Create a custom IP Block with AXI Interface

http://www.fpgadeveloper.com/2014/08/creating-a-custom-ipblock-in-vivado.html

Dr. Heinz Rongen Forschungszentrum Jülich GmbH Zentralinstitut Systeme der Elektronik (ZEA-2) <u>H.Rongen@fz-juelich.de</u>

Create IP Block

- Create a basic SOC Design •
 - Create Block diagram / Add IP / ZYNQ7 PS
 - Run Block Automation
- Now: Tools → Create and package IP
- Click Next
- Click "Create a new AXI4 peripheral"
- Give a name to your new IP Block, Description and location on the disc 🚴 Create and Package New IP



🯄 📄 🔳

् 🔀 🍃

Flow Navigator

Project Mana

Project \$

👌 Add Sou

IP Catale



microzed_custom_ip - [E:/Github/fpgadeveloper/microzed-custom-ip/Vivado/microzed_cu

Configure IP Block / AXI interface

- Configure the IP Block, the AXI bus interface
 - AXI Lite, a Slave, Bus width 32 bit (defaults are ok for this example)



- The next page is a summary
 - Select "Edit IP"
 - Click Finish

Another Vivado window will now open

| A Create and Package N | New IP |
|------------------------|--|
| | Create Peripheral |
| | Peripheral Generation Summary |
| | 1. IP (xilinx.com:user:my_multiplier:1.0) with <u>1_interface(s)</u> |
| | 2. Driver(v1_00_a) and testapp more info |
| | 3. AXI4 BFM Simulation demonstration design more info |
| | 4. AXI4 Debug Hardware Simulation demonstration design more info |
| | Peripheral created will be available in the catalog : |
| | E:/Github/fpgadeveloper/microzed-custom-ip/ip_repo |
| | Next Steps: |
| | C Add IP to the repository |
| | © Edit IP |
| | C Verify peripheral IP using AXI4 BFM Simulation interface |
| | C Verify peripheral IP using JTAG interface |
| | |
| VIVADO. | Click Finish to continue |
| | < Back New Concert |

Open the VHDL source

A second Vivado instance is open, to modify (describe) our IP Block

| and the second | | | | | | | | 12,101 |
|--|--|--------------------------|--|--|---|--|---|--|
| avigator (1 | Project Manager - edt_HR_MyIP1_v1_0 | | | | | | | × |
| | Sources | | X Project Summary X S Package IP - HR_HyIP1 X | | | 02 × | | |
| niect Manager | - ペニキ(単合)単置 | IP Packag | ing Steps 🛛 🔍 | IP Identification | | | | more info |
| Project Settings | Design Sources (2) | | Vendor: | | kfa-jueich de | | | |
| Add Sources P Catalog P Catalog Processor Provement Create Block Design Generate Block Design | () () () () () () () () () () () () () (| 100 | ✓ IP Compatibility | Lbrary: Name: Version: Daploy name: Description: | user | | | |
| | Constraints (i) - Constraints (ii) - Constraints | | | | HR MyIP1 | | | |
| | | V IP Fie | t Groupe | | 10 July 1 | | | |
| | | 🖌 IP Ca | stonization Parameters | | 1.0 | | | |
| | | V IP Por | rts and Interfaces | | HR_MyP1_v1.0 | | | |
| | | V IP Ad | dressing and Memory | | My new AXI IP | | | |
| | Hierarchy Lbraries Comple Order | | ✓ IP G.II Customization Company un | | nei | | | |
| | Sources P Templates | V IP GJ | | | | | | |
| | Properties | D L ^a X Revie | w and Package | Categories | AXI. Peripheral | | | |
| G Seulation Secondaria | + + 2 k | | Rot directory: d:/redisardevancies/or resolv_myin1.1.0 | | | | | |
| Analysis | | | | Root directory: Xml file name: | d:/tedboardexamples/p_repo/ d:/tedboardexamples/p_repo/ | hr_myip1_1.0 hr_myip1_1.0/component.x | m | |
| Analysis Den Baborated C thesis Synthesis Settings Ann Synthesis | | | | Root directory: 3ml file name: | 5:/redboardexamples.jp_repol 5:/redboardexamples.jp_repol | hr_myio1_1.0 hr_myip1_1.0/component.a | ni | |
| Analysis Pen Elaborated C sthesis Synthesis Settings Run Synthesis Coen Synthesized | Desgrifuns | | | Root drectory: 3ml file name: | d:/redboardexamples/ip_repol d:/redboardexamples/ip_repol | hr_myp1_L0(component.x | mi | - 0 ¢ X |
| Analysis Coren Elaborated C stratesis Synthesis Settings Pan Synthesis Coren Synthesized Jementation | Design Runs Name Constrants With | S THS WHS THS | 6 TPWS Faled Routes | Koot drectory: 3ml file name: UUT PF BRAM | 5:/redboarde:camples.jb_repol 5:/redboarde:camples.jb_repol 5:/redboarde:camples.jb_repol | hr_myp1_1.0 hr_myp1_1.0(cmponent.x Bapsed Status | eri Progress | - D & X Stategy |
| Analysis Analysis Coren Elaborated E thesis Synthesis Settings An Synthesis Coren Synthesized Jerrentation Coren Synthesized Jerrentation Se | Desgn Runs Mane Constraints WM Server 1 constraints | S THIS WHIS THIS | 5 TPWS Paled Routes | Root directory: 3ml file name: | d:/redbaarder.amplextjp_repol d:/redbaarder.amplextjp_repol d:/redbaarder.amplextjp_repol | hr_myp1_L.0 hr_myp1_L.0(component.x Bagsed Status Nut statute | Progress | - D & X Strategy Vivedo Synthesio Default |
| Analysis Analysis Brown Eaborated C sthesis Synthesis Settings Pan Synthesis Corn Synthesis Setterstation Setterstation Setterstation Set Pan Implementation Set | Desgrifuns Canstrants WM Stare Constraints WM Desgrifun Desgrifuns Desgrifuns Constraints WM Desgrifuns Desgrifuns Constraints WM | S THE WHE THE | 5 TPWS Paled Routes | Root drectory: 3ml file name: LUT PF BRAM | 6:/redbaarder.amplexilip_repol 6:/redbaarder.amplexilip_repol DSP Start | hr_myb1_1.6 fr_myb1_1.6(component.s Biopsed Status Nut started Not started | Progress | - C & X Strategy Vivedo Synthese Defaulte Vivedo Implementation D |
| Analysis Coen Elaborated C thesis Synthesis Settings Man Synthesis Coen Synthesis Coen Synthesis Encomposition Coen Synthesis Para Inglementato Coen Synthesis Coen | Descri Funs Name Constraints White Image: Synth_1 constra_1 Image: Synth_1 constra_1 | S THE WHS THE | TPWS Paled Routes | Root directory: Xeri file name: | 6.//edboarder.ampiekilo_repol 6.//edboarder.ampiekilo_repol 0.99 Start | hr Jmyb L.1.6 hr Jmyb L.1.6/component.a Elepsed Status Not started | Progress 0% | - D & X Strategy Wixedo Synthesis Default Wixedo Inglementation D |
| Analysis Com Babonated C thesis G Synthesis Settings Ann Synthesis Com Synthesis Com Synthesis Professional Setting Professional Professional Com Synthesis Professional Professional Com Synthesis Professional P | Design Funds | 5 TH5 NH6 TH5 | TPWS Paled Routes | Root directory: Jimi file name: | 6.//wdbaarder.ampiesito_repol dr./zetboarder.ampiesito_repol DSP Start | Pr_myo L_1.6 Pr_myo L_1.6/component.x Elected Status Not started | Progress 0% | - C & X Strategy Wrado Syntemic Default Wrado Inplementation Dr |
| Aralysis Com Baboxated C thesis Com Synthesis Settings Ann Synthesis Com Synthesised internation Com Synthesized internation P Ann Implementato P Ann Implementato Com Synthesized Com Synthesized Co | Desgn Runs Constraints WM State Propia constraint Propia constrain | S THS WHS THS | TPWS Paled Routes | Root directory: Juri file name: | 6:/redbaarder.amplexilg.repol 6:/redbaarder.amplexilg.repol DSP Start | In yoyo L.L.A In yoyo L.L.A (component.x Biosed Status Not started Not started | ed Progress 2% | - C 🖉 X Stateg Wvado Singlementation Dr Wvado Singlementation Dr |
| Analysis Analysis Com Elaborated C thesis Synthesis Settings In Aur Synthesis In Com Synthesis Aur Inplementator Aur Inplementator Aur Inplementator Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Synthesis Aurophysics Aurophysics Synthesis Aurophysics Auroph | Design Runs Name Constraints WMS Set Syndt_1 constra_1 Set Syndt_1 constra_3 Id Id Id | S THE WHIS THE | 5 TPWS Paled Routes | Reot directory: Just file name: | 6://edbaardexamplexitg.repol 6:/redbaardexamplexitg.repol DSP Start | In ynys LLâ In ynys LLâ(component.) Biosed Status Net started Net started | Progress 0% | - C (2 × Statecy Waado Synthese Default Waado Inglementation D |

- In the Flow Navigator
- Click "Add Source"
- Click on "Add or Create Design Sources"



Next

Define Entity

- Select VHDL
- Type a name for the VHDL source
- Click OK
- Click "Finish" (in the page before)

| Create a n | ew source file and add it to your project | | | | | |
|------------------------|---|----|--------|--|--|--|
| le type, name | and location | | | | | |
| File type: | WHDL | | | | | |
| File name: | HrMult_16x16_32 | | | | | |
| Fil <u>e</u> location: | 👸 <local project="" to=""></local> | | - | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | OK | Cancel | | | |

VHDL module creation wizard

- Define the entity
- as traditional VHDL

| Ports with biar | nk names will n | iot be w | intten. | | | |
|---------------------|-----------------|----------|---------|-----|-----|----------|
| Entity name: | HrMult_16x1 | 6_32 | | | | |
| Architecture name: | Behavioral | | | | | |
| I/O Port Definition | s | | | | | |
| Port Name | Directio | n | Bus | MSB | LSB | |
| Clk | in | ~ | | 0 | 0 | |
| A | in | ~ | ✓ | 15 | 0 | + |
| В | in | ~ | - | 15 | 0 | \times |
| Y | out | ~ | ✓ | 31 | 0 | + |
| | | | | | | 1 |
| | | | | | | 1 |



Instantiate the VHDL code (our behavior) inside the IP Core

• Double click on the "name_AXI_inst"

Declaration

- Find the line with the "begin" keyword
- add the following code just **above** it (in the VHDL declaration part)
- and declare the multiplier component

```
signal multiplier_out : std_logic_vector(31 downto 0);
component multiplier
port (
   clk: in std_logic;
   a: in std_logic_VECTOR(15 downto 0);
   b: in std_logic_VECTOR(15 downto 0);
   p: out std_logic_VECTOR(31 downto 0));
end component;
```

Instantiation

 Now find the line that says "- Add user logic here" and add the following code below it to instantiate the multiplier:

```
multiplier_0 : multiplier
port map (
    clk => S_AXI_ACLK,
    a => slv_reg0(31 downto 16),
    b => slv_reg0(15 downto 0),
    p => multiplier_out);
```

- Find this line of code "reg_data_out <= slv_reg1;"
- and replace it with "reg_data_out <= multiplier_out;".
- In the process statement just a few lines above, replace "slv_reg1" with "multiplier_out".
- Save the file.

You should notice that the "multiplier.vhd" file has been integrated into the hierarchy because we have instantiated it from within the peripheral.

VHDL source code for multiplier

Find VHDL code

https://github.com/fpgadeveloper/microzed-custom-ip/blob/master/Vivado/ip repo/my multiplier 1.0/src/multiplier.vhd

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std logic unsigned.all;
entity multiplier is
 port(
    clk : in std_logic;
    a : in std logic vector(15 downto 0);
      : in std_logic_vector(15 downto 0);
    b
    р
        : out std_logic_vector(31 downto 0)
  );
end multiplier;
architecture IMP of multiplier is
begin
  process (clk)
  begin
    if clk'event and clk = '1' then
      p <= a * b;
    end if;
  end process;
end IMP;
```