Lab #8: Verify Custom IP

05/03/2018

4190.309A: Hardware System Design
(Spring 2018)

Notations

- HOST$ XXX
  - Type XXX at the terminal of your Ubuntu-PC.
- BOARD$ YYY
  - Type YYY at the terminal of ZedBoard (on minicom).
- TCL$ ZZZ
  - Type ZZZ at the Tcl console of Vivado.

Overview

- Step-by-step procedure: implementing IP into the Linux system
  - Simulation verification
    - Test vector generation
    - Simulation with the BRAM model
  - Implementation
    - Instantiate the baseline block design
    - Edit custom IP
    - C-program test

- Get the source code before get into the practice

  -
    ```
    HOST$ git clone https://github.com/K16DIABLO/HSD_LAB8.git
    ```

Simulation Verification

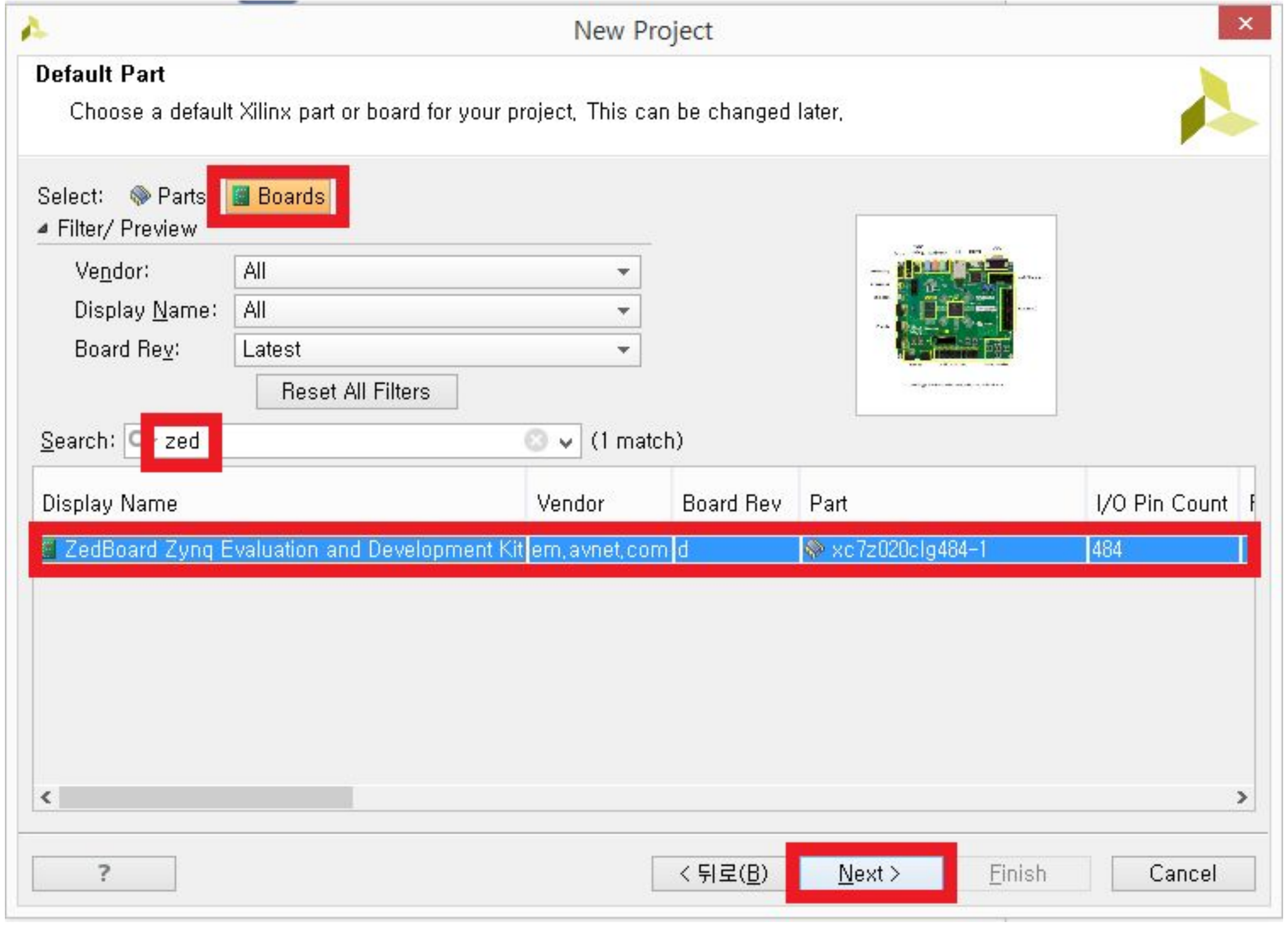
Test Vector Generation

- Get the sample test vector generator
  - `HOST$ cd $(git_clone_dir)/sw_1`
  - `HOST$ make`
    - See the sample test vectors ‘input.txt’ and the reference result ‘output.txt’
  - `HOST$ make debug`
    - Check the full intermediate results

Method	Simulation C-program	Simulation Verilog	Execution FPGA
Result	40889A69 4.268849	?	?

Create Vivado project

- Choose part or board
  - We are going to use ZedBoard



New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: Parts Boards

Filter/ Preview

Vendor: All

Display Name: All

Board Rev: Latest

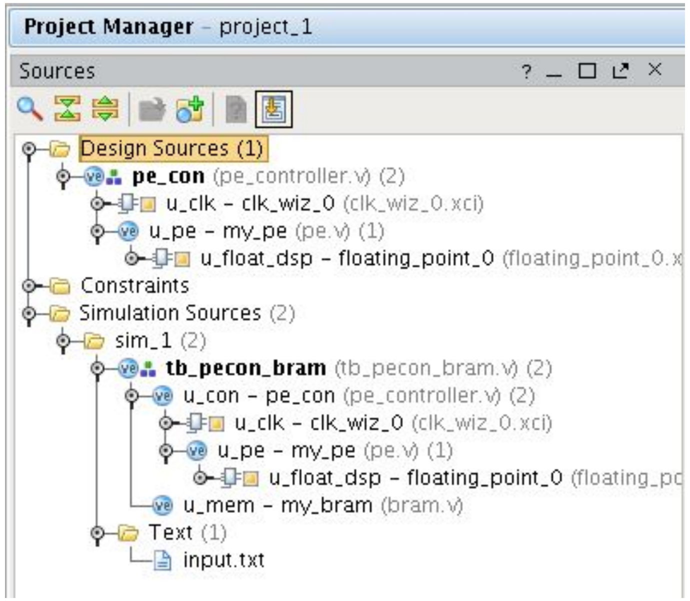
Reset All Filters

Search: zed (1 match)

Display Name	Vendor	Board Rev	Part	I/O Pin Count
ZedBoard Zynq Evaluation and Development Kit	em.avnet.com	d	xc7z020clg484-1	484

Simulation with the BRAM Model

- See the given lab4 code the TA sample code
  - `HOST$ cd $(git_clone_dir)/hw_1` <- *given LAB4 code*
  - `HOST$ cd $(git_clone_dir)/hw_2` <- *modified by TA*
    - Modification log)
      - Add BRAM interface
      - Add clock wizard IP
      - Add testbench
- Create Vivado project and import all hw_2 files
  - tb_pecon_bram.v, pe_controller.v, pe.v, bram.v, input.txt
  - Change input/output file directory path to your own environment.



Project Manager - project_1

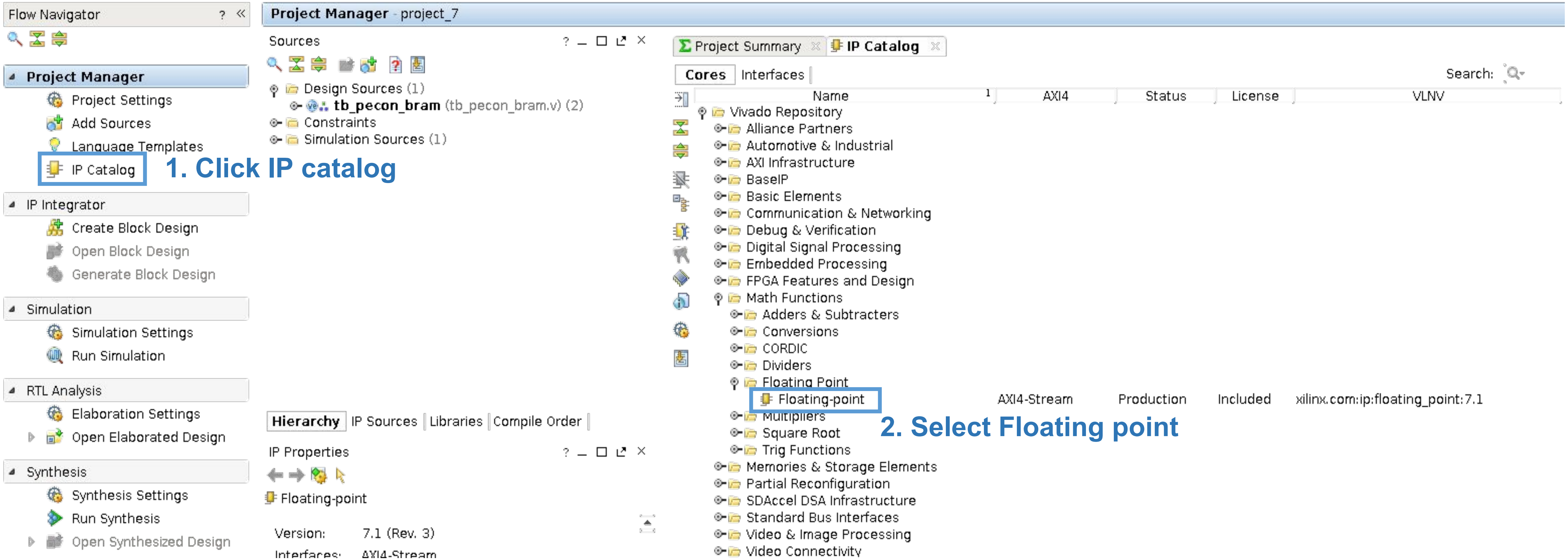
Sources

```
Design Sources (1)
  pe_con (pe_controller.v) (2)
    u_clk - clk_wiz_0 (clk_wiz_0.xci)
    u_pe - my_pe (pe.v) (1)
      u_float_dsp - floating_point_0 (floating_point_0.x
Constraints
Simulation Sources (2)
  sim_1 (2)
    tb_pecon_bram (tb_pecon_bram.v) (2)
      u_con - pe_con (pe_controller.v) (2)
        u_clk - clk_wiz_0 (clk_wiz_0.xci)
        u_pe - my_pe (pe.v) (1)
          u_float_dsp - floating_point_0 (floating_pc
      u_mem - my_bram (bram.v)
    Text (1)
      input.txt
```

- Generate missing IPs
  - floating_point_0, clk_wiz_0 (see next page)

Floating Point IP

- Click IP Catalog -> Math Functions -> Floating point



1. Click IP catalog

2. Select Floating point

Flow Navigator

- Project Manager
  - Project Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP Integrator
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- Simulation
  - Simulation Settings
  - Run Simulation
- RTL Analysis
  - Elaboration Settings
  - Open Elaborated Design
- Synthesis
  - Synthesis Settings
  - Run Synthesis
  - Open Synthesized Design

Project Manager - project_7

Sources

- Design Sources (1)
  - **tb_pecon_bram** (tb_pecon_bram.v) (2)
- Constraints
- Simulation Sources (1)

Hierarchy | IP Sources | Libraries | Compile Order

IP Properties

Floating-point

Version: 7.1 (Rev. 3)
Interfaces: AXI4-Stream

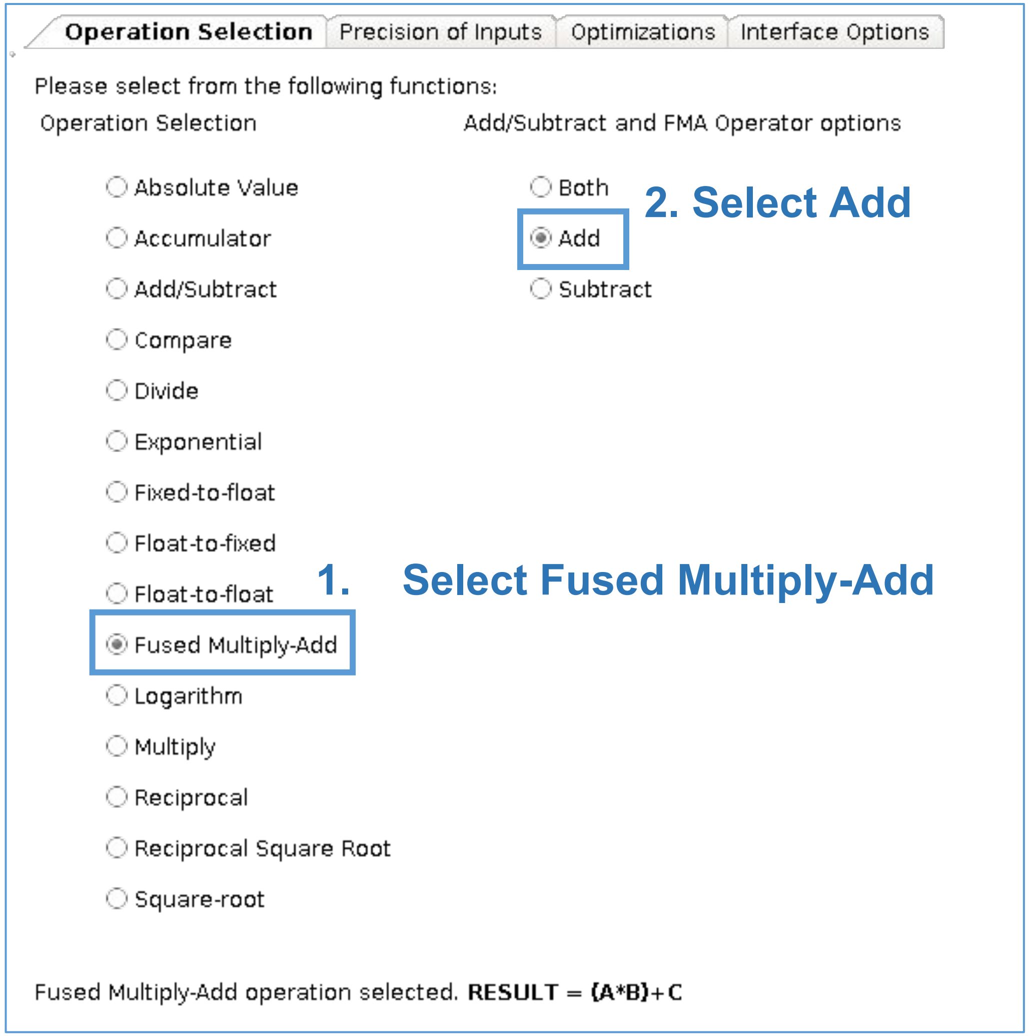
Project Summary | IP Catalog

Cores | Interfaces

Search:

Name	AXI4	Status	License	VLNV
Vivado Repository				
Alliance Partners				
Automotive & Industrial				
AXI Infrastructure				
BaseIP				
Basic Elements				
Communication & Networking				
Debug & Verification				
Digital Signal Processing				
Embedded Processing				
FPGA Features and Design				
Math Functions				
Adders & Subtracters				
Conversions				
CORDIC				
Dividers				
Floating Point				
Floating-point	AXI4-Stream	Production	Included	xilinx.com:ip:floating_point:7.1
Multipliers				
Square Root				
Trig Functions				
Memories & Storage Elements				
Partial Reconfiguration				
SDAccel DSA Infrastructure				
Standard Bus Interfaces				
Video & Image Processing				
Video Connectivity				

Floating Point IP



Operation Selection | Precision of Inputs | Optimizations | Interface Options

Please select from the following functions:

Operation Selection

○ Absolute Value
○ Accumulator
○ Add/Subtract
○ Compare
○ Divide
○ Exponential
○ Fixed-to-float
○ Float-to-fixed
○ Float-to-float
● Fused Multiply-Add
○ Logarithm
○ Multiply
○ Reciprocal
○ Reciprocal Square Root
○ Square-root

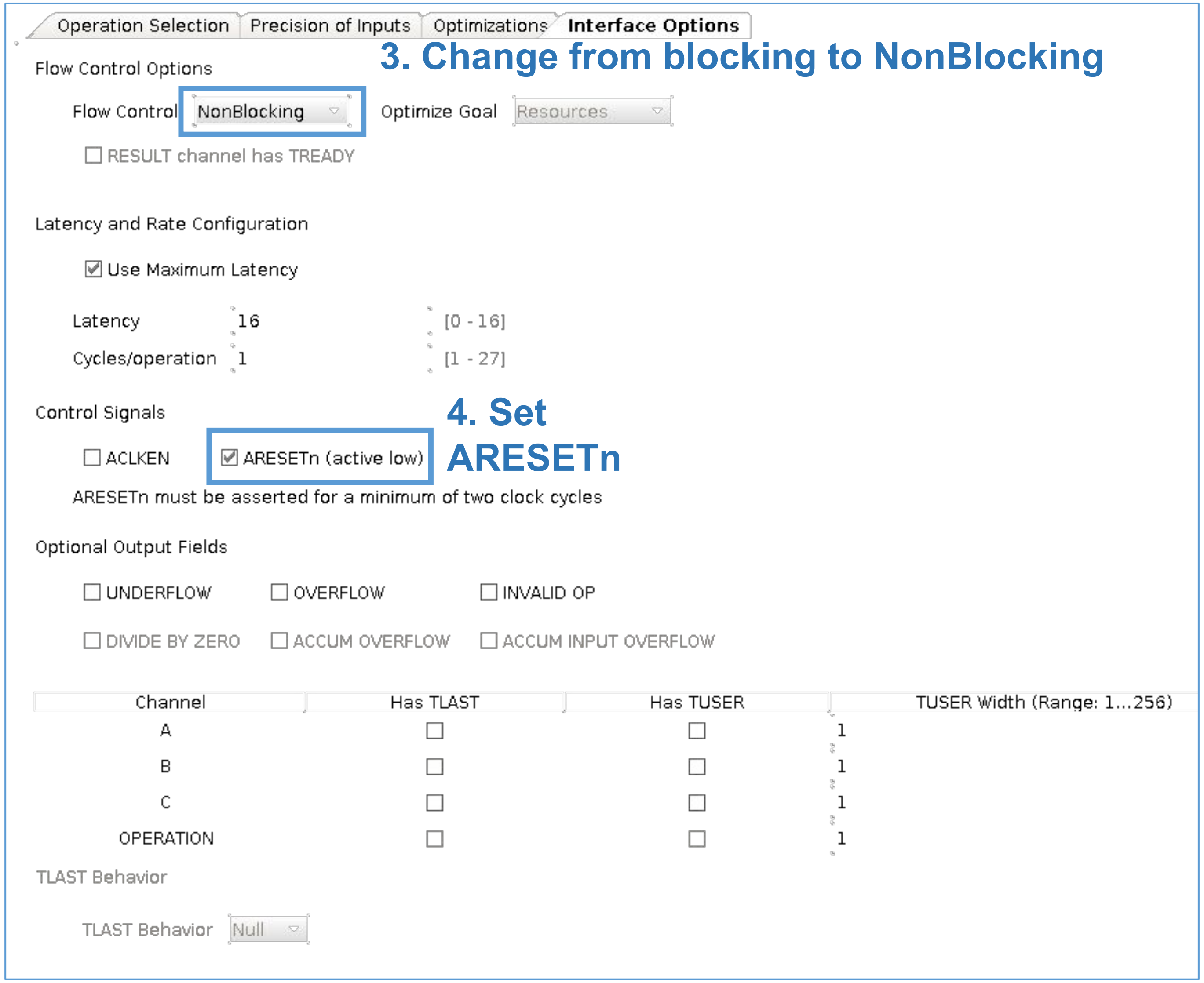
Add/Subtract and FMA Operator options

○ Both
● Add
○ Subtract

2. Select Add

1. Select Fused Multiply-Add

Fused Multiply-Add operation selected. **RESULT = (A*B)+C**



Operation Selection | Precision of Inputs | Optimizations | Interface Options

3. Change from blocking to NonBlocking

Flow Control Options

Flow Control: NonBlocking   Optimize Goal: Resources

□ RESULT channel has TREADY

Latency and Rate Configuration

☑ Use Maximum Latency

Latency: 16   [0 - 16]

Cycles/operation: 1   [1 - 27]

Control Signals

4. Set ARESETn

□ ACLKEN   ☑ ARESETn (active low)

ARESETn must be asserted for a minimum of two clock cycles

Optional Output Fields

□ UNDERFLOW   □ OVERFLOW   □ INVALID OP

□ DIVIDE BY ZERO   □ ACCUM OVERFLOW   □ ACCUM INPUT OVERFLOW

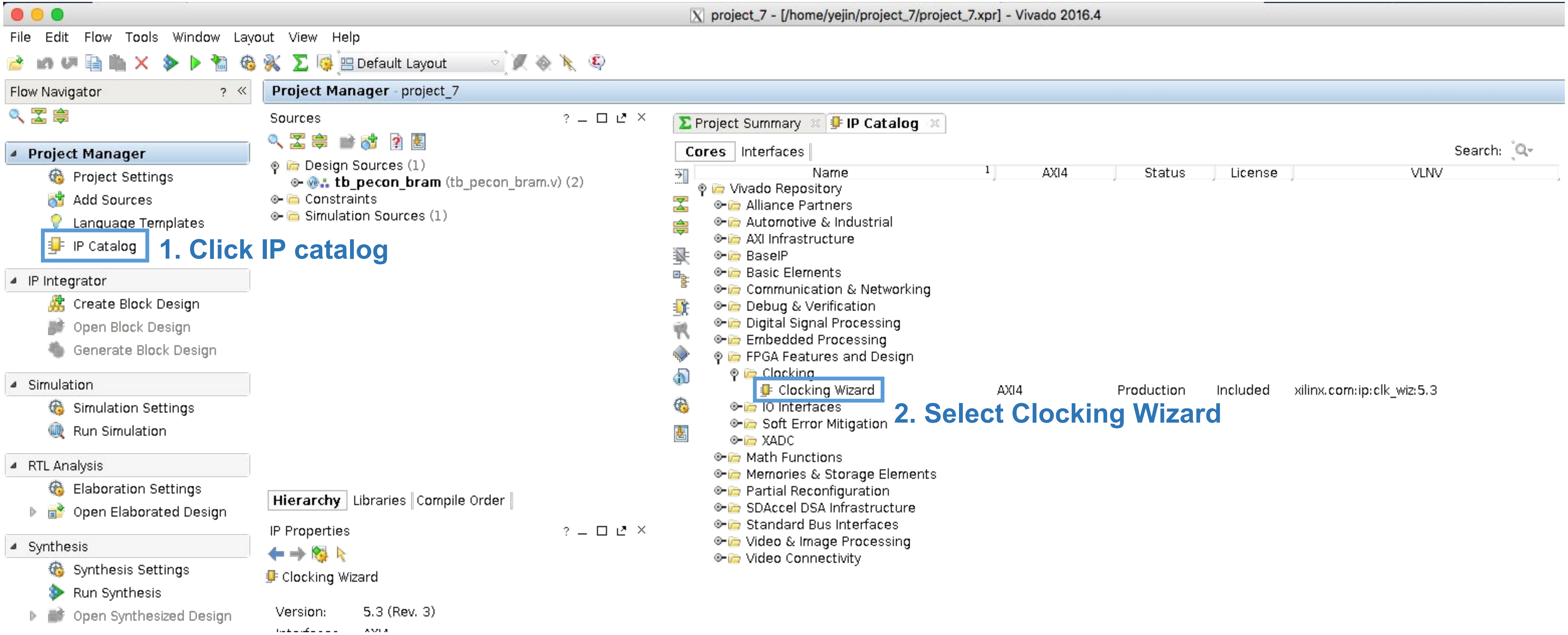
Channel	Has TLAST	Has TUSER	TUSER Width (Range: 1...256)
A	□	□	1
B	□	□	1
C	□	□	1
OPERATION	□	□	1

TLAST Behavior

TLAST Behavior: Null

Clocking Wizard IP

- Click IP Catalog -> FPGA Features and Design -> Clocking Wizard



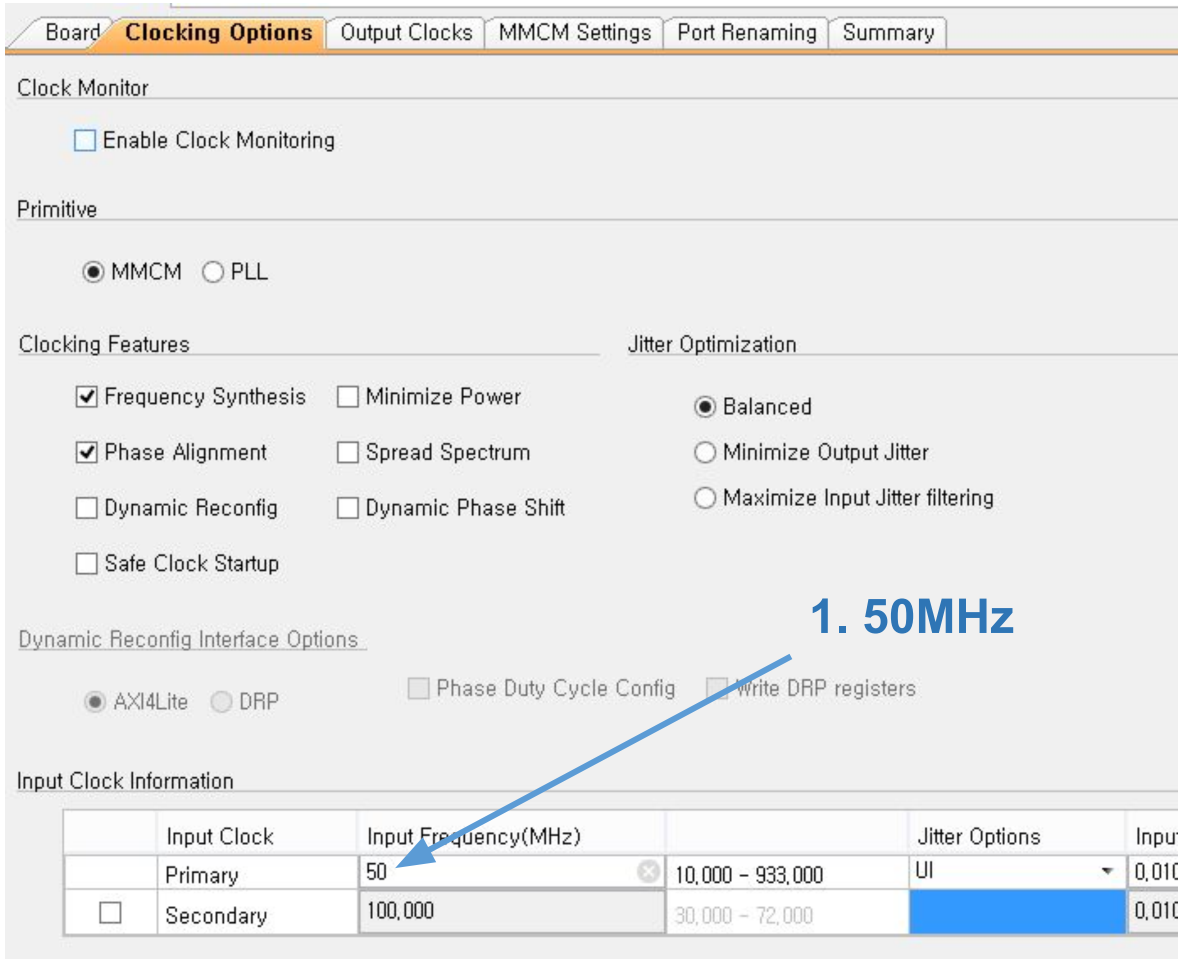
Name	AXI4	Status	License	VLNV
Clocking Wizard	AXI4	Production	Included	xilinx.com:ip:clk_wiz:5.3

Version: 5.3 (Rev. 3)

1. Click IP catalog

2. Select Clocking Wizard

Clocking Wizard IP



Clock Monitor

□ Enable Clock Monitoring

Primitive

● MMCM ○ PLL

Clocking Features

☑ Frequency Synthesis □ Minimize Power

☑ Phase Alignment □ Spread Spectrum

□ Dynamic Reconfig □ Dynamic Phase Shift

□ Safe Clock Startup

Jitter Optimization

● Balanced

○ Minimize Output Jitter

○ Maximize Input Jitter filtering

Dynamic Reconfig Interface Options

● AXI4Lite ○ DRP

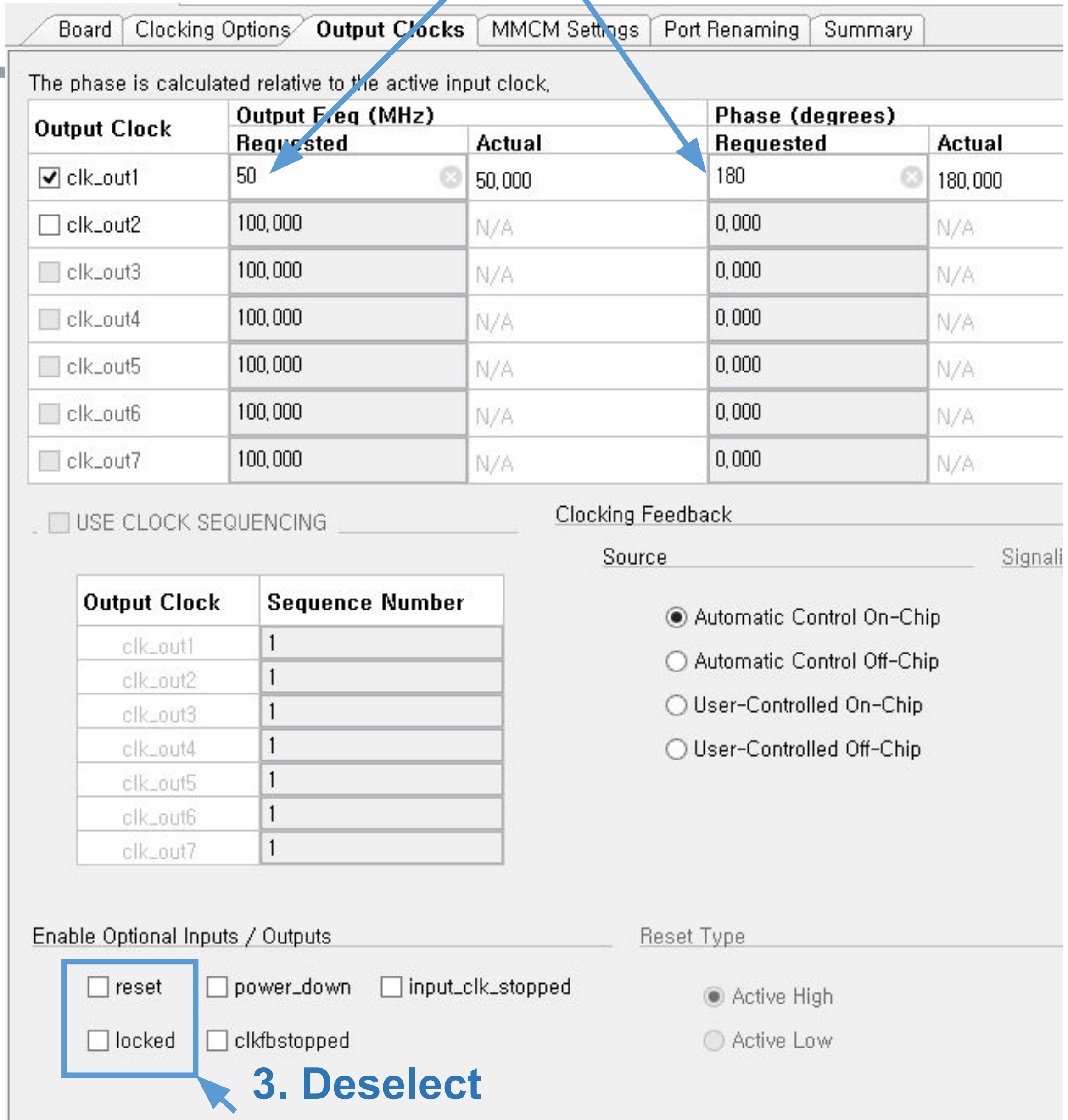
□ Phase Duty Cycle Config □ Write DRP registers

Input Clock Information

	Input Clock	Input Frequency(MHz)		Jitter Options	Inpu
	Primary	50	10,000 - 933,000	UI	0,010
□	Secondary	100,000	30,000 - 72,000		0,010

1. 50MHz

2. 50MHz, 180°



The phase is calculated relative to the active input clock.

Output Clock	Output Freq (MHz)		Phase (degrees)	
	Requested	Actual	Requested	Actual
☑ clk_out1	50	50,000	180	180,000
□ clk_out2	100,000	N/A	0,000	N/A
□ clk_out3	100,000	N/A	0,000	N/A
□ clk_out4	100,000	N/A	0,000	N/A
□ clk_out5	100,000	N/A	0,000	N/A
□ clk_out6	100,000	N/A	0,000	N/A
□ clk_out7	100,000	N/A	0,000	N/A

□ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Clocking Feedback

Source

Signali

● Automatic Control On-Chip

○ Automatic Control Off-Chip

○ User-Controlled On-Chip

○ User-Controlled Off-Chip

Enable Optional Inputs / Outputs

□ reset □ power_down □ input_clk_stopped

□ locked □ clkfbstopped

Reset Type

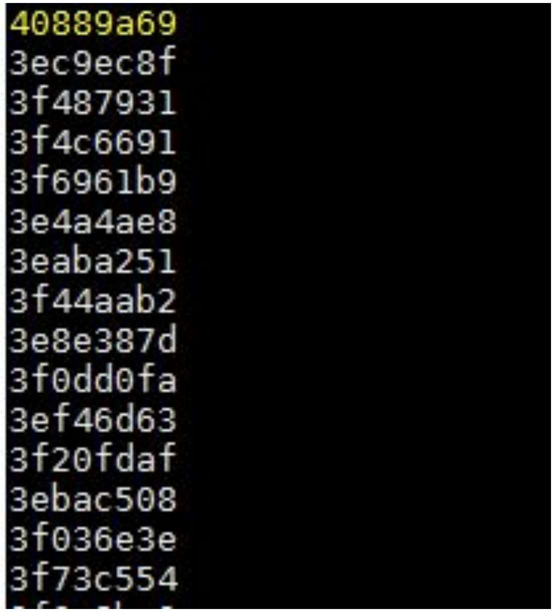
● Active High

○ Active Low

3. Deselect reset & locked

Simulation with the BRAM Model

- Get the behavioral simulation results and check output.txt
  - Result is written to address 0x0
  - Check the results from the different calculator
    - C program
    - Verilog simulator



```
40889a69
3ec9ec8f
3f487931
3f4c6691
3f6961b9
3e4a4ae8
3eaba251
3f44aab2
3e8e387d
3f0dd0fa
3ef46d63
3f20fdaf
3ebac508
3f036e3e
3f73c554
```

Method	Simulation C-program	Simulation Verilog	Execution FPGA
Result	40889A69 4.268849	40889A69	?

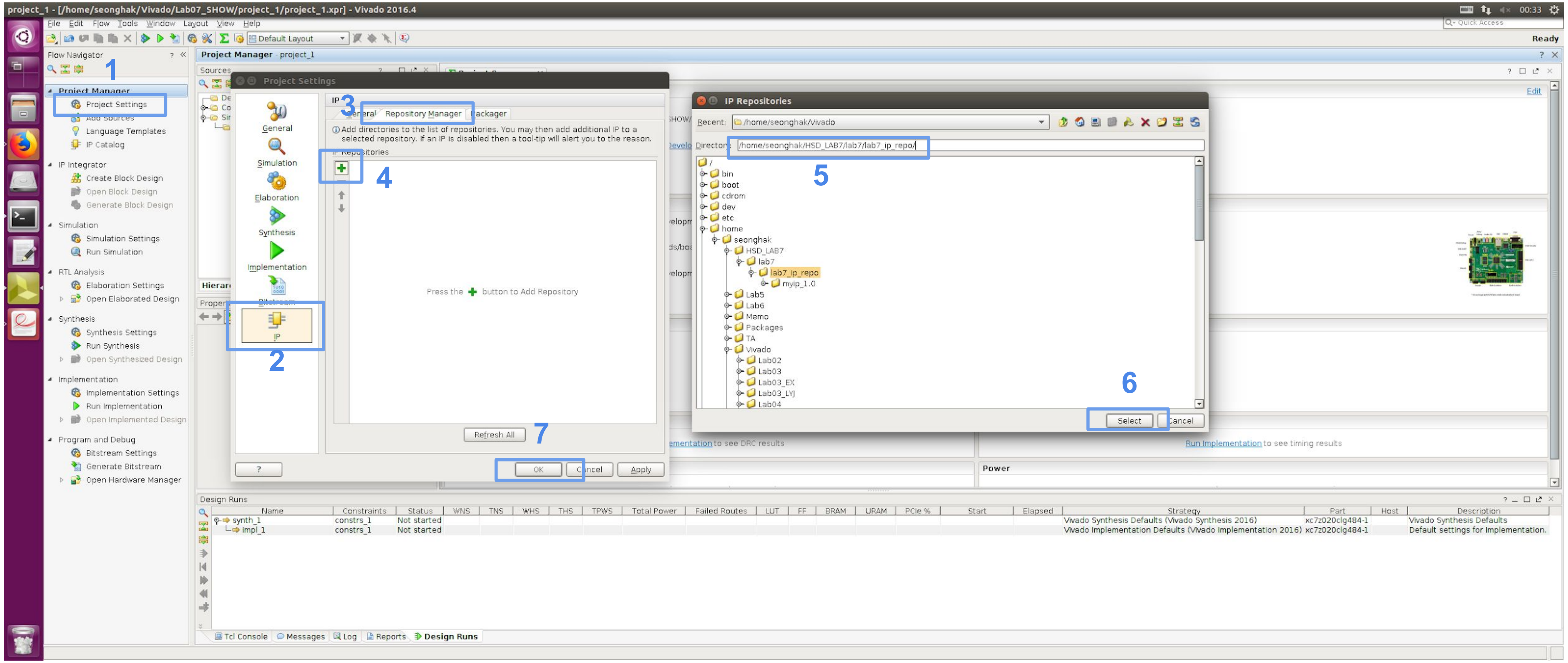
Implementation

Baseline Block Design

- You can get lab7 block design at:
  - `HOST$ cd $(git_clone_dir)/hw_3`

- Create another Vivado project and instantiate the block design
  - Add ip repository `$(git_clone_dir)/hw_3/lab7_ip_repo`
  - `TCL$ cd $(git_clone_dir)/hw_3`
  - `TCL$ source block_design.tcl`

Review : Apply Block Design (1)



project_1 - [/home/seonghak/Vivado/Lab07_SHOW/project_1/project_1.xpr] - Vivado 2016.4

Project Manager - project_1

Project Settings

IP

ⓘ Add directories to the list of repositories. You may then add additional IP to a selected repository. If an IP is disabled then a tool-tip will alert you to the reason.

IP Repositories

Recent: /home/seonghak/Vivado

Directory: /home/seonghak/HSD_LAB7/lab7/lab7_ip_repo/

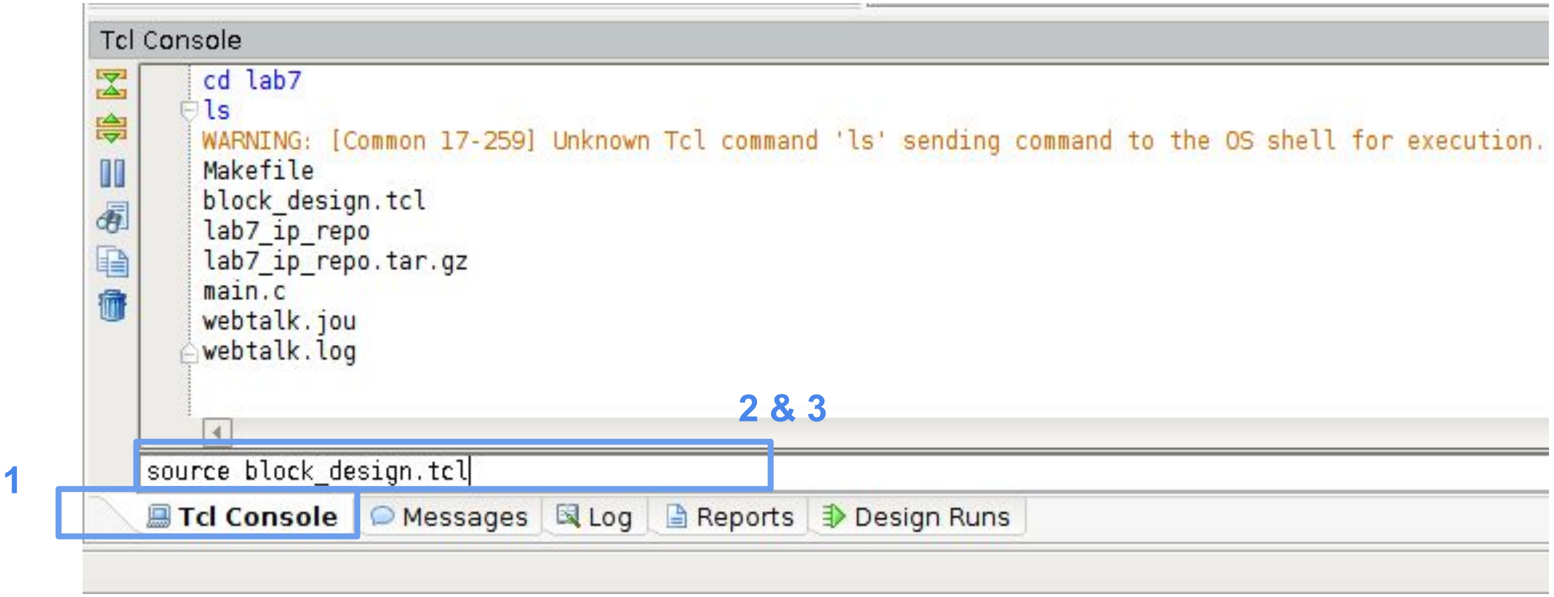
```
/
  bin
  boot
  cdrom
  dev
  etc
  home
    seonghak
      HSD_LAB7
        lab7
          lab7_ip_repo
            myip_1.0
      Lab5
      Lab6
      Memo
      Packages
      TA
      Vivado
        Lab02
        Lab03
        Lab03_EX
        Lab03_LYJ
        Lab04
```

Design Runs

Name	Constraints	Status	Strategy	Part	Description
synth_1	constrs_1	Not started	Vivado Synthesis Defaults (Vivado Synthesis 2016)	xc7z020clg484-1	Vivado Synthesis Defaults
impl_1	constrs_1	Not started	Vivado Implementation Defaults (Vivado Implementation 2016)	xc7z020clg484-1	Default settings for Implementation.

Review : Apply Block Design (2)

- Execute TCL scripts of the example block design.
  1. Select Tcl Console tab
  2. `TCL$ cd $(git_clone_dir)/hw_3`
  3. `TCL$ source block_design.tcl`



```
cd lab7
ls
WARNING: [Common 17-259] Unknown Tcl command 'ls' sending command to the OS shell for execution.
Makefile
block_design.tcl
lab7_ip_repo
lab7_ip_repo.tar.gz
main.c
webtalk.jou
webtalk.log
```

```
source block_design.tcl
```

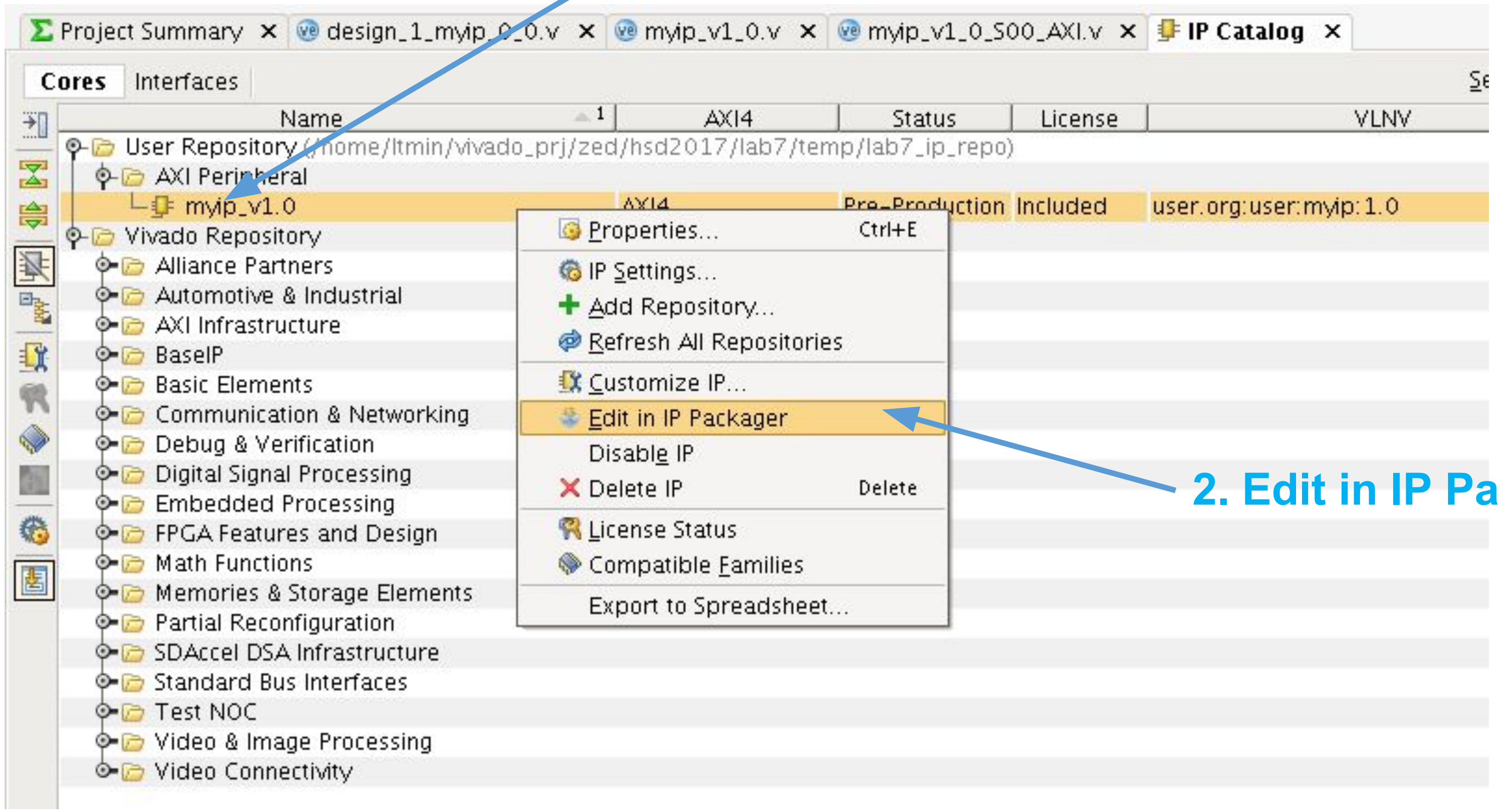
Editing Custom IP (1)

- See the sample IP by TA
  - Baseline custom IP: Lab7 shifter
  - The sample IP by TA: `$(git_clone_dir)/hw_3/lab7_ip_repo`
    - Modification log)
      - Import all source files
      - Instantiate pe_controller in the myip
      - Add BRAM_EN, BRAM_RST, …
      - Add start/done edge detectors

Editing Custom IP (2)

- IP Catalog -> Edit in IP Packager -> IP project

1. Right click



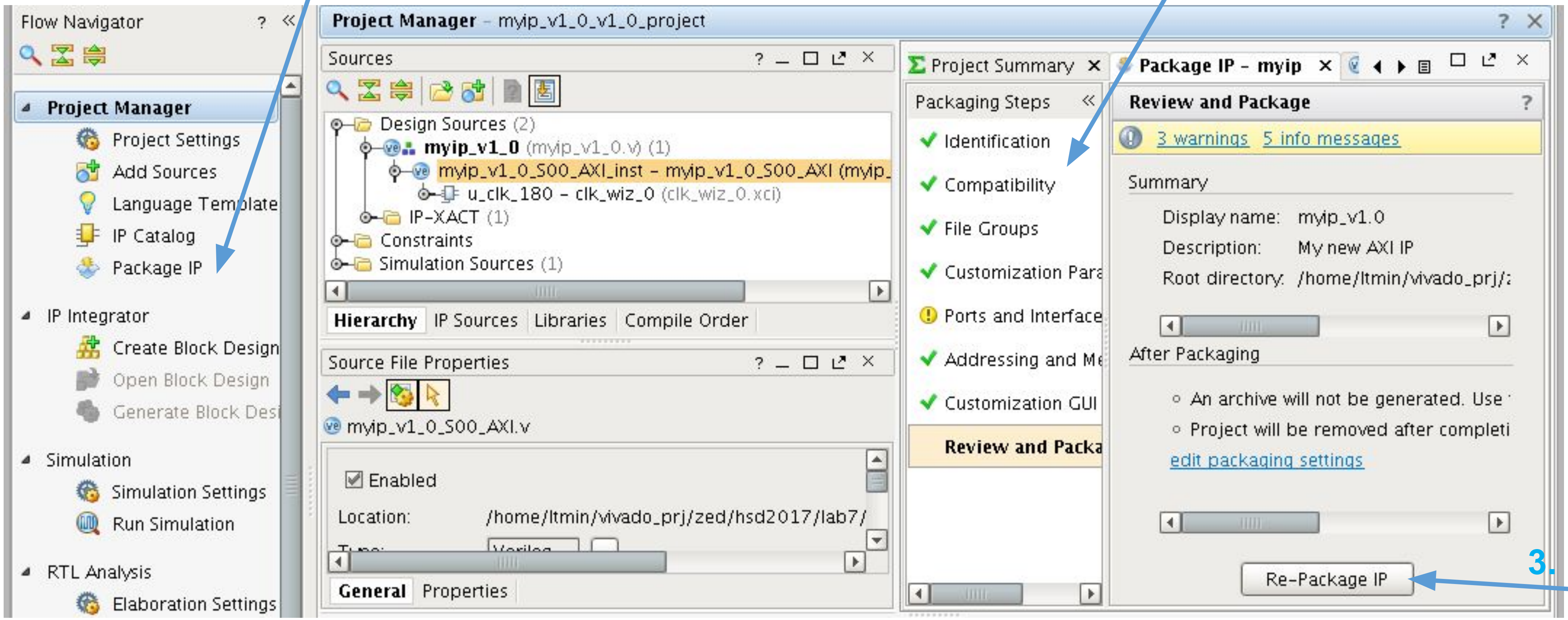
2. Edit in IP Packager

Editing Custom IP (3)

- Package IP -> Configurations -> Re-Package IP

1

2. Check IP Configurations



☑ Enabled

Location: /home/ltmin/vivado_prj/zed/hsd2017/lab7/

Type: Verilog

3 warnings 5 info messages

Display name: myip_v1.0

Description: My new AXI IP

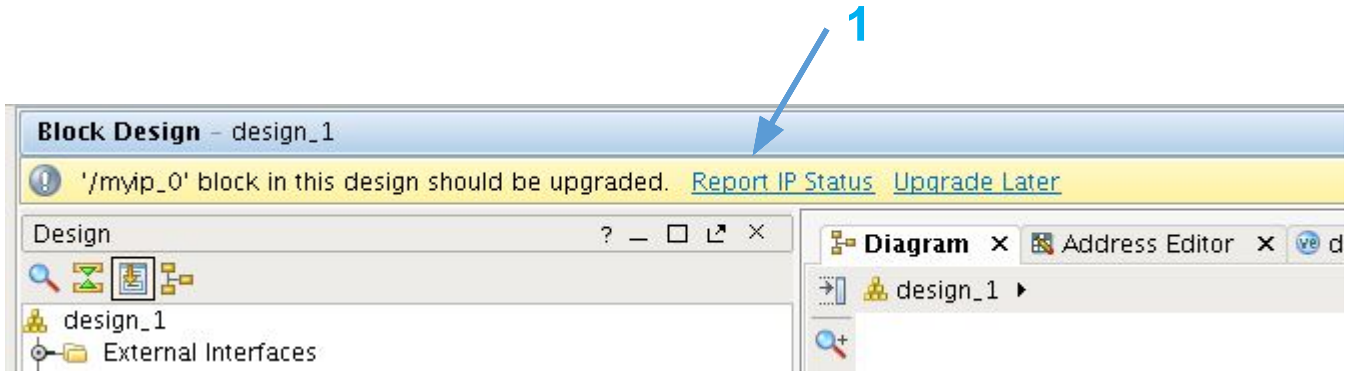
Root directory: /home/ltmin/vivado_prj/:

- An archive will not be generated. Use
- Project will be removed after completi

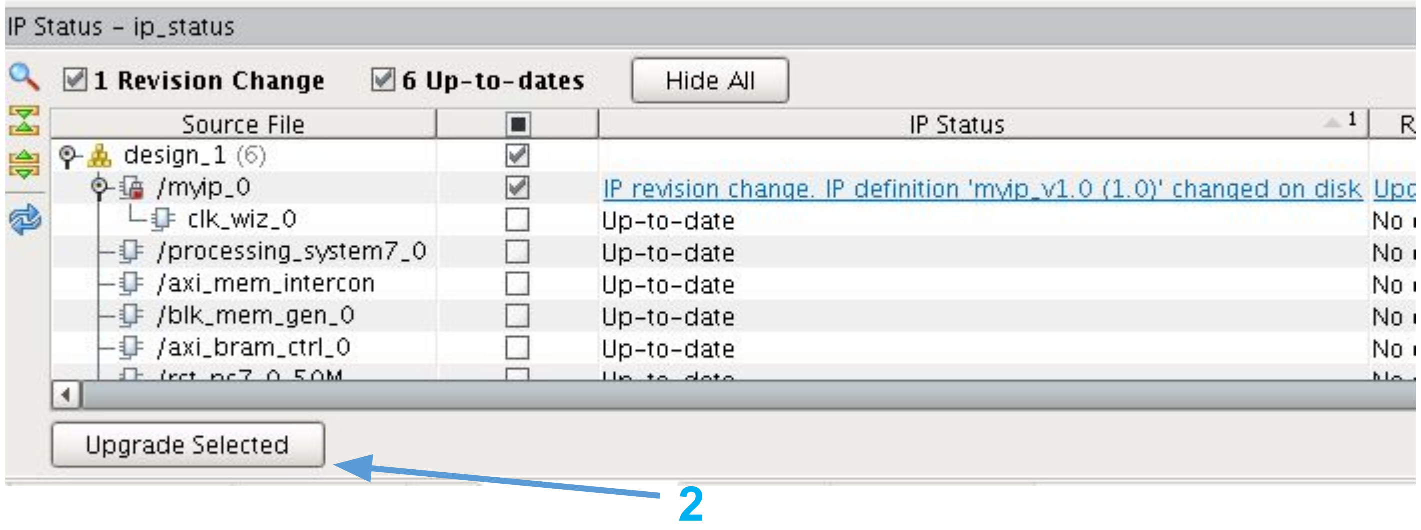
3. Re-packaging

Editing Custom IP (4)

- Report IP Status -> Upgrade Selected



'/myip_0' block in this design should be upgraded. Report IP Status Upgrade Later

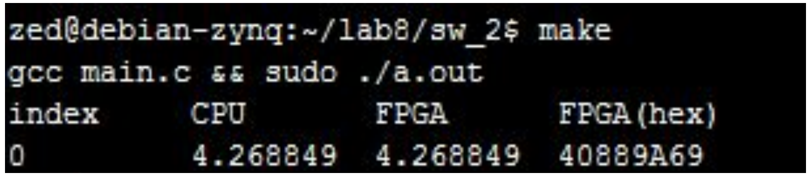


Source File	■	IP Status	R
design_1 (6)	☑		
/myip_0	☑	IP revision change. IP definition 'myip_v1.0 (1.0)' changed on disk	Upc
clk_wiz_0	☐	Up-to-date	No
/processing_system7_0	☐	Up-to-date	No
/axi_mem_intercon	☐	Up-to-date	No
/blk_mem_gen_0	☐	Up-to-date	No
/axi_bram_ctrl_0	☐	Up-to-date	No
/rst_ps7_0_50M	☐	Up-to-date	No

C-program Test

- Get the sample test c-program
  - BOARD$ git clone https://github.com/K16DIABLO/HSD_LAB8.git
  - BOARD$ cd $(git_clone_dir)/sw_2
  - BOARD$ make

- Check the results from the different calculators
  - C program
  - Verilog simulator
  - ZED board FPGA



```
zed@debian-zynq:~/lab8/sw_2$ make
gcc main.c && sudo ./a.out
index      CPU        FPGA       FPGA(hex)
0          4.268849   4.268849   40889A69
```

Method	Simulation C-program	Simulation Verilog	Execution FPGA
Result	40889A69 4.268849	40889A69	40889A69 4.268849

Grading policy

- Check lists
  - C Simulation output (20 points)
  - Verilog Simulation output (30 points)
  - FPGA Implementation output (50 points)

- Submit “L8.**pdf**” (with screenshots on it) on eTL
  - Due : 5/8 (Tue) PM 11:59