



Custom IP LED breathing light

Agenda

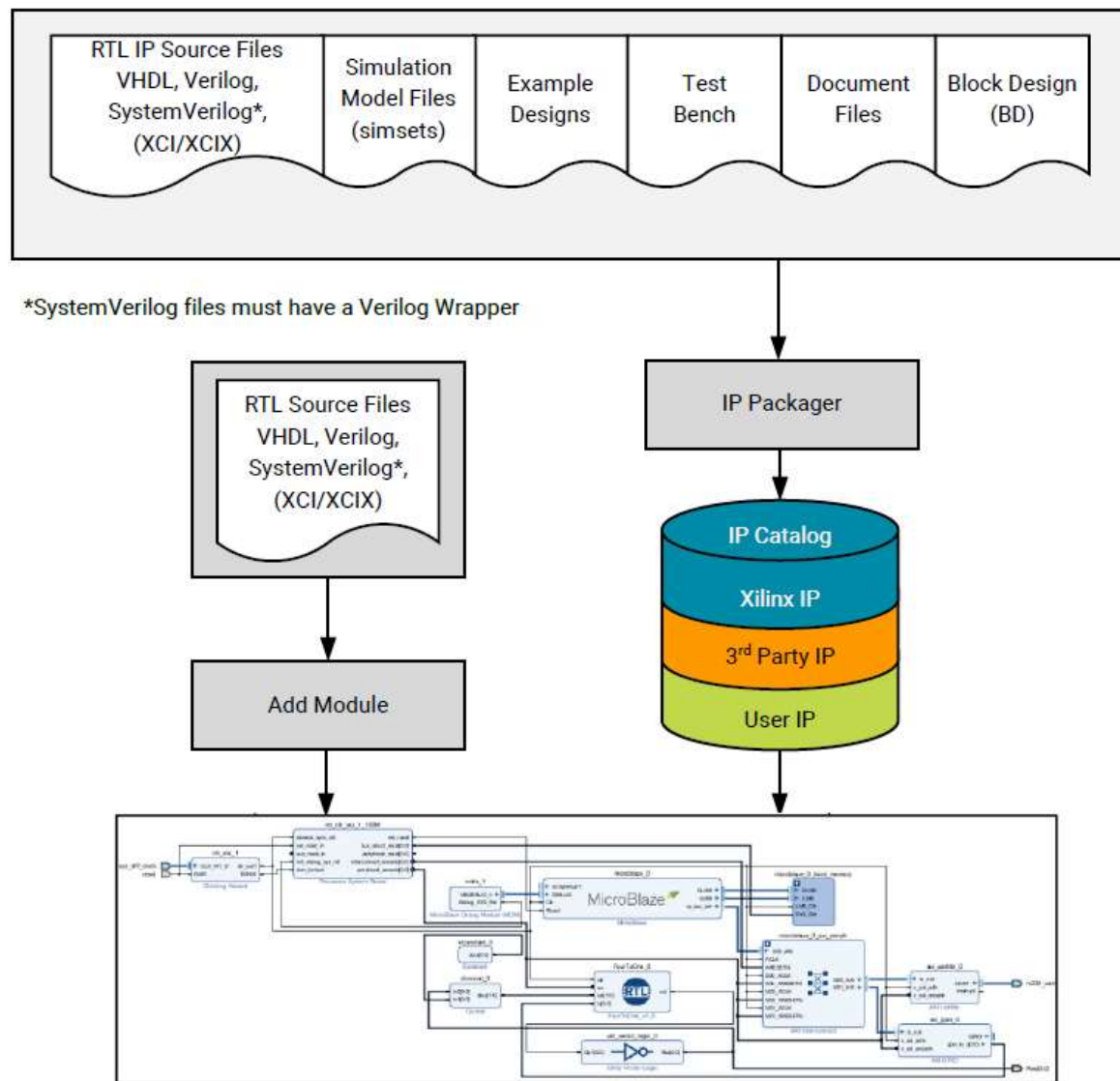
1. Custom IP introduction
2. Program code design

▼ Vivado Design Suite User Guide

Table of Contents

- Ch. 1: Creating and Packaging Custom IP
- Ch. 2: IP Packaging Basics
- Ch. 3: The Create and Package New IP Wizard
- Ch. 4: Packaging IP
- Ch. 5: Creating New Interface Definitions
- Ch. 6: Encrypting IP in Vivado
- Appx. A: Standard and Advanced File Groups
- Appx. B: Additional Resources and Legal Notices

Figure 1: IP Packaging and Usage Flow



X26893-071322

- **AXI4:** For memory-mapped interfaces, which allows burst of up to 256 data transfer cycles with a single address phase.
- **AXI4-Lite:** A light-weight, single transaction memory-mapped interface.
- **AXI4-Stream:** For high-speed streaming data.

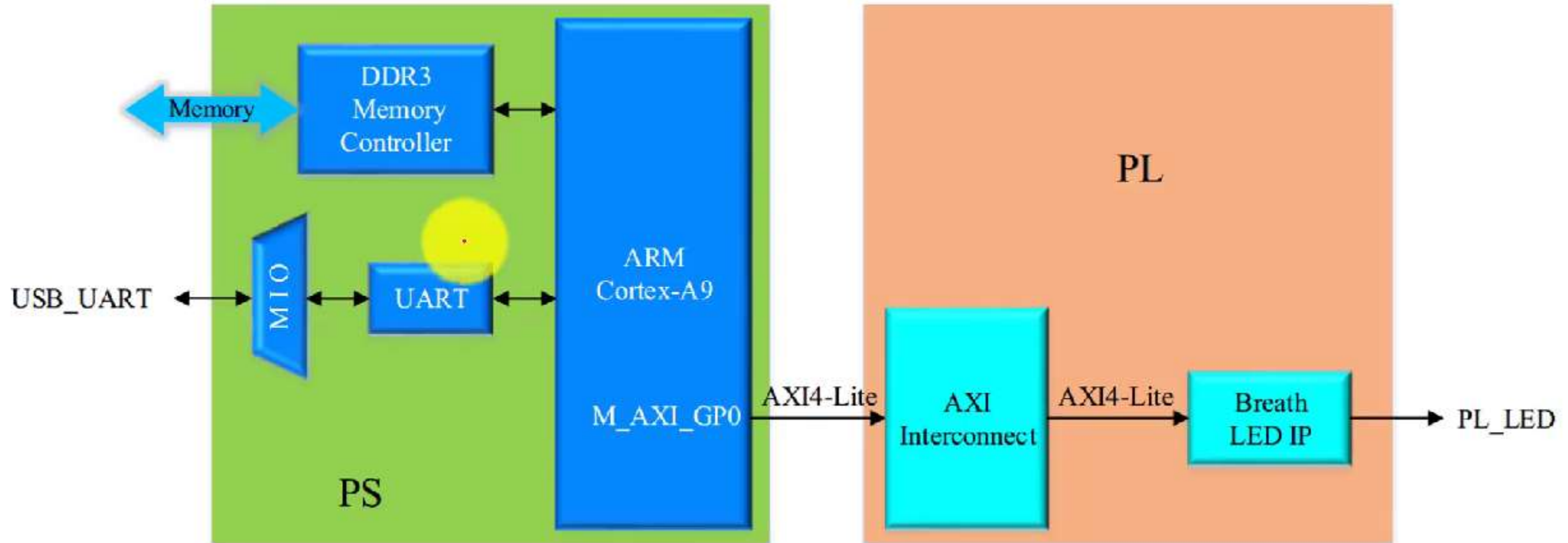
For more information on the Xilinx adoption of AXI, see the *Vivado Design Suite: AXI Reference Guide* ([UG1037](#)).

Note: For the simple purpose of adding custom RTL for use in IP integrator, the Module Reference feature is available and described in the *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* ([UG994](#)).

Experiment purpose

Custom LED IP, control PL LED to present breathing LED, and PS can use AXI interface to control the switch and Frequency.

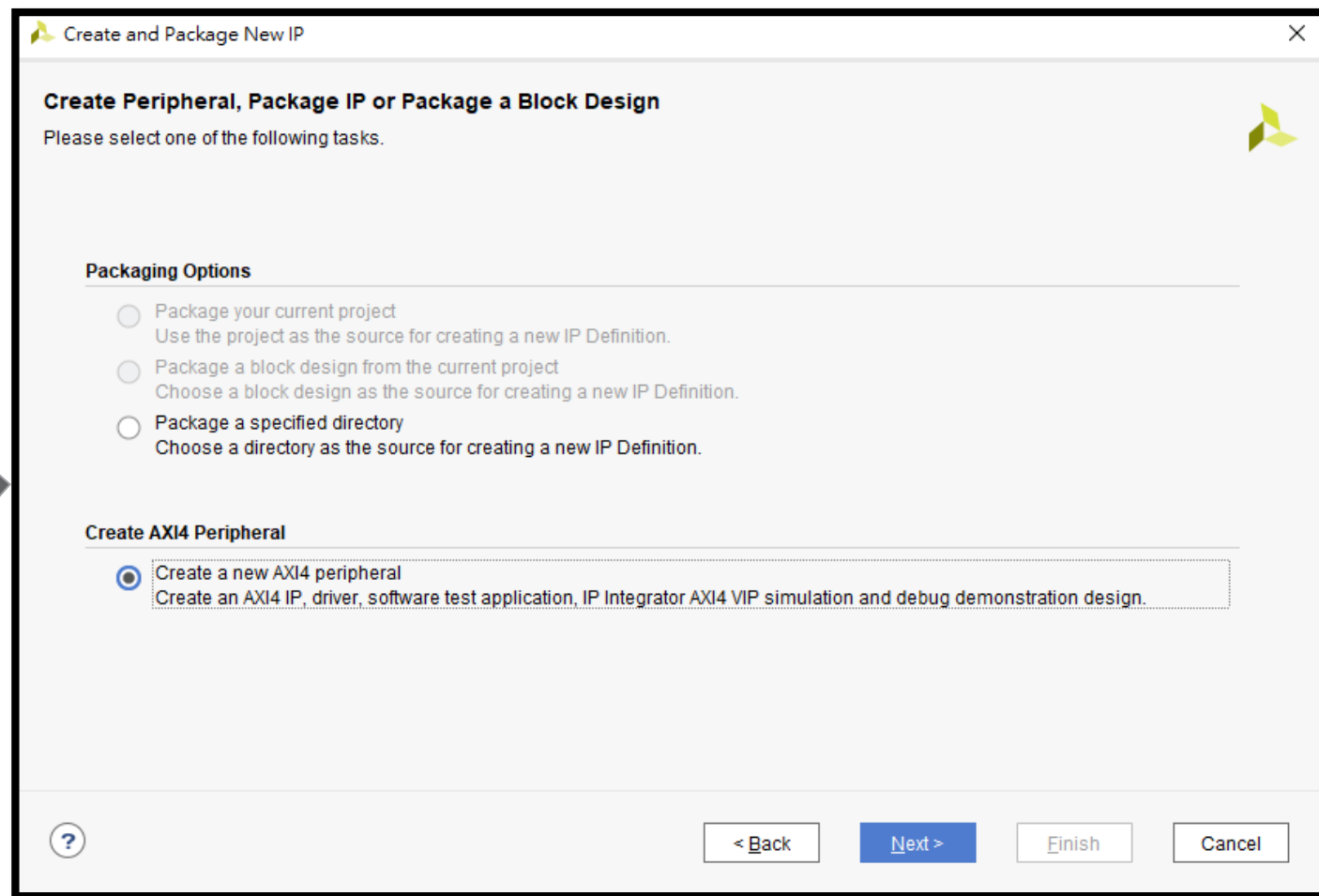
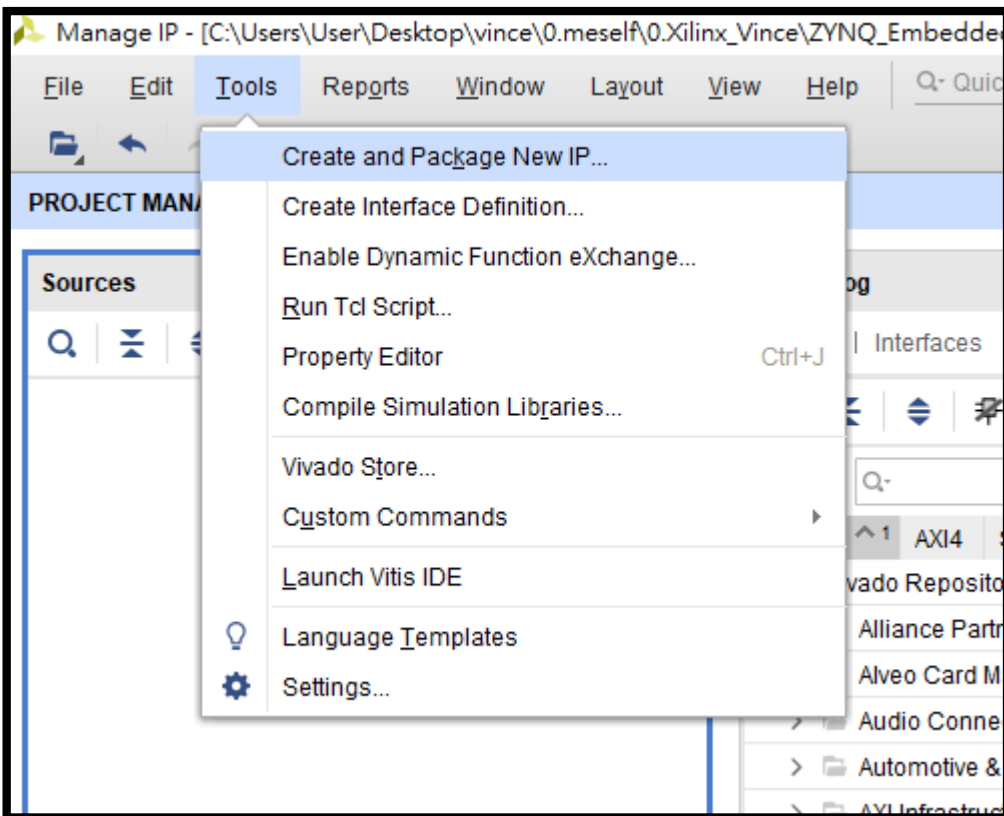
System Block diagram



Tasks

Manage IP >

- New IP Location... Manager >
- Open IP Location...



Create and Package New IP

Peripheral Details

Specify name, version and description for the new peripheral

Name:

Version:

Display name:

Description:

IP location: ...

☐ Overwrite existing



Create and Package New IP

Add Interfaces

Add AXI4 interfaces supported by your peripheral

☐ Enable Interrupt Support

Interfaces

- S00_AXI

S00_AXI

breath_led_ip_v1.0

Name:

Interface Type:

Interface Mode:

Data Width (Bits):

Memory Size (Bytes):

Number of Registers: [4..512]

< Back Next > Finish Cancel



Create and Package New IP

Create Peripheral

Peripheral Generation Summary

1. IP (xilinx.com:user:breath_led_ip:1.0) with 1 interface(s)
2. Driver(v1_00_a) and testapp [more info](#)
3. AXI4 VIP Simulation demonstration design [more info](#)
4. AXI4 Debug Hardware Simulation demonstration design [more info](#)

Peripheral created will be available in the catalog :

C:/Users/User/Desktop/vince/0.meself/0.Xilinx_Vince/ZYNQ_Embedded_system/4.Custom_IP_LED_breathing_

Next Steps:

- ☒ Add IP to the repository
- ☐ Edit IP
- ☐ Verify Peripheral IP using AXI4 VIP
- ☐ Verify peripheral IP using JTAG interface

Click Finish to continue

< Back Next > Finish Cancel

IP Catalog

Cores | Interfaces

Search:

This is the IP when user repository

Name	AXI4	Status	License	VLNV
<ul style="list-style-type: none"> User Repository (c:/Users/User/Desktop/Vince/0.meself/0.Xilinx_Vince/ZYNQ_Embedded_system/4.Custom_IP_LED_breathing_light/custom_ip/ip_repo/breath_led_ip_1_0) <ul style="list-style-type: none"> AXI Peripheral <ul style="list-style-type: none"> breath_led_ip_v1.0 Vivado Repository 	AXI4	Pre-Pro	Included	xilinx.cc

Name

- User Repository (c:/Users/User/Desktop/Vince/0.meself/0.Xilinx_Vince/ZYNQ_Embedded_system/4.Custom_IP_LED_breathing_light/custom_ip/ip_repo/breath_led_ip_1_0)
 - AXI Peripheral
 - breath_led_ip_v1.0
 - Vivado Repository

Details

Name: **breath_led_ip_v1.0**

Version: 1.0 (Rev. 1)

Interfaces: AXI4

Description: My new AXI IP

Status: **Pre-Production**

Context menu options:

- Properties... (Ctrl+E)
- IP Settings...
- Add Repository...
- Refresh All Repositories
- Customize IP...
- Edit in IP Packager**
- Disable IP
- Delete IP (Delete)
- License Status
- Compatible Families
- Compatible Simulators
- Answer Records
- Export to Spreadsheet...



Edit in IP Packager

Choose a project name and location for editing.

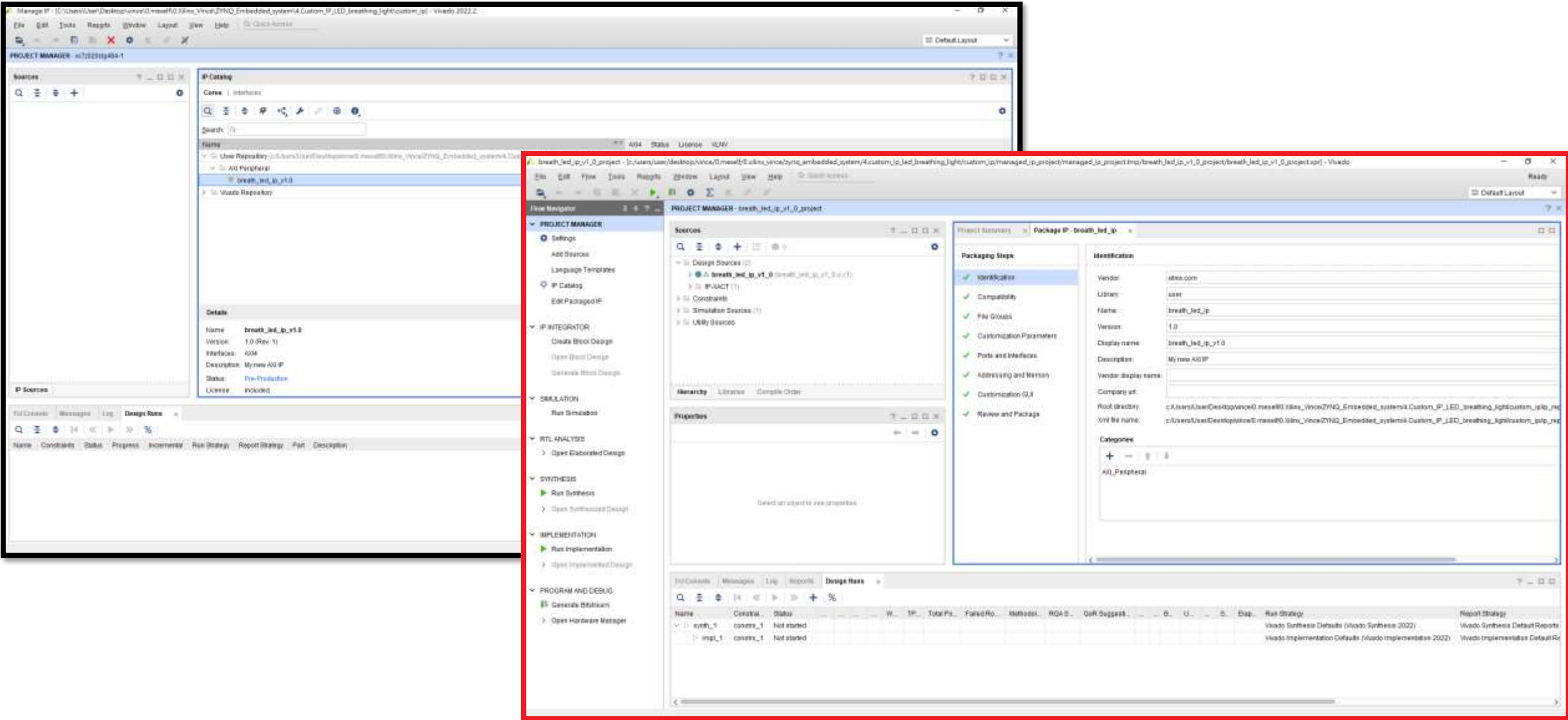
Project name:


Project location: ...

Edit IP project will be created at: ..._led_ip_v1_0_project

OK Cancel


Than mange ip will open other the Vivad project.






> ●  breath_led_ip_v1_0 (breath_led_ip_v1_0.v) (1)

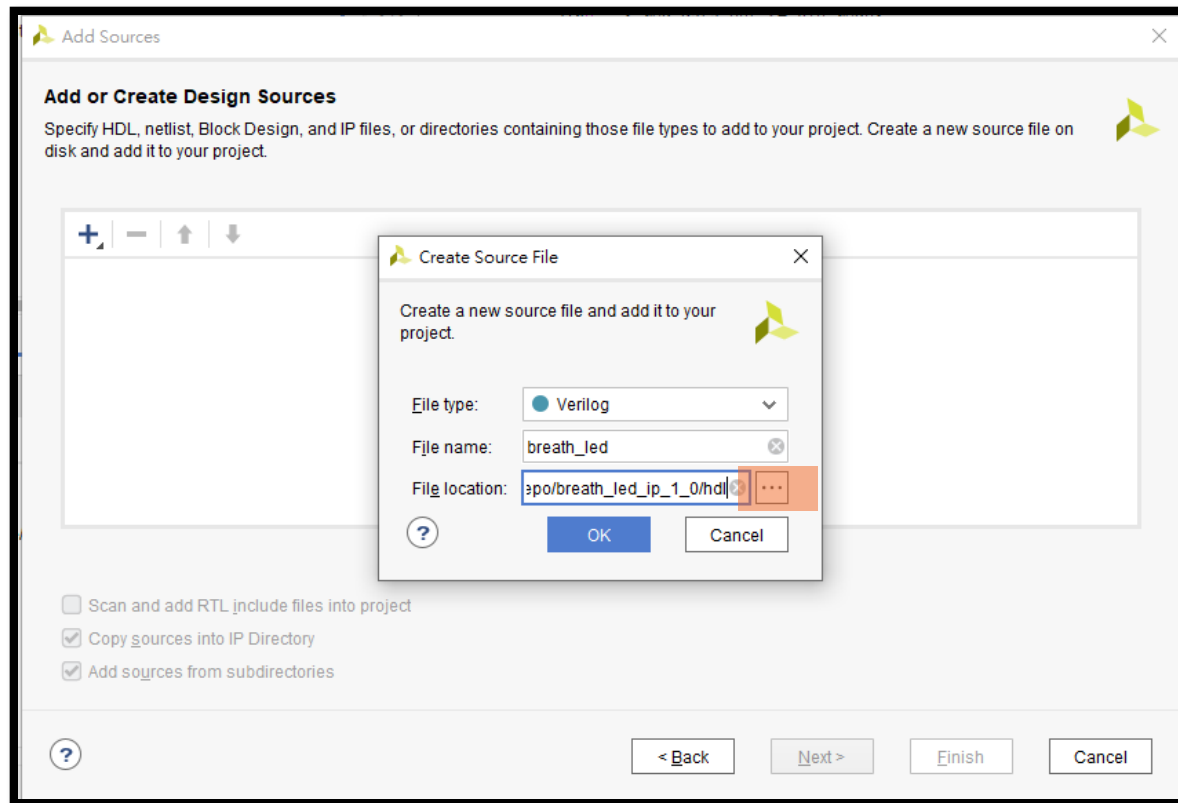
```
// Ports of Axi Slave Bus Interface S00_AXI
input wire  s00_axi_aclk,
input wire  s00_axi_aresetn,
input wire [C_S00_AXI_ADDR_WIDTH-1 : 0] s00_axi_awaddr,
input wire [2 : 0] s00_axi_awprot,
input wire  s00_axi_awvalid,
output wire  s00_axi_awready,
input wire [C_S00_AXI_DATA_WIDTH-1 : 0] s00_axi_wdata,
input wire [(C_S00_AXI_DATA_WIDTH/8)-1 : 0] s00_axi_wstrb,
input wire  s00_axi_wvalid,
output wire  s00_axi_wready,
output wire [1 : 0] s00_axi_bresp,
output wire  s00_axi_bvalid,
input wire  s00_axi_bready,
input wire [C_S00_AXI_ADDR_WIDTH-1 : 0] s00_axi_araddr,
input wire [2 : 0] s00_axi_arprot,
input wire  s00_axi_arvalid,
output wire  s00_axi_arready,
output wire [C_S00_AXI_DATA_WIDTH-1 : 0] s00_axi_rdata,
output wire [1 : 0] s00_axi_rresp,
output wire  s00_axi_rvalid,
input wire  s00_axi_rready
```

AXI Lite interface port

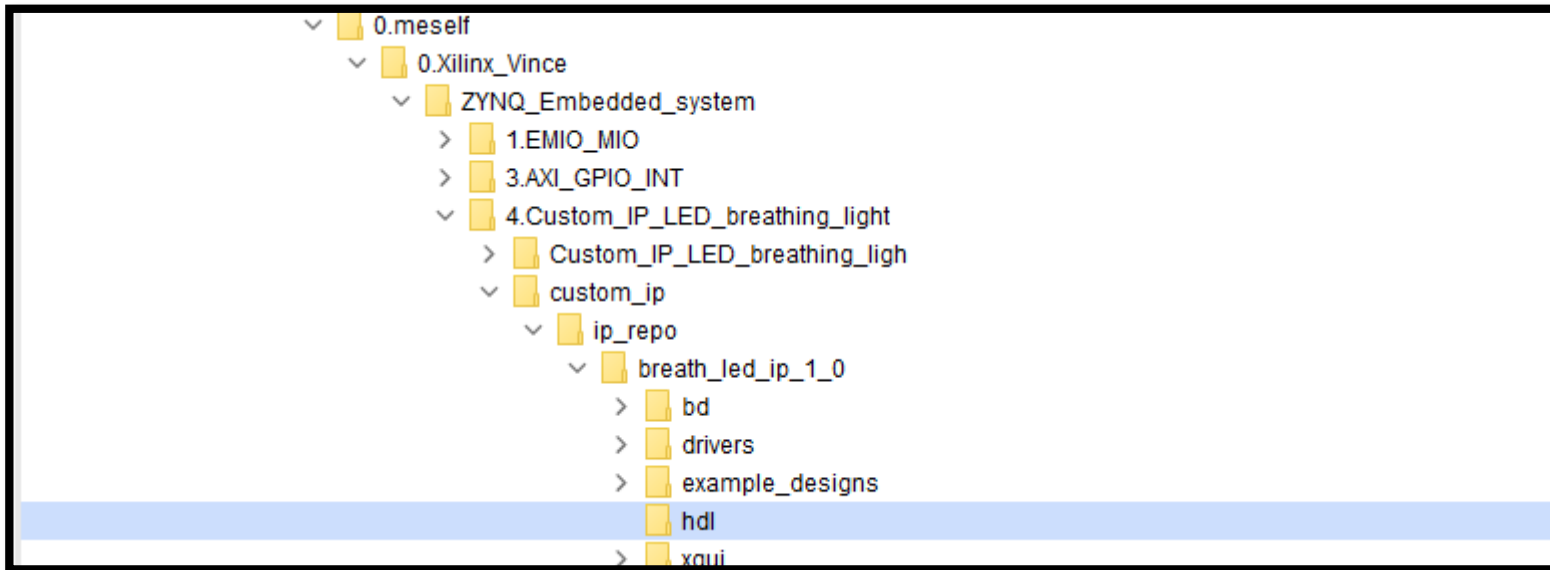
▼ ●  breath_led_ip_v1_0 (breath_led_ip_v1_0.v) (1)

● breath_led_ip_v1_0_S00_AXI_inst: breath_led_ip_v1_0_S00_AXI (breath_led_ip_v1_0.v) (1)

```
400  // Add user logic here
401 |
402  // User logic ends
403 |
404  endmodule
405 |
```



Create the breath_led, connect under the //add user logic here



breath_led_ip_v1_0.v(Top)

```
1
2 `timescale 1 ns / 1 ps
3
4 module breath_led_ip_v1_0 #
5 (
6     // Users to add parameters here
7     parameter START_FREA_STEP = 10'd100, // setup the frequency interval initial value
8     // User parameters ends
9     // Do not modify the parameters beyond this line
10
11
12     // Parameters of Axi Slave Bus Interface S00_AXI
13     parameter integer C_S00_AXI_DATA_WIDTH  = 32,
14     parameter integer C_S00_AXI_ADDR_WIDTH  = 4
15 )
16 (
17     // Users to add ports here
18     output led,
19     // User ports ends
20     // Do not modify the ports beyond this line
21 )
```

```
47 // Instantiation of Axi Bus Interface S00_AXI
48 breath_led_ip_v1_0_S00_AXI # (
49     .START_FREA_STEP(START_FREA_STEP),
50     .C_S_AXI_DATA_WIDTH(C_S00_AXI_DATA_WIDTH),
51     .C_S_AXI_ADDR_WIDTH(C_S00_AXI_ADDR_WIDTH)
52 ) breath_led_ip_v1_0_S00_AXI_inst (
53     .led(led),
54     .S_AXI_ACLK(s00_axi_aclk),
55     .S_AXI_ARESETN(s00_axi_aresetn),
56     .S_AXI_AWADDR(s00_axi_awaddr),
57     .S_AXI_AWPROT(s00_axi_awprot),
58     .S_AXI_AWVALID(s00_axi_awvalid),
59     .S_AXI_AWREADY(s00_axi_awready),
60     .S_AXI_WDATA(s00_axi_wdata),
61     .S_AXI_WSTRB(s00_axi_wstrb),
```

breath_led_ip_v1_0_S00_AXI.v

```
`timescale 1 ns / 1 ps

module breath_led_ip_v1_0_S00_AXI #
(
    // Users to add parameters here

    // User parameters ends

    parameter START_FREA_STEP = 10'd100, // setup the frequency interval initial value

    // Do not modify the parameters beyond this line

    // Width of S_AXI data bus
    parameter integer C_S_AXI_DATA_WIDTH    = 32,
    // Width of S_AXI address bus
    parameter integer C_S_AXI_ADDR_WIDTH    = 4

```

```
    // Add user logic here
    breath_led #(
        .START_FREA_STEP(START_FREA_STEP))

    u0_breath_led(
        .sys_clk      (S_AXI_ACLK      ),
        .sys_rst_n    (S_AXI_ARESETN  ),
        .sw_ctrl      (slv_reg0[0]    ),
        .set_en       (slv_reg0[1]    ),
        .set_freq_step(slv_reg0[11:2]  ),
        .led          (led             )
    );
    // User logic ends

```

Interface Mode: Slave

Data Width (Bits): 32

Memory Size (Bytes): 64

Number of Registers: 4 [4.512]

```
//-- Number of Slave Registers 4
reg [C_S_AXI_DATA_WIDTH-1:0]    slv_reg0;
reg [C_S_AXI_DATA_WIDTH-1:0]    slv_reg1;
reg [C_S_AXI_DATA_WIDTH-1:0]    slv_reg2;
reg [C_S_AXI_DATA_WIDTH-1:0]    slv_reg3;

```

Add code ,refer the figure.

In the top need to define parameter, because in the breath_led.v has defined the parameter.

breath_led_ip_v1_0.v & breath_led_ip_v1_0_S00_AXI.v

had defined START_FREA_STEP, need to write the breath_led #().

Use the Axi clk and reset, in the beginning, had set the AXI interface, data width is 32 bit, number of registers is 4, (slv_reg0; slv_reg1; slv_reg2; slv_reg3;)

Add port

```
20 // Users to add ports here
21 output led,
22 // User ports ends
23 // Do not modify the ports beyond this line
24

```

SYNTHESIS

[Run Synthesis](#)

> Open Synthesized Design

Design Sources (2)

- breath_led_ip_v1_0 (breath_led_ip_v1_0)
 - breath_led_ip_v1_0_S00_AXI_in
 - u0_breath_led : breath_led (1)
- IP-XACT (1)
 - component.xml

Packaging Steps

- Identification
- Compatibility
- File Groups
- Customization Parameters
- Ports and Interfaces
- Addressing and Memory
- Customization GUI
- Review and Package

Compatibility

Refer to the RTL Kernel Developer's Guide

☐ Package for Vitis

Control protocol: ap_ctrl

☐ Package for IPI

☐ Ignore Freq_Hz

Family | Simulator

+ - ↑ ↓ Auto

Fa Add Life Cycle

zynq Pre-Production

Add the board family

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- Identification
- Compatibility
- File Groups
- Customization Parameters
- Ports and Interfaces
- Addressing and Memory
- Customization GUI
- Review and Package

Identification

Vendor: xilinx.com

Library: user

Name: breath_led_ip

Version: 1.0

Display name: breath_led_ip_v1.0

Description: My new AXI IP

Vendor display name:

Company url:

Root directory: c:/Users/User/Desktop/Custom IP/ip_repo/breath_led_ip_1_0

Xml file name: c:/Users/User/Desktop/Custom IP/ip_repo/breath_led_ip_1_0/component.xml

Categories

+ - ↑ ↓

AXI_Peripheral

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- File Groups**
- Customization Parameters
- Ports and Interfaces
- ✓ Addressing and Memory
- Customization GUI
- Review and Package

File Groups

! Merge changes from File Groups Wizard

Q [Icons] + C

Name	Library Name	Type	Is Include	File Group Name	Model Name
Standard			<input type="checkbox"/>		
Advanced			<input type="checkbox"/>		
Verilog Synthesis (2)			<input type="checkbox"/>		breath_led_ip_v1_0
hdl/breath_led_ip_v1_0_S00_AXI.v		verilogSource	<input type="checkbox"/>	xilinx_verilog	
hdl/breath_led_ip_v1_0.v		verilogSource	<input type="checkbox"/>	xilinx_verilog	
Verilog Simulation (2)			<input type="checkbox"/>		breath_led_ip_v1_0
Software Driver (6)			<input type="checkbox"/>		
UI Layout (1)			<input type="checkbox"/>		
Block Diagram (1)			<input type="checkbox"/>		

Sources

Q [Icons] + ? 0

- Design Sources (2)
 - breath_led_ip_v1_0 (breath_led_ip_v1_0.v) (1)
 - breath_led_ip_v1_0_S00_AXI_inst: breath_led_ip_v1_0_S00_AXI (breath_led_ip_v1_0_S00_AXI.v) (1)
 - u0_breath_led: breath_led (breath_led.v)
 - IP-XACT (1)
 - component.xml
 - Constraints
 - Simulation Sources (1)
 - Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- File Groups**
- Customization Parameters
- Ports and Interfaces
- ✓ Addressing and Memory
- Customization GUI
- Review and Package

File Groups

Q [Icons] + C

Name	Library
Standard	
Advanced	
Verilog Synthesis (3)	
hdl/breath_led.v	
hdl/breath_led_ip_v1_0_S00_AXI.v	
hdl/breath_led_ip_v1_0.v	
Verilog Simulation (3)	
Software Driver (6)	
UI Layout (1)	
Block Diagram (1)	

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✎ Customization Parameters
- ✎ Ports and Interfaces
- ✓ Addressing and Memory
- ✎ Customization GUI
- ✎ Review and Package

Customization Parameters

! Merge changes from Customization Parameters Wizard

Q [] [] [] [] [] []

Name	Description	Display Name	Value	Value Bit String Length	Value Format
Customization Parameters					
⚙ C_S00_AXI_DATA_WIDTH	Width of S_AXI data bus	C S00 AXI DATA WIDTH	32	0	long
⚙ C_S00_AXI_ADDR_WIDTH	Width of S_AXI address bus	C S00 AXI ADDR WIDTH	4	0	long
⚙ C_S00_AXI_BASEADDR		C S00 AXI BASEADDR	0xFFFFFFFF	32	bitString
⚙ C_S00_AXI_HIGHADDR		C S00 AXI HIGHADDR	0x00000000	32	bitString

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory
- ✓ Customization GUI
- ✎ Review and Package

Customization Parameters

Q [] [] [] [] [] []

Name	Description	Display Name	Value
Customization Parameters			
⚙ C_S00_AXI_DATA_WIDTH	Width of S_AXI data bus	C S00 AXI DATA WIDTH	32
⚙ C_S00_AXI_ADDR_WIDTH	Width of S_AXI address bus	C S00 AXI ADDR WIDTH	4
⚙ C_S00_AXI_BASEADDR		C S00 AXI BASEADDR	0xFFFFFFFF
⚙ C_S00_AXI_HIGHADDR		C S00 AXI HIGHADDR	0x00000000
Hidden Parameters			
⚙ START_FREA_STEP		Start Frea Step	"0001100100"

```

1
2 `timescale 1 ns / 1 ps
3
4 module breath_led_ip_v1_0 #
5 (
6     // Users to add parameters here
7     parameter START_FREA_STEP = 10'd100, // setup the frequency interval initial value
8     // User parameters ends
9     // Do not modify the parameters beyond this line
10
11
12     // Parameters of Axi Slave Bus Interface S00_AXI
13     parameter integer C_S00_AXI_DATA_WIDTH = 32,
14     parameter integer C_S00_AXI_ADDR_WIDTH = 4
15 )
16 (
17     // Users to add ports here
18     output led,
19     // User ports ends
20     // Do not modify the ports beyond this line
21 )
  
```


Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory
- ✓ Customization GUI
- ✎ Review and Package

Customization Parameters

Name	Description	Display Name	Value
▼ Customization Parameters			
⚙ C_S00_AXI_DATA_WIDTH	Width of S_AXI data bus	C S00 AXI DATA WIDTH	32
⚙ C_S00_AXI_ADDR_WIDTH	Width of S_AXI address bus	C S00 AXI ADDR WIDTH	4
⚙ C_S00_AXI_BASEADDR		C S00 AXI BASEADDR	0xFFFFFFFF
⚙ C_S00_AXI_HIGHADDR		C S00 AXI HIGHADDR	0x00000000
▼ Hidden Parameters			
⚙ START_FREA_STEP		Start Frea Step	"0001100100"

Click twice

Edit IP Parameter

Use the options below to customize how the parameter will appear in the Customization GUI for users of the IP.

Name: START_FREA_STEP

☒ Visible in Customization GUI

☒ Show Name

Display Name: Start Frea Step

Tooltip:

Format: long

Editable: Yes

Dependency: No

☒ Specify Range

Type: Range of integers

Minimum: 1

Maximum: 1000

☒ Show Range

Show As: Not Applicable

Layout: Not Applicable

Default Value: 100

OK Cancel

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory
- ✓ Customization GUI
- Review and Package

Review and Package

IP has been modified.

Summary

Display name: breath_led_ip_v1.0

Description: My new AXI IP

Root directory: c:/Users/User/Desktop/Custom IP/ip_repo/breath_led_ip_1_0

After Packaging

An archive will not be generated. Use the settings link below to change your preference

Project will be removed after completion

[Edit packaging settings](#)

Project Summary x Package IP - breath_led_ip x breath_led.v x breath_led_ip_v1_0.v x breath_led_ip_v1_0_S00_AXI.v x

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- ✓ Customization Parameters
- ✓ Ports and Interfaces
- ✓ Addressing and Memory
- ✓ Customization GUI
- Review and Package

Review and Package

Summary

Display name: breath_led_ip_v1.0

Description: My new AXI IP

Root directory: c:/Users/User/Desktop/Custom IP/ip_repo/breath_led_ip_1_0

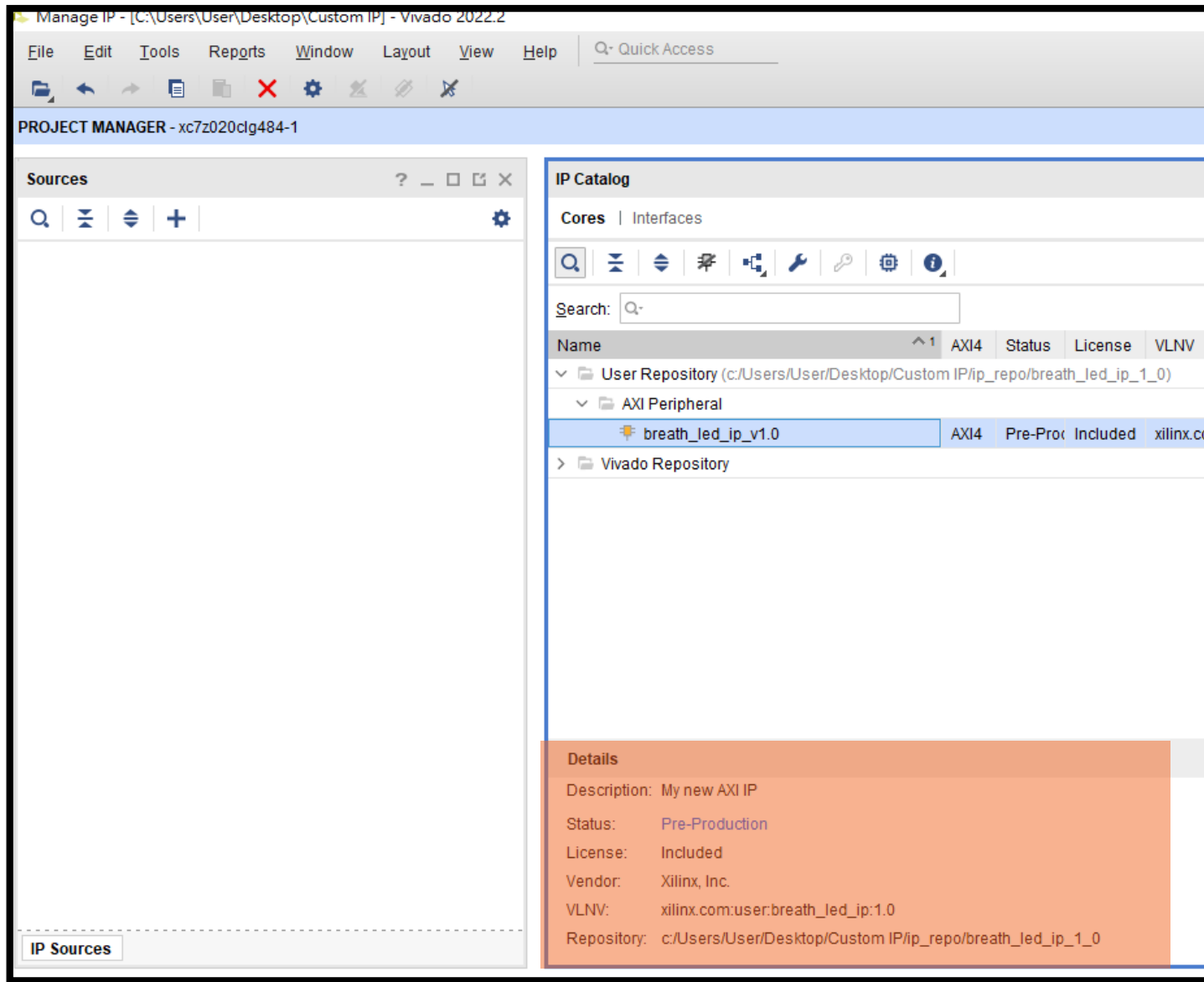
After Packaging

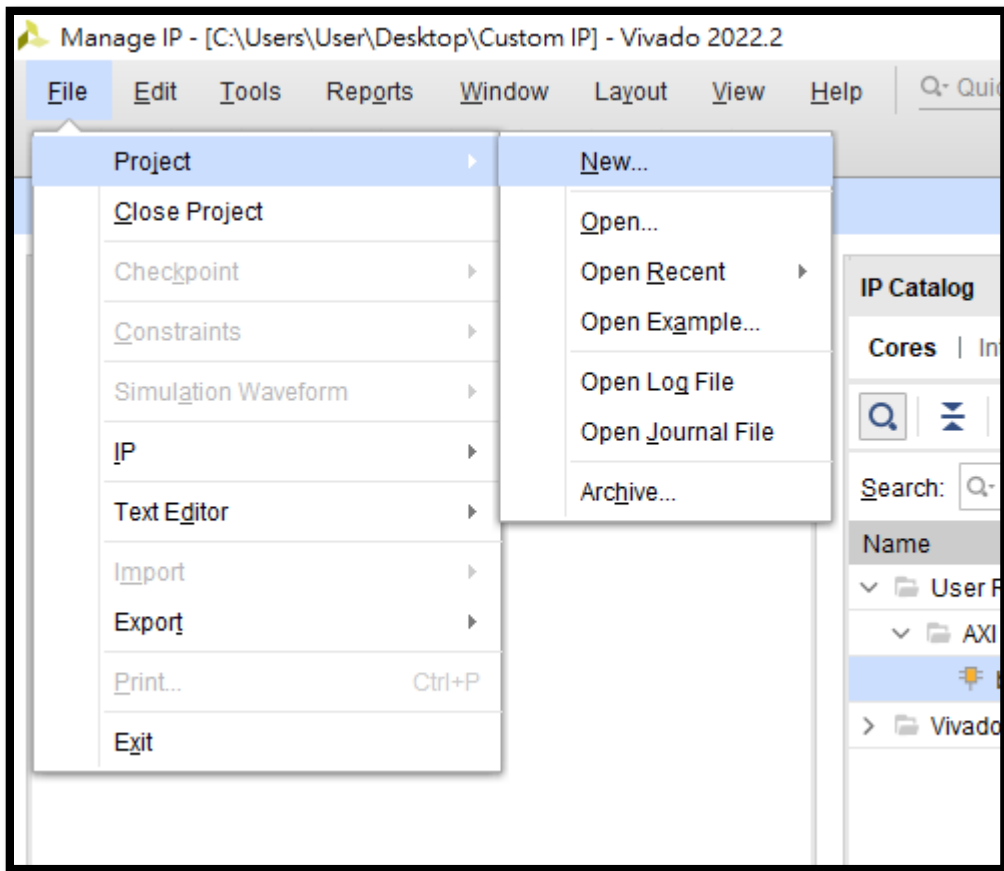
An archive will not be generated. Use the settings link below to change your preference

Project will be removed after completion

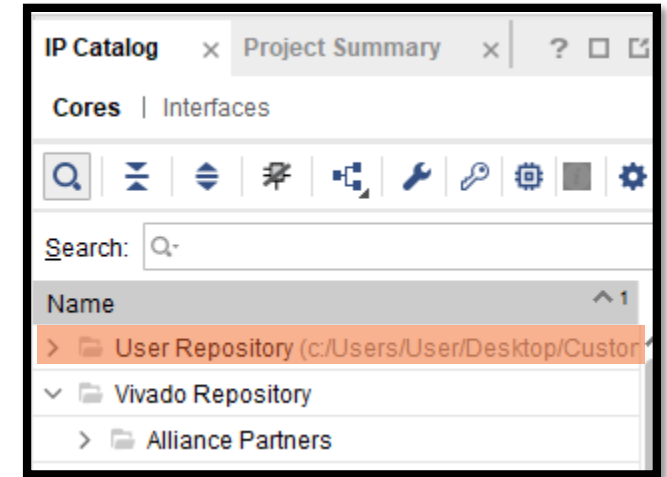
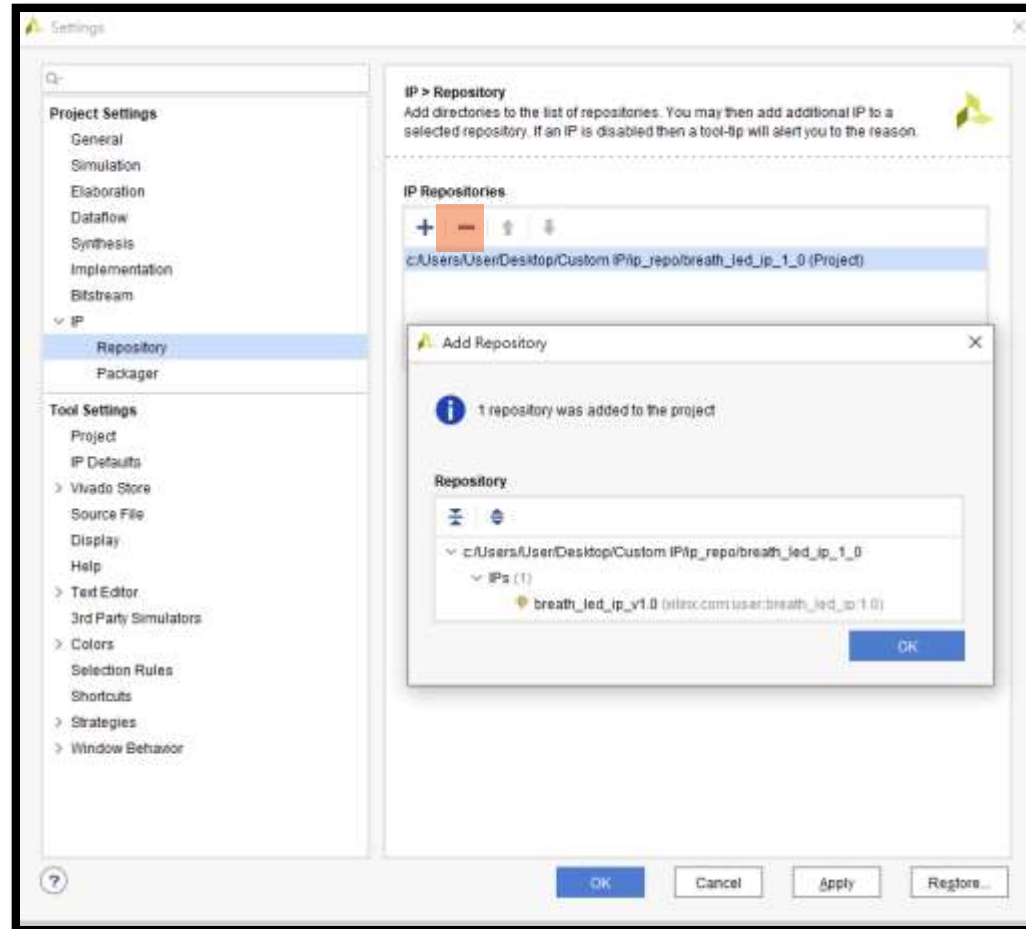
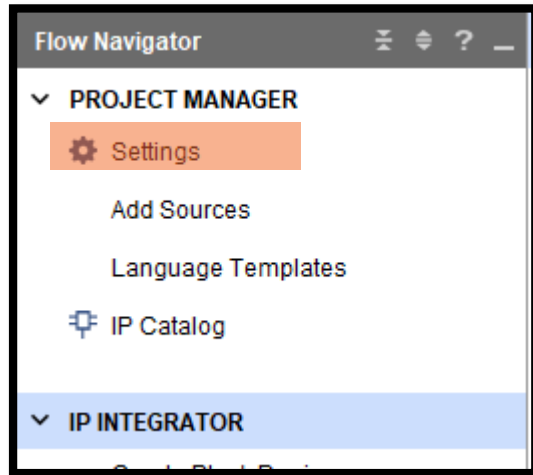
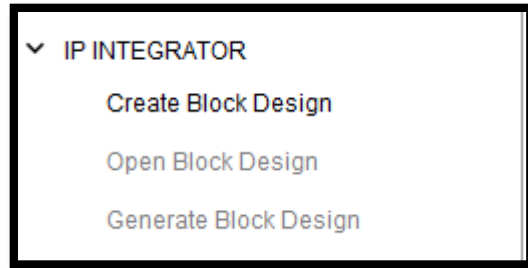
[Edit packaging settings](#)

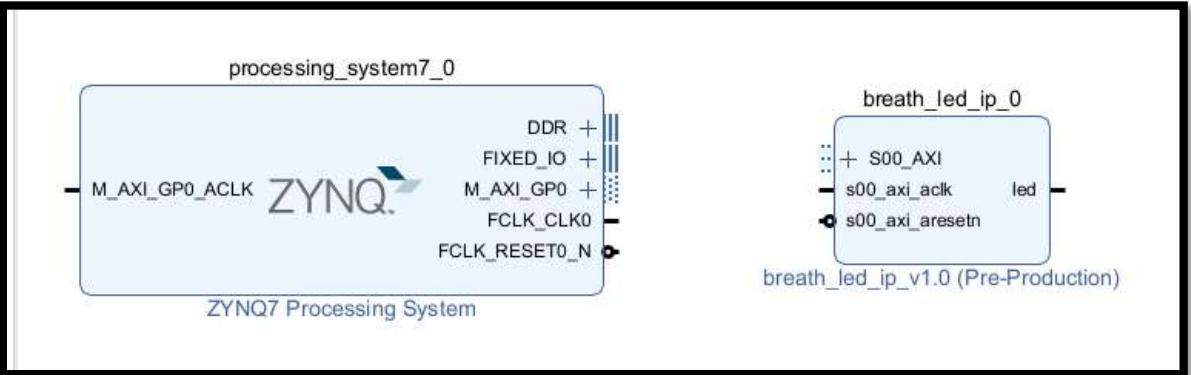
Re-Package IP





New project





Re-customize IP

ZYNQ7 Processing System (5.5)

Documentation Presets IP Location

Page Navigator

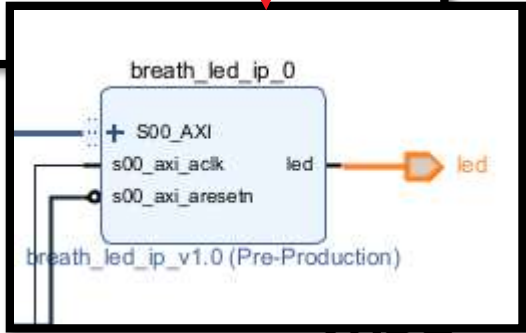
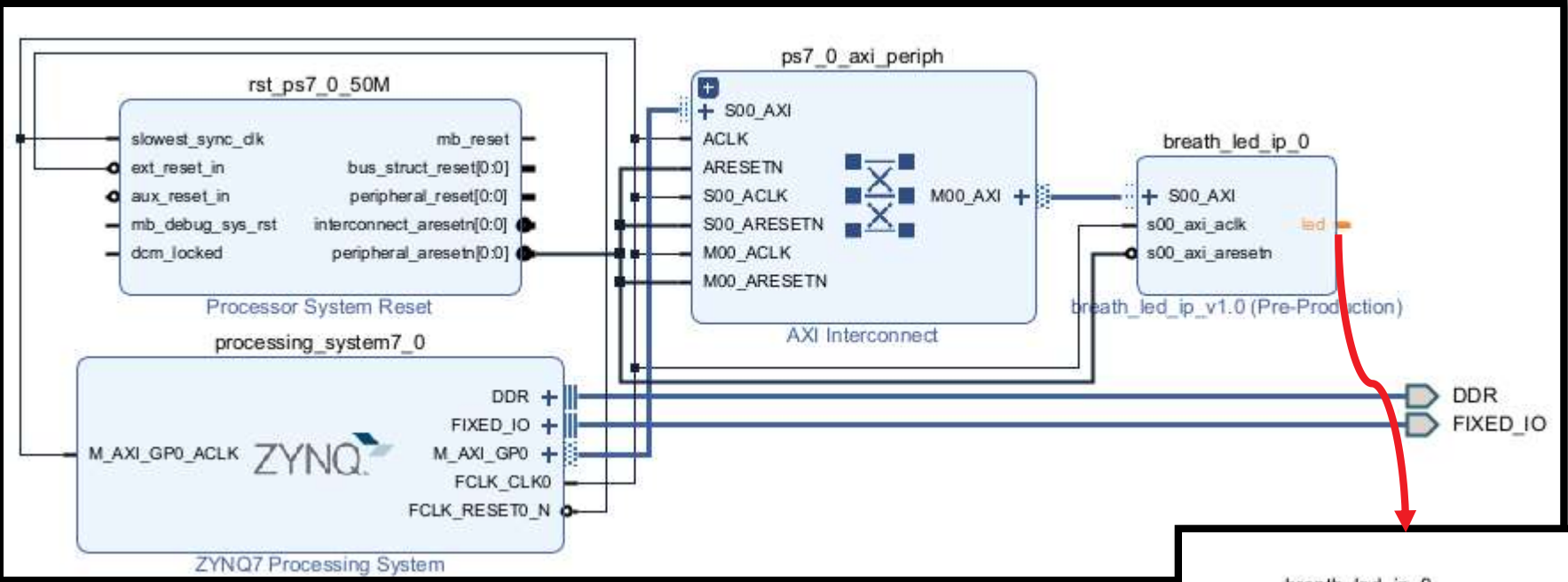
- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

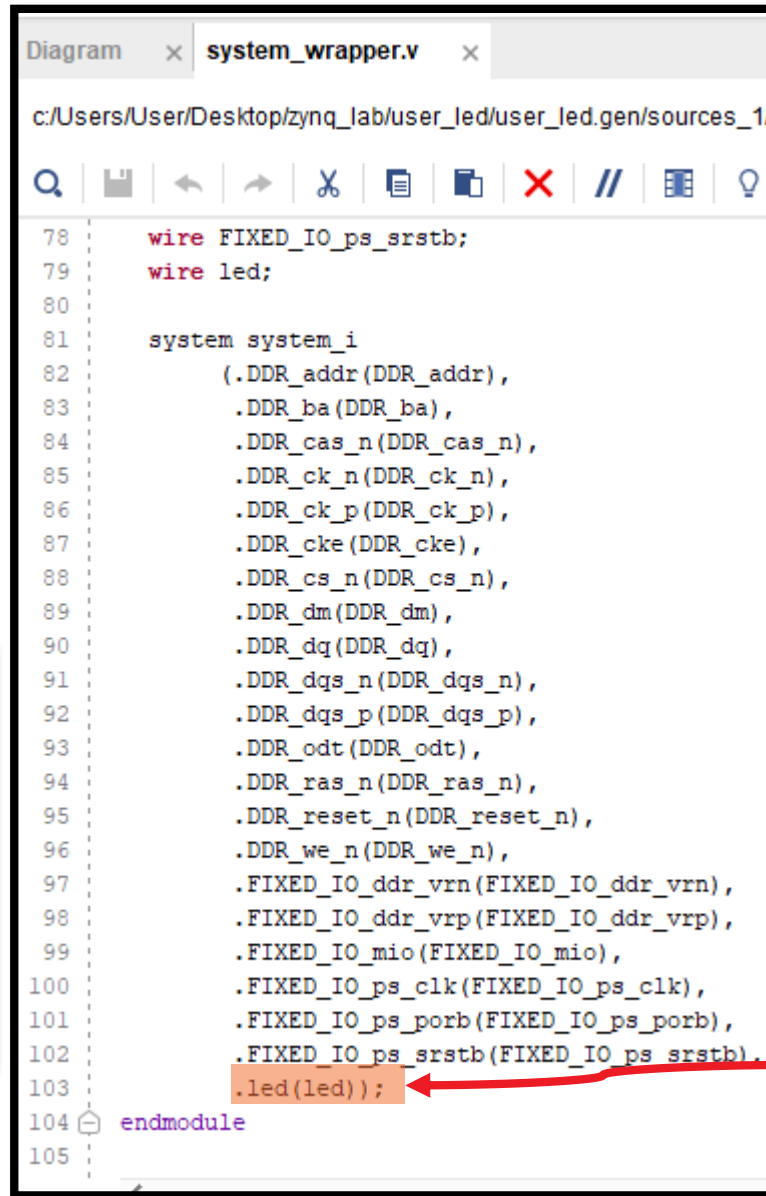
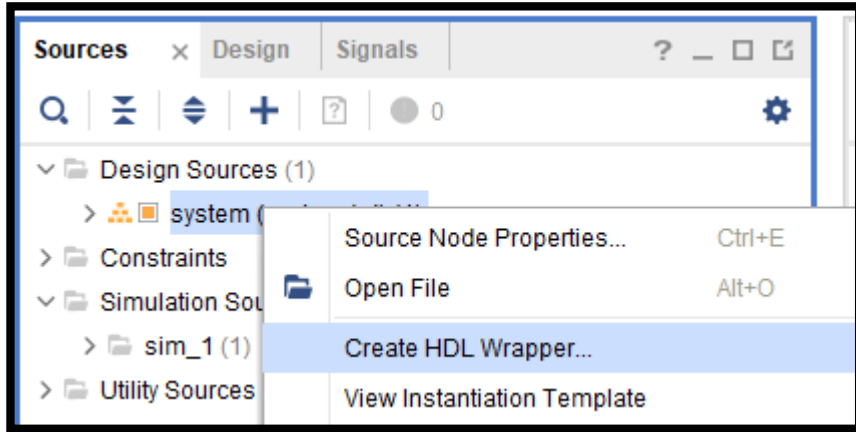
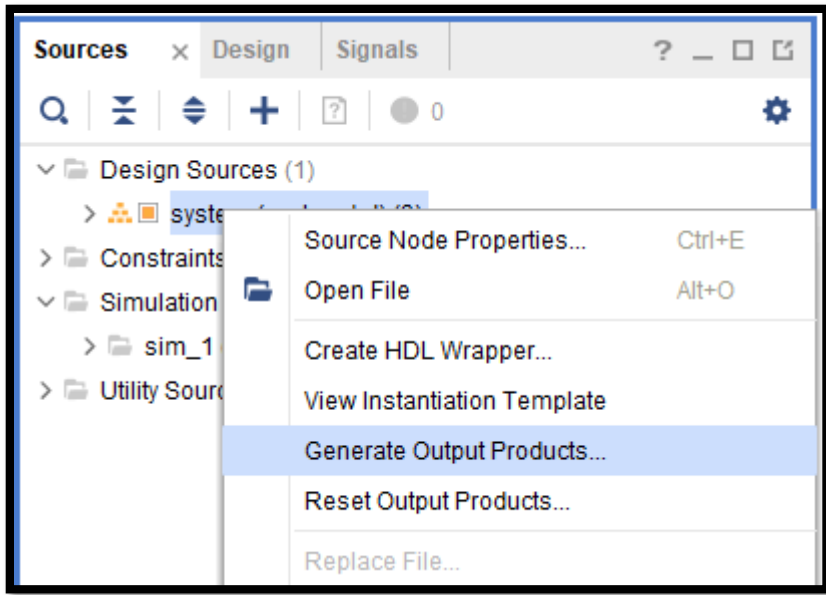
Peripheral I/O Pins

Search:

Peripherals

- ☐ SD 1
- ☐ SPI 0
- ☐ SPI 1
- ☐ UART 0
- ☒ UART 1
- ☐ I2C 0





Report Utilization

Report Power
 Schematic

IMPLEMENTATION

Run Implementation

> Open Implemented Design

Tcl Console

Messages

Log

Reports

Design Runs

Package Pins

I/O Ports x

Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	V
<div> <div> <div>▼</div> <div>📁 All ports (131)</div> </div> <div> <div>></div> <div>📁 DDR_12642 (71)</div> </div> </div>	INOUT					✓	502	(Multiple)*	
<div> <div>></div> <div>📁 FIXED_IO_12642 (59)</div> </div>	INOUT					✓	(Multiple)	(Multiple)*	
<div> <div> <div>▼</div> <div>📁 Scalar ports (1)</div> </div> <div> <div>✓</div> <div>led</div> </div> </div>	OUT				G19	▼	✓	35	LVC MOS3: ▼

▼ PROGRAM AND DEBUG

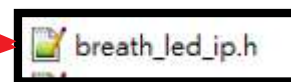
Generate Bitstream

> Open Hardware Manager



```
*/  
#define BREATH_LED_IP_mWriteReg(BaseAddress, RegOffset, Data) \  
    Xil_Out32((BaseAddress) + (RegOffset), (u32)(Data))  
/**
```

```
*/  
#define BREATH_LED_IP_mReadReg(BaseAddress, RegOffset) \  
    Xil_In32((BaseAddress) + (RegOffset))
```



```
#ifndef BREATH_LED_IP_H  
#define BREATH_LED_IP_H  
  
/***** Include Files *****/  
#include "xil_types.h"  
#include "xstatus.h"  
  
#define BREATH_LED_IP_S00_AXI_SLV_REG0_OFFSET 0  
#define BREATH_LED_IP_S00_AXI_SLV_REG1_OFFSET 4  
#define BREATH_LED_IP_S00_AXI_SLV_REG2_OFFSET 8  
#define BREATH_LED_IP_S00_AXI_SLV_REG3_OFFSET 12
```

Define the AXI_SLV_REG0~REG4

We can use the function to Read/Write the reg



INCLUDEFILES=*.h
LIBSOURCES=*.c
OUTS = *.o

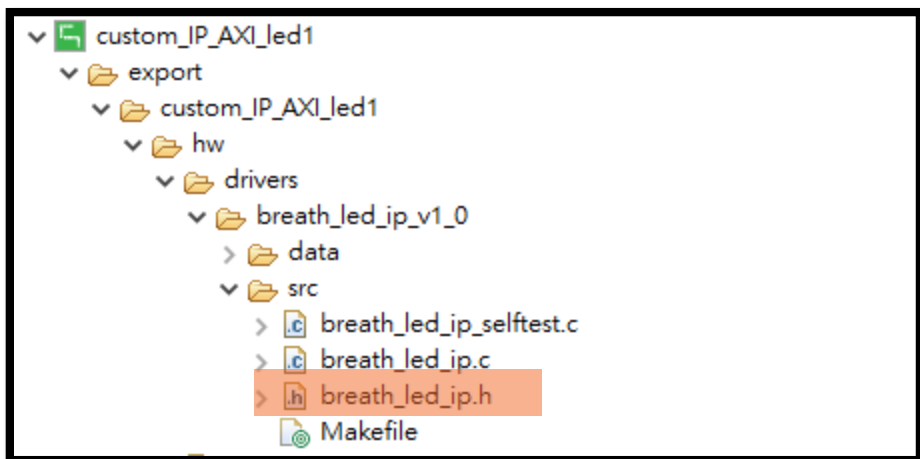


INCLUDEFILES=\$(wildcard *.h)
LIBSOURCES=\$(wildcard *.c)
OUTS = \$(wildcard *.o)

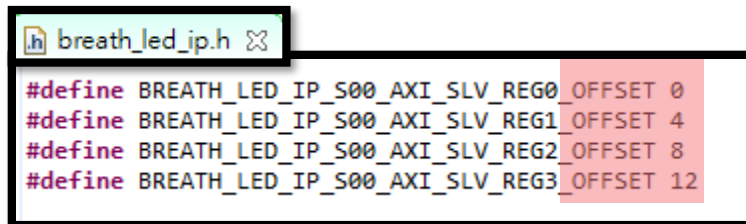
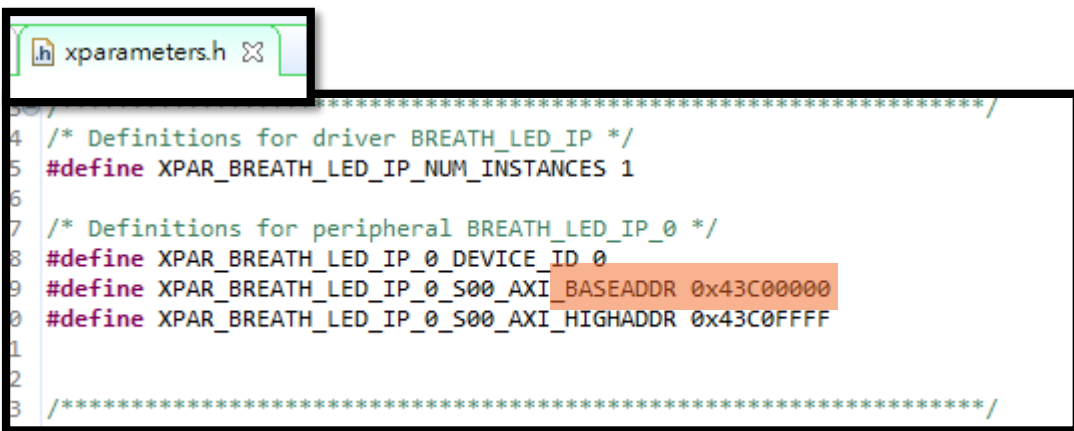
```

1  COMPILER=
2  ARCHIVER=
3  CP=cp
4  COMPILER_FLAGS=
5  EXTRA_COMPILER_FLAGS=
6  LIB=libxil.a
7
8  RELEASEDIR=../../lib
9  INCLUDEDIR=../../include
10 INCLUDES=-I./ -I${INCLUDEDIR}
11
12 INCLUDEFILES=$(wildcard *.h)
13 LIBSOURCES=$(wildcard *.c)
14 OUTS = $(wildcard *.o)
15
16 libs:
17     echo "Compiling axiled..."
18     $(COMPILER) $(COMPILER_FLAGS) $(EXTRA_COMPILER_FLAGS) $(INCLUDES) $(LIBSOURCES)
19     $(ARCHIVER) -r ${RELEASEDIR}/${LIB} ${OUTS}
20     make clean
21
22 include:
23     ${CP} $(INCLUDEFILES) $(INCLUDEDIR)
24
25 clean:
26     rm -rf ${OUTS}
27

```



```
#define BREATH_LED_IP_mWriteReg(BaseAddress, RegOffset, Data)
```



References

UG1118

