VC707 BIST Flow

Field Application Engineer
Adaptive and Embedded Computing Group (AECG)

AMD
together we advance_

Revision History

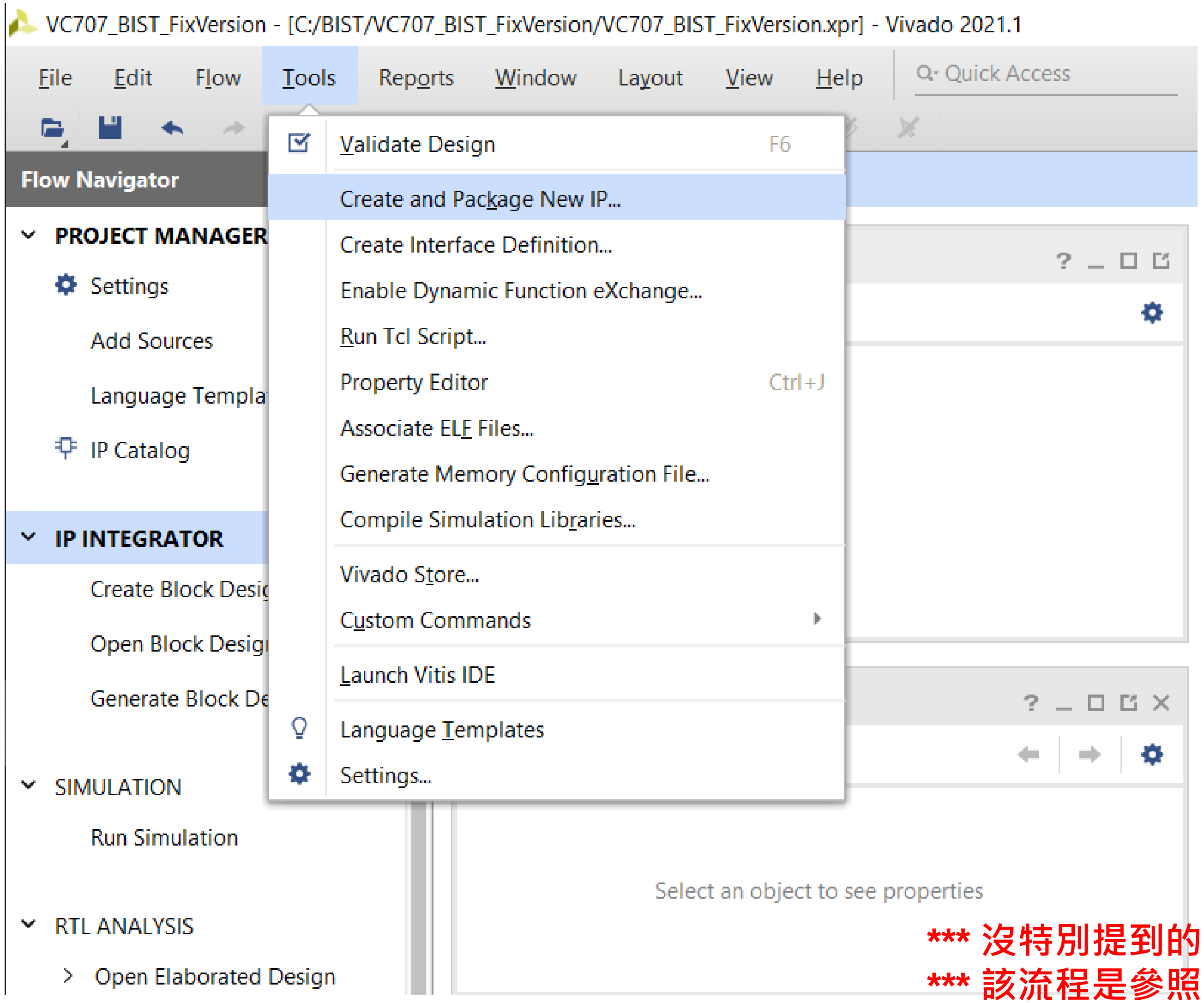
Date	Version	Description
10/25/23	1.0	Initial version for flow introduction.

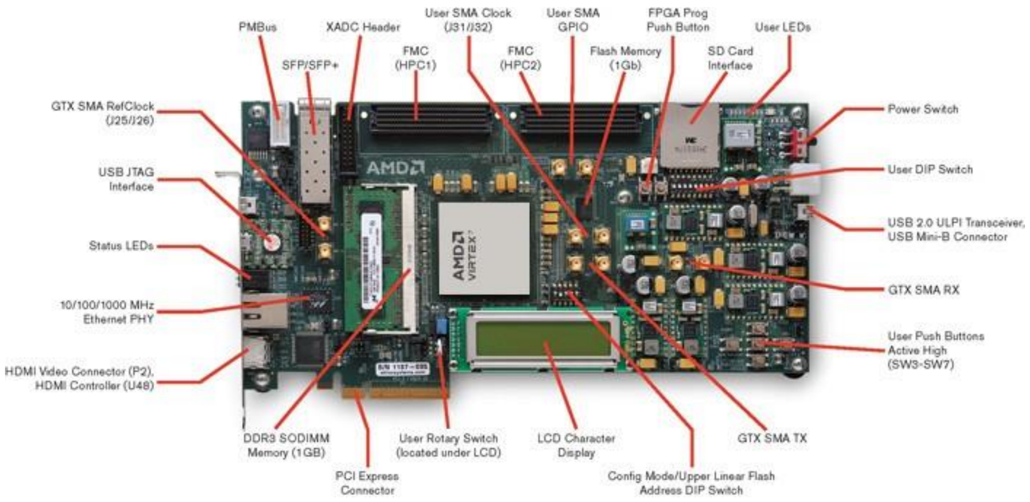
NOTICE OF DISCLAIMER: The information disclosed to you hereunder (the “Information”) is provided “AS-IS” with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

Vivado 2021.1 Part

VC707 BIST Flow

vc707_bist.zip



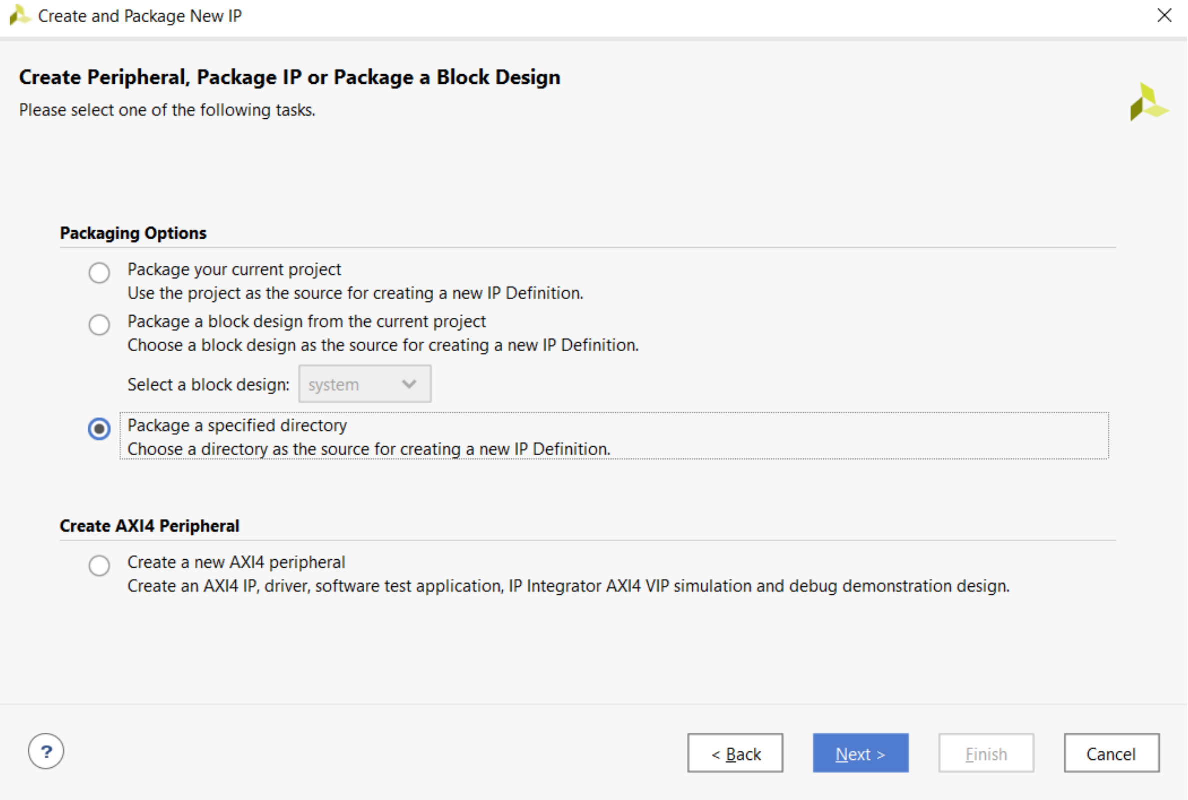


*** 沒特別提到的選項就是按 OK 或是 Next, Finish 之類的一直按下去

*** 該流程是參照 Vivado 2014.1 移植到 2021.1，若有其他版本請自己移植

together we advance_

VC707 BIST Flow



Create and Package New IP

Create Peripheral, Package IP or Package a Block Design

Please select one of the following tasks.

Packaging Options

○ Package your current project
Use the project as the source for creating a new IP Definition.

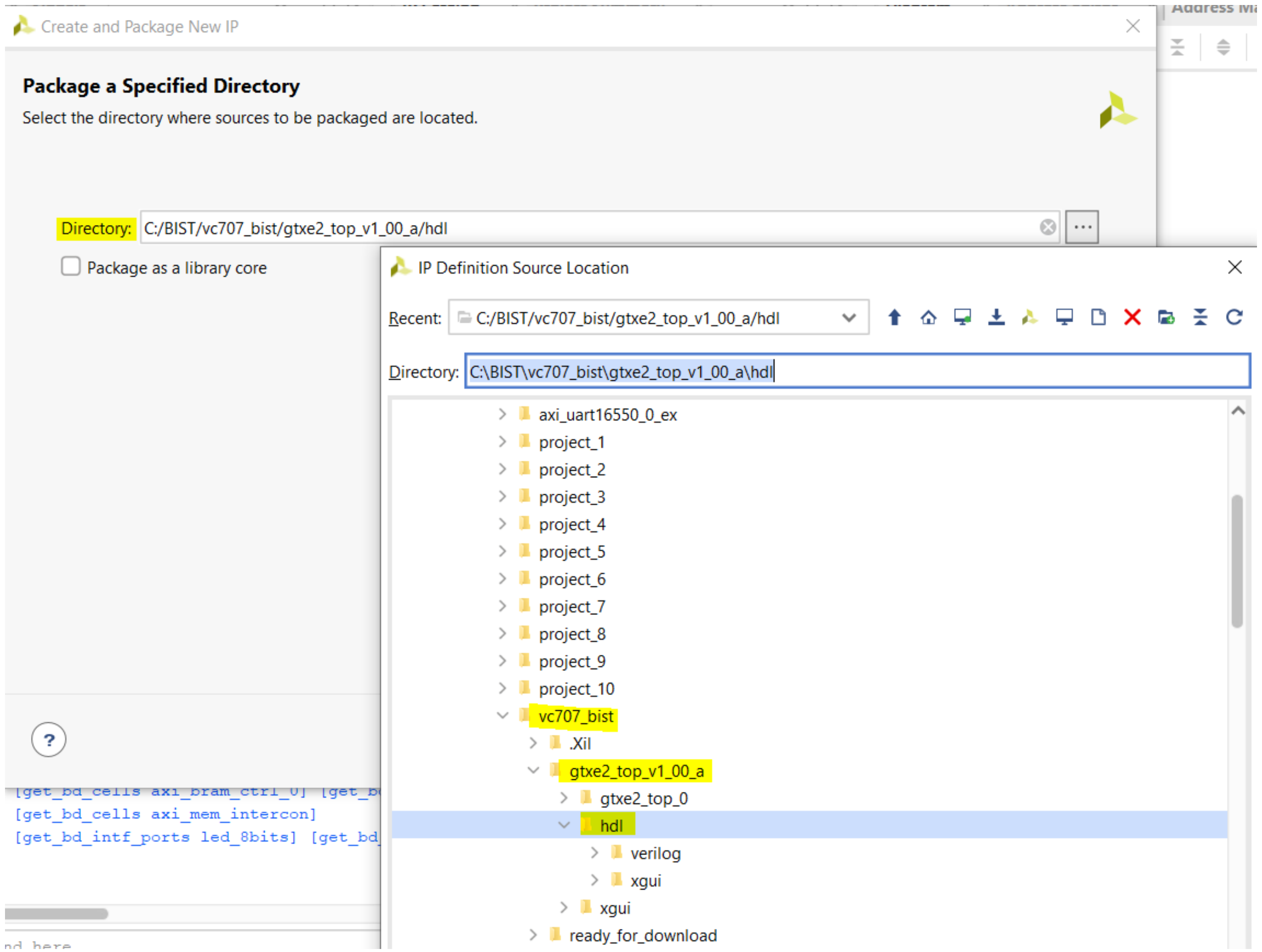
○ Package a block design from the current project
Choose a block design as the source for creating a new IP Definition.

Select a block design: system

● Package a specified directory
Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

○ Create a new AXI4 peripheral
Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.



Create and Package New IP

Package a Specified Directory

Select the directory where sources to be packaged are located.

Directory: C:/BIST/vc707_bist/gtxe2_top_v1_00_a/hdl

□ Package as a library core

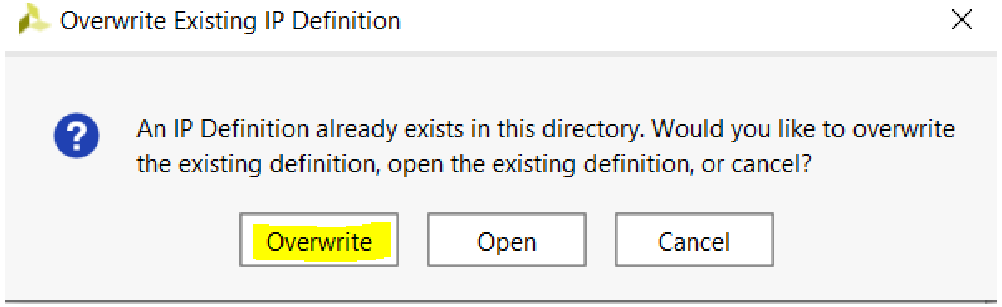
```
[get_bd_cells axi_bram_ctrl_0] [get_b
[get_bd_cells axi_mem_intercon]
[get_bd_intf_ports led_8bits] [get_bd
```

IP Definition Source Location

Recent: C:/BIST/vc707_bist/gtxe2_top_v1_00_a/hdl

Directory: C:\BIST\vc707_bist\gtxe2_top_v1_00_a\hdl

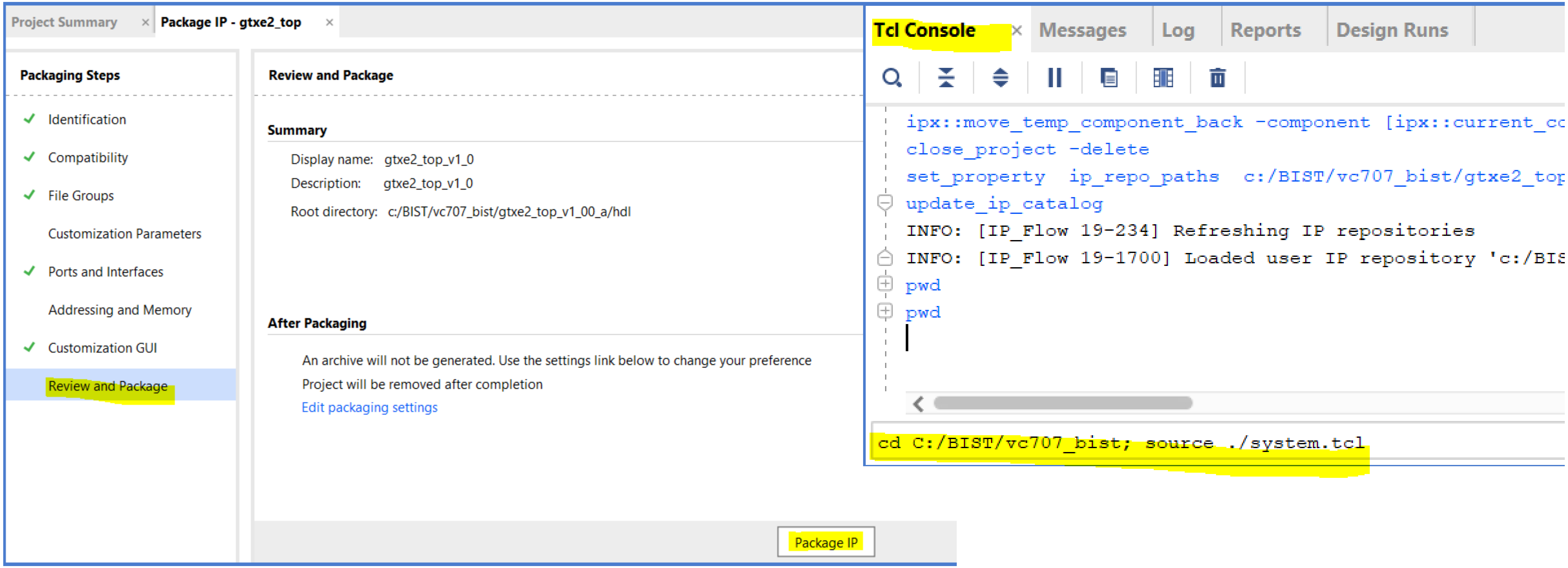
```
> axi_uart16550_0_ex
> project_1
> project_2
> project_3
> project_4
> project_5
> project_6
> project_7
> project_8
> project_9
> project_10
v vc707_bist
    > .Xil
    v gtxe2_top_v1_00_a
        > gtxe2_top_0
        v hdl
            > verilog
            > xgui
        > xgui
    > ready_for_download
```



Overwrite Existing IP Definition

An IP Definition already exists in this directory. Would you like to overwrite the existing definition, open the existing definition, or cancel?

VC707 BIST Flow



Review and Package

Summary

Display name: gtxe2_top_v1_0

Description: gtxe2_top_v1_0

Root directory: c:/BIST/vc707_bist/gtxe2_top_v1_00_a/hdl

After Packaging

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

Project will be removed after completion

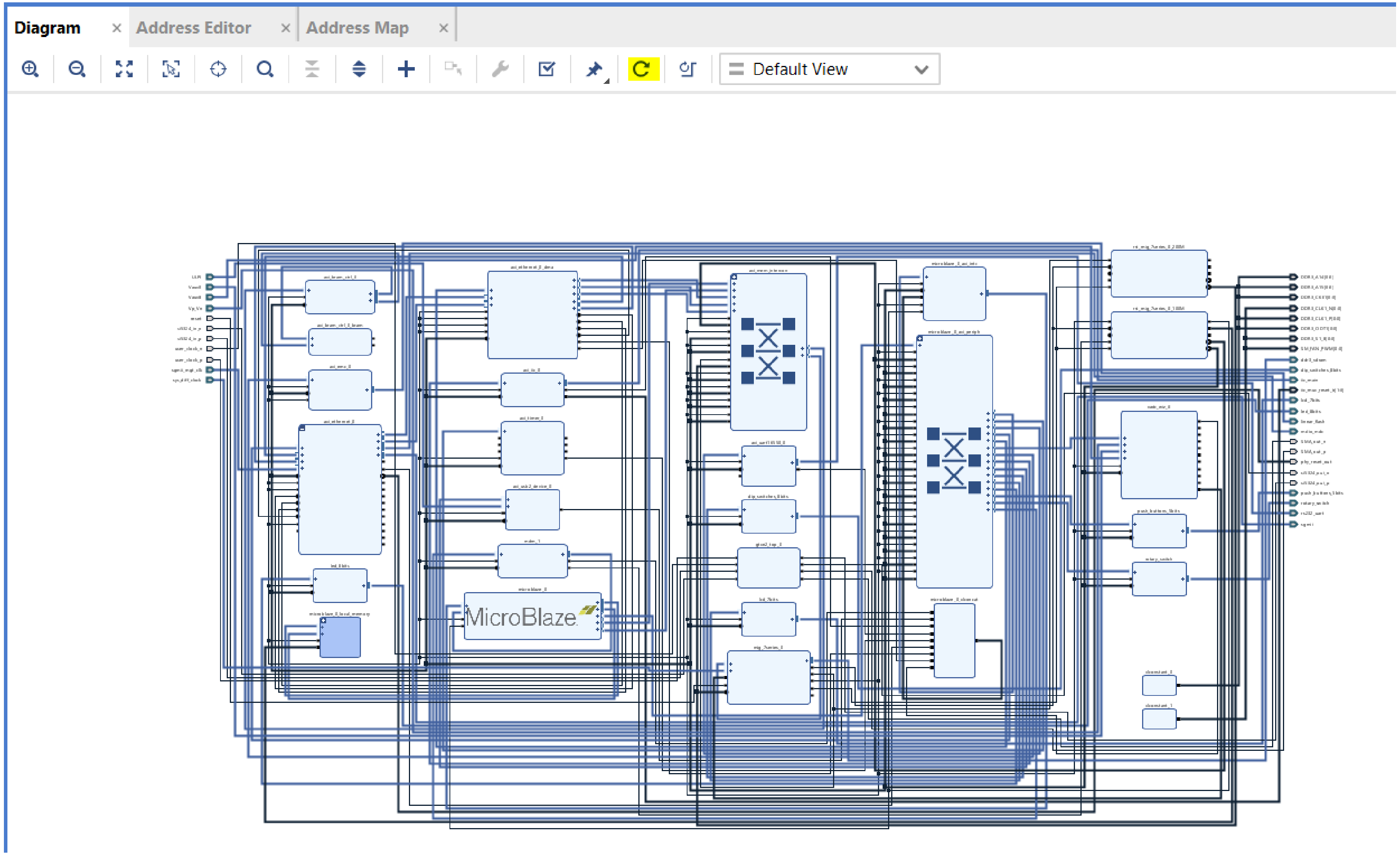
Edit packaging settings

```
ipx::move_temp_component_back -component [ipx::current_cc
close_project -delete
set_property  ip_repo_paths  c:/BIST/vc707_bist/gtxe2_top
update_ip_catalog
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1700] Loaded user IP repository 'c:/BIS
pwd
pwd
```

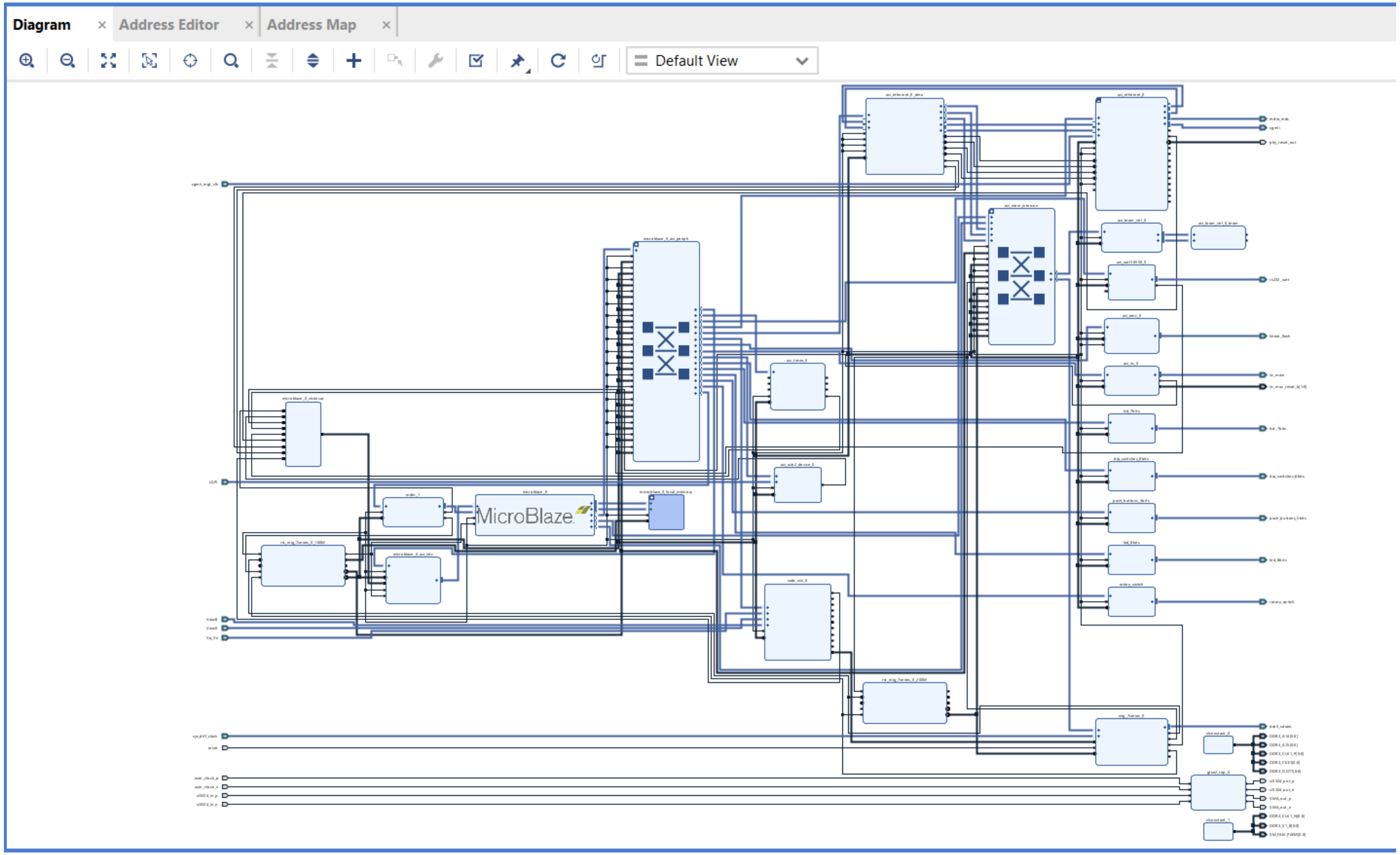
```
cd C:/BIST/vc707_bist; source ./system.tcl
```

Wait a minute…

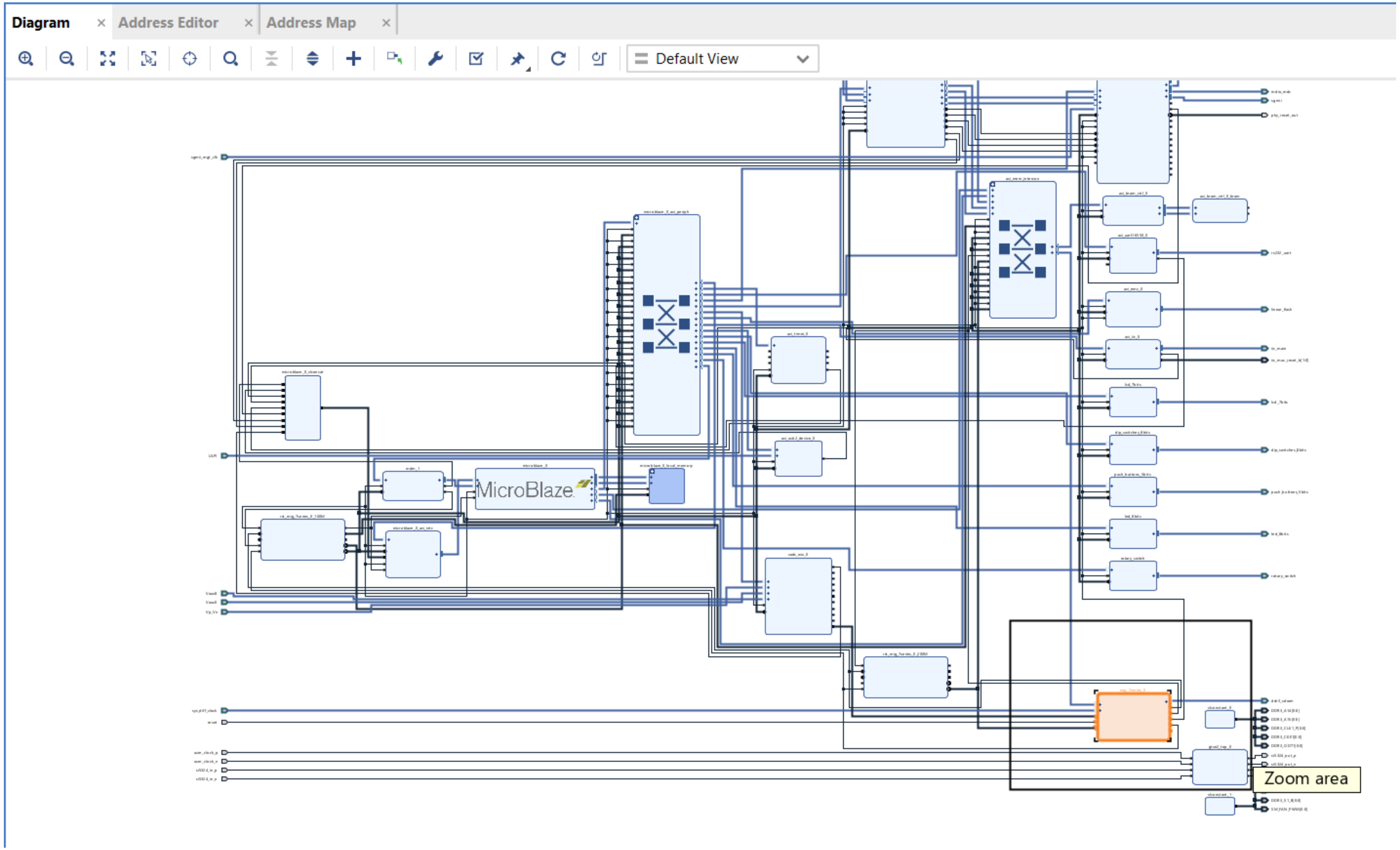
VC707 BIST Flow



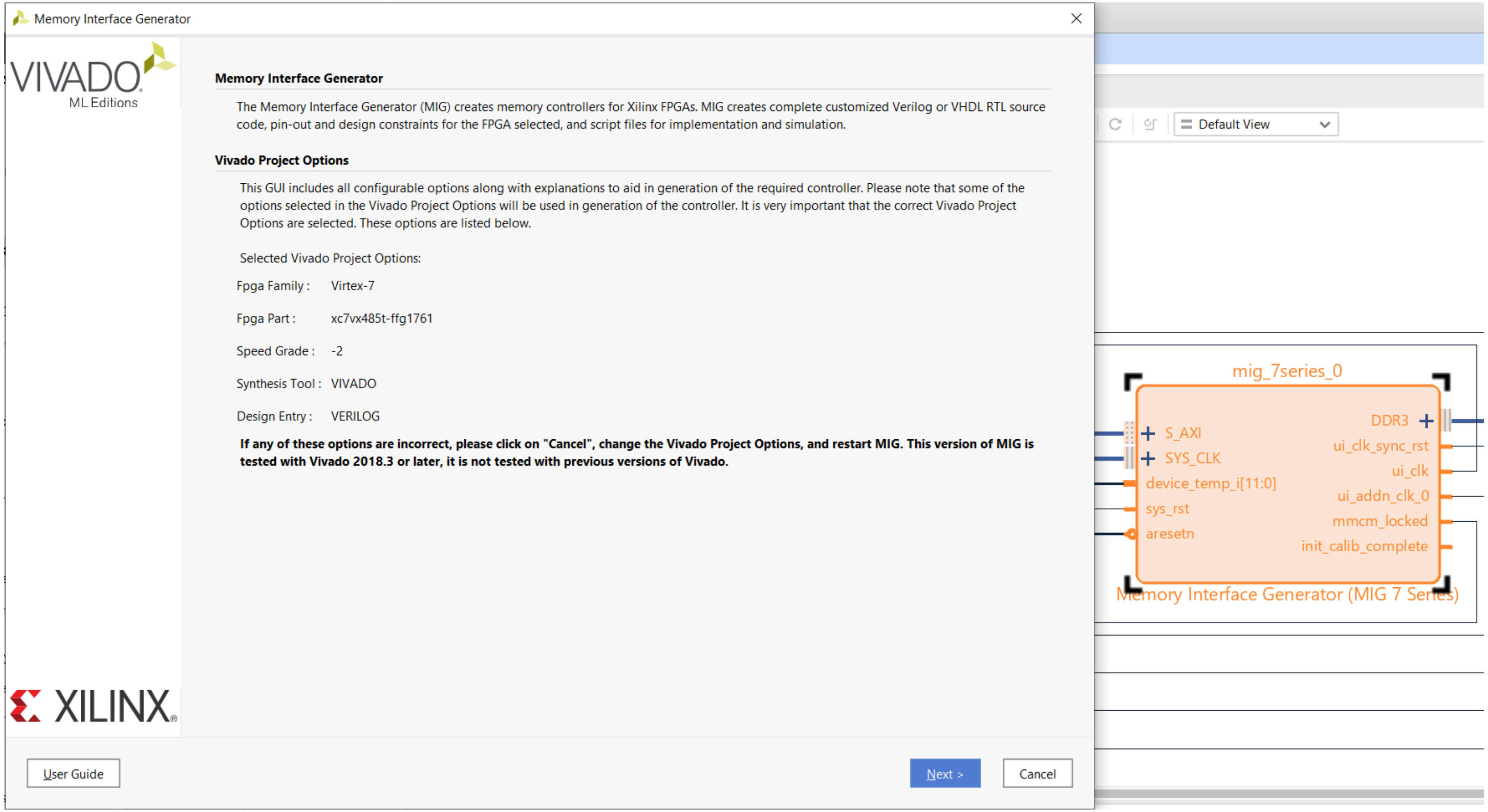
VC707 BIST Flow



VC707 BIST Flow



VC707 BIST Flow



Memory Interface Generator

The Memory Interface Generator (MIG) creates memory controllers for Xilinx FPGAs. MIG creates complete customized Verilog or VHDL RTL source code, pin-out and design constraints for the FPGA selected, and script files for implementation and simulation.

Vivado Project Options

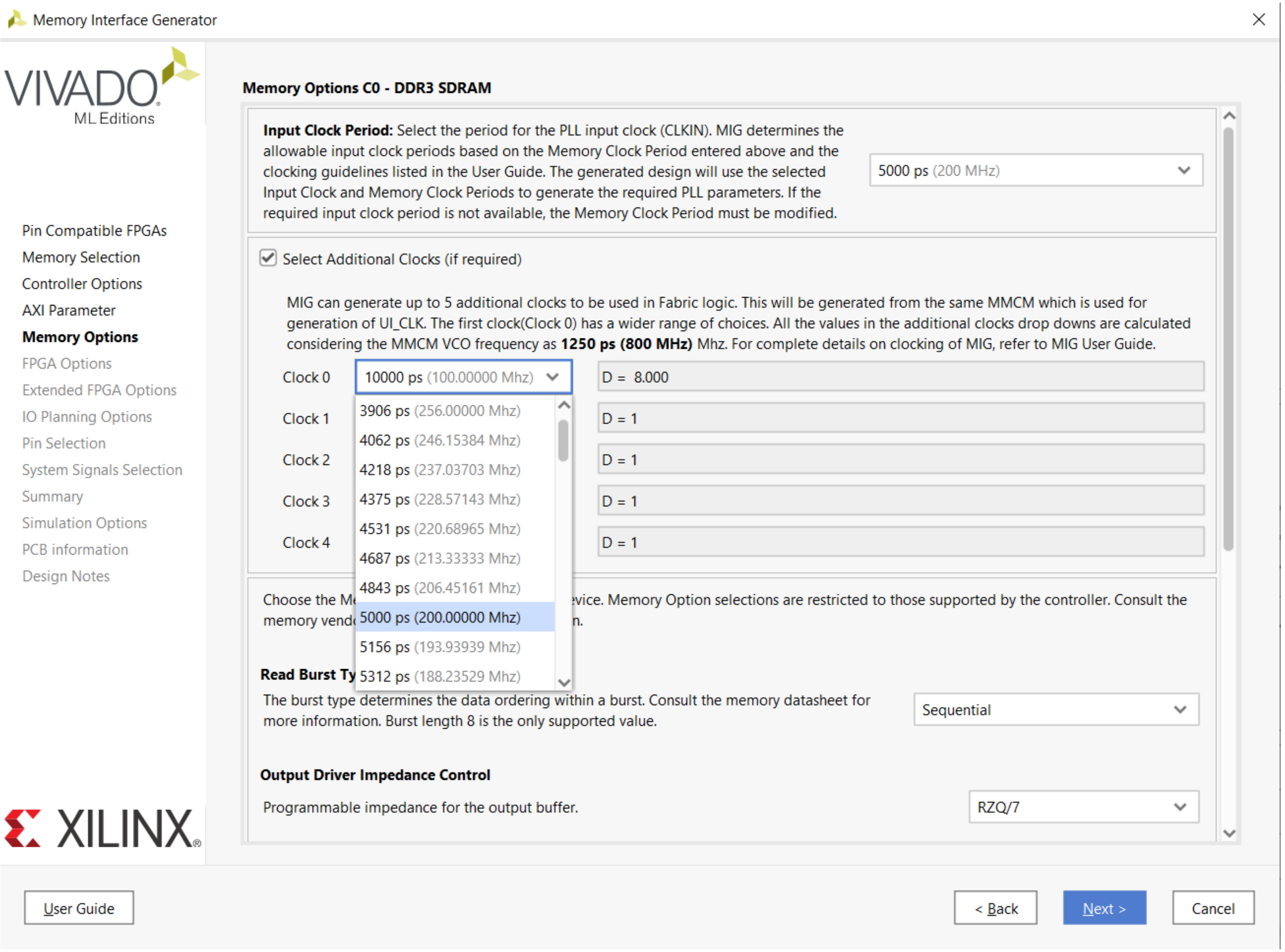
This GUI includes all configurable options along with explanations to aid in generation of the required controller. Please note that some of the options selected in the Vivado Project Options will be used in generation of the controller. It is very important that the correct Vivado Project Options are selected. These options are listed below.

Selected Vivado Project Options:

Fpga Family :	Virtex-7
Fpga Part :	xc7vx485t-ffg1761
Speed Grade :	-2
Synthesis Tool :	VIVADO
Design Entry :	VERILOG

If any of these options are incorrect, please click on "Cancel", change the Vivado Project Options, and restart MIG. This version of MIG is tested with Vivado 2018.3 or later, it is not tested with previous versions of Vivado.

VC707 BIST Flow



Memory Options C0 - DDR3 SDRAM

Input Clock Period: Select the period for the PLL input clock (CLKIN). MIG determines the allowable input clock periods based on the Memory Clock Period entered above and the clocking guidelines listed in the User Guide. The generated design will use the selected Input Clock and Memory Clock Periods to generate the required PLL parameters. If the required input clock period is not available, the Memory Clock Period must be modified.

5000 ps (200 MHz)

☑ Select Additional Clocks (if required)

MIG can generate up to 5 additional clocks to be used in Fabric logic. This will be generated from the same MMCM which is used for generation of UI_CLK. The first clock(Clock 0) has a wider range of choices. All the values in the additional clocks drop downs are calculated considering the MMCM VCO frequency as **1250 ps (800 MHz)** Mhz. For complete details on clocking of MIG, refer to MIG User Guide.

Clock 0 10000 ps (100.00000 Mhz) D = 8.000

Clock 1 D = 1

Clock 2 D = 1

Clock 3 D = 1

Clock 4 D = 1

```
3906 ps (256.00000 Mhz)
4062 ps (246.15384 Mhz)
4218 ps (237.03703 Mhz)
4375 ps (228.57143 Mhz)
4531 ps (220.68965 Mhz)
4687 ps (213.33333 Mhz)
4843 ps (206.45161 Mhz)
5000 ps (200.00000 Mhz)
5156 ps (193.93939 Mhz)
5312 ps (188.23529 Mhz)
```

Choose the M… …vice. Memory Option selections are restricted to those supported by the controller. Consult the memory vend… …n.

Read Burst Ty…

The burst type determines the data ordering within a burst. Consult the memory datasheet for more information. Burst length 8 is the only supported value.

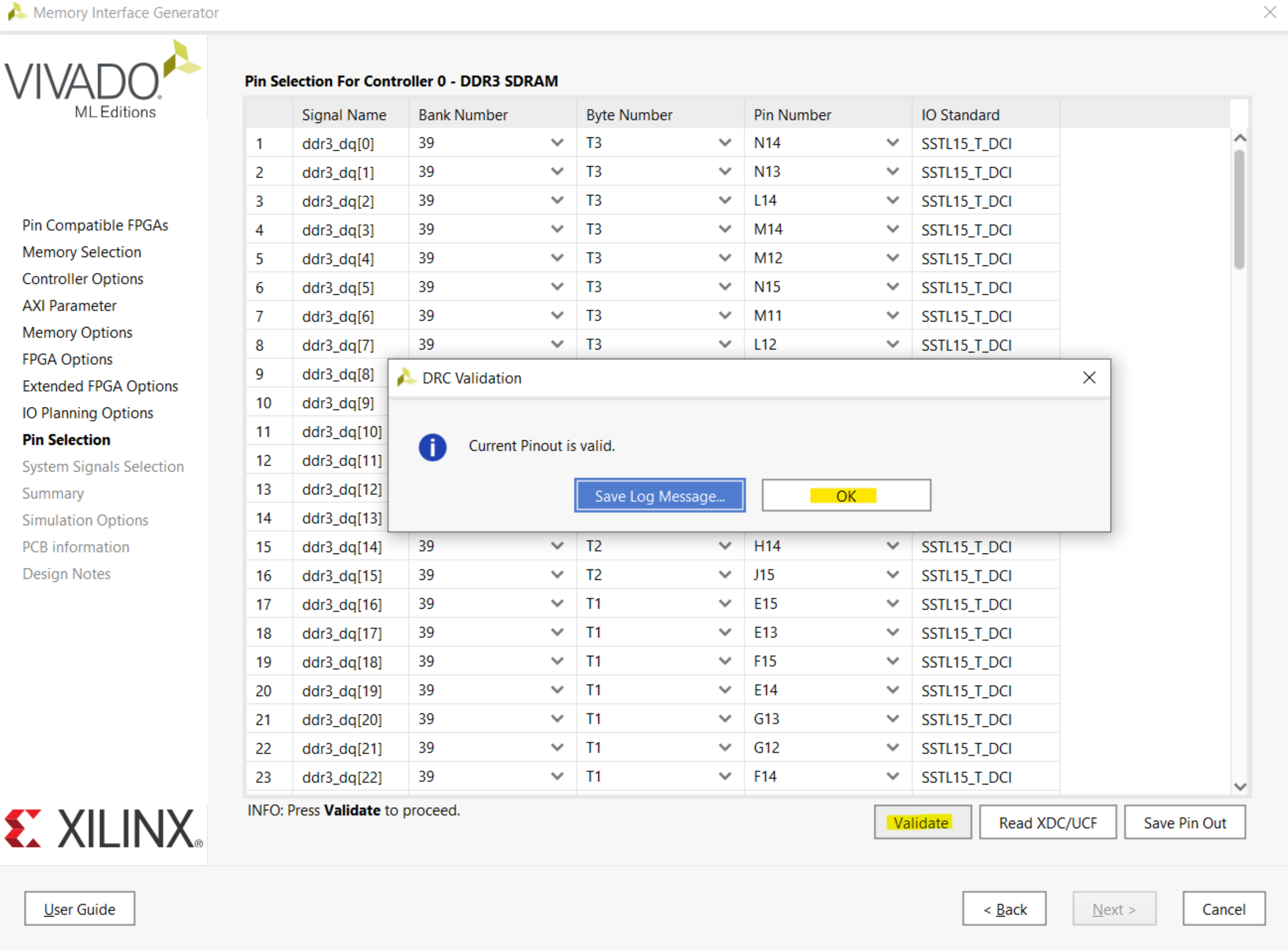
Sequential

Output Driver Impedance Control

Programmable impedance for the output buffer.

RZQ/7

VC707 BIST Flow



Pin Selection For Controller 0 - DDR3 SDRAM

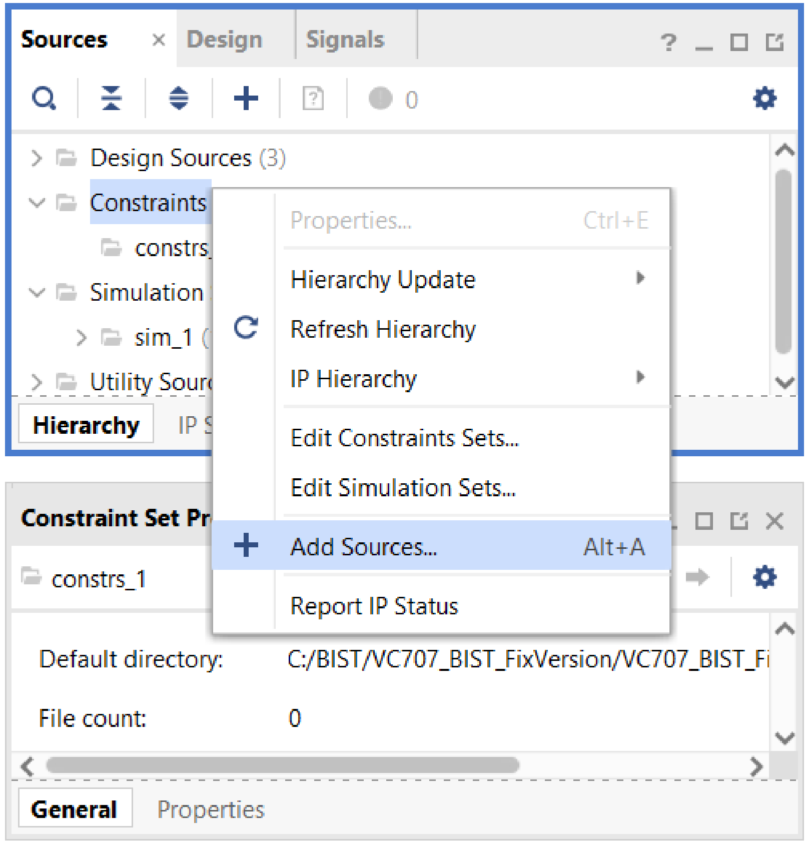
	Signal Name	Bank Number	Byte Number	Pin Number	IO Standard
1	ddr3_dq[0]	39	T3	N14	SSTL15_T_DCI
2	ddr3_dq[1]	39	T3	N13	SSTL15_T_DCI
3	ddr3_dq[2]	39	T3	L14	SSTL15_T_DCI
4	ddr3_dq[3]	39	T3	M14	SSTL15_T_DCI
5	ddr3_dq[4]	39	T3	M12	SSTL15_T_DCI
6	ddr3_dq[5]	39	T3	N15	SSTL15_T_DCI
7	ddr3_dq[6]	39	T3	M11	SSTL15_T_DCI
8	ddr3_dq[7]	39	T3	L12	SSTL15_T_DCI
9	ddr3_dq[8]				
10	ddr3_dq[9]				
11	ddr3_dq[10]				
12	ddr3_dq[11]				
13	ddr3_dq[12]				
14	ddr3_dq[13]				
15	ddr3_dq[14]	39	T2	H14	SSTL15_T_DCI
16	ddr3_dq[15]	39	T2	J15	SSTL15_T_DCI
17	ddr3_dq[16]	39	T1	E15	SSTL15_T_DCI
18	ddr3_dq[17]	39	T1	E13	SSTL15_T_DCI
19	ddr3_dq[18]	39	T1	F15	SSTL15_T_DCI
20	ddr3_dq[19]	39	T1	E14	SSTL15_T_DCI
21	ddr3_dq[20]	39	T1	G13	SSTL15_T_DCI
22	ddr3_dq[21]	39	T1	G12	SSTL15_T_DCI
23	ddr3_dq[22]	39	T1	F14	SSTL15_T_DCI

DRC Validation

Current Pinout is valid.

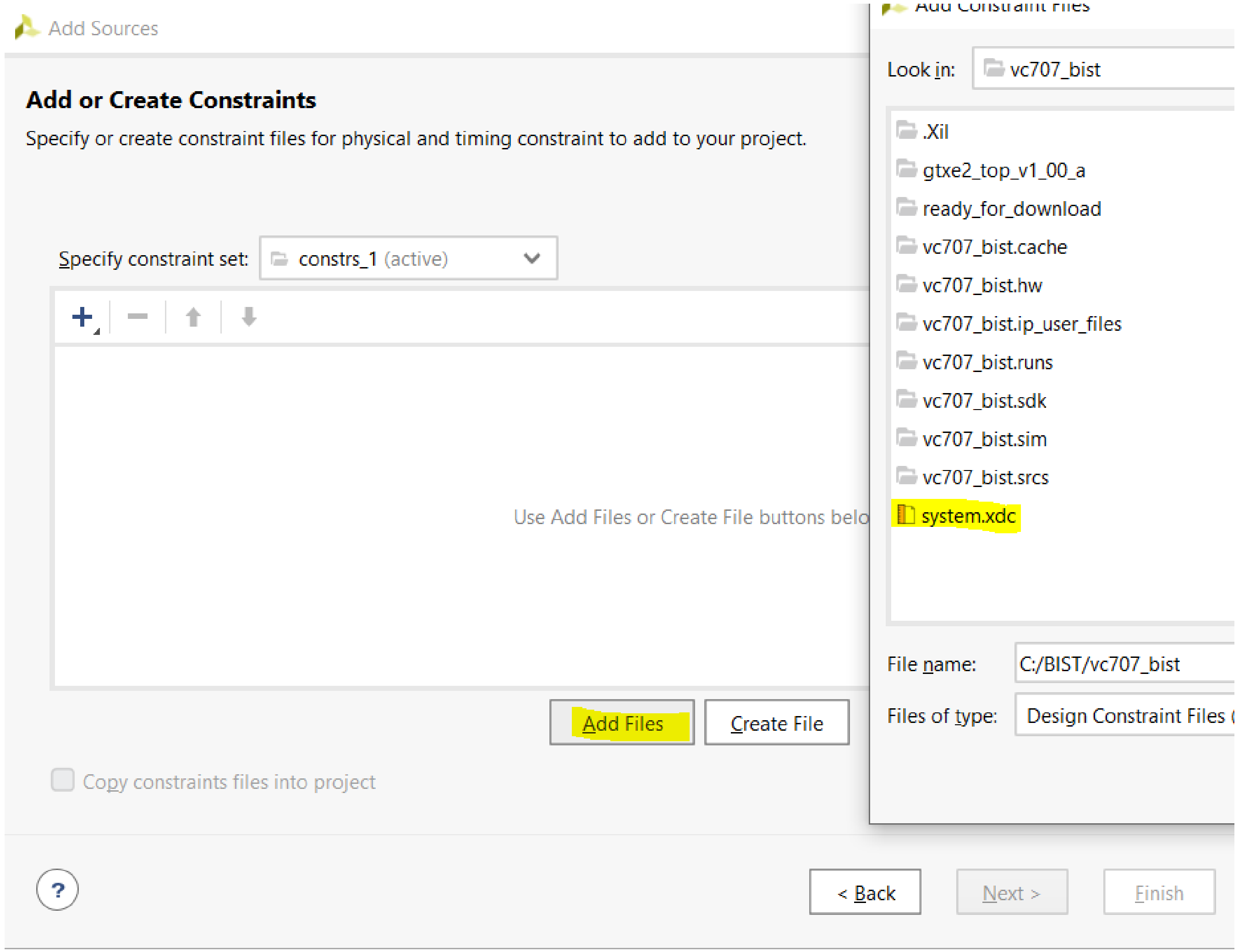
INFO: Press **Validate** to proceed.

VC707 BIST Flow



Default directory: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_Fi

File count: 0



Add or Create Constraints

Specify or create constraint files for physical and timing constraint to add to your project.

Specify constraint set: constrs_1 (active)

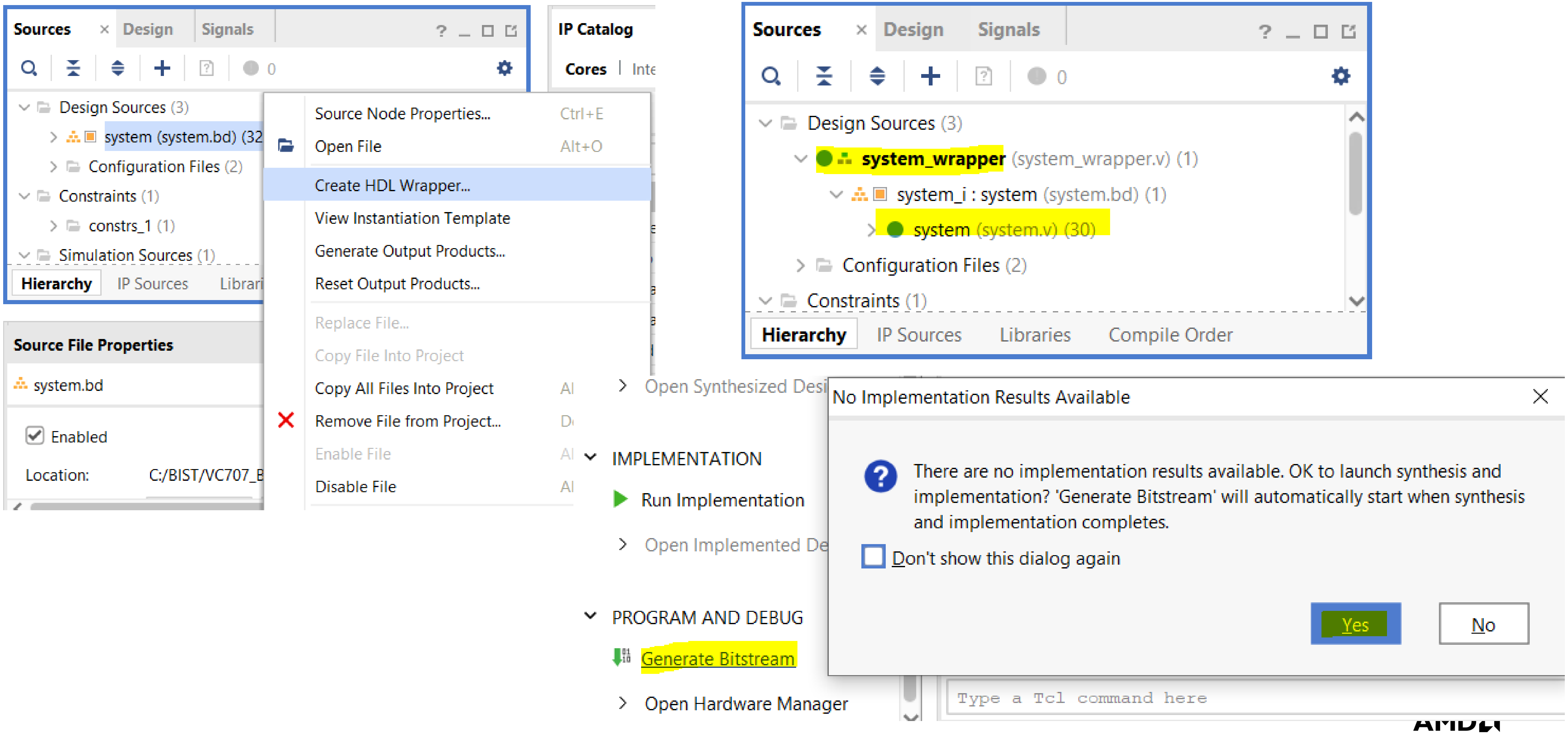
Use Add Files or Create File buttons belo

Look in: vc707_bist

File name: C:/BIST/vc707_bist

Files of type: Design Constraint Files (

VC707 BIST Flow



No Implementation Results Available

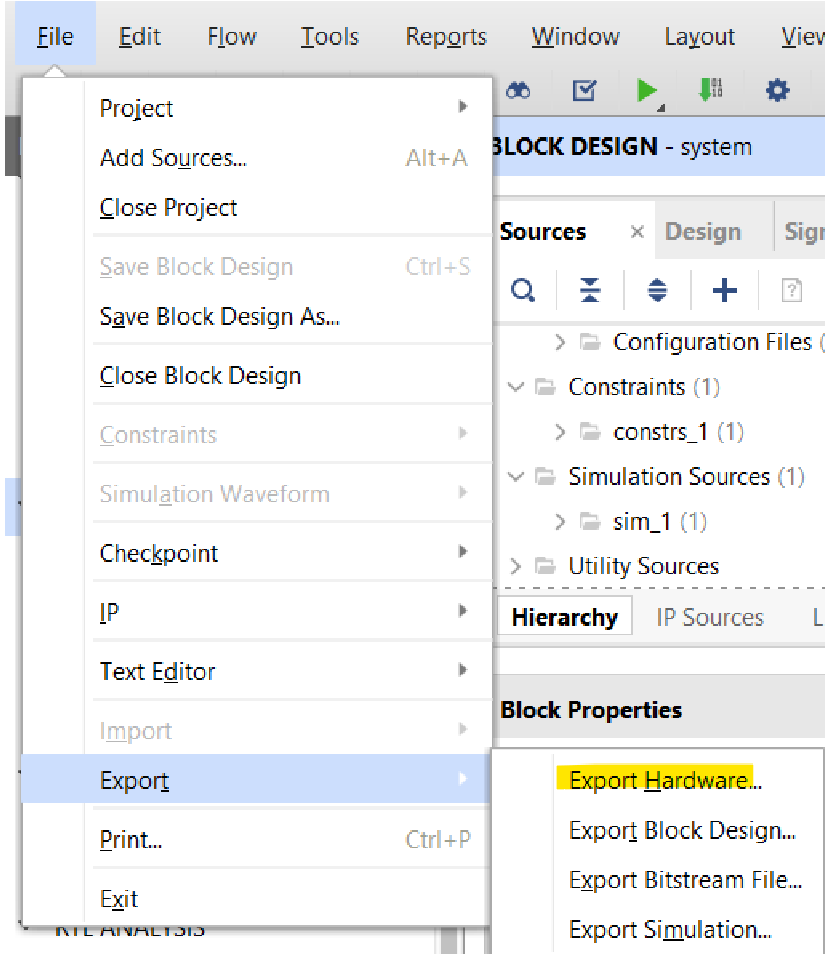
There are no implementation results available. OK to launch synthesis and implementation? 'Generate Bitstream' will automatically start when synthesis and implementation completes.

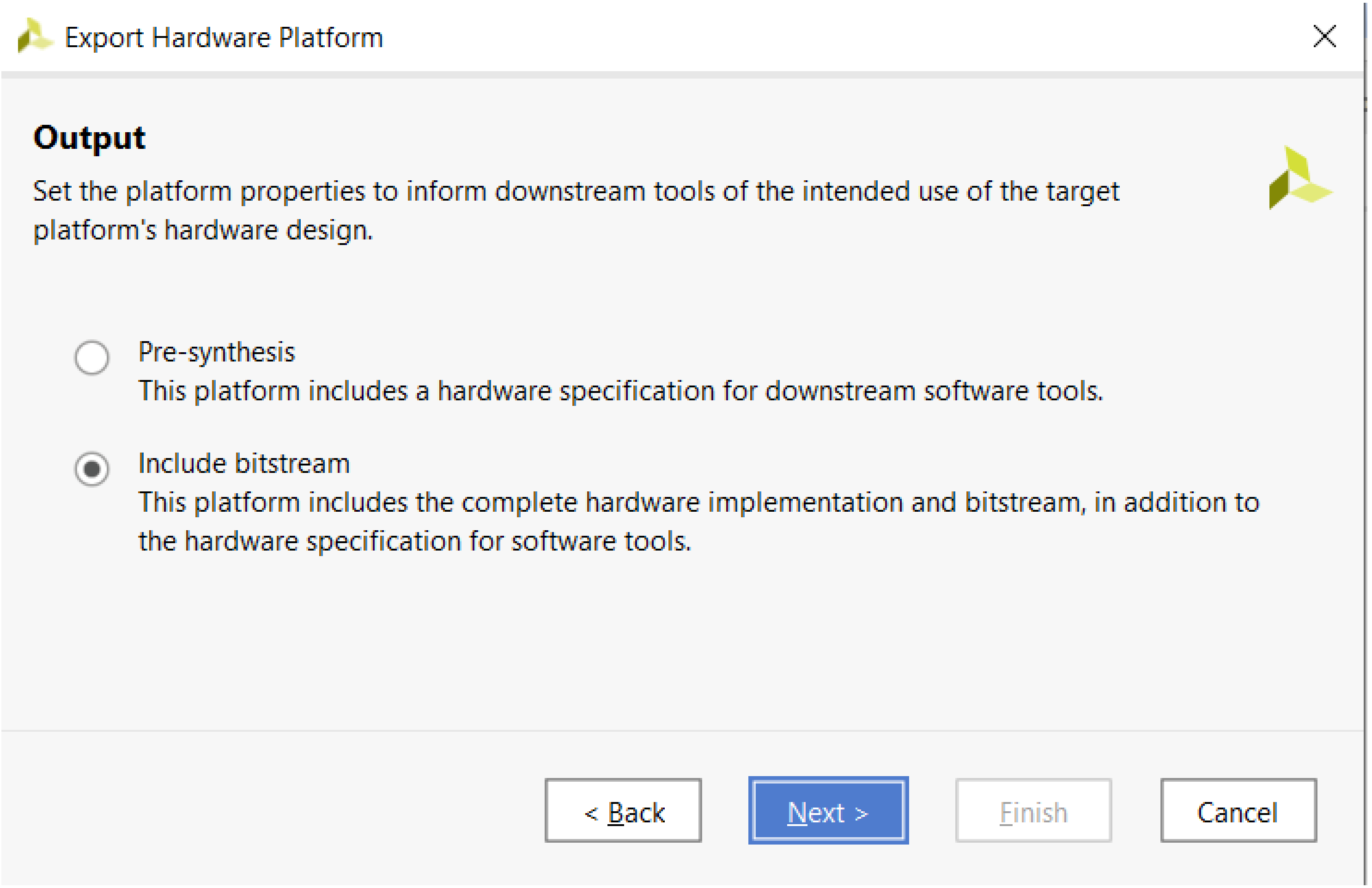
□ Don't show this dialog again

Yes No

THIRTY-SEVEN MINUTES LATER...

VC707 BIST Flow




Export Hardware Platform

Output

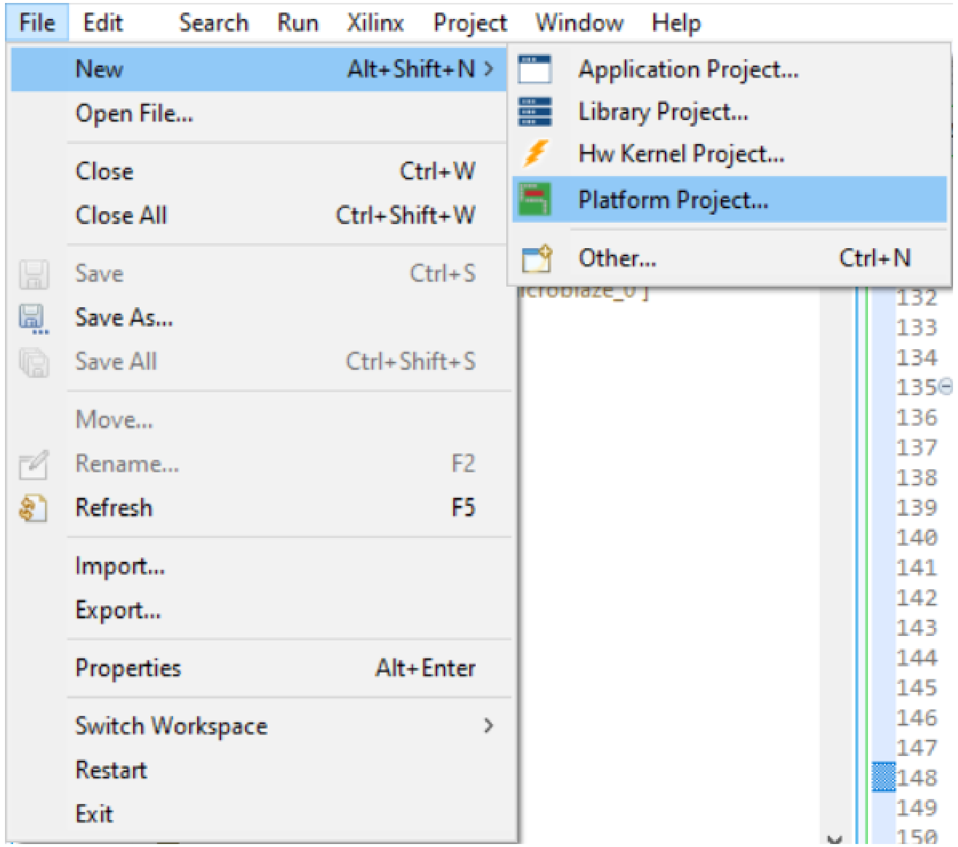
Set the platform properties to inform downstream tools of the intended use of the target platform's hardware design.

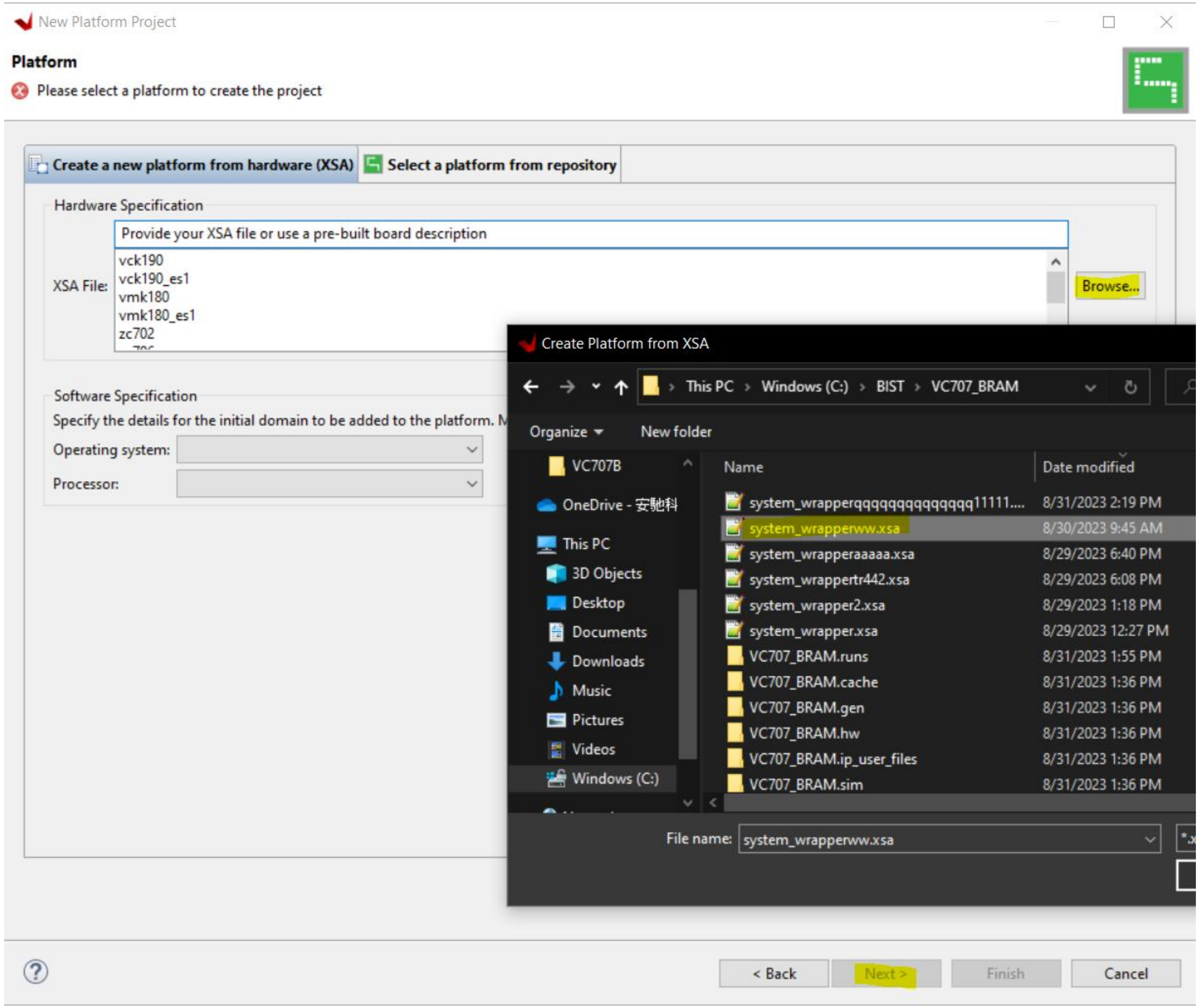
○ Pre-synthesis
This platform includes a hardware specification for downstream software tools.

● Include bitstream
This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for software tools.

Vitis 2021.1 Part

VC707 BIST Flow





Platform

Please select a platform to create the project

Hardware Specification

XSA File: Provide your XSA file or use a pre-built board description

vck190
vck190_es1
vmk180
vmk180_es1
zc702

Software Specification

Specify the details for the initial domain to be added to the platform.

Operating system:

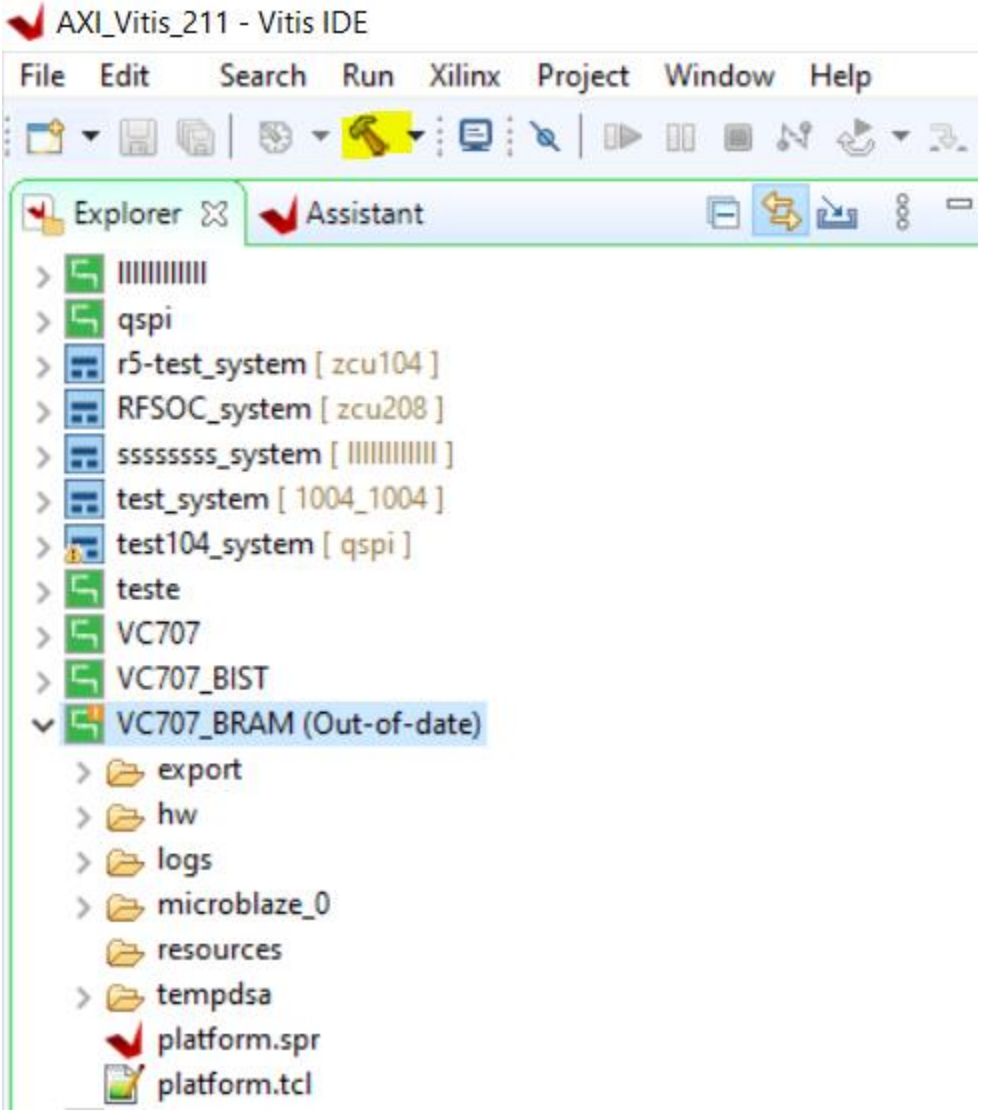
Processor:

Name	Date modified
system_wrapperqqqqqqqqqqqqqqq11111....	8/31/2023 2:19 PM
system_wrapperww.xsa	8/30/2023 9:45 AM
system_wrapperaaaaa.xsa	8/29/2023 6:40 PM
system_wrappertr442.xsa	8/29/2023 6:08 PM
system_wrapper2.xsa	8/29/2023 1:18 PM
system_wrapper.xsa	8/29/2023 12:27 PM
VC707_BRAM.runs	8/31/2023 1:55 PM
VC707_BRAM.cache	8/31/2023 1:36 PM
VC707_BRAM.gen	8/31/2023 1:36 PM
VC707_BRAM.hw	8/31/2023 1:36 PM
VC707_BRAM.ip_user_files	8/31/2023 1:36 PM
VC707_BRAM.sim	8/31/2023 1:36 PM

File name: system_wrapperww.xsa

*** 實際檔案請按照自己設定的位置與名稱去開啟

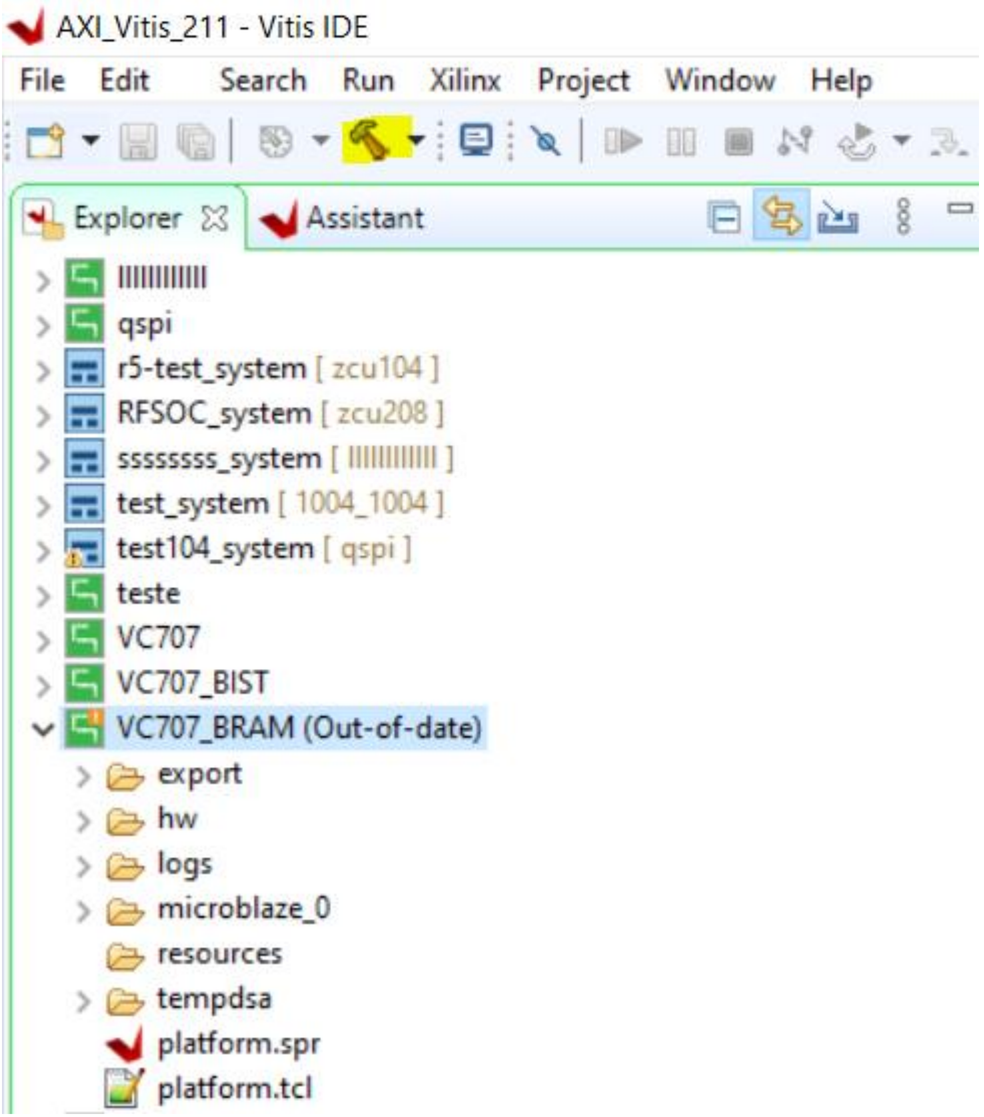
VC707 BIST Flow

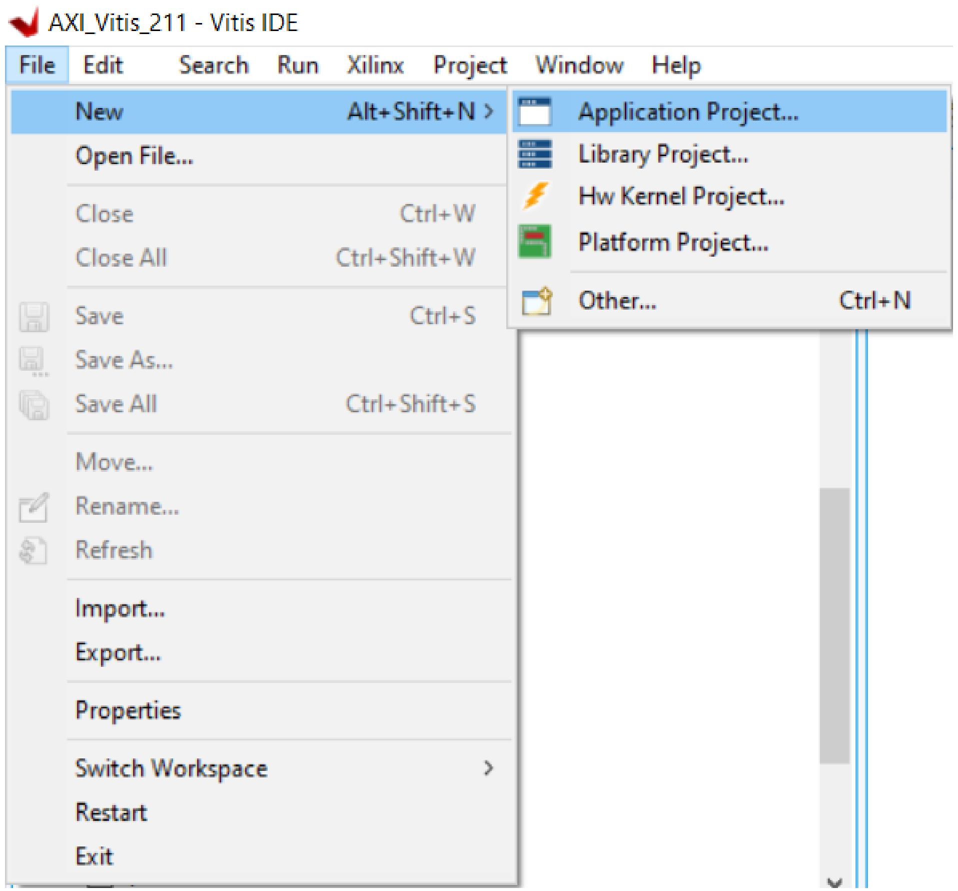


AXI_Vitis_211 - Vitis IDE

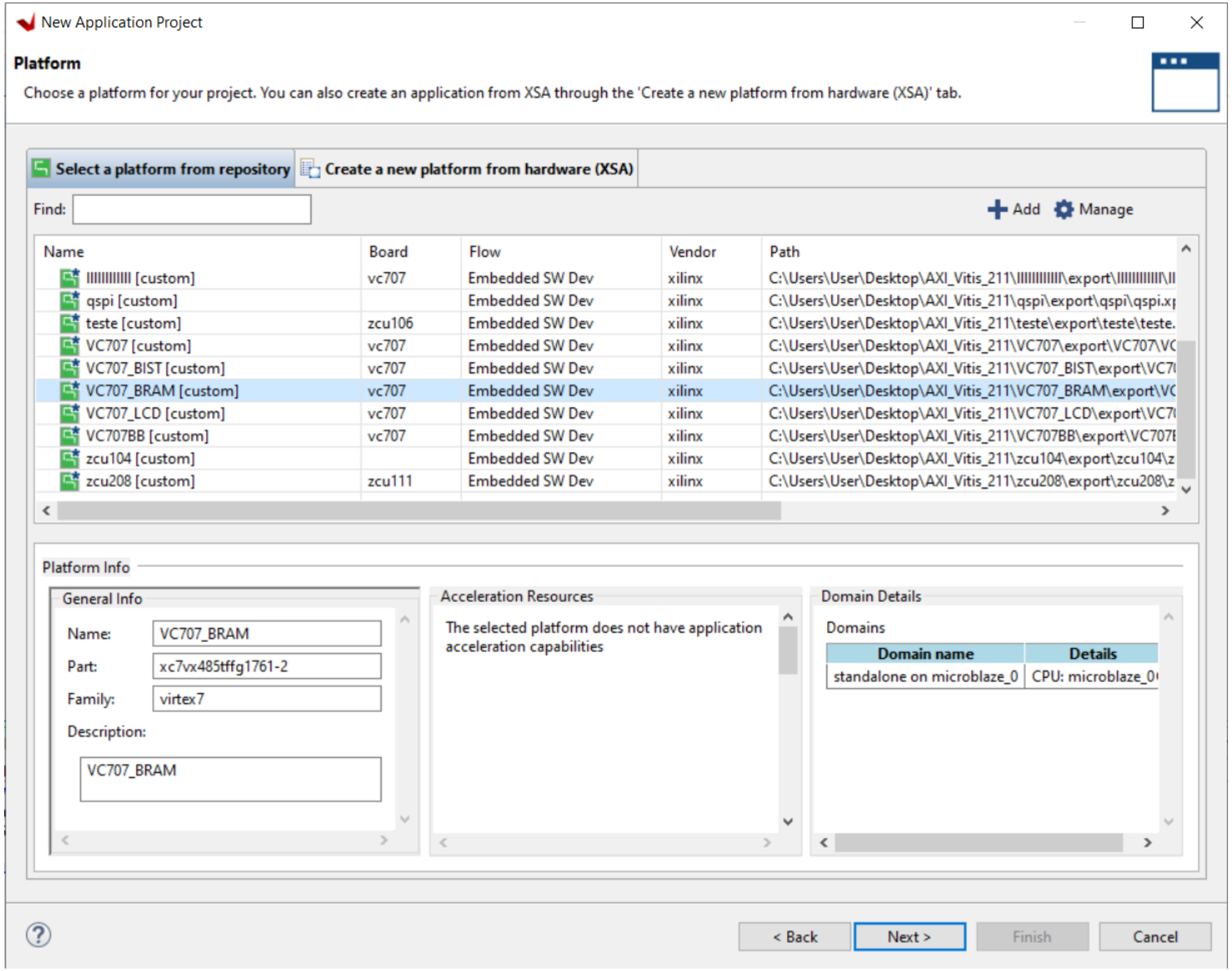
- IIIIIIIIIII
- qspi
- r5-test_system [zcu104]
- RFSOC_system [zcu208]
- ssssssss_system [IIIIIIIIIII]
- test_system [1004_1004]
- test104_system [qspi]
- teste
- VC707
- VC707_BIST
- VC707_BRAM (Out-of-date)
  - export
  - hw
  - logs
  - microblaze_0
  - resources
  - tempdsa
  - platform.spr
  - platform.tcl

VC707 BIST Flow





VC707 BIST Flow



New Application Project

Platform

Choose a platform for your project. You can also create an application from XSA through the 'Create a new platform from hardware (XSA)' tab.

Select a platform from repository
Create a new platform from hardware (XSA)
Find:
Add
Manage

Name	Board	Flow	Vendor	Path
IIIIIIIIIII [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\IIIIIIIIIII\export\IIIIIIIIIII\II
qspi [custom]		Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\qspi\export\qspi\qspi.x
teste [custom]	zcu106	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\teste\export\teste\teste.
VC707 [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\VC707\export\VC707\VC
VC707_BIST [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\VC707_BIST\export\VC7
VC707_BRAM [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\VC707_BRAM\export\VC
VC707_LCD [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\VC707_LCD\export\VC7
VC707BB [custom]	vc707	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\VC707BB\export\VC707
zcu104 [custom]		Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\zcu104\export\zcu104\z
zcu208 [custom]	zcu111	Embedded SW Dev	xilinx	C:\Users\User\Desktop\AXI_Vitis_211\zcu208\export\zcu208\z

Platform Info

General Info

Name: VC707_BRAM

Part: xc7vx485tffg1761-2

Family: virtex7

Description:

VC707_BRAM

Acceleration Resources

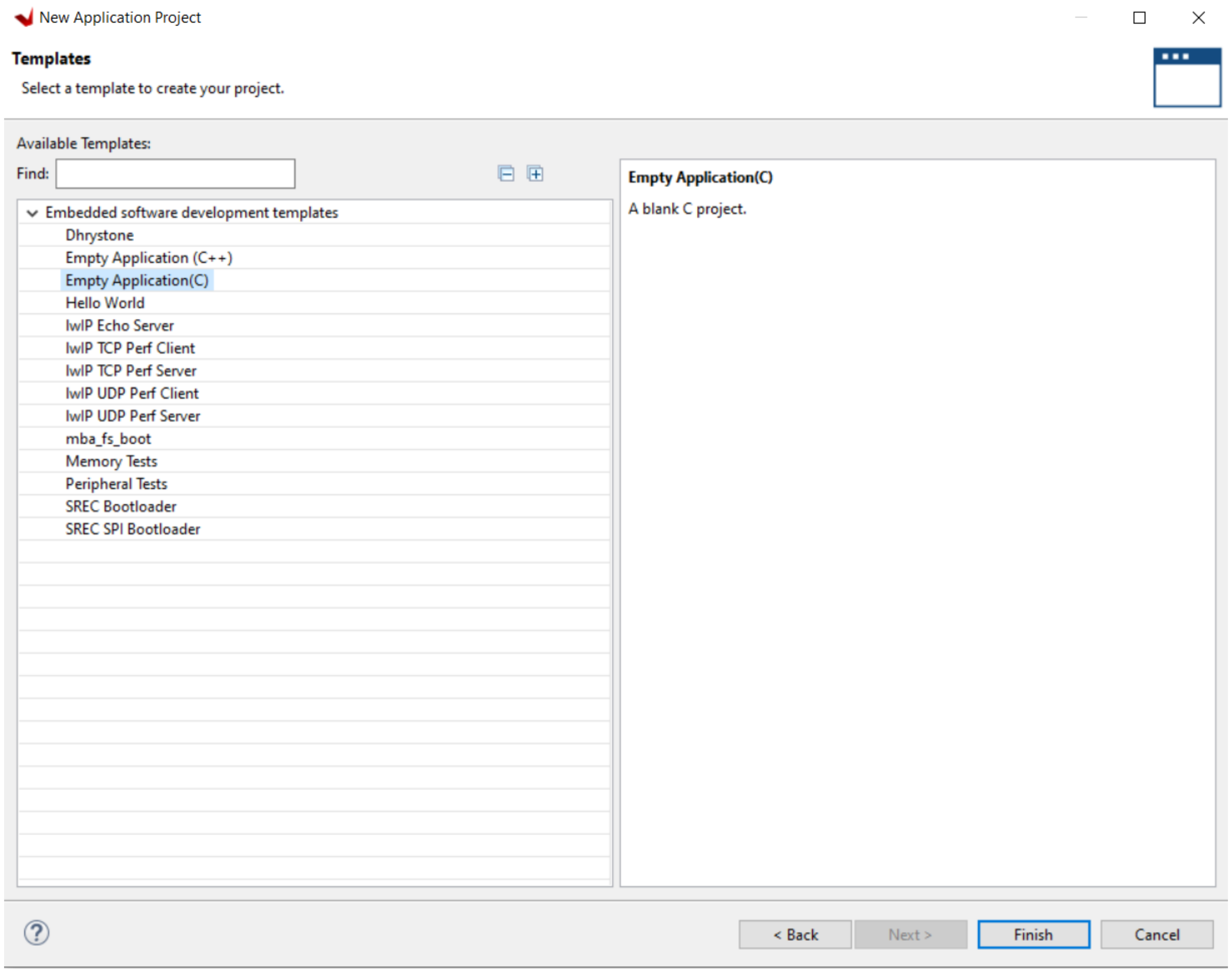
The selected platform does not have application acceleration capabilities

Domain Details

Domains

Domain name	Details
standalone on microblaze_0	CPU: microblaze_0

VC707 BIST Flow



Templates

Select a template to create your project.

Available Templates:

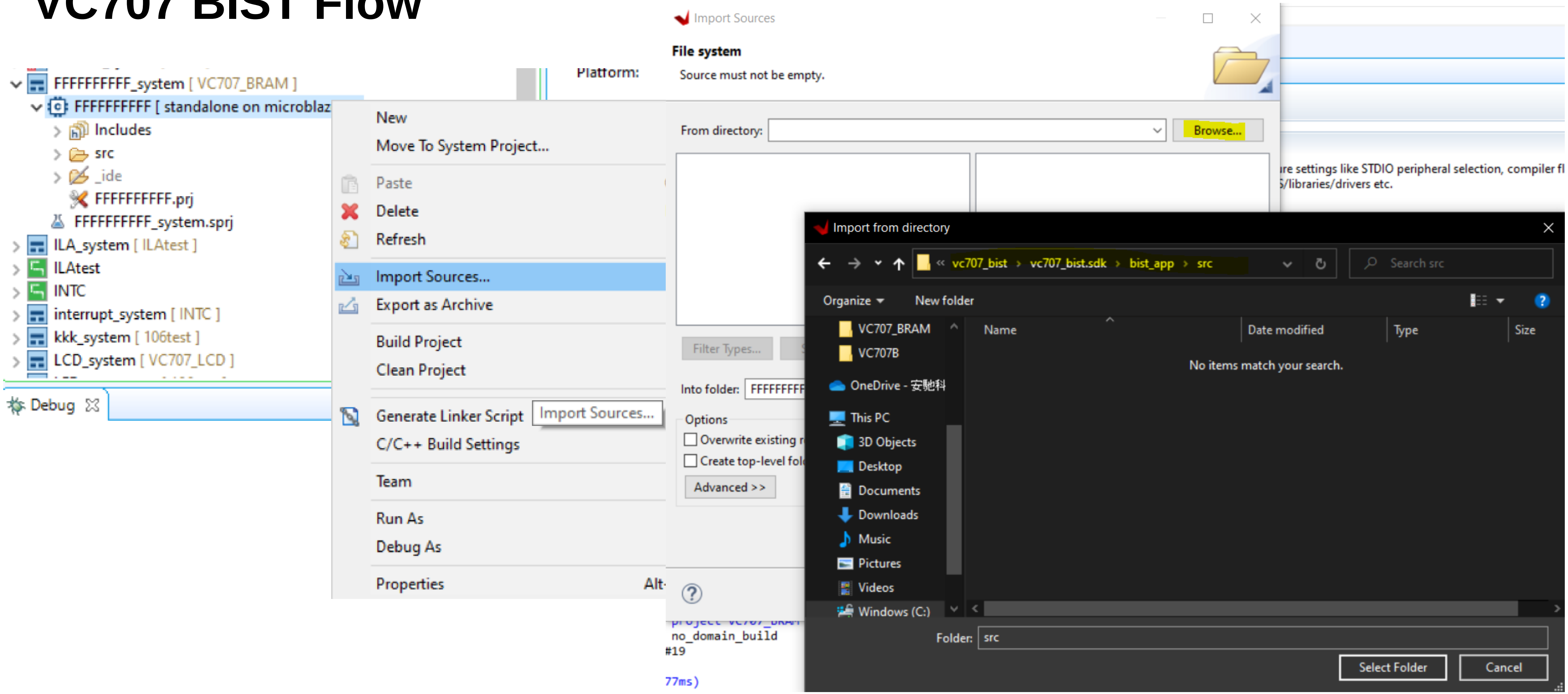
Find:

- Embedded software development templates
  - Dhrystone
  - Empty Application (C++)
  - Empty Application(C)
  - Hello World
  - lwIP Echo Server
  - lwIP TCP Perf Client
  - lwIP TCP Perf Server
  - lwIP UDP Perf Client
  - lwIP UDP Perf Server
  - mba_fs_boot
  - Memory Tests
  - Peripheral Tests
  - SREC Bootloader
  - SREC SPI Bootloader

Empty Application(C)

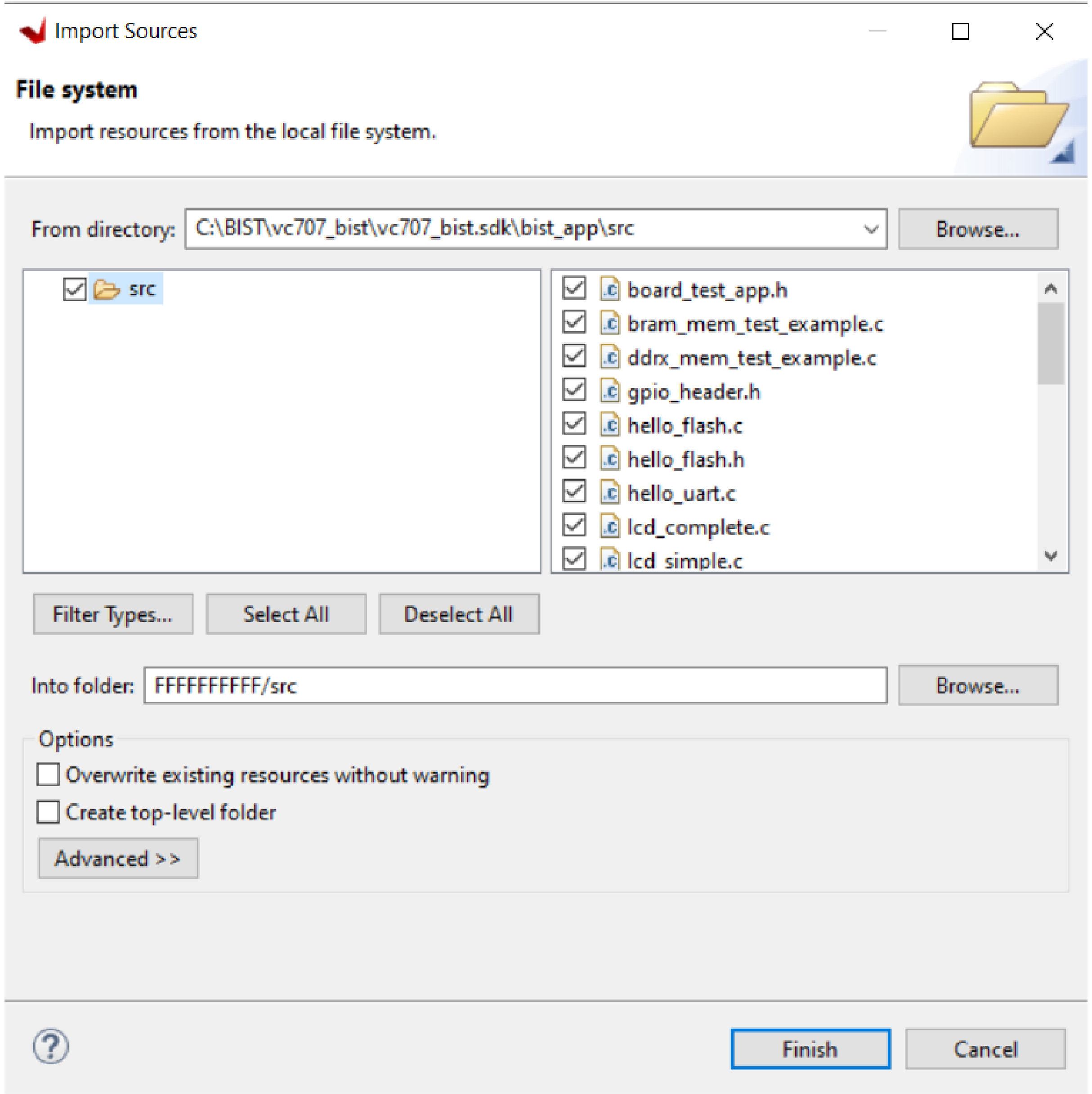
A blank C project.

VC707 BIST Flow



AMD
together we advance_

VC707 BIST Flow



Import Sources

File system

Import resources from the local file system.

From directory: C:\BIST\vc707_bist\vc707_bist.sdk\bist_app\src

☑ src

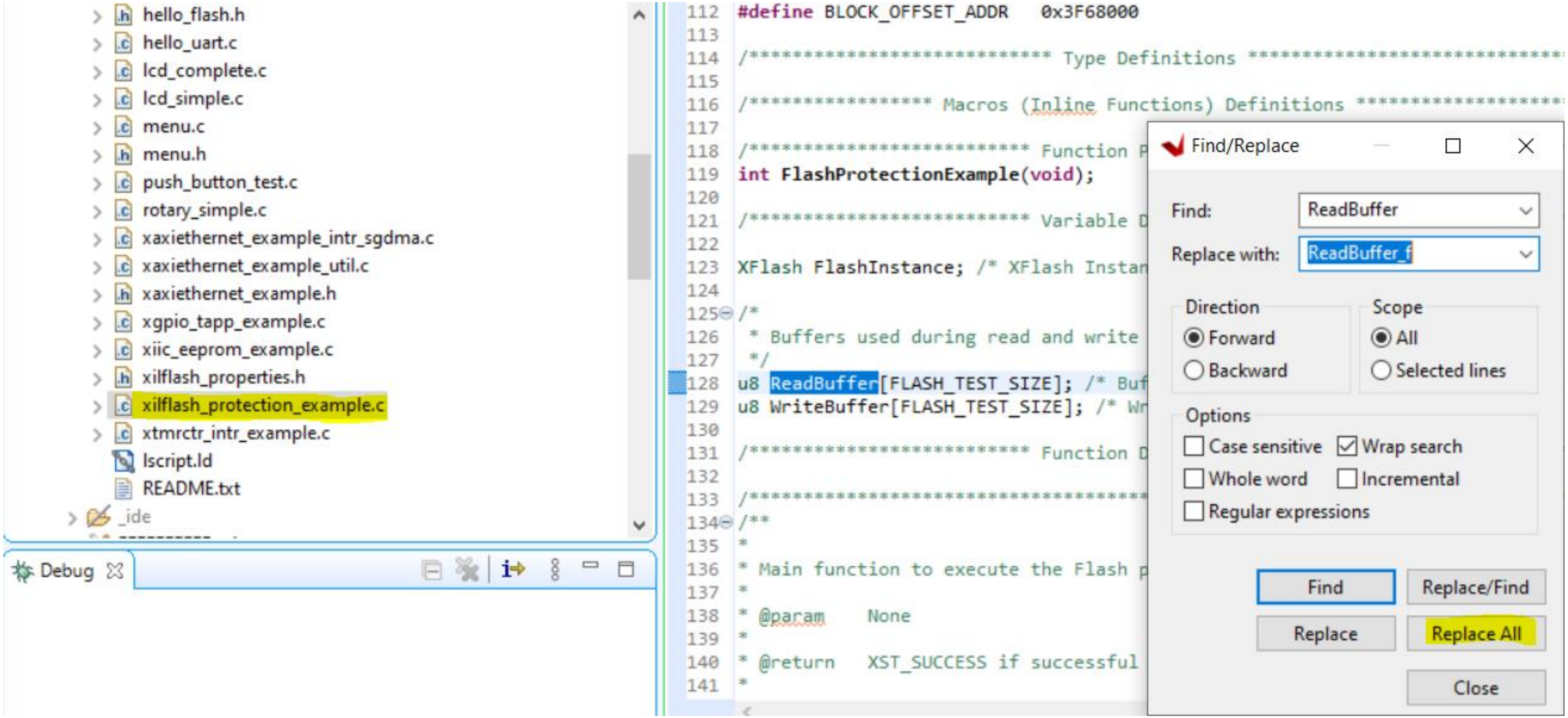
☑ board_test_app.h
☑ bram_mem_test_example.c
☑ ddrx_mem_test_example.c
☑ gpio_header.h
☑ hello_flash.c
☑ hello_flash.h
☑ hello_uart.c
☑ lcd_complete.c
☑ lcd_simple.c

Into folder: FFFFFFFFFF/src

Options

☐ Overwrite existing resources without warning
☐ Create top-level folder

VC707 BIST Flow



- hello_flash.h
- hello_uart.c
- lcd_complete.c
- lcd_simple.c
- menu.c
- menu.h
- push_button_test.c
- rotary_simple.c
- xaxiethernet_example_intr_sgdma.c
- xaxiethernet_example_util.c
- xaxiethernet_example.h
- xgpio_tapp_example.c
- xiic_eeprom_example.c
- xilflash_properties.h
- xilflash_protection_example.c
- xtmrctr_intr_example.c
- lscript.ld
- README.txt
- _ide

```
#define BLOCK_OFFSET_ADDR   0x3F68000

/**************************** Type Definitions ******************************

/**************** Macros (Inline Functions) Definitions ******************

/************************** Function P
int FlashProtectionExample(void);

/************************** Variable D

XFlash FlashInstance; /* XFlash Instan

/*
 * Buffers used during read and write
 */
u8 ReadBuffer[FLASH_TEST_SIZE]; /* Buf
u8 WriteBuffer[FLASH_TEST_SIZE]; /* Wr

/************************** Function D

/************************************
/**
 *
 * Main function to execute the Flash p
 *
 * @param    None
 *
 * @return   XST_SUCCESS if successful
 *
```

Find/Replace

Find: ReadBuffer

Replace with: ReadBuffer_f

Direction

● Forward
○ Backward

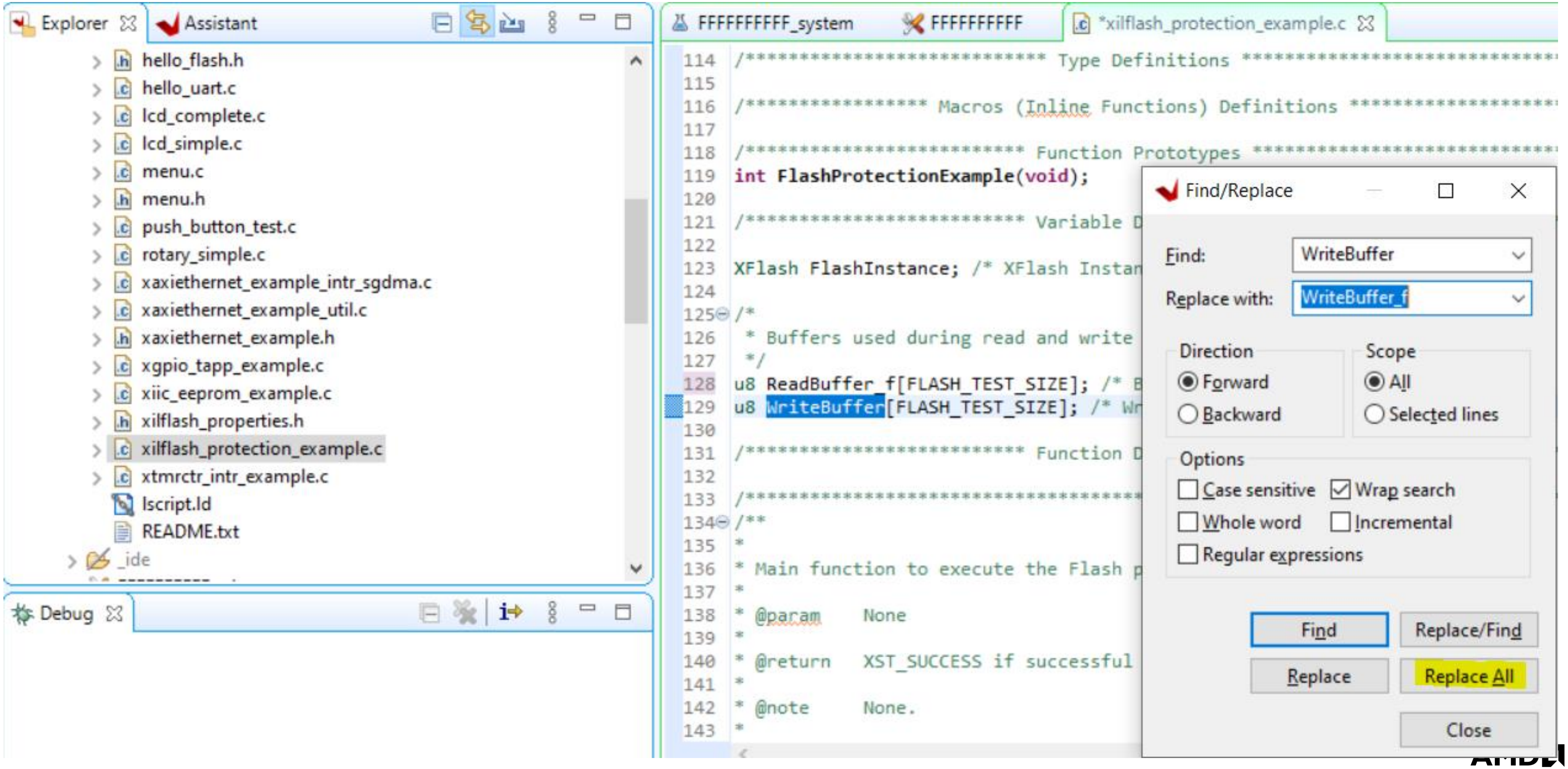
Scope

● All
○ Selected lines

Options

☐ Case sensitive
☑ Wrap search
☐ Whole word
☐ Incremental
☐ Regular expressions

VC707 BIST Flow



- hello_flash.h
- hello_uart.c
- lcd_complete.c
- lcd_simple.c
- menu.c
- menu.h
- push_button_test.c
- rotary_simple.c
- xaxiethernet_example_intr_sgdma.c
- xaxiethernet_example_util.c
- xaxiethernet_example.h
- xgpio_tapp_example.c
- xiic_eeprom_example.c
- xilflash_properties.h
- xilflash_protection_example.c
- xtmrctr_intr_example.c
- lscript.ld
- README.txt
- _ide

```
/**************************** Type Definitions ******************************

/***************** Macros (Inline Functions) Definitions *******************

/************************** Function Prototypes *****************************
int FlashProtectionExample(void);

/************************** Variable D

XFlash FlashInstance; /* XFlash Instan

/*
 * Buffers used during read and write
 */
u8 ReadBuffer_f[FLASH_TEST_SIZE]; /* B
u8 WriteBuffer[FLASH_TEST_SIZE]; /* Wr

/************************** Function D

/************************************

/**
*
* Main function to execute the Flash p
*
* @param    None
*
* @return   XST_SUCCESS if successful
*
* @note     None.
*
```

Find: WriteBuffer

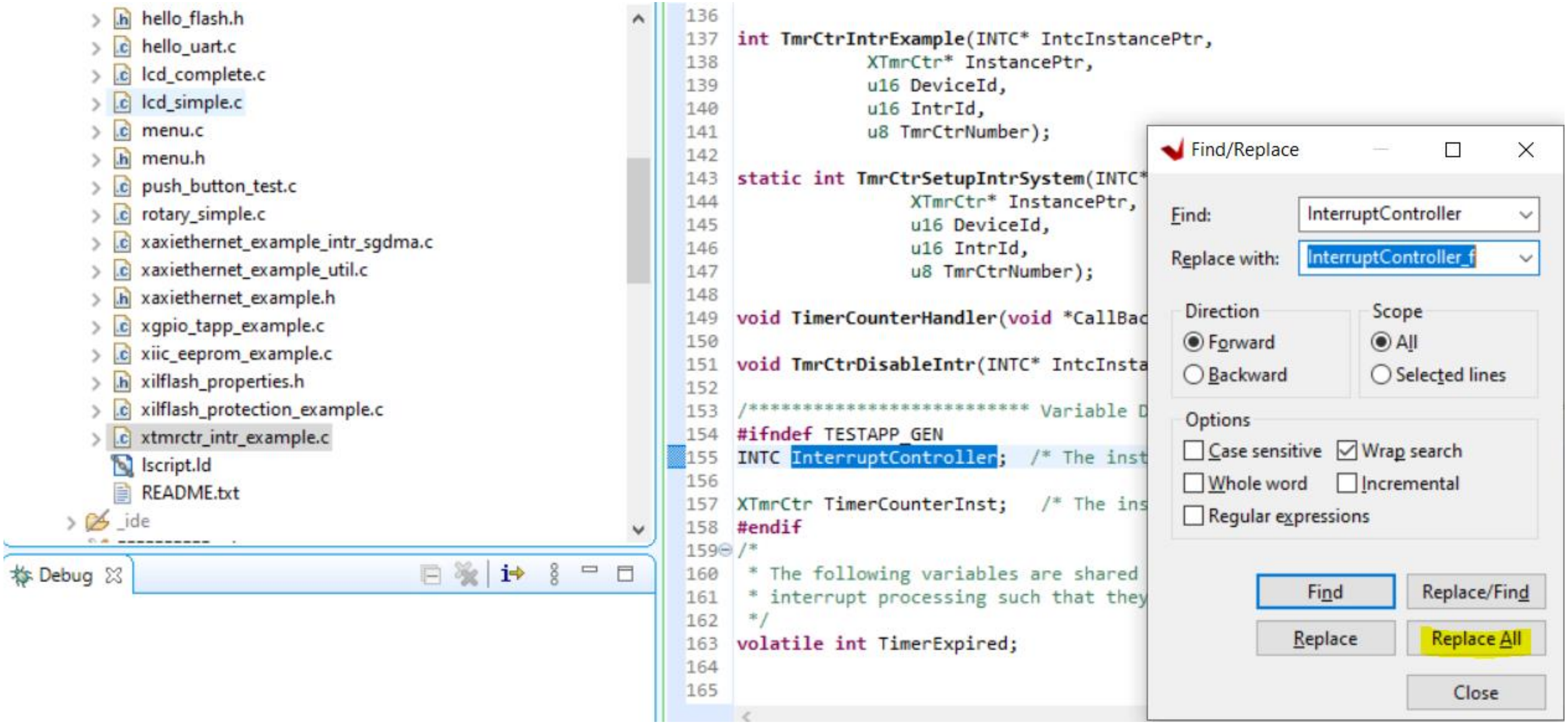
Replace with: WriteBuffer_f

Direction: ● Forward ○ Backward

Scope: ● All ○ Selected lines

Options:
- ☐ Case sensitive
- ☑ Wrap search
- ☐ Whole word
- ☐ Incremental
- ☐ Regular expressions

VC707 BIST Flow



- hello_flash.h
- hello_uart.c
- lcd_complete.c
- lcd_simple.c
- menu.c
- menu.h
- push_button_test.c
- rotary_simple.c
- xaxiethernet_example_intr_sgdma.c
- xaxiethernet_example_util.c
- xaxiethernet_example.h
- xgpio_tapp_example.c
- xiic_eeprom_example.c
- xilflash_properties.h
- xilflash_protection_example.c
- xtmrctr_intr_example.c
- lscript.ld
- README.txt
- _ide

```
int TmrCtrIntrExample(INTC* IntcInstancePtr,
            XTmrCtr* InstancePtr,
            u16 DeviceId,
            u16 IntrId,
            u8 TmrCtrNumber);

static int TmrCtrSetupIntrSystem(INTC*
              XTmrCtr* InstancePtr,
              u16 DeviceId,
              u16 IntrId,
              u8 TmrCtrNumber);

void TimerCounterHandler(void *CallBac

void TmrCtrDisableIntr(INTC* IntcInsta

/************************** Variable D
#ifndef TESTAPP_GEN
INTC InterruptController;   /* The inst

XTmrCtr TimerCounterInst;   /* The ins
#endif
/*
 * The following variables are shared
 * interrupt processing such that they
 */
volatile int TimerExpired;
```

Find/Replace

Find: InterruptController

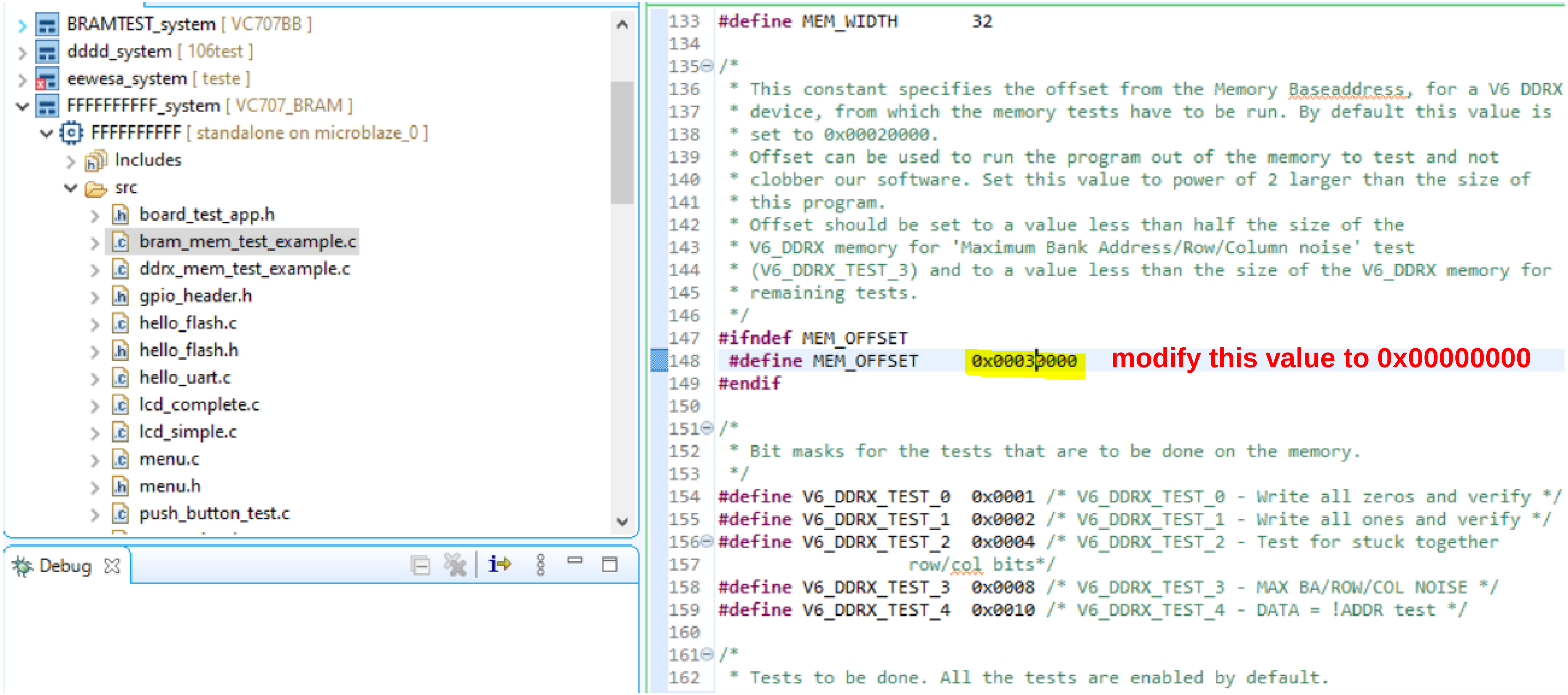
Replace with: InterruptController_f

Direction: ● Forward ○ Backward

Scope: ● All ○ Selected lines

Options: □ Case sensitive ☑ Wrap search □ Whole word □ Incremental □ Regular expressions

VC707 BIST Flow



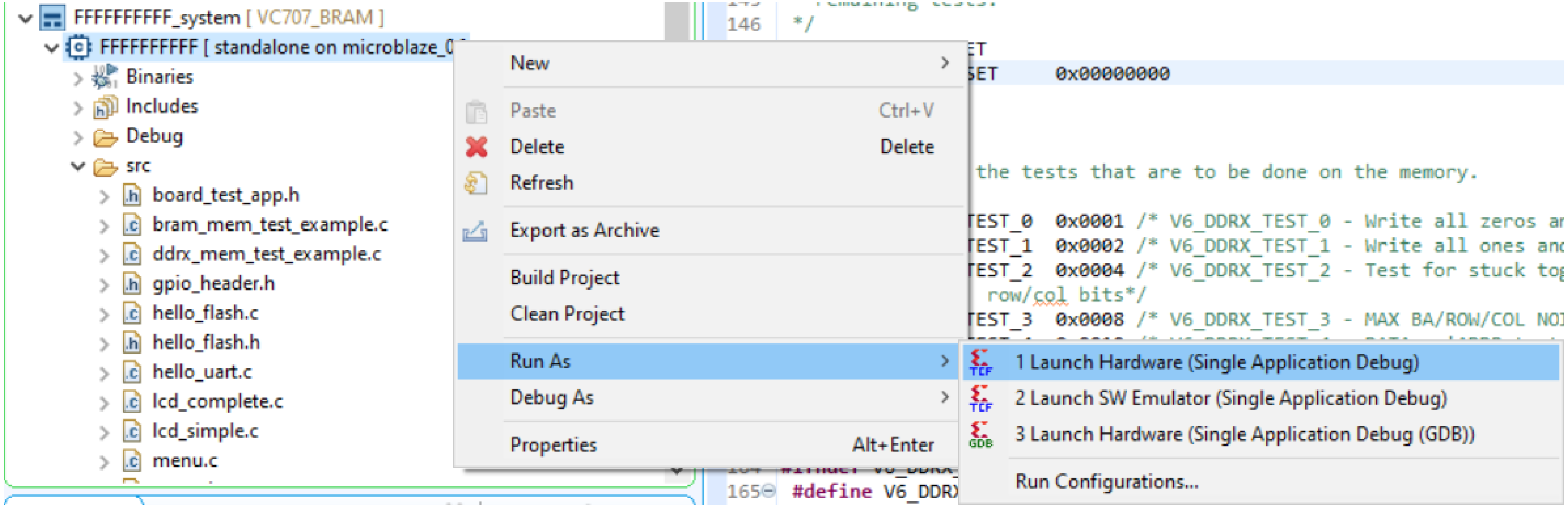
```c
#define MEM_WIDTH       32

/*
 * This constant specifies the offset from the Memory Baseaddress, for a V6 DDRX
 * device, from which the memory tests have to be run. By default this value is
 * set to 0x00020000.
 * Offset can be used to run the program out of the memory to test and not
 * clobber our software. Set this value to power of 2 larger than the size of
 * this program.
 * Offset should be set to a value less than half the size of the
 * V6_DDRX memory for 'Maximum Bank Address/Row/Column noise' test
 * (V6_DDRX_TEST_3) and to a value less than the size of the V6_DDRX memory for
 * remaining tests.
 */
#ifndef MEM_OFFSET
 #define MEM_OFFSET        0x00030000
#endif

/*
 * Bit masks for the tests that are to be done on the memory.
 */
#define V6_DDRX_TEST_0  0x0001 /* V6_DDRX_TEST_0 - Write all zeros and verify */
#define V6_DDRX_TEST_1  0x0002 /* V6_DDRX_TEST_1 - Write all ones and verify */
#define V6_DDRX_TEST_2  0x0004 /* V6_DDRX_TEST_2 - Test for stuck together
                                   row/col bits*/
#define V6_DDRX_TEST_3  0x0008 /* V6_DDRX_TEST_3 - MAX BA/ROW/COL NOISE */
#define V6_DDRX_TEST_4  0x0010 /* V6_DDRX_TEST_4 - DATA = !ADDR test */

/*
 * Tests to be done. All the tests are enabled by default.
```

modify this value to 0x00000000

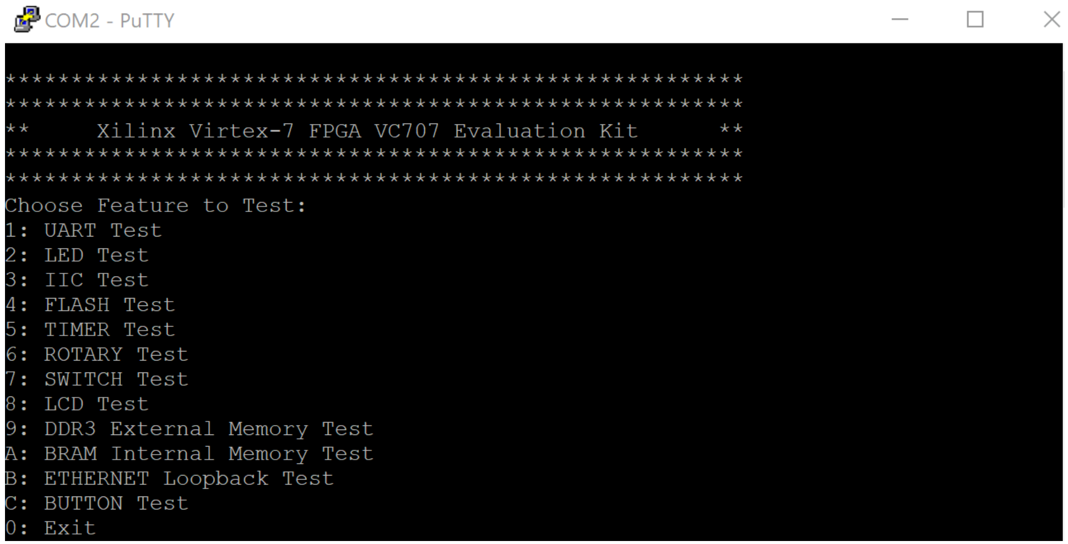
Then build the whole application

AMD
together we advance_

VC707 BIST Flow



```
 */
ET
SET      0x00000000

 the tests that are to be done on the memory.

TEST_0  0x0001 /* V6_DDRX_TEST_0 - Write all zeros ar
TEST_1  0x0002 /* V6_DDRX_TEST_1 - Write all ones and
TEST_2  0x0004 /* V6_DDRX_TEST_2 - Test for stuck tog
  row/col bits*/
TEST_3  0x0008 /* V6_DDRX_TEST_3 - MAX BA/ROW/COL NOI
#define V6_DDRX
```

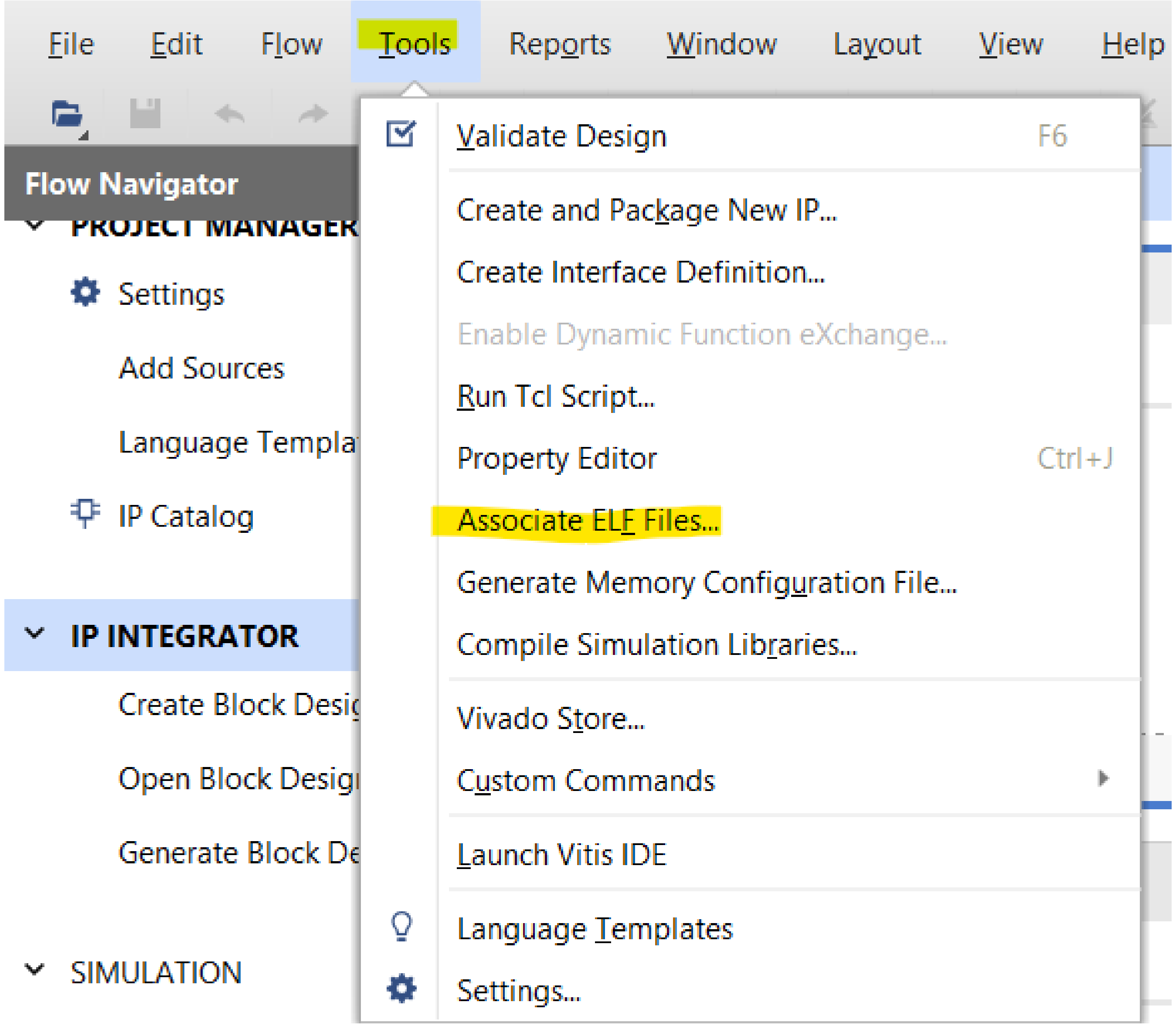


```
*********************************************************
*********************************************************
**     Xilinx Virtex-7 FPGA VC707 Evaluation Kit      **
*********************************************************
*********************************************************
Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: FLASH Test
5: TIMER Test
6: ROTARY Test
7: SWITCH Test
8: LCD Test
9: DDR3 External Memory Test
A: BRAM Internal Memory Test
B: ETHERNET Loopback Test
C: BUTTON Test
0: Exit
```

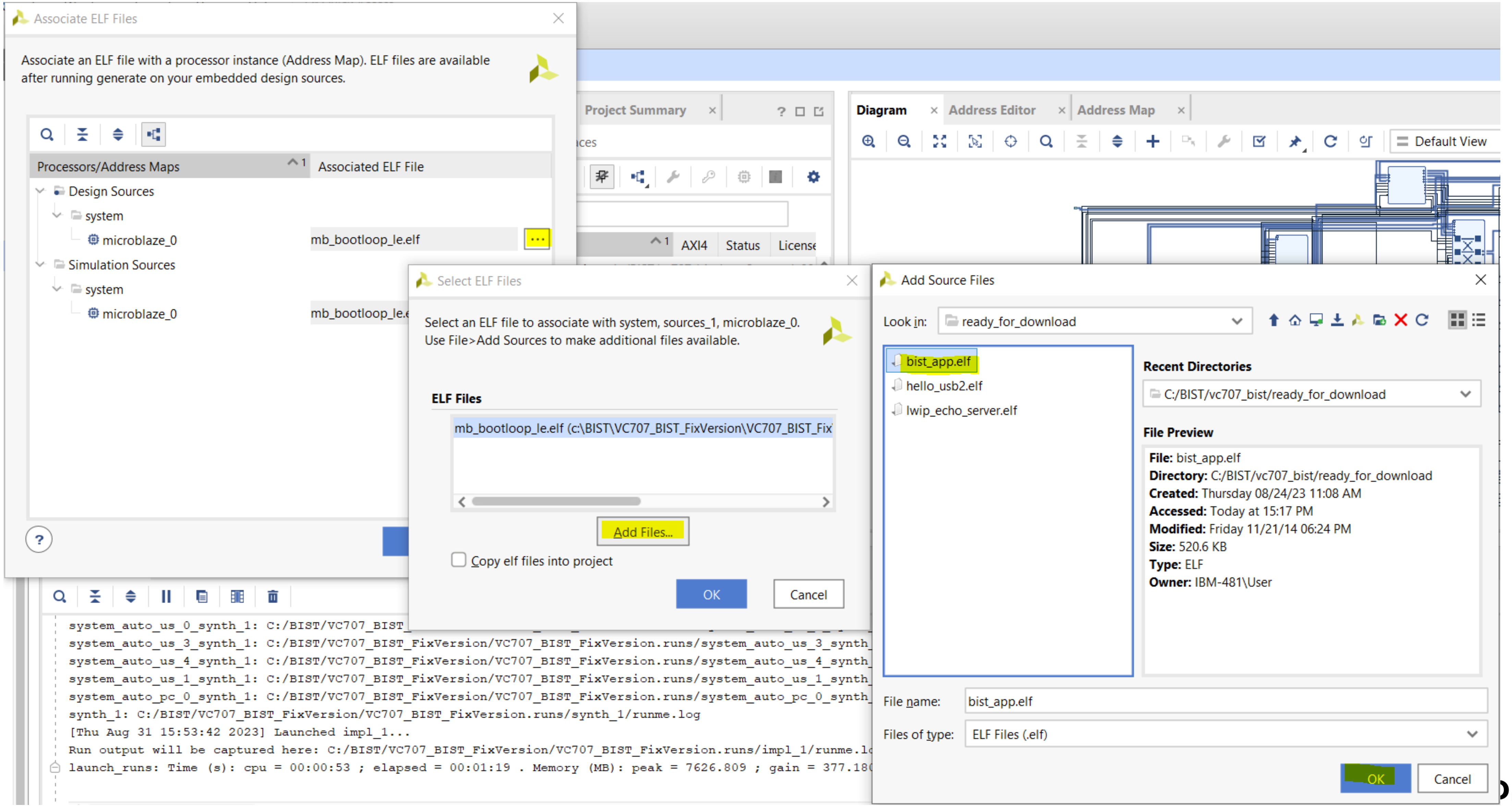
It works!

AMD

APPENDIX A: Combine bitstream with .elf file(Only for MicroBlaze)



APPENDIX A: Combine bitstream with .elf file(Only for MicroBlaze)



Associate ELF Files

Associate an ELF file with a processor instance (Address Map). ELF files are available after running generate on your embedded design sources.

Processors/Address Maps	Associated ELF File
Design Sources	
system	
microblaze_0	mb_bootloop_le.elf
Simulation Sources	
system	
microblaze_0	mb_bootloop_le.e

Select ELF Files

Select an ELF file to associate with system, sources_1, microblaze_0.
Use File>Add Sources to make additional files available.

ELF Files

mb_bootloop_le.elf (c:\BIST\VC707_BIST_FixVersion\VC707_BIST_Fix

□ Copy elf files into project

Add Source Files

Look in: ready_for_download

bist_app.elf
hello_usb2.elf
lwip_echo_server.elf

Recent Directories

C:/BIST/vc707_bist/ready_for_download

File Preview

File: bist_app.elf
Directory: C:/BIST/vc707_bist/ready_for_download
Created: Thursday 08/24/23 11:08 AM
Accessed: Today at 15:17 PM
Modified: Friday 11/21/14 06:24 PM
Size: 520.6 KB
Type: ELF
Owner: IBM-481\User

File name: bist_app.elf

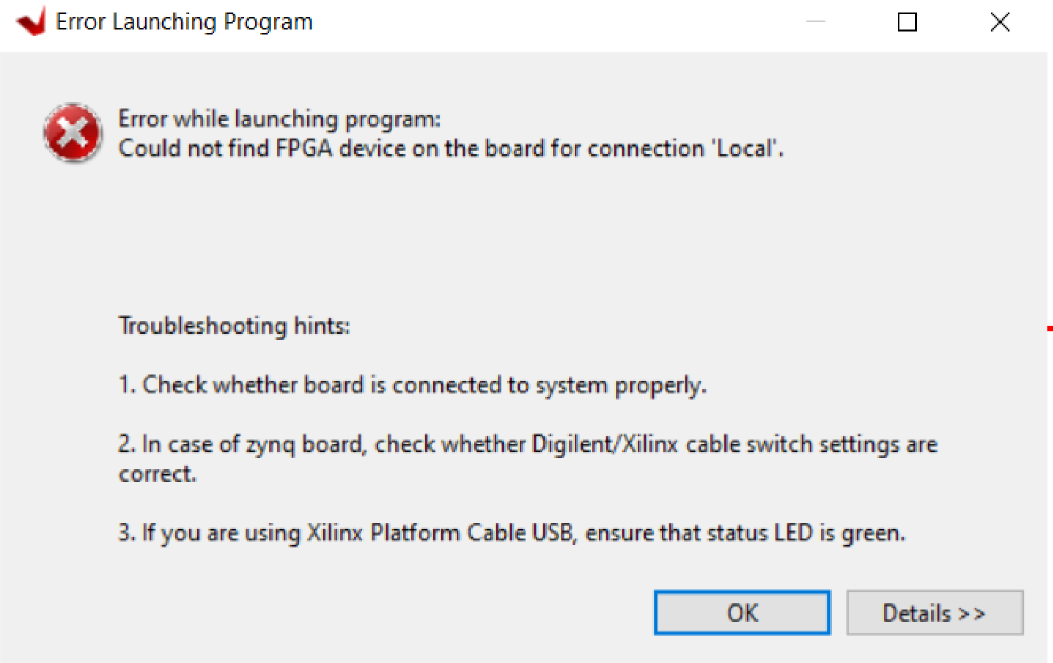
Files of type: ELF Files (.elf)

```
system_auto_us_0_synth_1: C:/BIST/VC707_BIST_
system_auto_us_3_synth_1: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/system_auto_us_3_synth_
system_auto_us_4_synth_1: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/system_auto_us_4_synth_
system_auto_us_1_synth_1: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/system_auto_us_1_synth_
system_auto_pc_0_synth_1: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/system_auto_pc_0_synth_
synth_1: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/synth_1/runme.log
[Thu Aug 31 15:53:42 2023] Launched impl_1...
Run output will be captured here: C:/BIST/VC707_BIST_FixVersion/VC707_BIST_FixVersion.runs/impl_1/runme.lo
launch_runs: Time (s): cpu = 00:00:53 ; elapsed = 00:01:19 . Memory (MB): peak = 7626.809 ; gain = 377.18
```

APPENDIX A: Combine bitstream with .elf file(Only for MicroBlaze)

Then Regenerate the bitstream, you will get a .bit file which including .elf.

APPENDIX B: Program Error



Error Launching Program

Error while launching program:
Could not find FPGA device on the board for connection 'Local'.

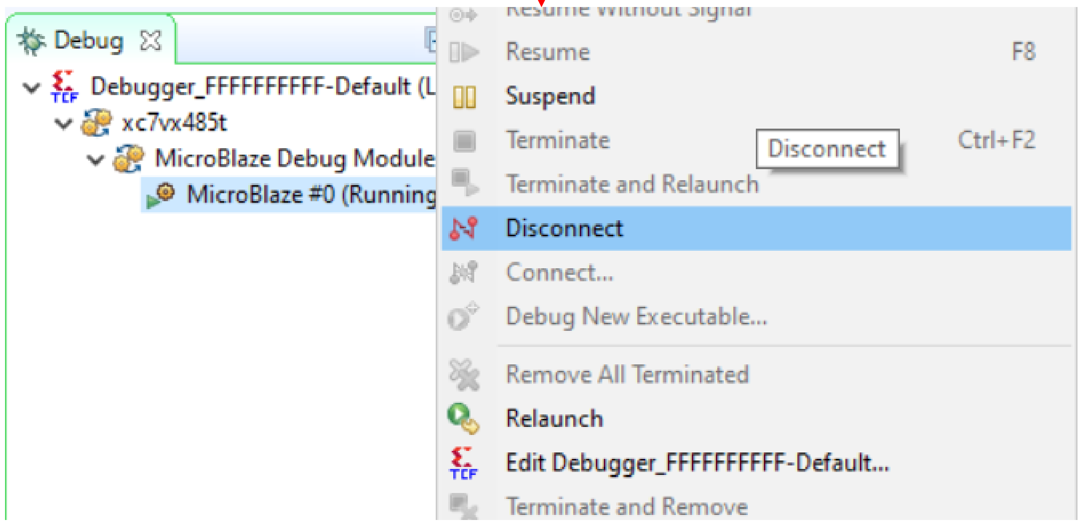
Troubleshooting hints:

1. Check whether board is connected to system properly.

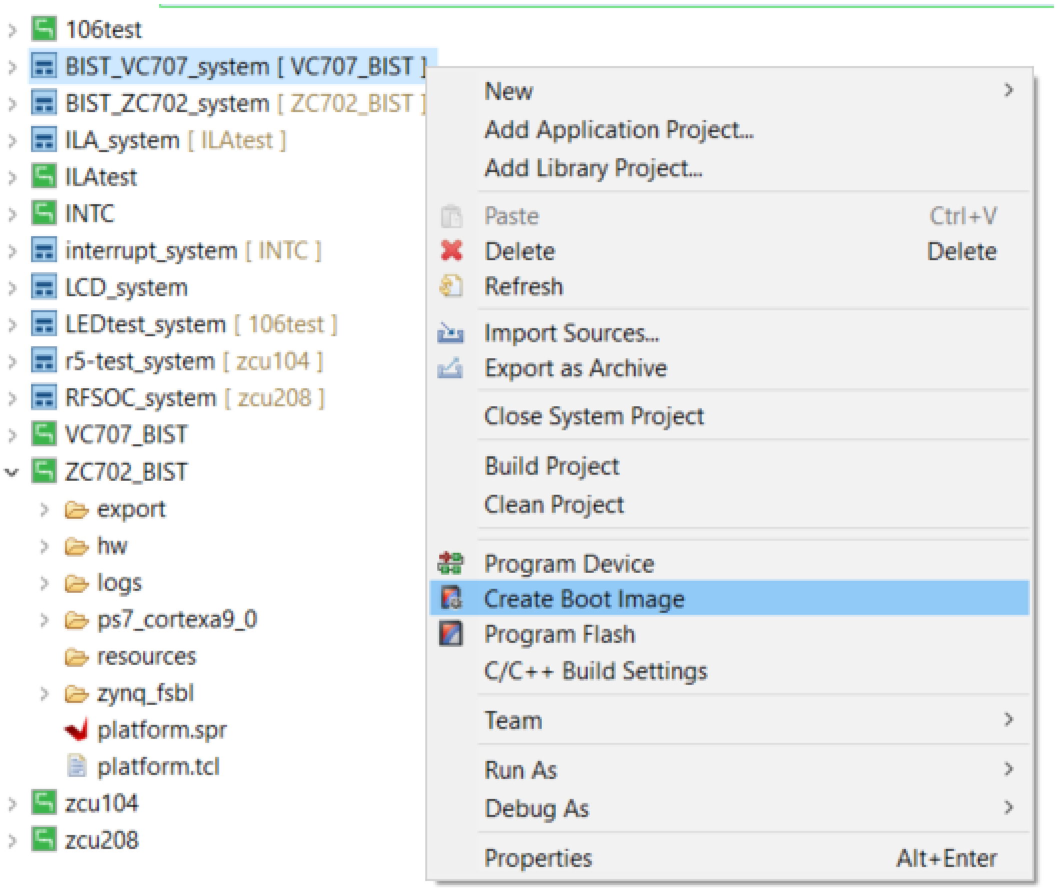
2. In case of zynq board, check whether Digilent/Xilinx cable switch settings are correct.

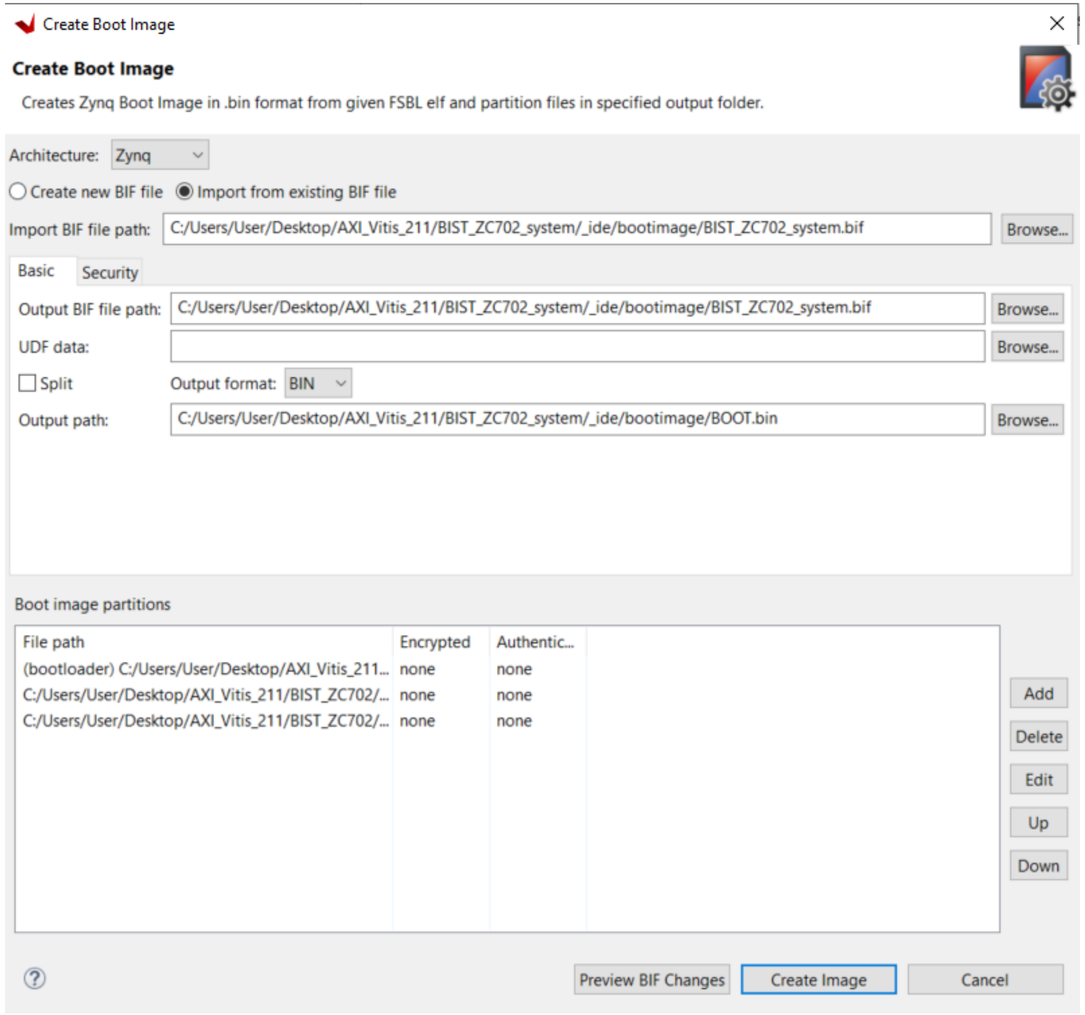
3. If you are using Xilinx Platform Cable USB, ensure that status LED is green.

Solution



APPENDIX C: Program ZYNQ Series(ZC702)





Create Boot Image

Creates Zynq Boot Image in .bin format from given FSBL elf and partition files in specified output folder.

Architecture: Zynq

○ Create new BIF file ● Import from existing BIF file

Import BIF file path: C:/Users/User/Desktop/AXI_Vitis_211/BIST_ZC702_system/_ide/bootimage/BIST_ZC702_system.bif

Output BIF file path: C:/Users/User/Desktop/AXI_Vitis_211/BIST_ZC702_system/_ide/bootimage/BIST_ZC702_system.bif

UDF data:

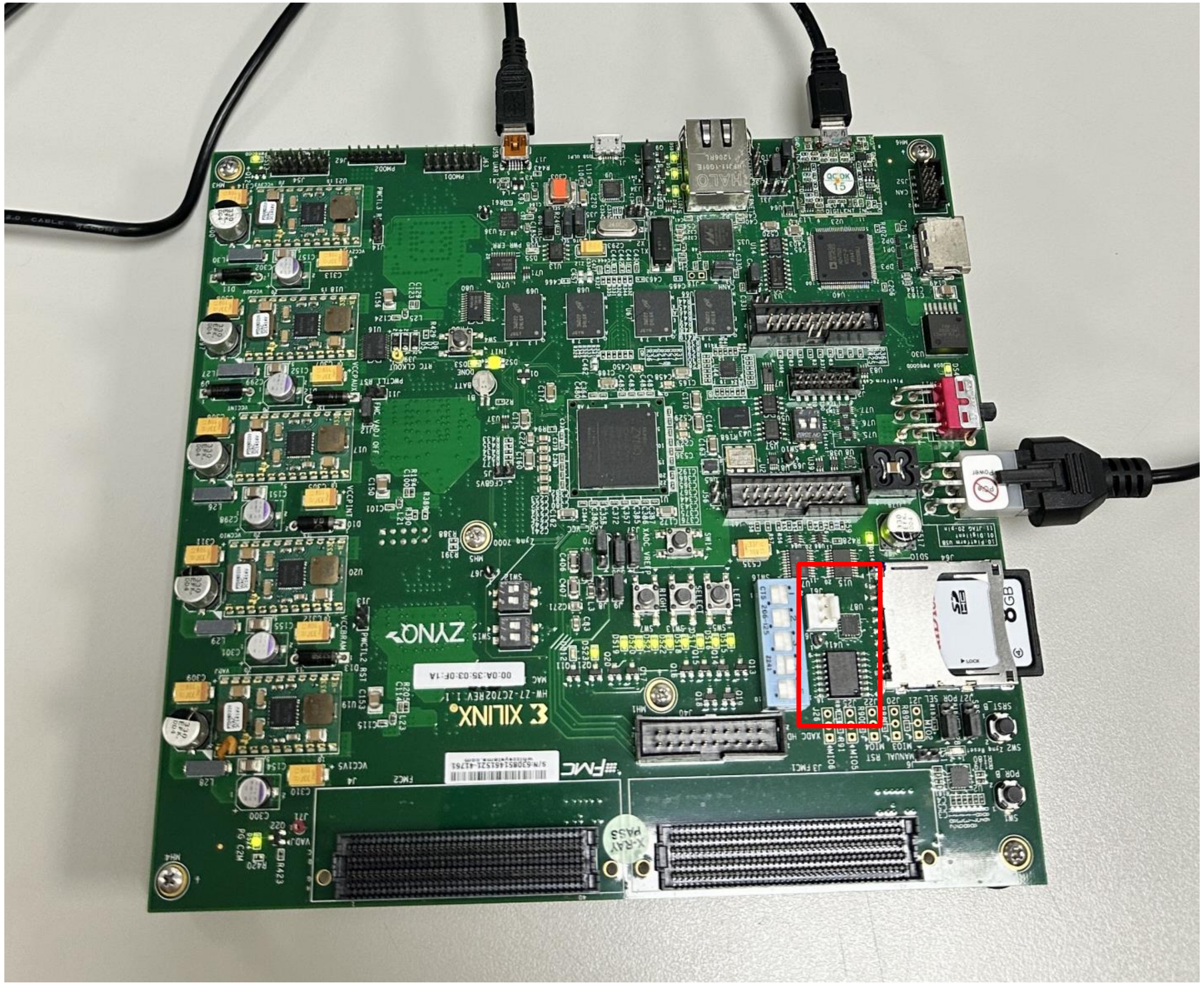
□ Split Output format: BIN

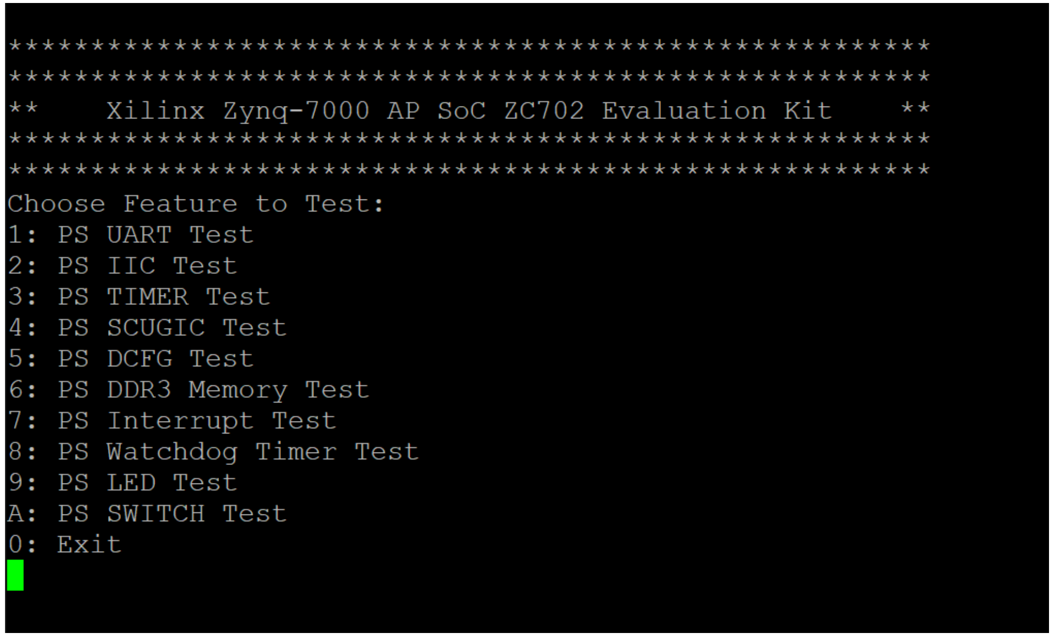
Output path: C:/Users/User/Desktop/AXI_Vitis_211/BIST_ZC702_system/_ide/bootimage/BOOT.bin

Boot image partitions

File path	Encrypted	Authentic...
(bootloader) C:/Users/User/Desktop/AXI_Vitis_211...	none	none
C:/Users/User/Desktop/AXI_Vitis_211/BIST_ZC702/...	none	none
C:/Users/User/Desktop/AXI_Vitis_211/BIST_ZC702/...	none	none

APPENDIX C: Program ZYNQ Series(ZC702)





```
*********************************************************
*********************************************************
**    Xilinx Zynq-7000 AP SoC ZC702 Evaluation Kit     **
*********************************************************
*********************************************************
Choose Feature to Test:
1: PS UART Test
2: PS IIC Test
3: PS TIMER Test
4: PS SCUGIC Test
5: PS DCFG Test
6: PS DDR3 Memory Test
7: PS Interrupt Test
8: PS Watchdog Timer Test
9: PS LED Test
A: PS SWITCH Test
0: Exit
```