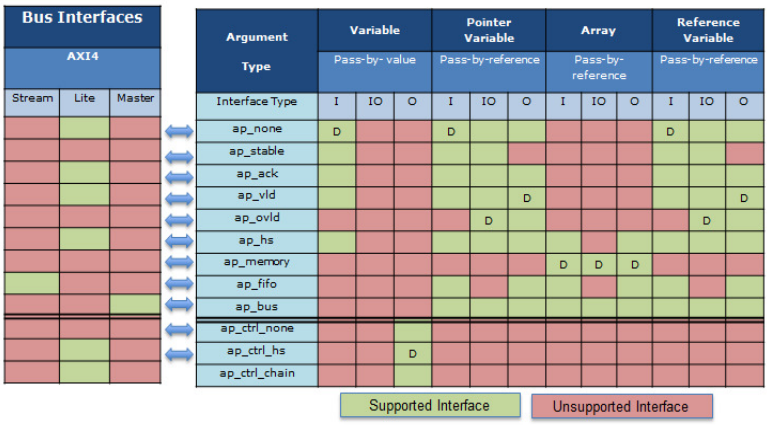


- **Add adapters to the RTL and create a bus interface**
  - The principle is the same
  - We define a resource for the port
    - Plus there are a few other options we’ll cover

© Copyright 2014 Xilinx

Adding Adapters 1) Check the RTL & C IO

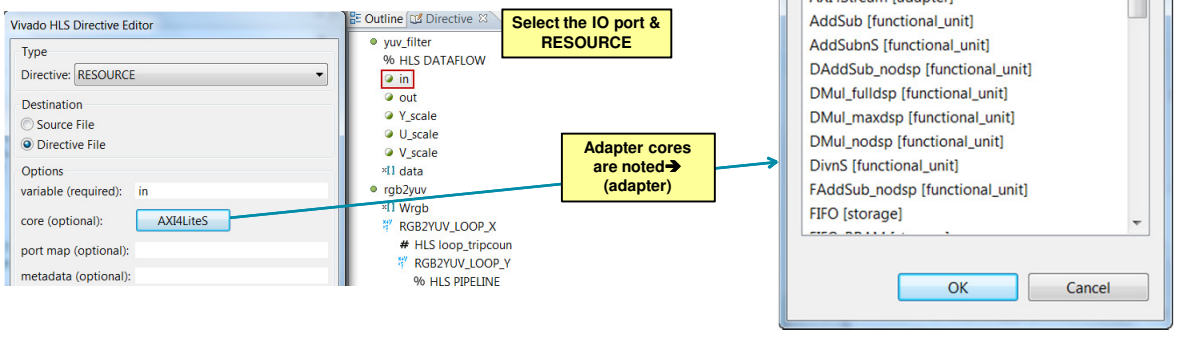
- **The first step in adding an adapter**
  - Make sure you have a suitable port on the RTL
  - Example,
    - If you want an AXI Stream interface → must have an ap_fifo on the RTL
    - If an ap_fifo on the RTL is needed → pass pointer variable, an array, or a reference variable (in, out, but not in/out)



© Copyright 2014 Xilinx

2) Assign an adapter Core to each Port

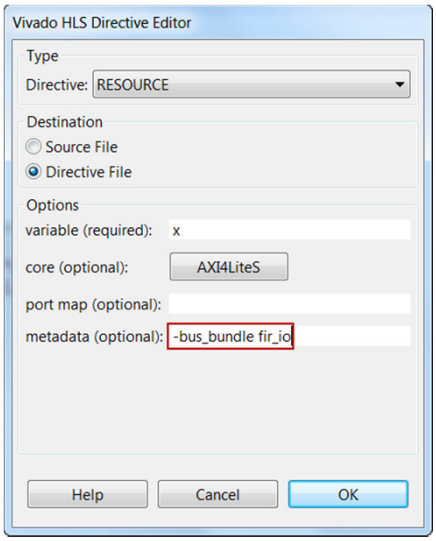
- **Once the I/O adapter is known**
  - Add the core as a resource on the RTL port
- **The cores are listed in the Vivado HLS Library Guide**
  - And available from the menu in the GUI



© Copyright 2014 Xilinx

3) Optionally group and rename buses

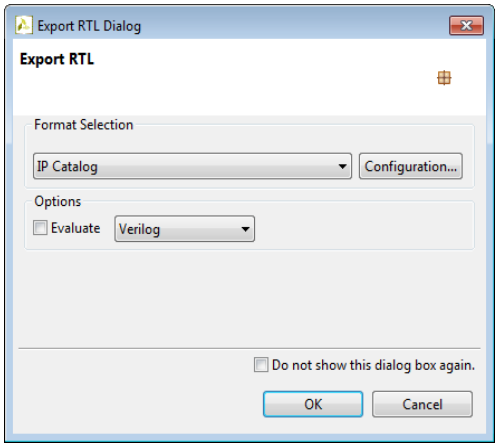
- **Metadata bundle buses**
  - Allow buses to be grouped with new names
  - Multiple resources can have the same name
    - Will be grouped into the same bus
  - In this example, port “x” will be
    - AXI4 Slave interface
    - And renamed fir_io
    - Any other adapter named fir_io will be grouped with this port
- **Port Name Mappings**
  - Optionally, port may be name mapped in the port map field
    - {{output} {output_hs_vld, output_ap_ack}}
  - The RTL port names are renamed on the adaper interface
    - In the above example, it will be renamed as output



© Copyright 2014 Xilinx

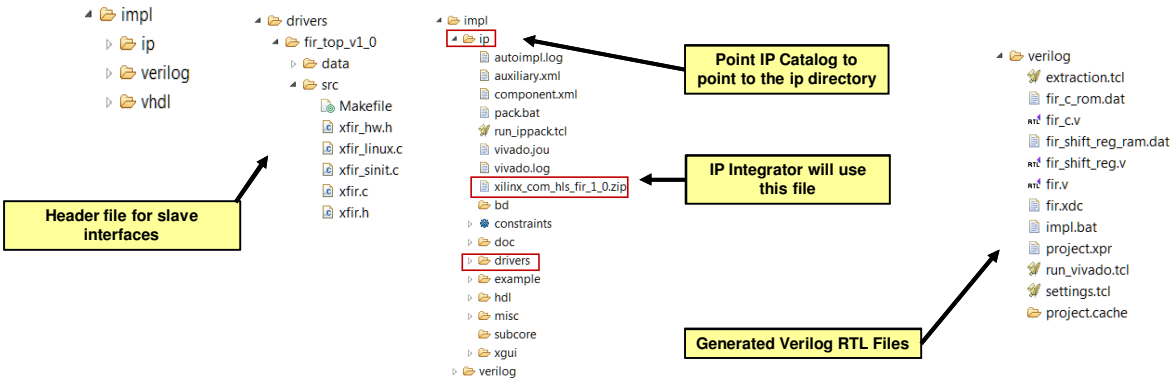
4) Generate the Adapter

- **Select Solution > Export RTL**
- **Select IP Catalog, System Generator for Vivado or PCore for EDK**
- **Click on Configuration… if you want to change the version number or other information**
  - Default is v1_00_a
- **Click on OK**
  - The directory (ip) will be generated under the impl folder under the current project directory and current solution
  - Only RTL code in Verilog will be generated, even if you select VHDL language



© Copyright 2014 Xilinx

Generated impl Directory



© Copyright 2014 Xilinx

Outline

- Creating IP-XACT Adapters
- *Integrating the IP-XACT Adapter in AXI System*
- Summary

© Copyright 2014 Xilinx

Integrate the IP-XACT Adapter in AXI System

- Create a new Vivado project, or open an existing project
- Set IP setting to point to the exported IP using Project Settings
- Invoke IP Integrator
- Construct(modify) the hardware portion of the embedded design by adding the IP-XACT adapter created in Vivado HLS
- Create (Update) top level HDL wrapper
- Synthesize and implement in Vivado
- Export the hardware description, and launch SDK

© Copyright 2014 Xilinx

Integrate the IP-XACT Adapter in AXI System

- Set software repository to point to the created IP if needed
- Create a new [or update an existing] software board support package to include the driver for the IP
- Create an application project in the SDK
- Compile the software with the GNU cross-compiler in SDK
- Download the programmable logic's completed bitstream
- Use SDK to download the program (the ELF file)

© Copyright 2014 Xilinx

Outline

- Creating IP-XACT Adapters
- Integrating the IP-XACT Adapter in AXI System
- *Summary*

© Copyright 2014 Xilinx

Summary

- **Vivado HLS provides wide support of AXI interfaces, System Generator design, and Pcore for EDK**
  - Assign as an external resource, just like a RAM
  - The choice of adapter is a function of the C variable type (pointer, etc.)
- **Start with the correct C argument type**
  - Verify the design at the C level
  - Accept the default block-level I/O protocol
  - Select the port-level I/O protocol that gives the required pcore adapter interface
  - Specify the port to have the appropriate adapter RESOURCE
  - Optionally group and rename ports
- **Export the design**
- **Add the IP in Vivado's IP Integrator**

© Copyright 2014 Xilinx