The University of Aizu

DN-SoC: FPGA Implementation of Doanh Neuromorphic System-on-Chip

Ngo-Doanh NGUYEN

2023.09.19

https://www.u-aizu.ac.jp/misc/neuro-eng/

Overview

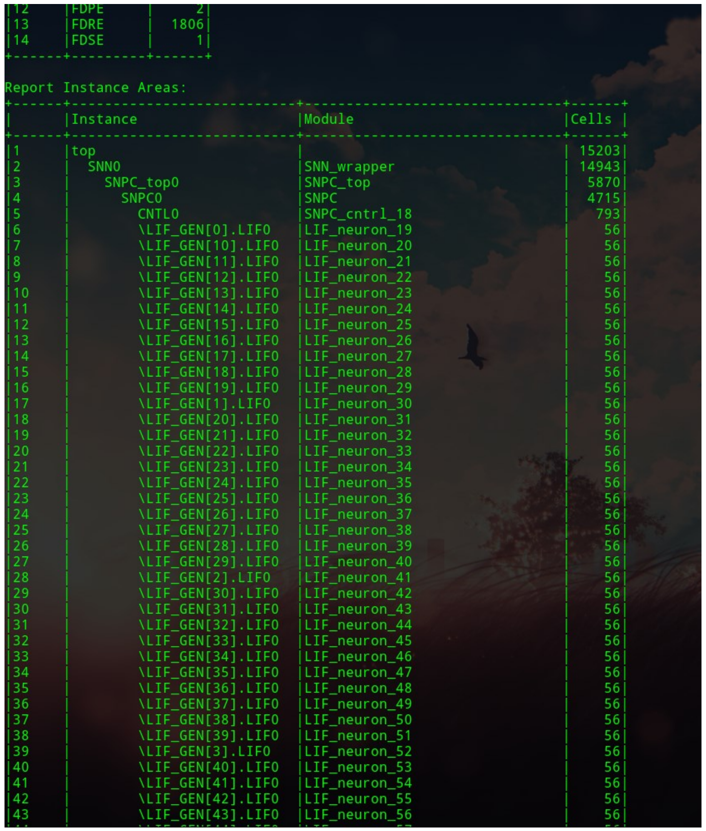
1. Package SNN IP
2. SoC Integration with MicroBlaze
3. Software Implementation
4. Flash memory configuration

Overview

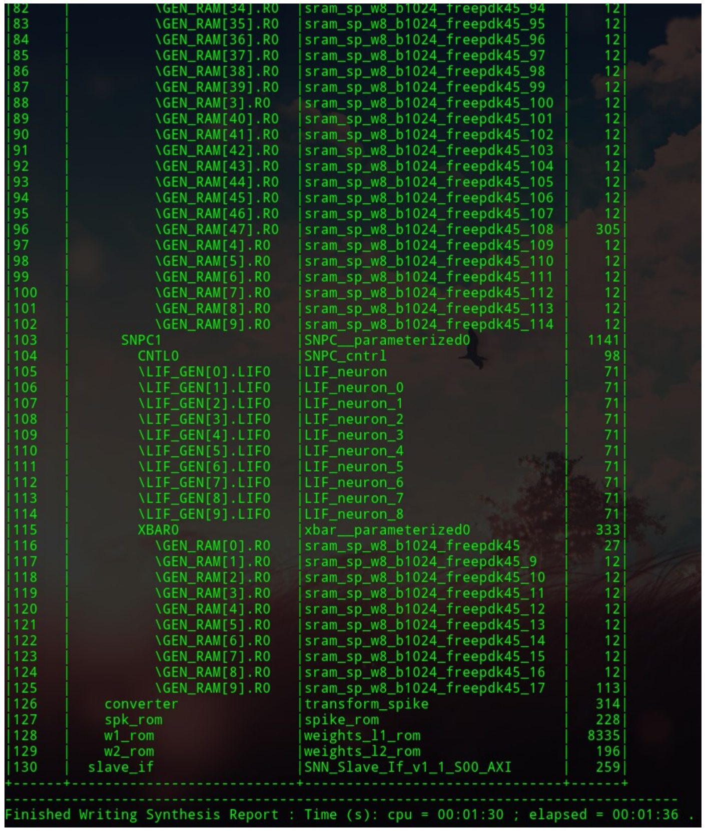
1. Package SNN IP
2. SoC Integration with MicroBlaze
3. Software Implementation
4. Flash memory configuration

Succesful Synthesizing SNN IP

- Success = This Instance Area Report
- Success = No Error in .log file



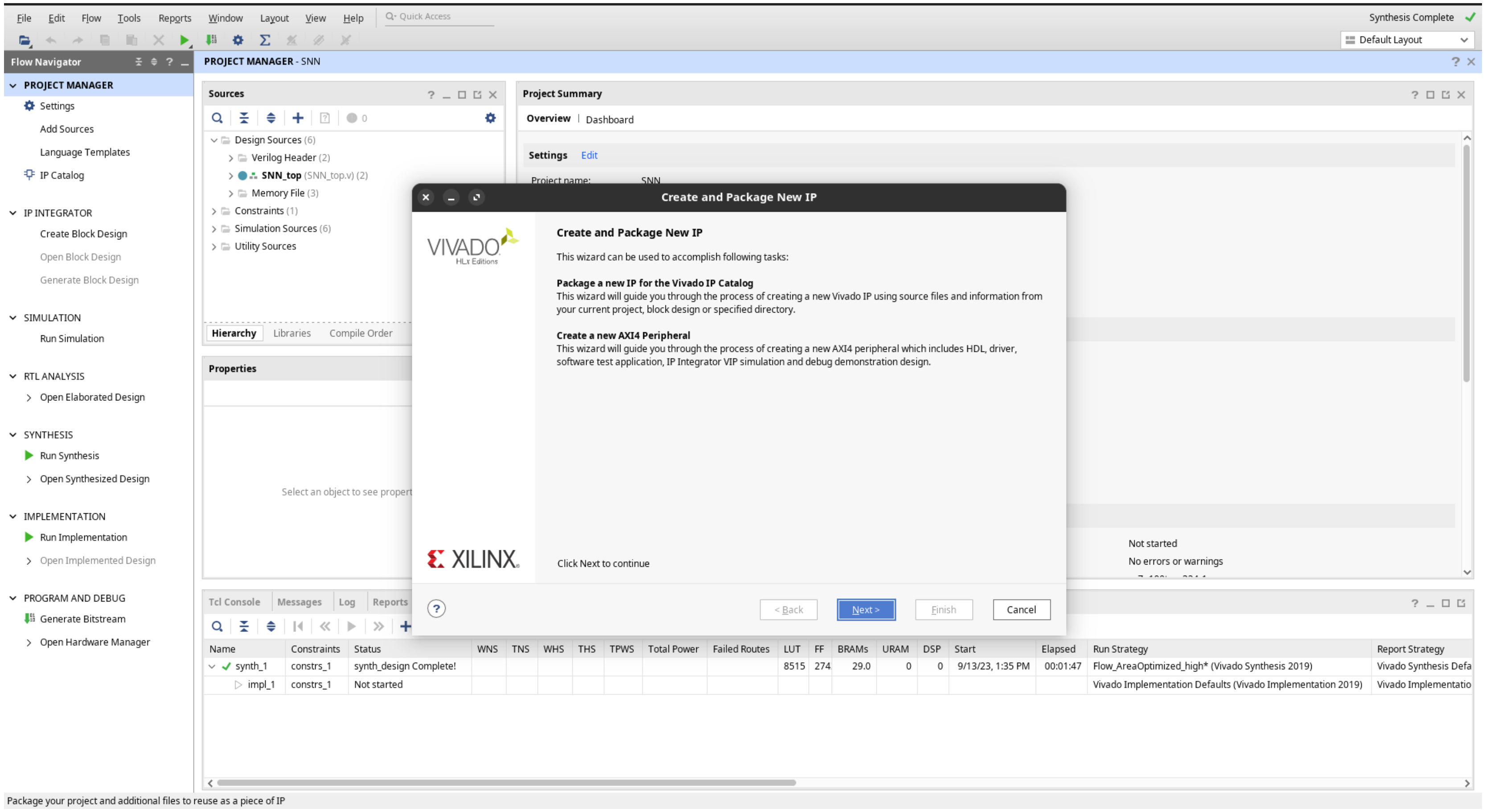
```
|12    |FDPE     |     2|
|13    |FDRE     |  1806|
|14    |FDSE     |     1|
+------+---------+------+

Report Instance Areas:
+------+---------------------------+--------------------------------+------+
|      |Instance                   |Module                          |Cells |
+------+---------------------------+--------------------------------+------+
|1     |top                        |                                | 15203|
|2     |  SNN0                     |SNN_wrapper                     | 14943|
|3     |    SNPC_top0              |SNPC_top                        |  5870|
|4     |      SNPC0                |SNPC                            |  4715|
|5     |        CNTL0              |SNPC_cntrl_18                   |   793|
|6     |        \LIF_GEN[0].LIF0   |LIF_neuron_19                   |    56|
|7     |        \LIF_GEN[10].LIF0  |LIF_neuron_20                   |    56|
|8     |        \LIF_GEN[11].LIF0  |LIF_neuron_21                   |    56|
|9     |        \LIF_GEN[12].LIF0  |LIF_neuron_22                   |    56|
|10    |        \LIF_GEN[13].LIF0  |LIF_neuron_23                   |    56|
|11    |        \LIF_GEN[14].LIF0  |LIF_neuron_24                   |    56|
|12    |        \LIF_GEN[15].LIF0  |LIF_neuron_25                   |    56|
|13    |        \LIF_GEN[16].LIF0  |LIF_neuron_26                   |    56|
|14    |        \LIF_GEN[17].LIF0  |LIF_neuron_27                   |    56|
|15    |        \LIF_GEN[18].LIF0  |LIF_neuron_28                   |    56|
|16    |        \LIF_GEN[19].LIF0  |LIF_neuron_29                   |    56|
|17    |        \LIF_GEN[1].LIF0   |LIF_neuron_30                   |    56|
|18    |        \LIF_GEN[20].LIF0  |LIF_neuron_31                   |    56|
|19    |        \LIF_GEN[21].LIF0  |LIF_neuron_32                   |    56|
|20    |        \LIF_GEN[22].LIF0  |LIF_neuron_33                   |    56|
|21    |        \LIF_GEN[23].LIF0  |LIF_neuron_34                   |    56|
|22    |        \LIF_GEN[24].LIF0  |LIF_neuron_35                   |    56|
|23    |        \LIF_GEN[25].LIF0  |LIF_neuron_36                   |    56|
|24    |        \LIF_GEN[26].LIF0  |LIF_neuron_37                   |    56|
|25    |        \LIF_GEN[27].LIF0  |LIF_neuron_38                   |    56|
|26    |        \LIF_GEN[28].LIF0  |LIF_neuron_39                   |    56|
|27    |        \LIF_GEN[29].LIF0  |LIF_neuron_40                   |    56|
|28    |        \LIF_GEN[2].LIF0   |LIF_neuron_41                   |    56|
|29    |        \LIF_GEN[30].LIF0  |LIF_neuron_42                   |    56|
|30    |        \LIF_GEN[31].LIF0  |LIF_neuron_43                   |    56|
|31    |        \LIF_GEN[32].LIF0  |LIF_neuron_44                   |    56|
|32    |        \LIF_GEN[33].LIF0  |LIF_neuron_45                   |    56|
|33    |        \LIF_GEN[34].LIF0  |LIF_neuron_46                   |    56|
|34    |        \LIF_GEN[35].LIF0  |LIF_neuron_47                   |    56|
|35    |        \LIF_GEN[36].LIF0  |LIF_neuron_48                   |    56|
|36    |        \LIF_GEN[37].LIF0  |LIF_neuron_49                   |    56|
|37    |        \LIF_GEN[38].LIF0  |LIF_neuron_50                   |    56|
|38    |        \LIF_GEN[39].LIF0  |LIF_neuron_51                   |    56|
|39    |        \LIF_GEN[3].LIF0   |LIF_neuron_52                   |    56|
|40    |        \LIF_GEN[40].LIF0  |LIF_neuron_53                   |    56|
|41    |        \LIF_GEN[41].LIF0  |LIF_neuron_54                   |    56|
|42    |        \LIF_GEN[42].LIF0  |LIF_neuron_55                   |    56|
|43    |        \LIF_GEN[43].LIF0  |LIF_neuron_56                   |    56|
```



```
|82    |          \GEN_RAM[34].R0  |sram_sp_w8_b1024_freepdk45_94   |    12|
|83    |          \GEN_RAM[35].R0  |sram_sp_w8_b1024_freepdk45_95   |    12|
|84    |          \GEN_RAM[36].R0  |sram_sp_w8_b1024_freepdk45_96   |    12|
|85    |          \GEN_RAM[37].R0  |sram_sp_w8_b1024_freepdk45_97   |    12|
|86    |          \GEN_RAM[38].R0  |sram_sp_w8_b1024_freepdk45_98   |    12|
|87    |          \GEN_RAM[39].R0  |sram_sp_w8_b1024_freepdk45_99   |    12|
|88    |          \GEN_RAM[3].R0   |sram_sp_w8_b1024_freepdk45_100  |    12|
|89    |          \GEN_RAM[40].R0  |sram_sp_w8_b1024_freepdk45_101  |    12|
|90    |          \GEN_RAM[41].R0  |sram_sp_w8_b1024_freepdk45_102  |    12|
|91    |          \GEN_RAM[42].R0  |sram_sp_w8_b1024_freepdk45_103  |    12|
|92    |          \GEN_RAM[43].R0  |sram_sp_w8_b1024_freepdk45_104  |    12|
|93    |          \GEN_RAM[44].R0  |sram_sp_w8_b1024_freepdk45_105  |    12|
|94    |          \GEN_RAM[45].R0  |sram_sp_w8_b1024_freepdk45_106  |    12|
|95    |          \GEN_RAM[46].R0  |sram_sp_w8_b1024_freepdk45_107  |    12|
|96    |          \GEN_RAM[47].R0  |sram_sp_w8_b1024_freepdk45_108  |   305|
|97    |          \GEN_RAM[4].R0   |sram_sp_w8_b1024_freepdk45_109  |    12|
|98    |          \GEN_RAM[5].R0   |sram_sp_w8_b1024_freepdk45_110  |    12|
|99    |          \GEN_RAM[6].R0   |sram_sp_w8_b1024_freepdk45_111  |    12|
|100   |          \GEN_RAM[7].R0   |sram_sp_w8_b1024_freepdk45_112  |    12|
|101   |          \GEN_RAM[8].R0   |sram_sp_w8_b1024_freepdk45_113  |    12|
|102   |          \GEN_RAM[9].R0   |sram_sp_w8_b1024_freepdk45_114  |    12|
|103   |      SNPC1                |SNPC__parameterized0            |  1141|
|104   |        CNTL0              |SNPC_cntrl                      |    98|
|105   |        \LIF_GEN[0].LIF0   |LIF_neuron                      |    71|
|106   |        \LIF_GEN[1].LIF0   |LIF_neuron_0                    |    71|
|107   |        \LIF_GEN[2].LIF0   |LIF_neuron_1                    |    71|
|108   |        \LIF_GEN[3].LIF0   |LIF_neuron_2                    |    71|
|109   |        \LIF_GEN[4].LIF0   |LIF_neuron_3                    |    71|
|110   |        \LIF_GEN[5].LIF0   |LIF_neuron_4                    |    71|
|111   |        \LIF_GEN[6].LIF0   |LIF_neuron_5                    |    71|
|112   |        \LIF_GEN[7].LIF0   |LIF_neuron_6                    |    71|
|113   |        \LIF_GEN[8].LIF0   |LIF_neuron_7                    |    71|
|114   |        \LIF_GEN[9].LIF0   |LIF_neuron_8                    |    71|
|115   |        XBAR0              |xbar__parameterized0            |   333|
|116   |          \GEN_RAM[0].R0   |sram_sp_w8_b1024_freepdk45      |    27|
|117   |          \GEN_RAM[1].R0   |sram_sp_w8_b1024_freepdk45_9    |    12|
|118   |          \GEN_RAM[2].R0   |sram_sp_w8_b1024_freepdk45_10   |    12|
|119   |          \GEN_RAM[3].R0   |sram_sp_w8_b1024_freepdk45_11   |    12|
|120   |          \GEN_RAM[4].R0   |sram_sp_w8_b1024_freepdk45_12   |    12|
|121   |          \GEN_RAM[5].R0   |sram_sp_w8_b1024_freepdk45_13   |    12|
|122   |          \GEN_RAM[6].R0   |sram_sp_w8_b1024_freepdk45_14   |    12|
|123   |          \GEN_RAM[7].R0   |sram_sp_w8_b1024_freepdk45_15   |    12|
|124   |          \GEN_RAM[8].R0   |sram_sp_w8_b1024_freepdk45_16   |    12|
|125   |          \GEN_RAM[9].R0   |sram_sp_w8_b1024_freepdk45_17   |   113|
|126   |    converter              |transform_spike                 |   314|
|127   |    spk_rom                |spike_rom                       |   228|
|128   |    w1_rom                 |weights_l1_rom                  |  8335|
|129   |    w2_rom                 |weights_l2_rom                  |   196|
|130   |  slave_if                 |SNN_Slave_If_v1_1_S00_AXI       |   259|
+------+---------------------------+--------------------------------+------+
---------------------------------------------------------------------------------
Finished Writing Synthesis Report : Time (s): cpu = 00:01:30 ; elapsed = 00:01:36 .
```

Packaging SNN IP (1)

- Find Tools/Create and Package New IP



Create and Package New IP

Create and Package New IP

This wizard can be used to accomplish following tasks:

Package a new IP for the Vivado IP Catalog
This wizard will guide you through the process of creating a new Vivado IP using source files and information from your current project, block design or specified directory.

Create a new AXI4 Peripheral
This wizard will guide you through the process of creating a new AXI4 peripheral which includes HDL, driver, software test application, IP Integrator VIP simulation and debug demonstration design.

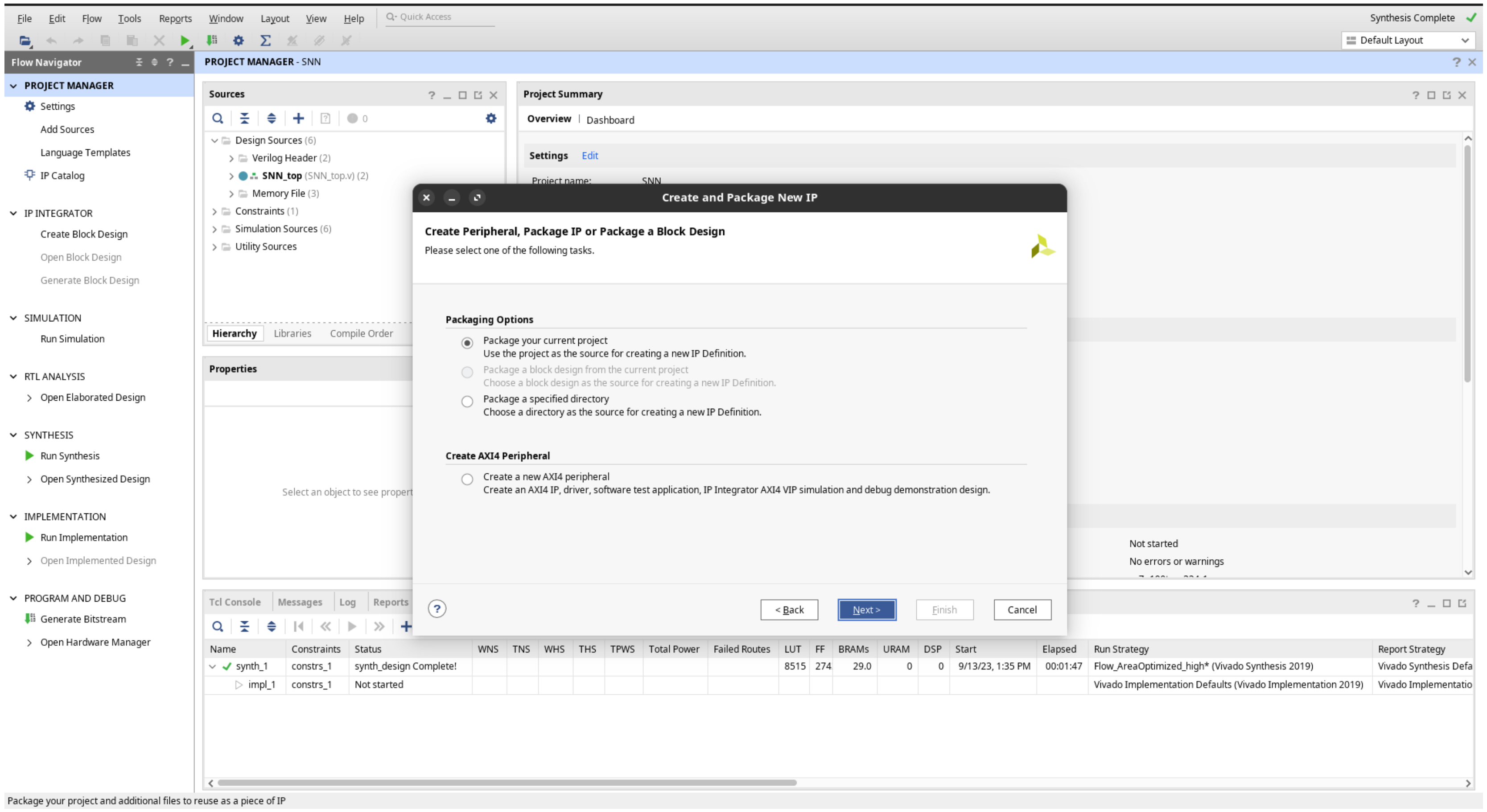
Click Next to continue

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
✓ synth_1	constrs_1	synth_design Complete!								8515	274	29.0	0	0	9/13/23, 1:35 PM	00:01:47	Flow_AreaOptimized_high* (Vivado Synthesis 2019)	Vivado Synthesis Defa
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Implementatio

Package your project and additional files to reuse as a piece of IP

Packaging SNN IP (2)

- Choose “Package your current project” & Finish



File Edit Flow Tools Reports Window Layout View Help | Quick Access | Synthesis Complete ✓ | Default Layout

Flow Navigator | PROJECT MANAGER - SNN

PROJECT MANAGER
- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR
- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION
- Run Simulation

RTL ANALYSIS
- Open Elaborated Design

SYNTHESIS
- Run Synthesis
- Open Synthesized Design

IMPLEMENTATION
- Run Implementation
- Open Implemented Design

PROGRAM AND DEBUG
- Generate Bitstream
- Open Hardware Manager

Sources
- Design Sources (6)
  - Verilog Header (2)
  - SNN_top (SNN_top.v) (2)
  - Memory File (3)
- Constraints (1)
- Simulation Sources (6)
- Utility Sources

Hierarchy | Libraries | Compile Order

Properties

Select an object to see propert

Project Summary

Overview | Dashboard

Settings Edit

Project name: SNN

Not started

No errors or warnings

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.

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.

< Back | Next > | Finish | Cancel

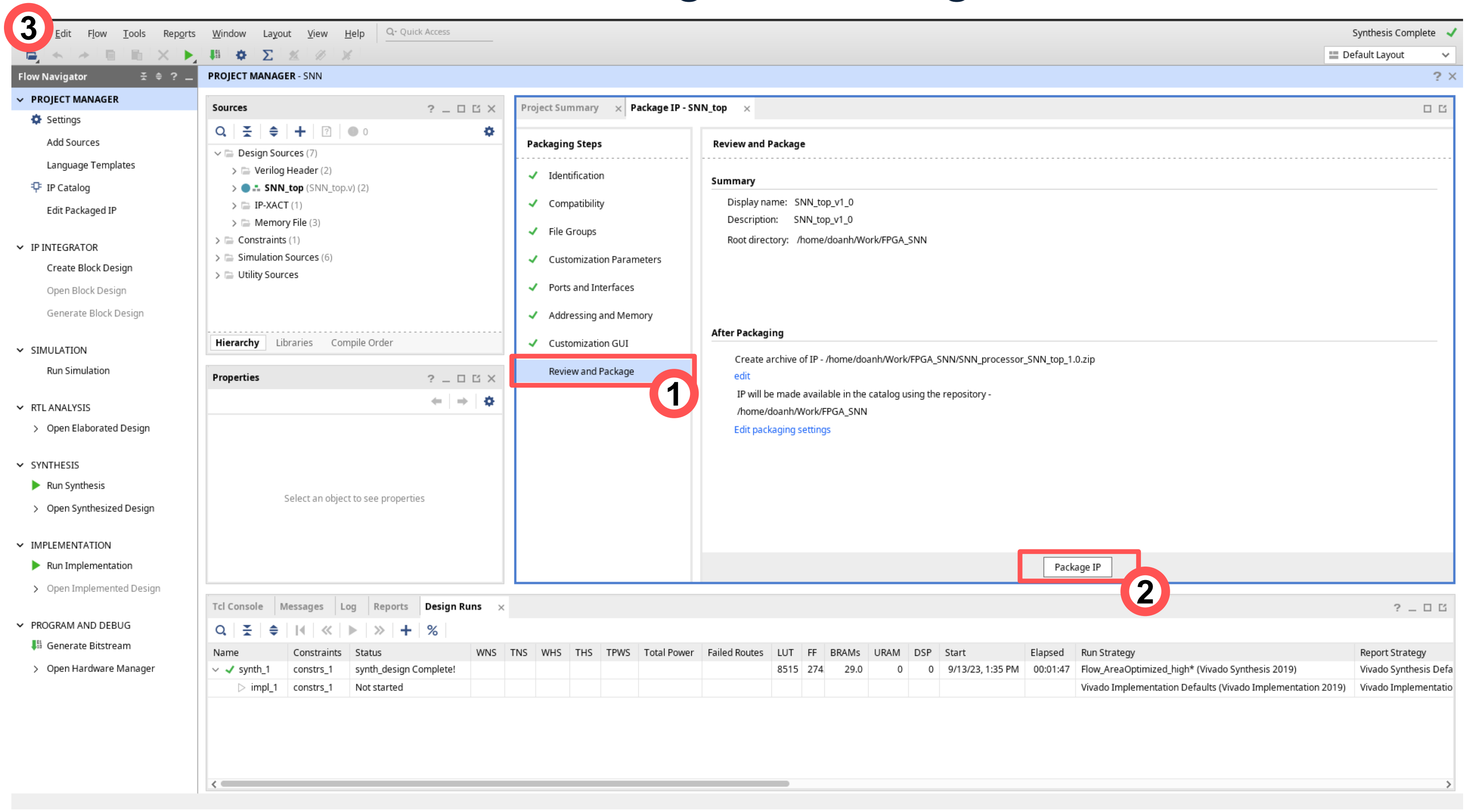
Tcl Console | Messages | Log | Reports

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
✓ synth_1	constrs_1	synth_design Complete!								8515	274	29.0	0	0	9/13/23, 1:35 PM	00:01:47	Flow_AreaOptimized_high* (Vivado Synthesis 2019)	Vivado Synthesis Defa
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Implementatio

Package your project and additional files to reuse as a piece of IP

Packaging SNN IP (3)

- Choose “Review and Package”. Package IP. Exit Vivado.



Design Sources (7)
Verilog Header (2)
SNN_top (SNN_top.v) (2)
IP-XACT (1)
Memory File (3)
Constraints (1)
Simulation Sources (6)
Utility Sources

Select an object to see properties

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:	SNN_top_v1_0
Description:	SNN_top_v1_0
Root directory:	/home/doanh/Work/FPGA_SNN

After Packaging

Create archive of IP - /home/doanh/Work/FPGA_SNN/SNN_processor_SNN_top_1.0.zip
edit
IP will be made available in the catalog using the repository - /home/doanh/Work/FPGA_SNN
Edit packaging settings

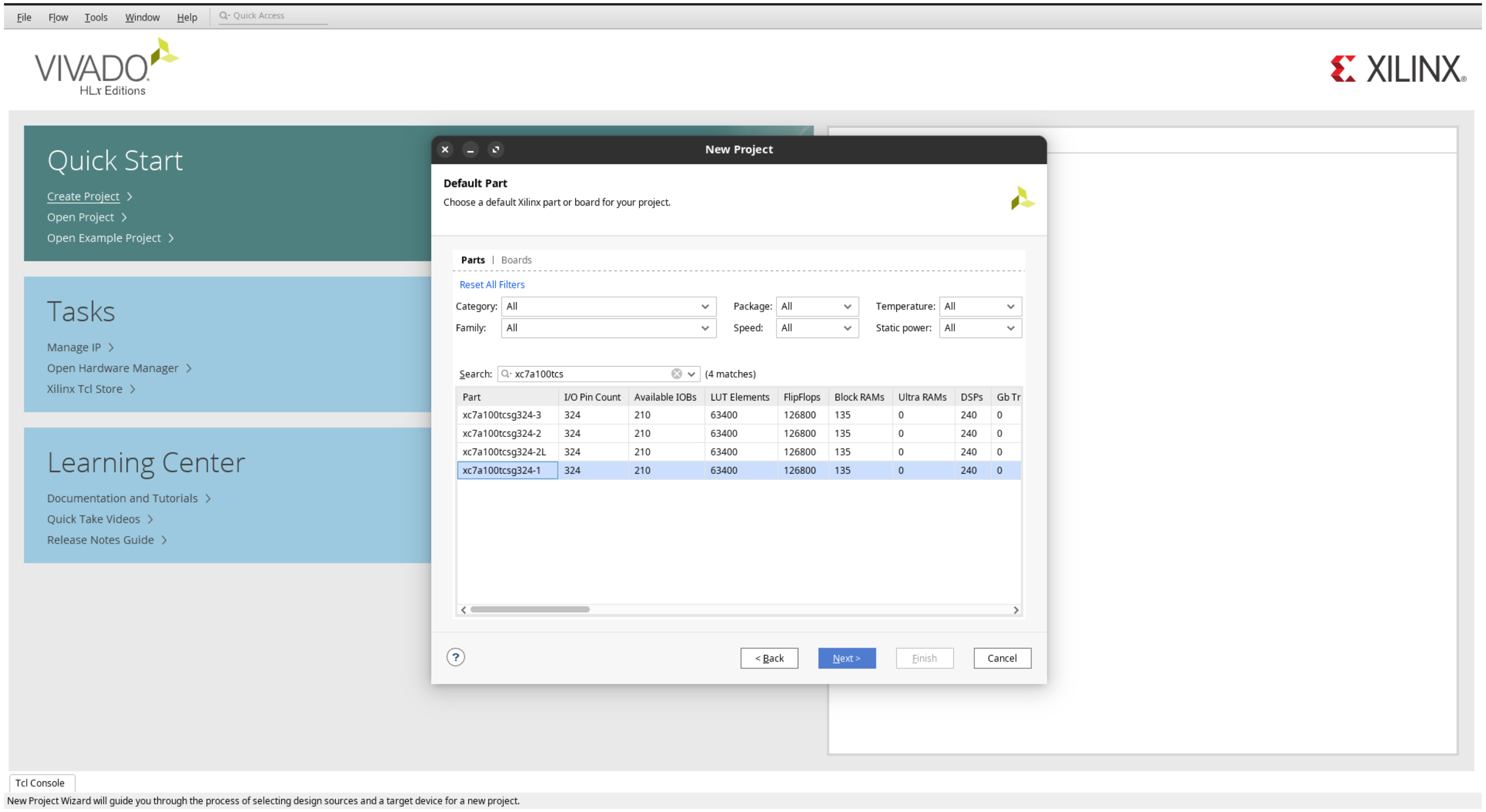
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
✓ synth_1	constrs_1	synth_design Complete!								8515	274	29.0	0	0	9/13/23, 1:35 PM	00:01:47	Flow_AreaOptimized_high* (Vivado Synthesis 2019)	Vivado Synthesis Defa
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Implementatio

Overview

1. Package SNN IP
2. SoC Integration with MicroBlaze
3. Software Implementation
4. Flash memory configuration

SoC Integration (1)

- Create a new Vivado project. Choose xc7a100tcsg324-1



Default Part

Choose a default Xilinx part or board for your project.

Parts | Boards

Reset All Filters

Category: All — Package: All — Temperature: All

Family: All — Speed: All — Static power: All

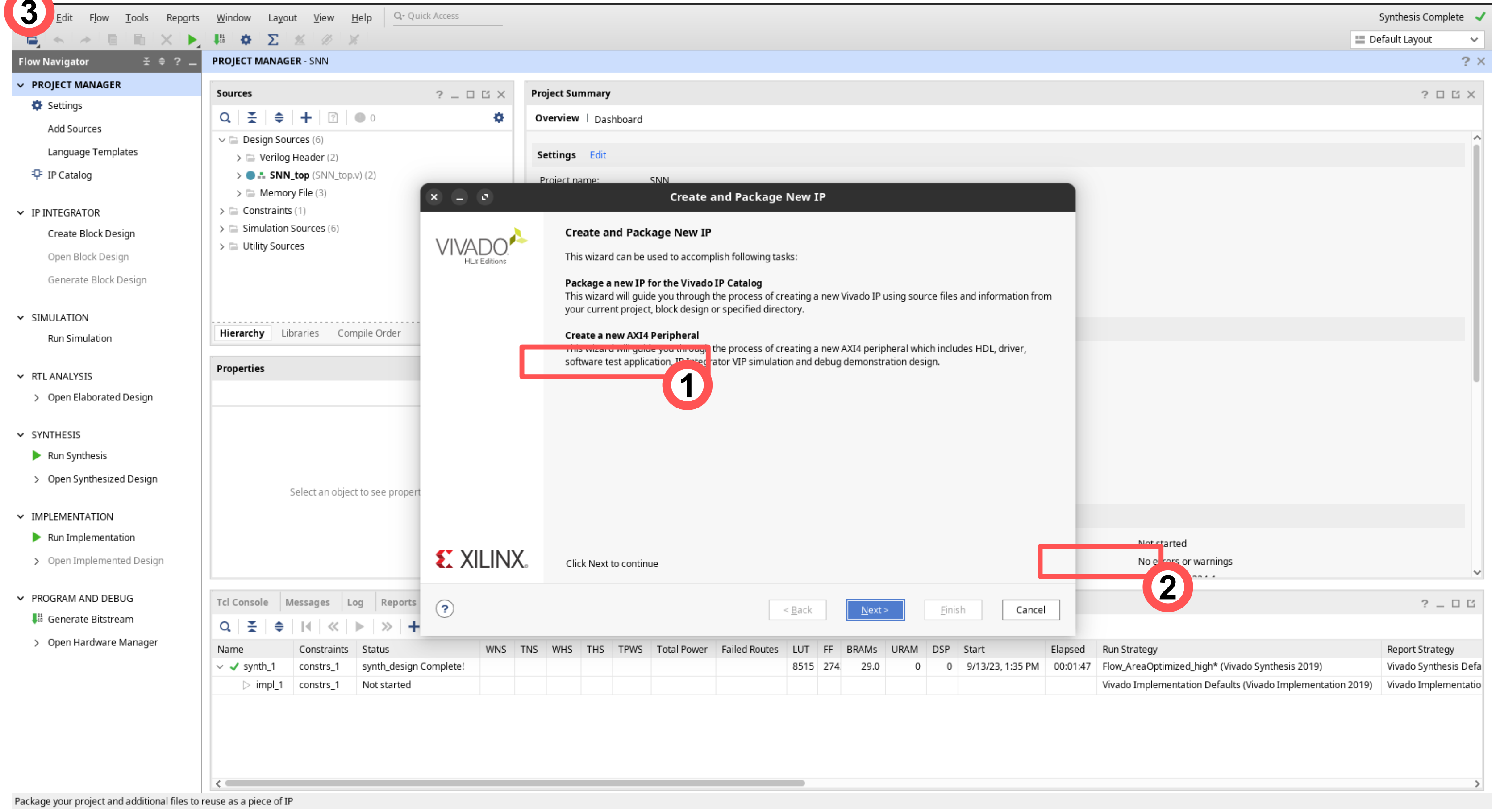
Search: xc7a100tcs (4 matches)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Tr
xc7a100tcsg324-3	324	210	63400	126800	135	0	240	0
xc7a100tcsg324-2	324	210	63400	126800	135	0	240	0
xc7a100tcsg324-2L	324	210	63400	126800	135	0	240	0
xc7a100tcsg324-1	324	210	63400	126800	135	0	240	0

New Project Wizard will guide you through the process of selecting design sources and a target device for a new project.

SoC Integration (1)

- Create a new Vivado project. Choose xc7a100tcsg324-1



Create and Package New IP

Create and Package New IP

This wizard can be used to accomplish following tasks:

Package a new IP for the Vivado IP Catalog
This wizard will guide you through the process of creating a new Vivado IP using source files and information from your current project, block design or specified directory.

Create a new AXI4 Peripheral
This wizard will guide you through the process of creating a new AXI4 peripheral which includes HDL, driver, software test application, IP Integrator VIP simulation and debug demonstration design.

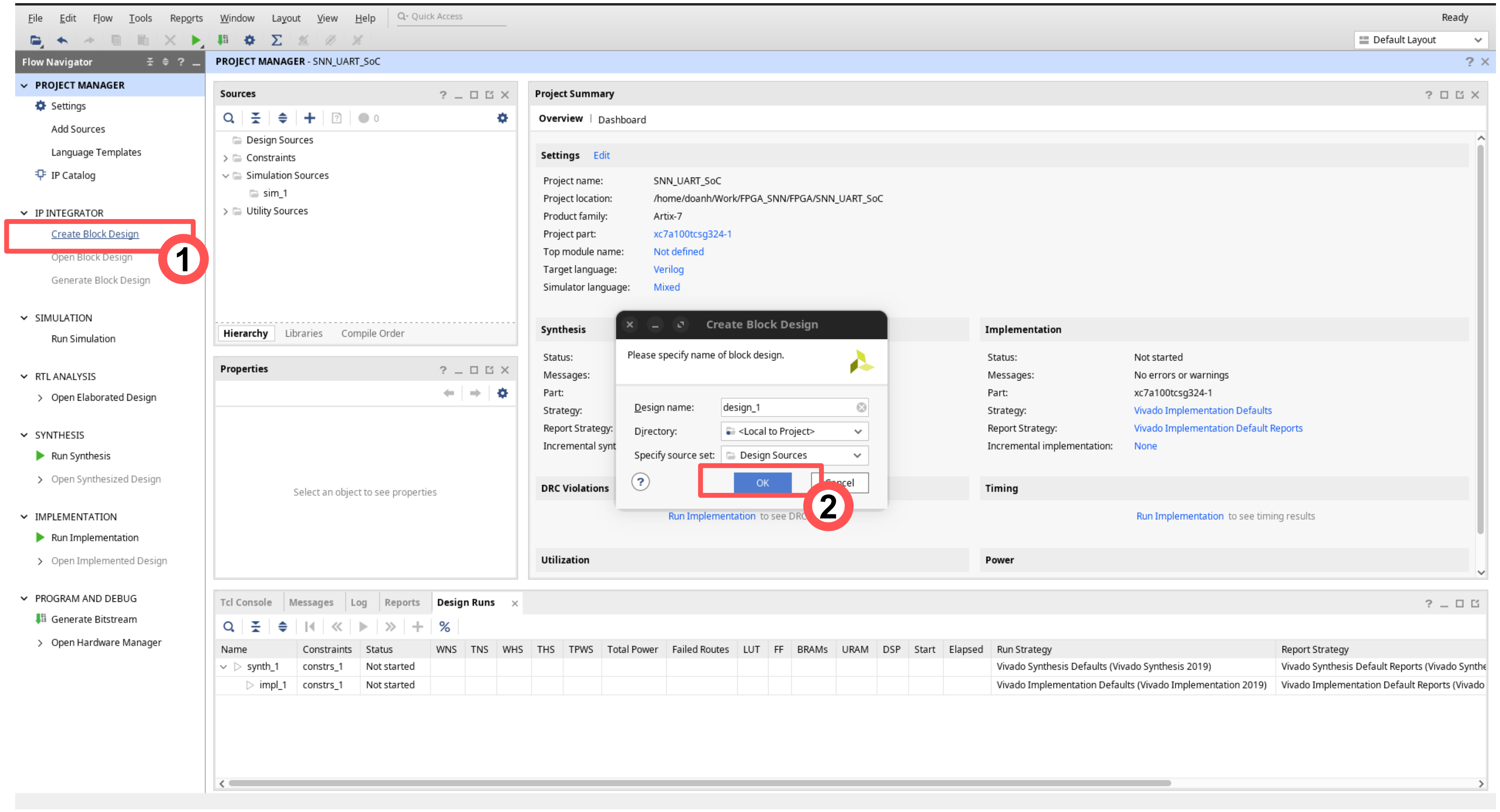
Click Next to continue

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
✓ synth_1	constrs_1	synth_design Complete!								8515	274	29.0	0	0	9/13/23, 1:35 PM	00:01:47	Flow_AreaOptimized_high* (Vivado Synthesis 2019)	Vivado Synthesis Defa
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Implementatio

Package your project and additional files to reuse as a piece of IP

SoC Integration (2)

- Create new block design



Settings Edit

Project name:	SNN_UART_SoC
Project location:	/home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Implementation

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

Create Block Design

Please specify name of block design.

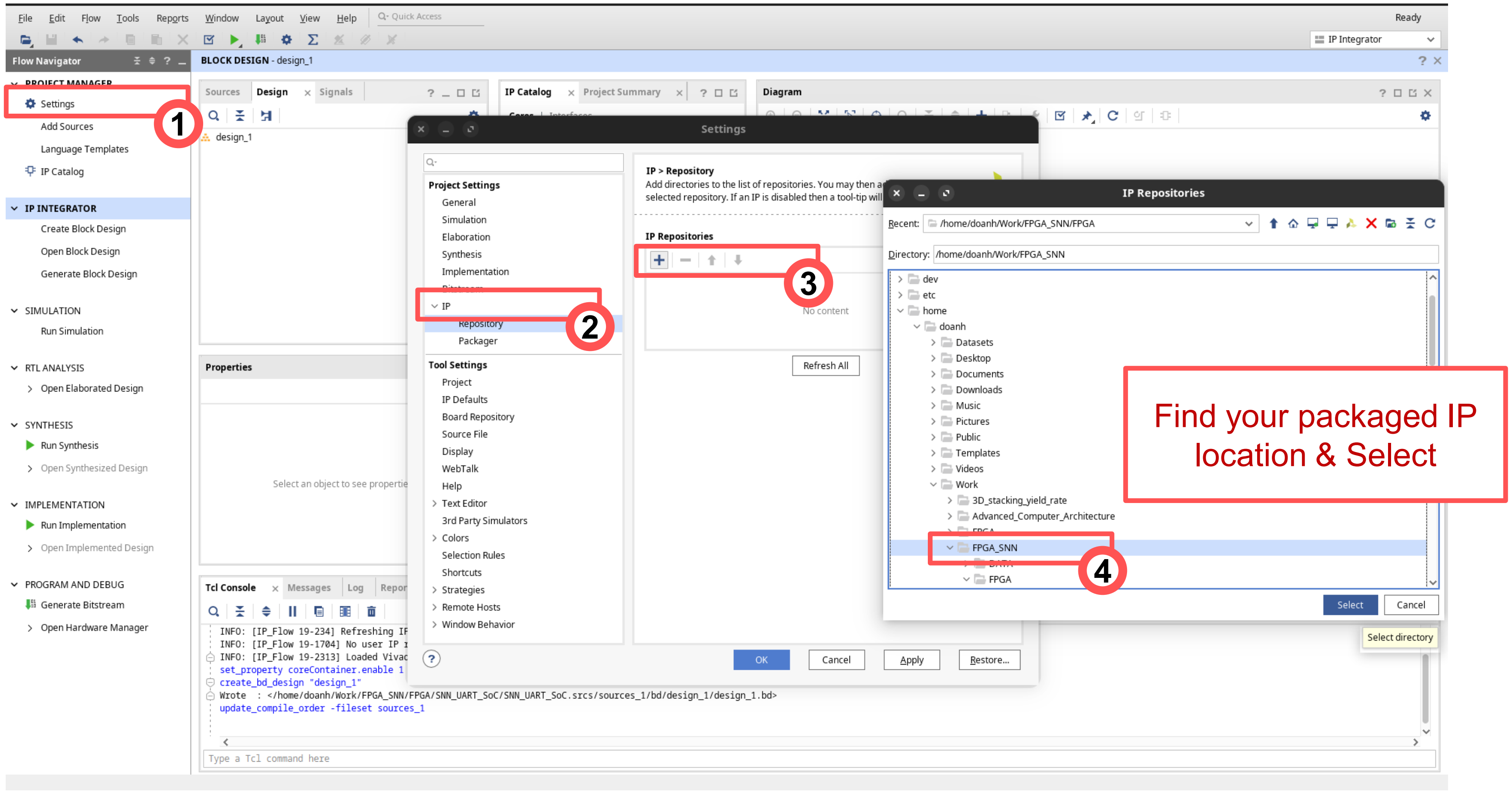
Design name:	design_1
Directory:	<Local to Project>
Specify source set:	Design Sources

OK Cancel

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2019)	Vivado Synthesis Default Reports (Vivado Synthe
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Implementation Default Reports (Vivado

SoC Integration (3)

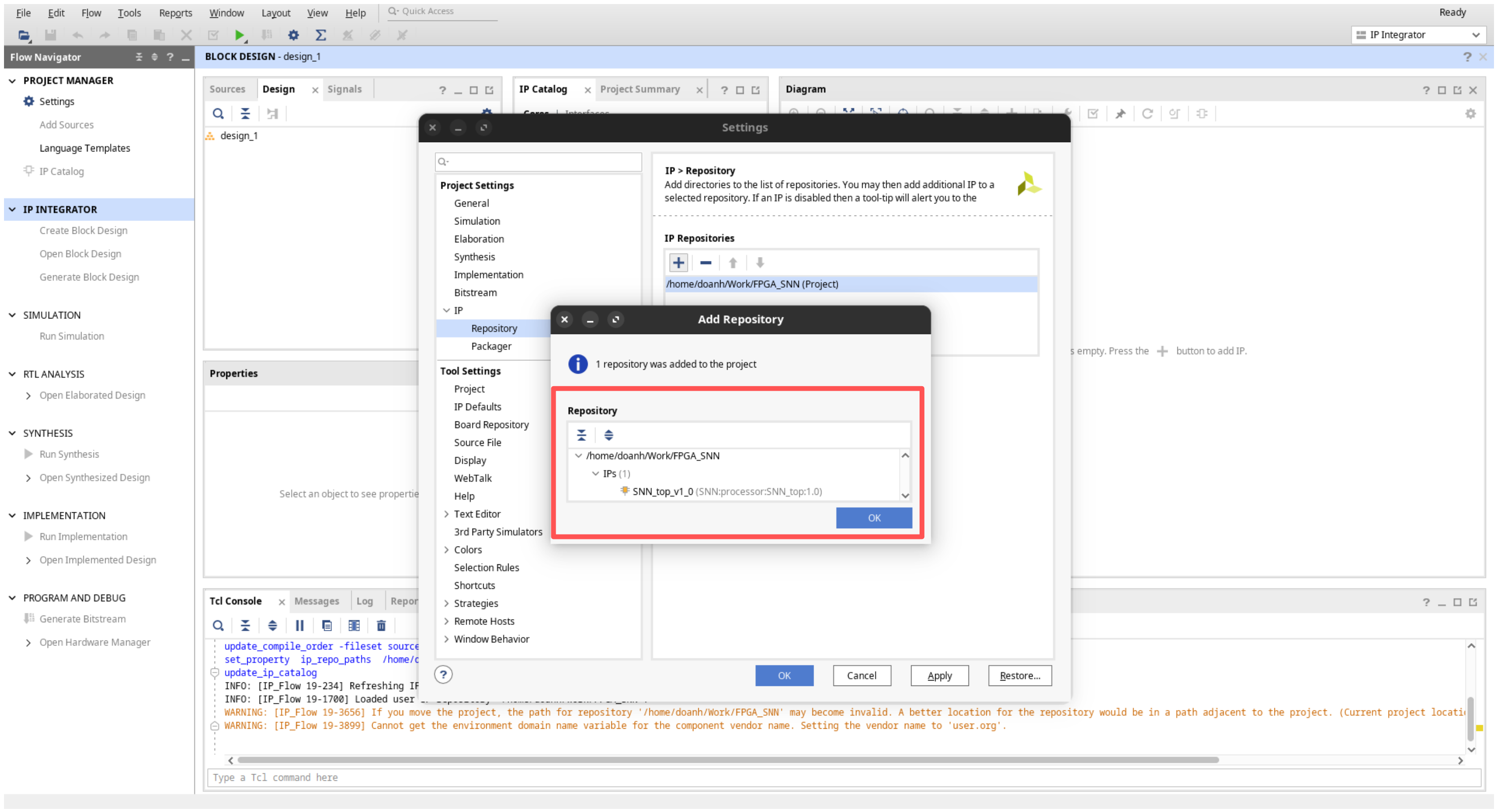
- Add packaged SNN IP into project



```
INFO: [IP_Flow 19-234] Refreshing IF
INFO: [IP_Flow 19-1704] No user IP r
INFO: [IP_Flow 19-2313] Loaded Vivac
set_property coreContainer.enable 1
create_bd_design "design_1"
Wrote  : </home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/design_1.bd>
update_compile_order -fileset sources_1
```

SoC Integration (4)

- Successful added IP



IP > Repository
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

IP Repositories
/home/doanh/Work/FPGA_SNN (Project)

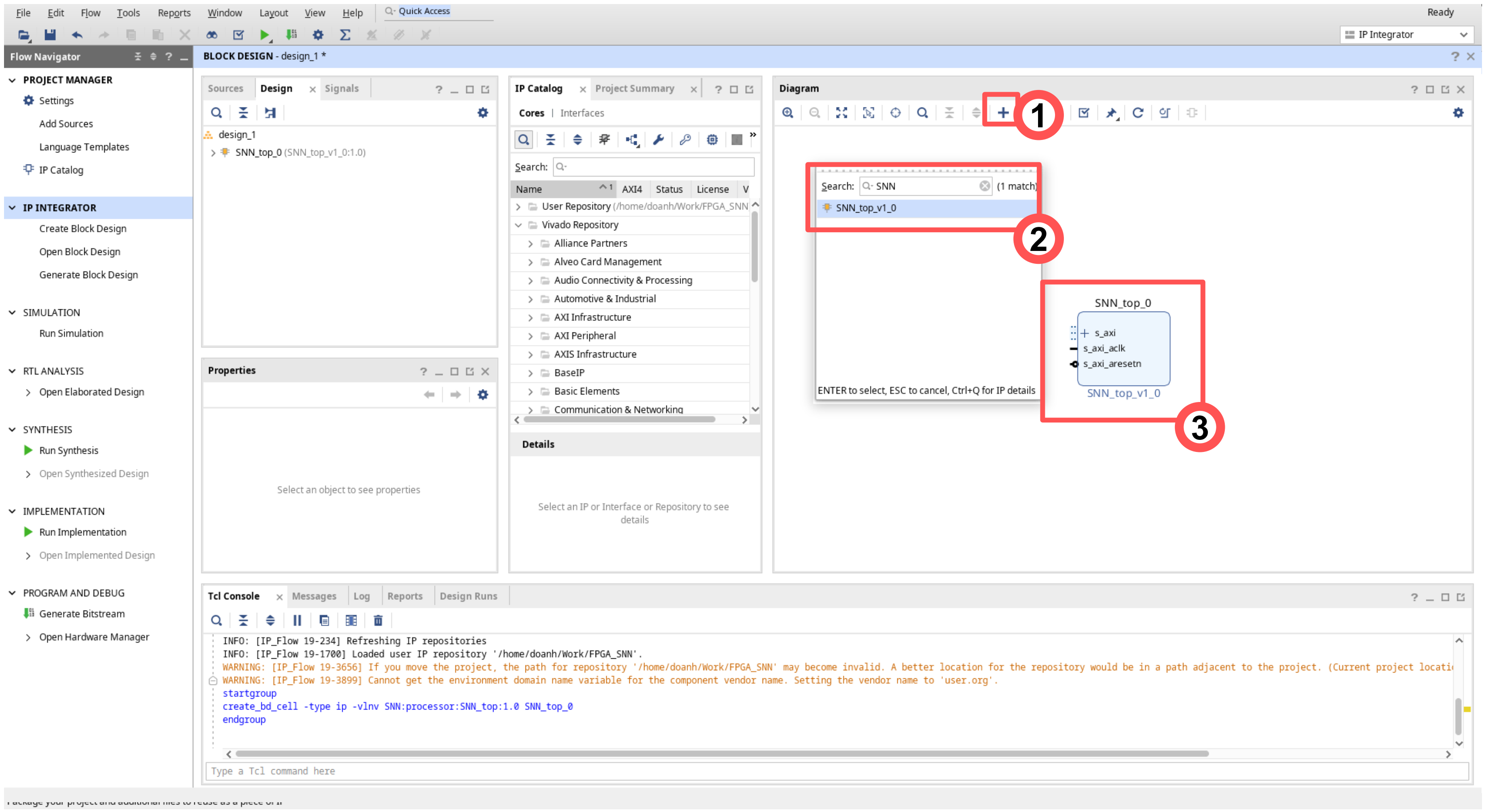
Add Repository
1 repository was added to the project

Repository
/home/doanh/Work/FPGA_SNN
IPs (1)
SNN_top_v1_0 (SNN:processor:SNN_top:1.0)

```
update_compile_order -fileset source
set_property  ip_repo_paths  /home/c
update_ip_catalog
INFO: [IP_Flow 19-234] Refreshing IF
INFO: [IP_Flow 19-1700] Loaded user
WARNING: [IP_Flow 19-3656] If you move the project, the path for repository '/home/doanh/Work/FPGA_SNN' may become invalid. A better location for the repository would be in a path adjacent to the project. (Current project locati
WARNING: [IP_Flow 19-3899] Cannot get the environment domain name variable for the component vendor name. Setting the vendor name to 'user.org'.
```

SoC Integration (5)

- Add SNN IP into Block Design

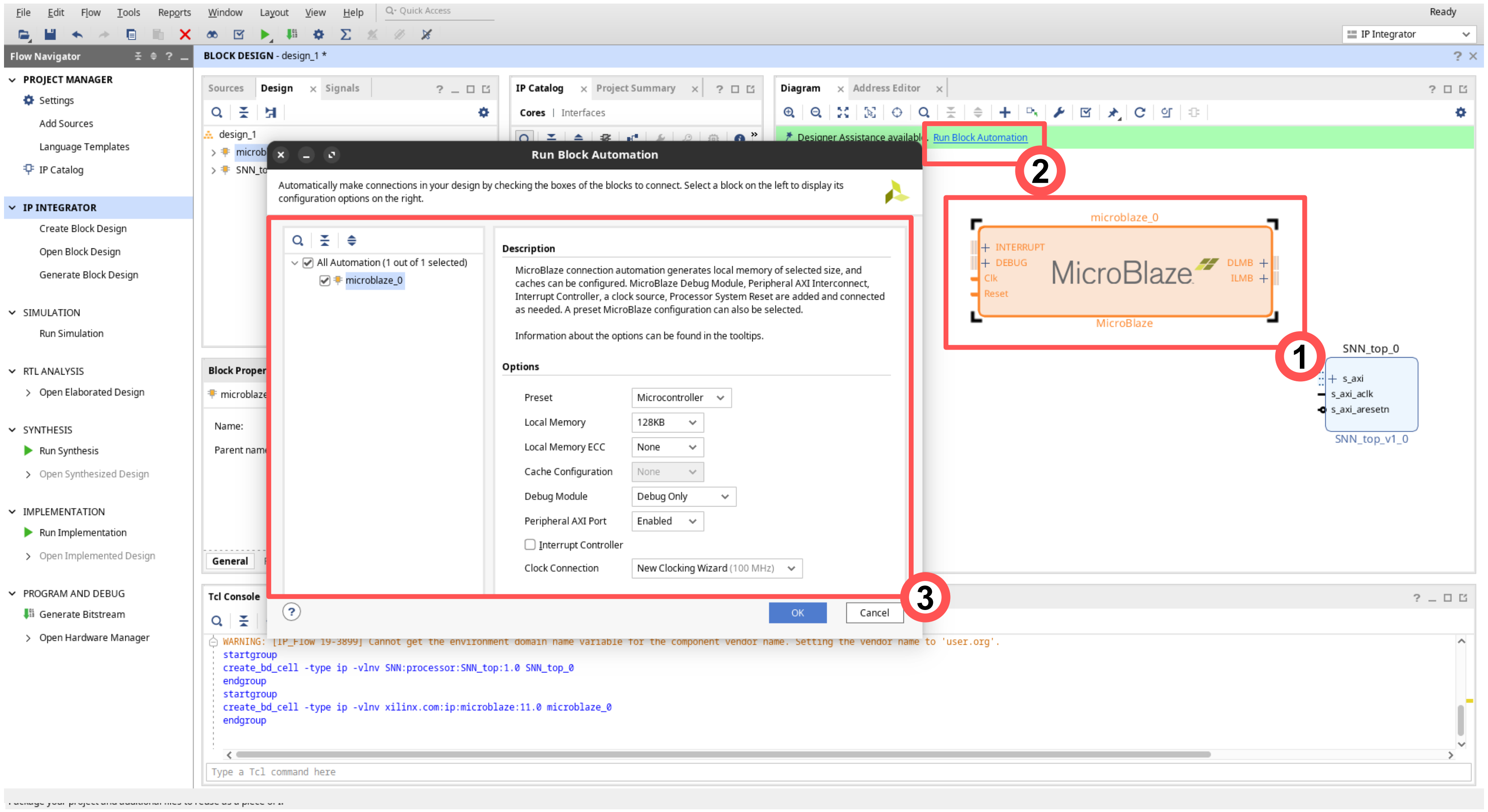


```
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1700] Loaded user IP repository '/home/doanh/Work/FPGA_SNN'.
WARNING: [IP_Flow 19-3656] If you move the project, the path for repository '/home/doanh/Work/FPGA_SNN' may become invalid. A better location for the repository would be in a path adjacent to the project. (Current project locati
WARNING: [IP_Flow 19-3899] Cannot get the environment domain name variable for the component vendor name. Setting the vendor name to 'user.org'.
startgroup
create_bd_cell -type ip -vlnv SNN:processor:SNN_top:1.0 SNN_top_0
endgroup
```

Type a Tcl command here

SoC Integration (6)

- Config SoC - MicroBlaze



Run Block Automation

Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.

- [✓] All Automation (1 out of 1 selected)
  - [✓] microblaze_0

Description

MicroBlaze connection automation generates local memory of selected size, and caches can be configured. MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset are added and connected as needed. A preset MicroBlaze configuration can also be selected.

Information about the options can be found in the tooltips.

Options

Preset	Microcontroller
Local Memory	128KB
Local Memory ECC	None
Cache Configuration	None
Debug Module	Debug Only
Peripheral AXI Port	Enabled
☐ Interrupt Controller	
Clock Connection	New Clocking Wizard (100 MHz)

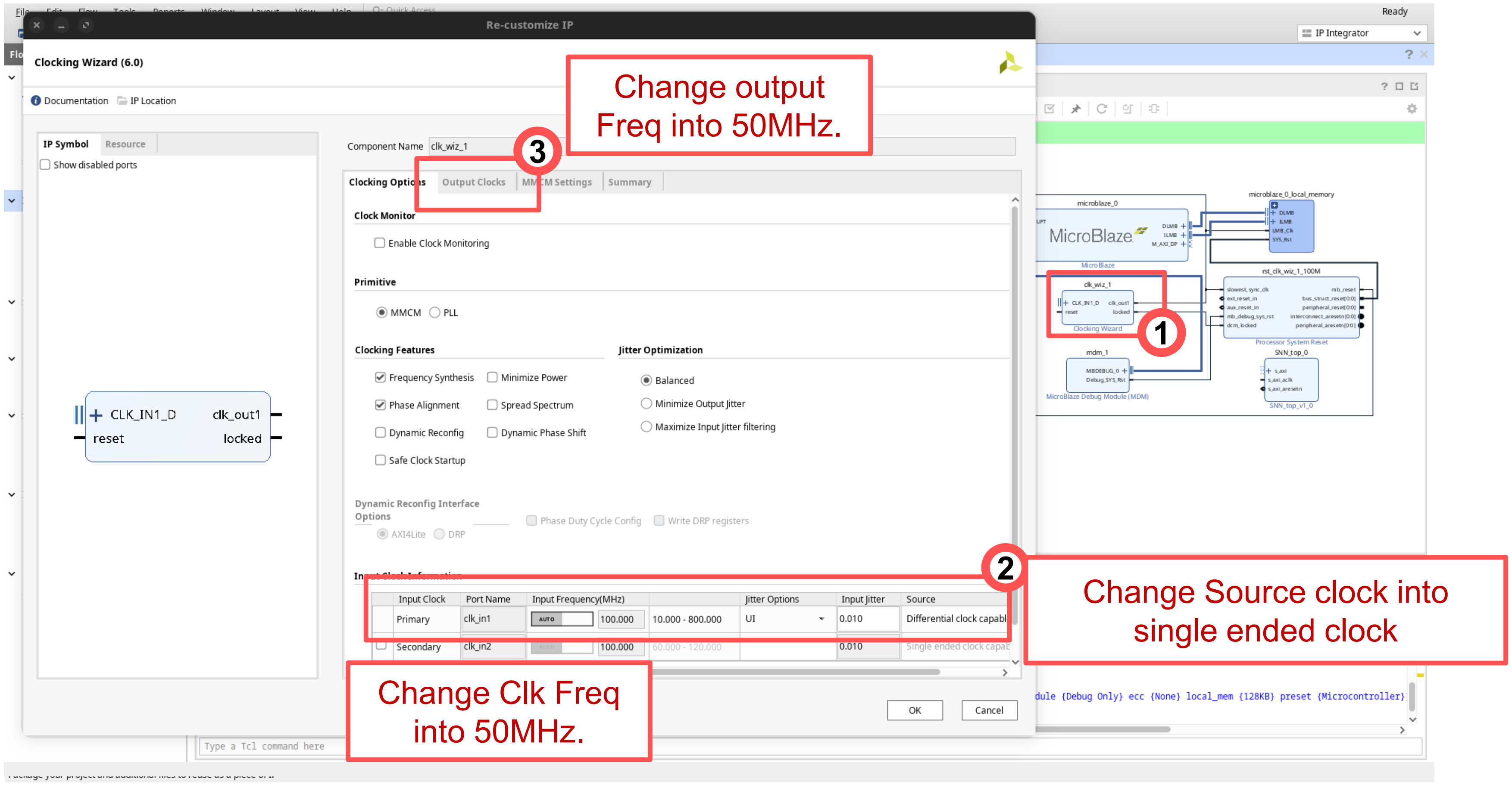
OK Cancel

```
WARNING: [IP_Flow 19-3899] Cannot get the environment domain name variable for the component vendor name. Setting the vendor name to 'user.org'.
startgroup
create_bd_cell -type ip -vlnv SNN:processor:SNN_top:1.0 SNN_top_0
endgroup
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:microblaze:11.0 microblaze_0
endgroup
```

Type a Tcl command here

SoC Integration (7)

- Config SoC - Clock Wizard



Re-customize IP

Clocking Wizard (6.0)

Documentation | IP Location

IP Symbol | Resource

☐ Show disabled ports

Component Name: clk_wiz_1

Clocking Options | Output Clocks | MMCM Settings | Summary

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	Port Name	Input Frequency(MHz)		Jitter Options	Input Jitter	Source
Primary	clk_in1	AUTO 100.000	10.000 - 800.000	UI	0.010	Differential clock capabl
Secondary	clk_in2	AUTO 100.000	60.000 - 120.000		0.010	Single ended clock capab

OK | Cancel

Change output Freq into 50MHz.

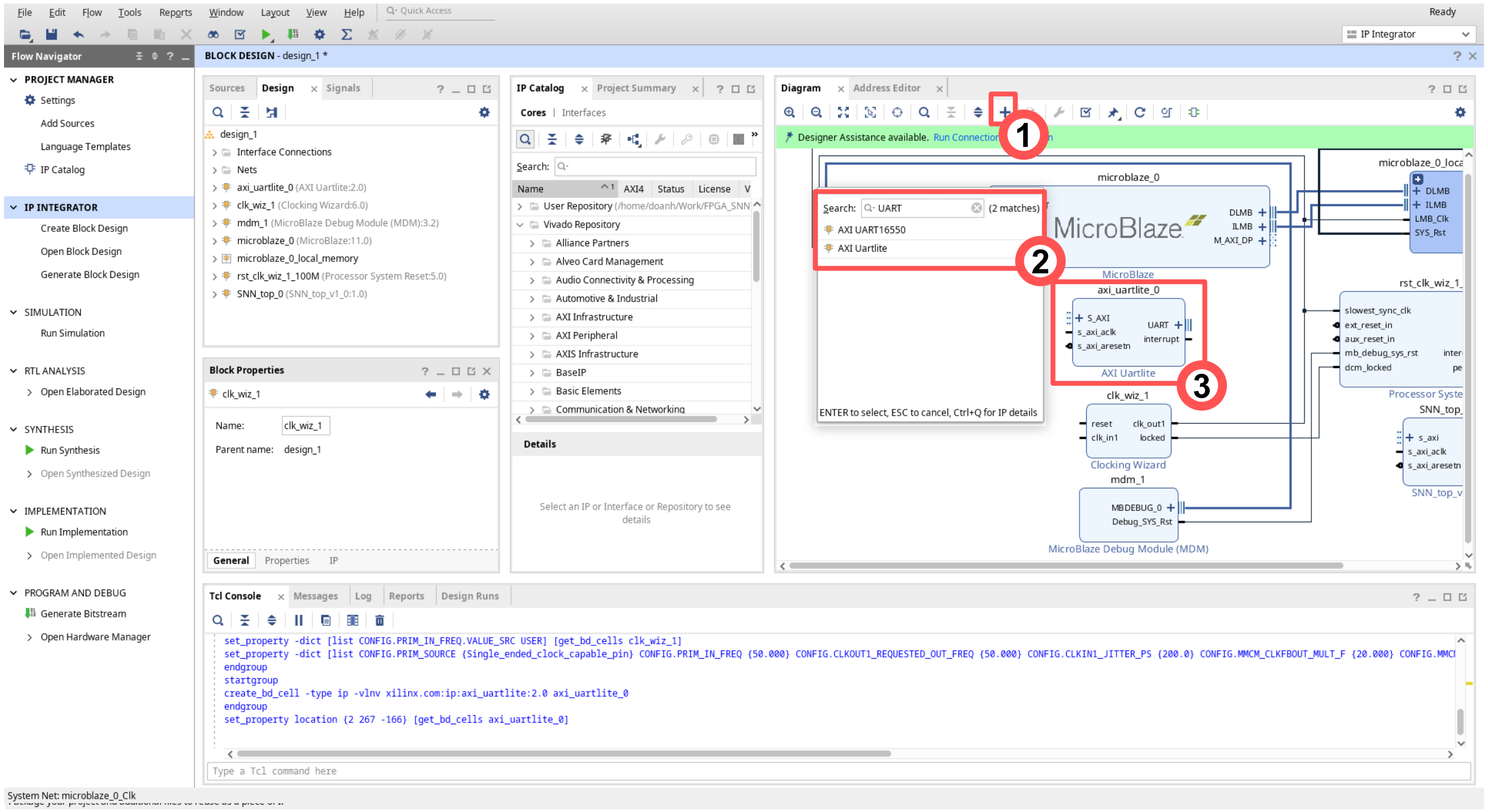
Change Source clock into single ended clock

Change Clk Freq into 50MHz.

```
dule {Debug Only} ecc {None} local_mem {128KB} preset {Microcontroller}
```

SoC Integration (8)

- Config SoC - Add UART Lite



Search: UART (2 matches)

AXI UART16550

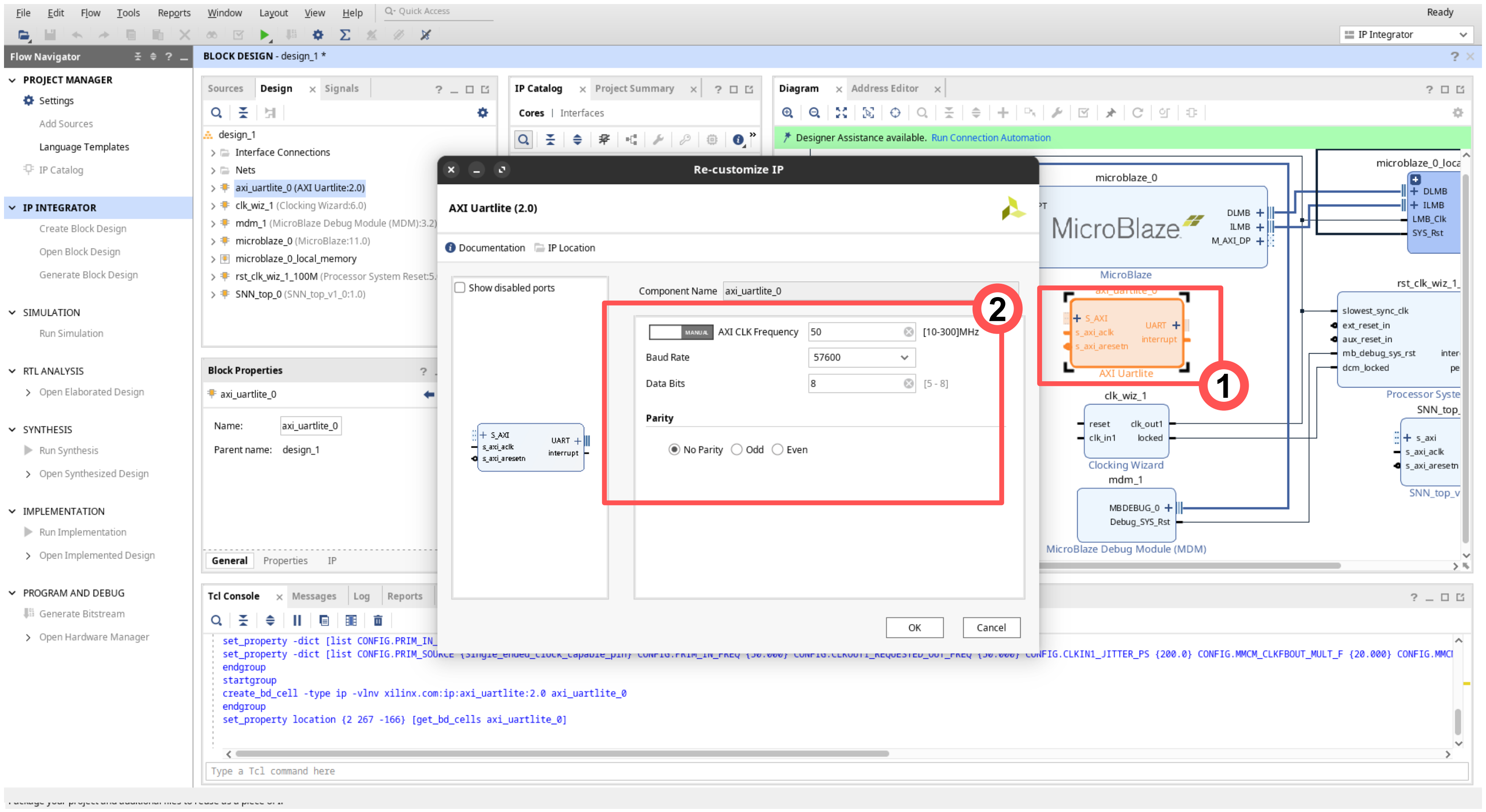
AXI Uartlite

ENTER to select, ESC to cancel, Ctrl+Q for IP details

```
set_property -dict [list CONFIG.PRIM_IN_FREQ.VALUE_SRC USER] [get_bd_cells clk_wiz_1]
set_property -dict [list CONFIG.PRIM_SOURCE {Single_ended_clock_capable_pin} CONFIG.PRIM_IN_FREQ {50.000} CONFIG.CLKOUT1_REQUESTED_OUT_FREQ {50.000} CONFIG.CLKIN1_JITTER_PS {200.0} CONFIG.MMCM_CLKFBOUT_MULT_F {20.000} CONFIG.MMCl
endgroup
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_uartlite:2.0 axi_uartlite_0
endgroup
set_property location {2 267 -166} [get_bd_cells axi_uartlite_0]
```

SoC Integration (8)

- Config SoC - Config UART Lite



Re-customize IP — AXI Uartlite (2.0)

Component Name: axi_uartlite_0

AXI CLK Frequency	50	[10-300]MHz
Baud Rate	57600	
Data Bits	8	[5 - 8]

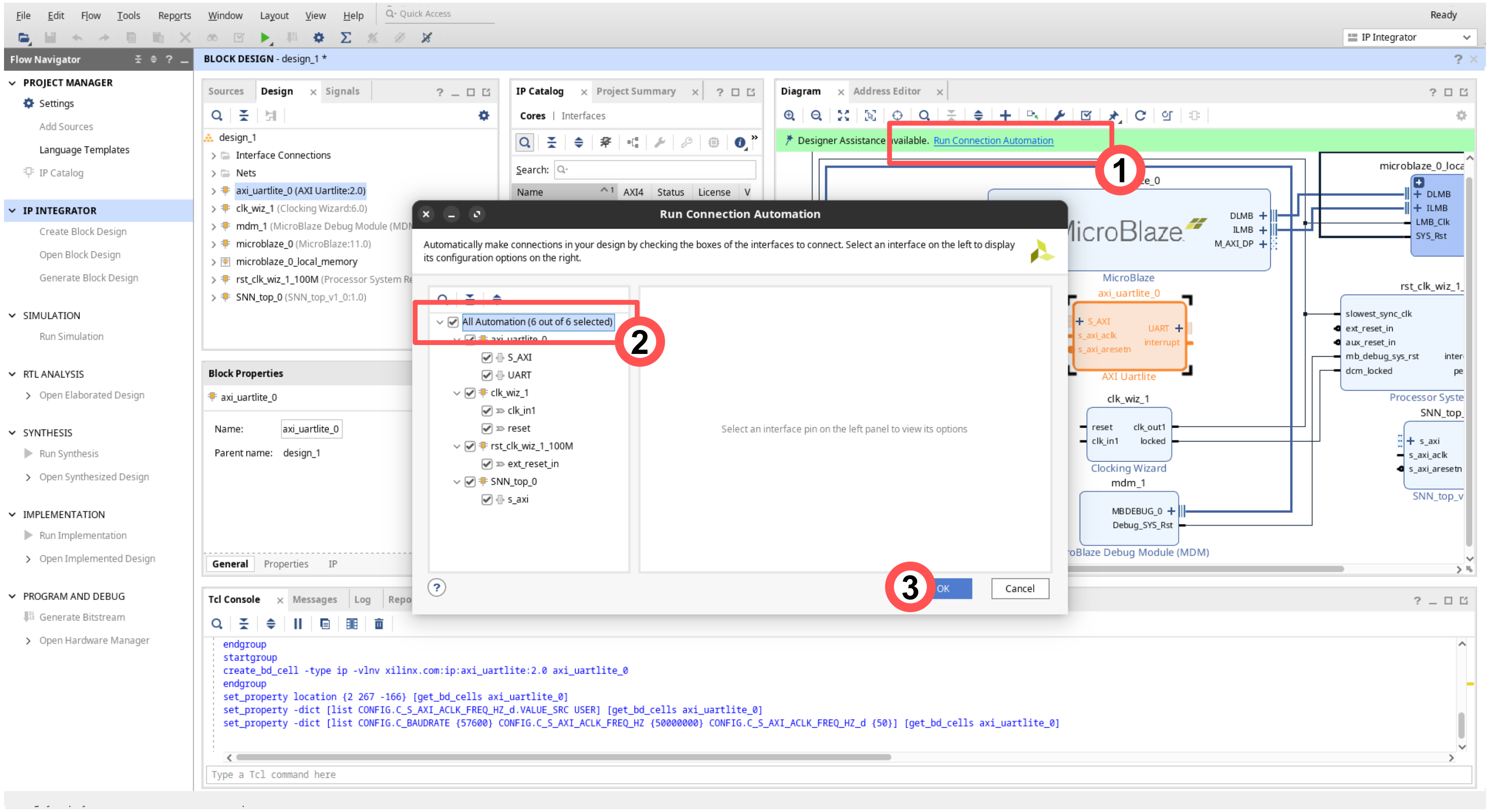
Parity: [✓ No Parity] Odd Even

```
set_property -dict [list CONFIG.PRIM_IN_
set_property -dict [list CONFIG.PRIM_SOURCE {Single_ended_clock_capable_pin} CONFIG.PRIM_IN_FREQ {50.000} CONFIG.CLKOUT1_REQUESTED_OUT_FREQ {50.000} CONFIG.CLKIN1_JITTER_PS {200.0} CONFIG.MMCM_CLKFBOUT_MULT_F {20.000} CONFIG.MMCI
endgroup
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_uartlite:2.0 axi_uartlite_0
endgroup
set_property location {2 267 -166} [get_bd_cells axi_uartlite_0]
```

1
2
Flow Navigator
PROJECT MANAGER
IP INTEGRATOR
BLOCK DESIGN - design_1 *
Sources
Design
Signals
design_1
axi_uartlite_0 (AXI Uartlite:2.0)
clk_wiz_1 (Clocking Wizard:6.0)
mdm_1 (MicroBlaze Debug Module (MDM):3.2)
microblaze_0 (MicroBlaze:11.0)
microblaze_0_local_memory
rst_clk_wiz_1_100M (Processor System Reset:5.
SNN_top_0 (SNN_top_v1_0:1.0)
IP Catalog
Diagram
Address Editor
Designer Assistance available. Run Connection Automation
microblaze_0
MicroBlaze
AXI Uartlite
clk_wiz_1
Clocking Wizard
mdm_1
MicroBlaze Debug Module (MDM)
rst_clk_wiz_1
Processor System Reset
SNN_top
Tcl Console
OK
Cancel

SoC Integration (9)

- Config SoC - Auto Connection



Run Connection Automation

Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.

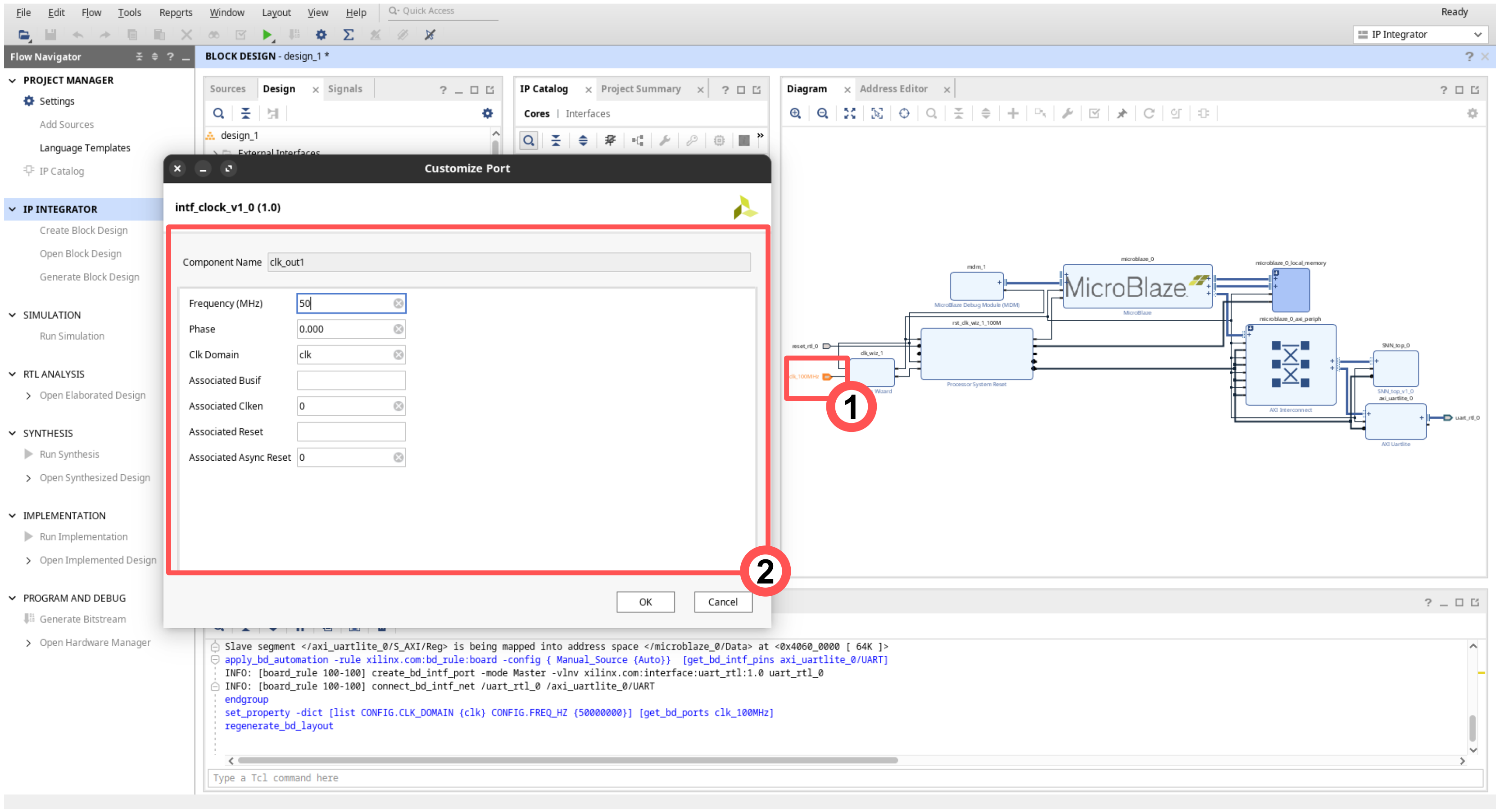
- ✓ All Automation (6 out of 6 selected)
  - ✓ axi_uartlite_0
    - ✓ S_AXI
    - ✓ UART
  - ✓ clk_wiz_1
    - ✓ clk_in1
    - ✓ reset
  - ✓ rst_clk_wiz_1_100M
    - ✓ ext_reset_in
  - ✓ SNN_top_0
    - ✓ s_axi

Select an interface pin on the left panel to view its options

```
endgroup
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_uartlite:2.0 axi_uartlite_0
endgroup
set_property location {2 267 -166} [get_bd_cells axi_uartlite_0]
set_property -dict [list CONFIG.C_S_AXI_ACLK_FREQ_HZ_d.VALUE_SRC USER] [get_bd_cells axi_uartlite_0]
set_property -dict [list CONFIG.C_BAUDRATE {57600} CONFIG.C_S_AXI_ACLK_FREQ_HZ {50000000} CONFIG.C_S_AXI_ACLK_FREQ_HZ_d {50}] [get_bd_cells axi_uartlite_0]
```

SoC Integration (10)

- Config SoC - Config Clock Pin



Customize Port

intf_clock_v1_0 (1.0)

Component Name	clk_out1
Frequency (MHz)	50
Phase	0.000
Clk Domain	clk
Associated Busif	
Associated Clken	0
Associated Reset	
Associated Async Reset	0

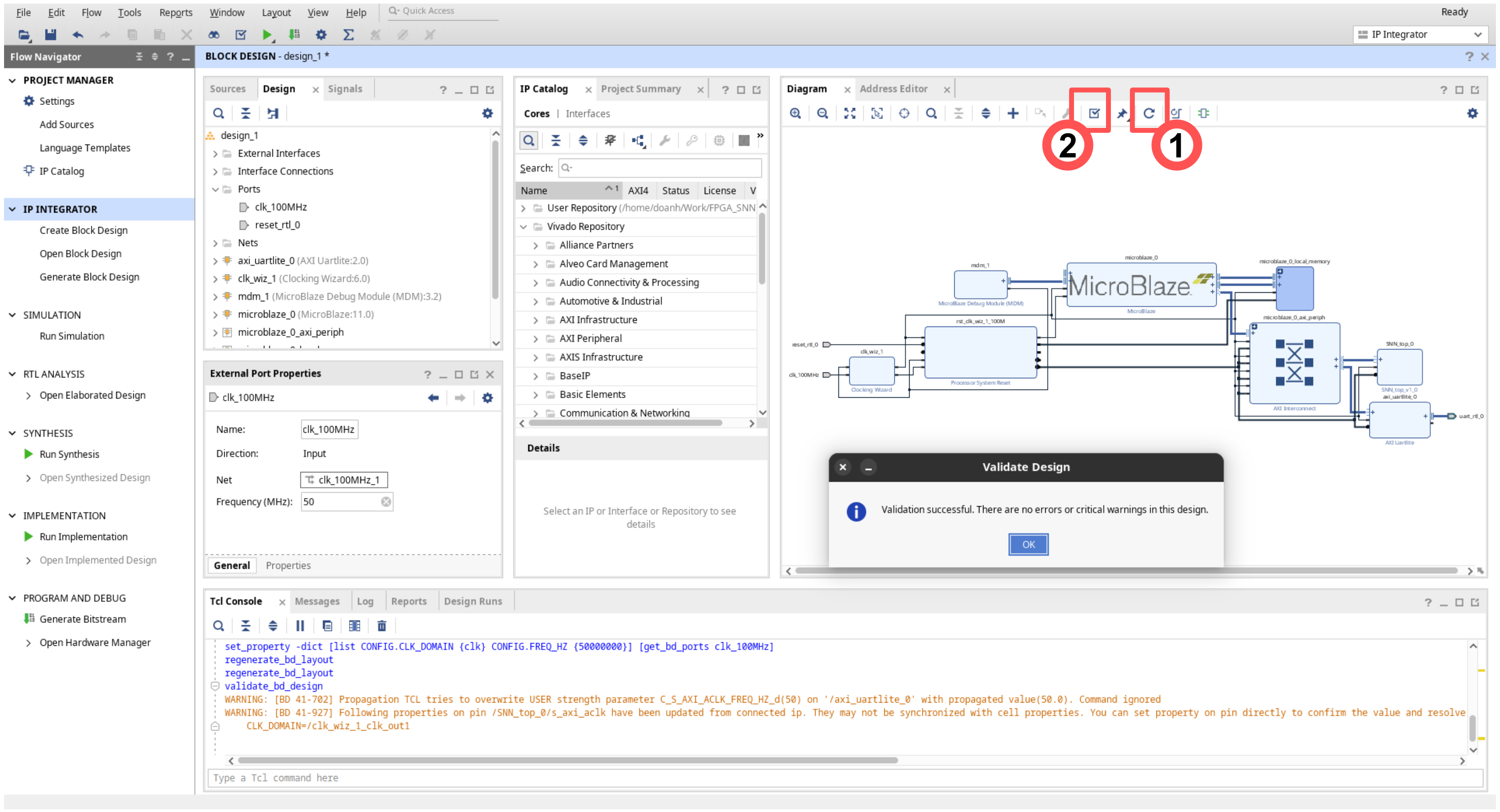
OK Cancel

```
Slave segment </axi_uartlite_0/S_AXI/Reg> is being mapped into address space </microblaze_0/Data> at <0x4060_0000 [ 64K ]>
apply_bd_automation -rule xilinx.com:bd_rule:board -config { Manual_Source {Auto}}  [get_bd_intf_pins axi_uartlite_0/UART]
INFO: [board_rule 100-100] create_bd_intf_port -mode Master -vlnv xilinx.com:interface:uart_rtl:1.0 uart_rtl_0
INFO: [board_rule 100-100] connect_bd_intf_net /uart_rtl_0 /axi_uartlite_0/UART
endgroup
set_property -dict [list CONFIG.CLK_DOMAIN {clk} CONFIG.FREQ_HZ {50000000}] [get_bd_ports clk_100MHz]
regenerate_bd_layout
```

Type a Tcl command here

SoC Integration (11)

- Validate block design



clk_100MHz

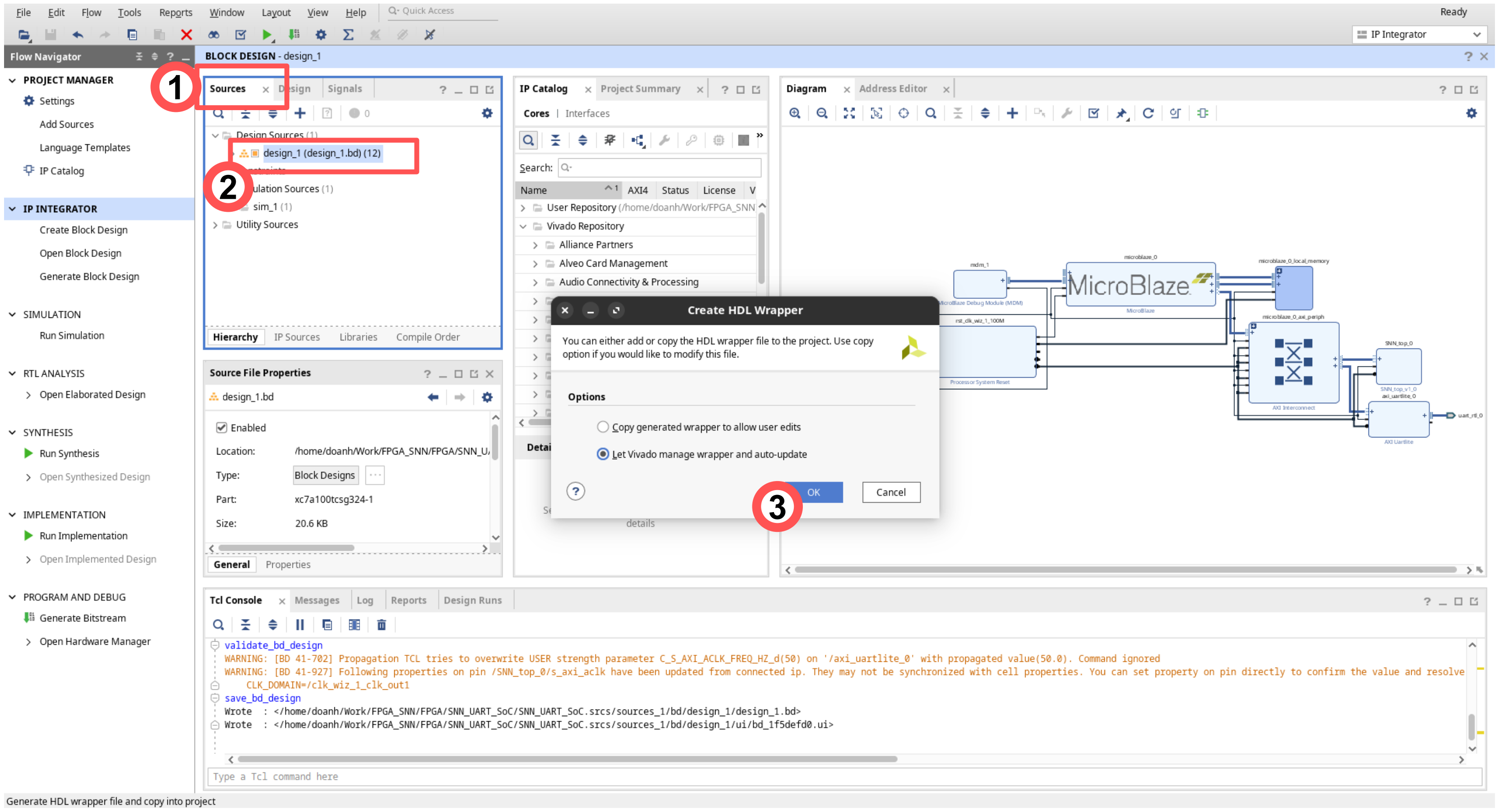
Name:	clk_100MHz
Direction:	Input
Net	clk_100MHz_1
Frequency (MHz):	50

Validation successful. There are no errors or critical warnings in this design.

```
set_property -dict [list CONFIG.CLK_DOMAIN {clk} CONFIG.FREQ_HZ {50000000}] [get_bd_ports clk_100MHz]
regenerate_bd_layout
regenerate_bd_layout
validate_bd_design
WARNING: [BD 41-702] Propagation TCL tries to overwrite USER strength parameter C_S_AXI_ACLK_FREQ_HZ_d(50) on '/axi_uartlite_0' with propagated value(50.0). Command ignored
WARNING: [BD 41-927] Following properties on pin /SNN_top_0/s_axi_aclk have been updated from connected ip. They may not be synchronized with cell properties. You can set property on pin directly to confirm the value and resolve
    CLK_DOMAIN=/clk_wiz_1_clk_out1
```

SoC Integration (12)

- Create HDL wrapper for block design



Enabled: ✓
Location: /home/doanh/Work/FPGA_SNN/FPGA/SNN_U
Type: Block Designs
Part: xc7a100tcsg324-1
Size: 20.6 KB

Create HDL Wrapper

You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.

Options

- ( ) Copy generated wrapper to allow user edits
- (●) Let Vivado manage wrapper and auto-update

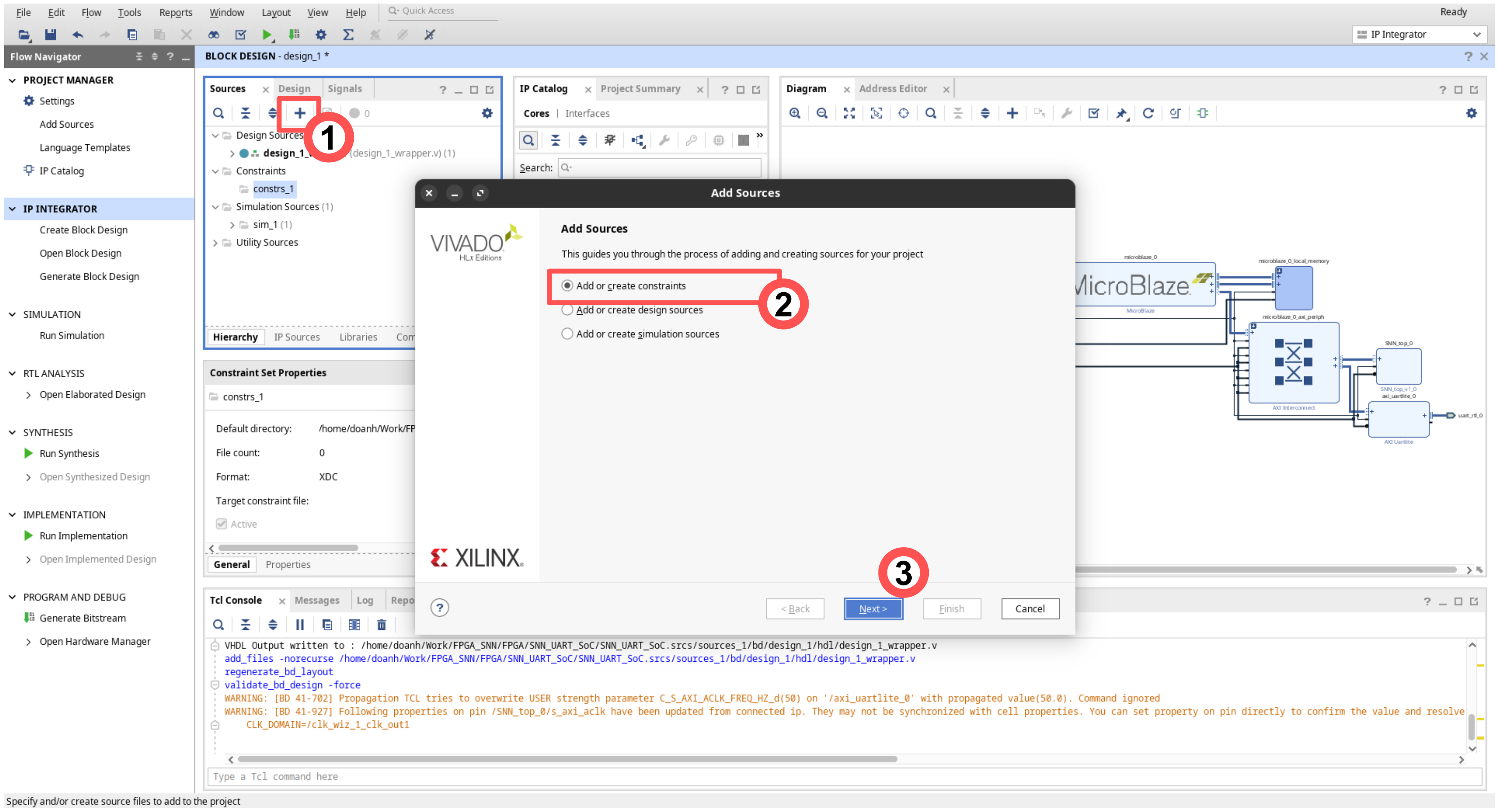
OK | Cancel

Tcl Console

```
validate_bd_design
WARNING: [BD 41-702] Propagation TCL tries to overwrite USER strength parameter C_S_AXI_ACLK_FREQ_HZ_d(50) on '/axi_uartlite_0' with propagated value(50.0). Command ignored
WARNING: [BD 41-927] Following properties on pin /SNN_top_0/s_axi_aclk have been updated from connected ip. They may not be synchronized with cell properties. You can set property on pin directly to confirm the value and resolve
    CLK_DOMAIN=/clk_wiz_1_clk_out1
save_bd_design
Wrote  : </home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/design_1.bd>
Wrote  : </home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/ui/bd_1f5defd0.ui>
```

SoC Integration (13)

- Create Design Constraint



Add Sources

This guides you through the process of adding and creating sources for your project

Constraint Set Properties

constrs_1

Default directory: /home/doanh/Work/FP

File count: 0

Format: XDC

Target constraint file:

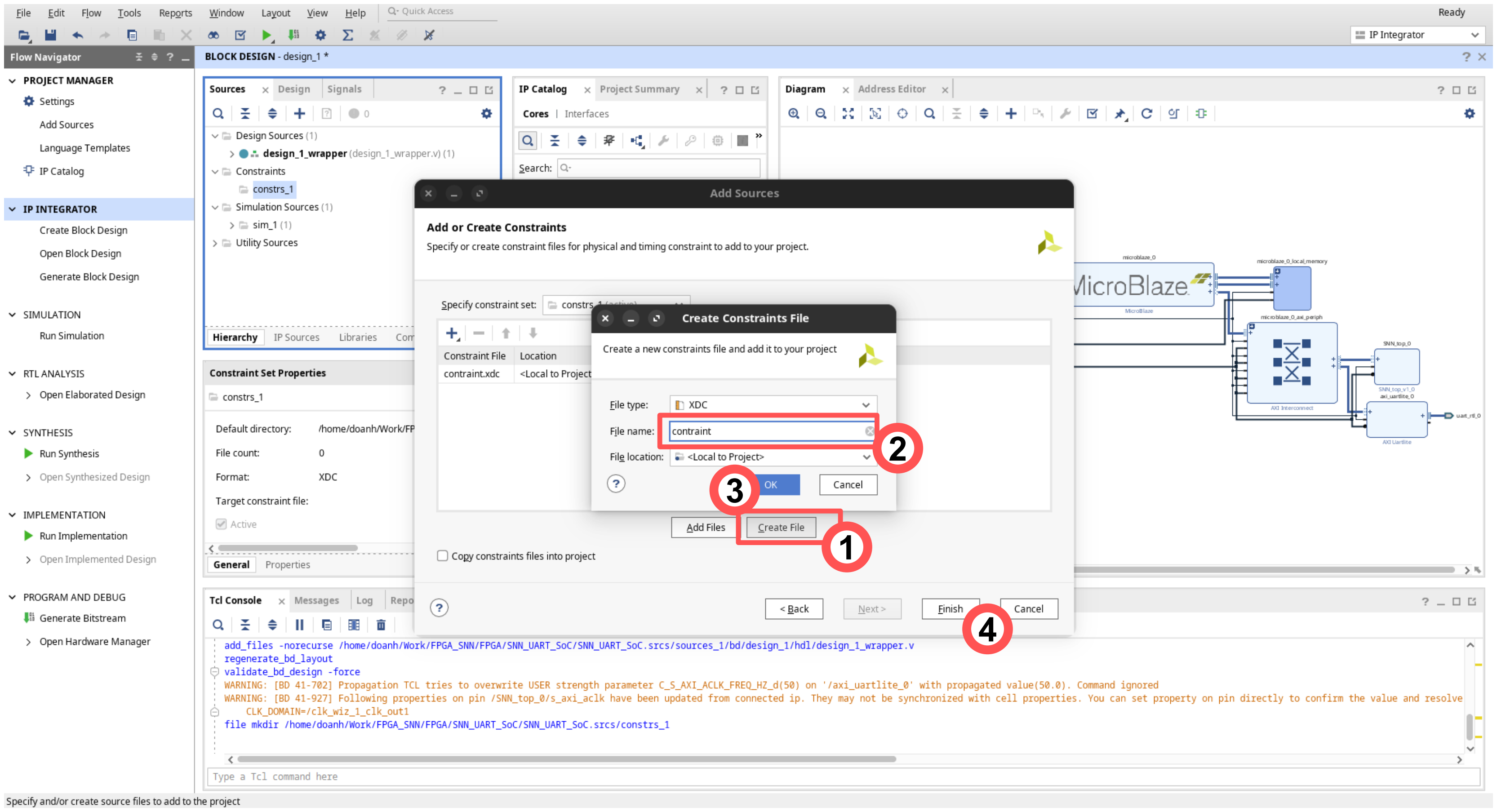
Active

```
VHDL Output written to : /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/hdl/design_1_wrapper.v
add_files -norecurse /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/hdl/design_1_wrapper.v
regenerate_bd_layout
validate_bd_design -force
WARNING: [BD 41-702] Propagation TCL tries to overwrite USER strength parameter C_S_AXI_ACLK_FREQ_HZ_d(50) on '/axi_uartlite_0' with propagated value(50.0). Command ignored
WARNING: [BD 41-927] Following properties on pin /SNN_top_0/s_axi_aclk have been updated from connected ip. They may not be synchronized with cell properties. You can set property on pin directly to confirm the value and resolve
    CLK_DOMAIN=/clk_wiz_1_clk_out1
```

Specify and/or create source files to add to the project

SoC Integration (14)

- Create Design Constraint



Add Sources

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)

Constraint File	Location
contraint.xdc	<Local to Project>

Copy constraints files into project

Create Constraints File

Create a new constraints file and add it to your project

File type: XDC

File name: contraint

File location: <Local to Project>

Constraint Set Properties

constrs_1

Default directory: /home/doanh/Work/FP

File count: 0

Format: XDC

Target constraint file:

Active

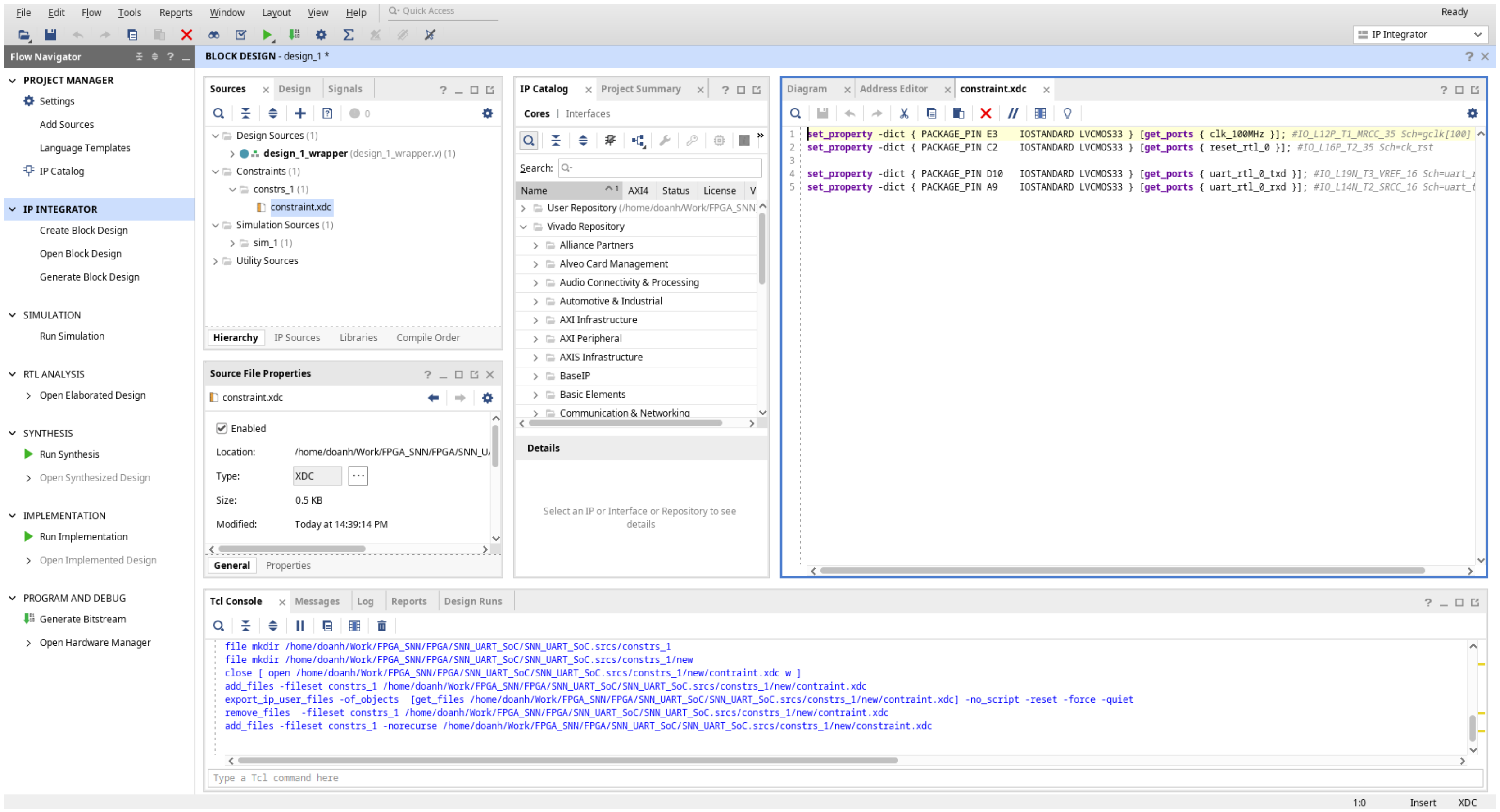
Tcl Console

```
add_files -norecurse /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/sources_1/bd/design_1/hdl/design_1_wrapper.v
regenerate_bd_layout
validate_bd_design -force
WARNING: [BD 41-702] Propagation TCL tries to overwrite USER strength parameter C_S_AXI_ACLK_FREQ_HZ_d(50) on '/axi_uartlite_0' with propagated value(50.0). Command ignored
WARNING: [BD 41-927] Following properties on pin /SNN_top_0/s_axi_aclk have been updated from connected ip. They may not be synchronized with cell properties. You can set property on pin directly to confirm the value and resolve
    CLK_DOMAIN=/clk_wiz_1_clk_out1
file mkdir /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1
```

Specify and/or create source files to add to the project

SoC Integration (15)

- Define design constraints (Mapping I/O Pin)



```
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk_100MHz }]; #IO_L12P_T1_MRCC_35 Sch=gclk[100]
set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { reset_rtl_0 }]; #IO_L16P_T2_35 Sch=ck_rst

set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_txd }]; #IO_L19N_T3_VREF_16 Sch=uart_
set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_rxd }]; #IO_L14N_T2_SRCC_16 Sch=uart_
```

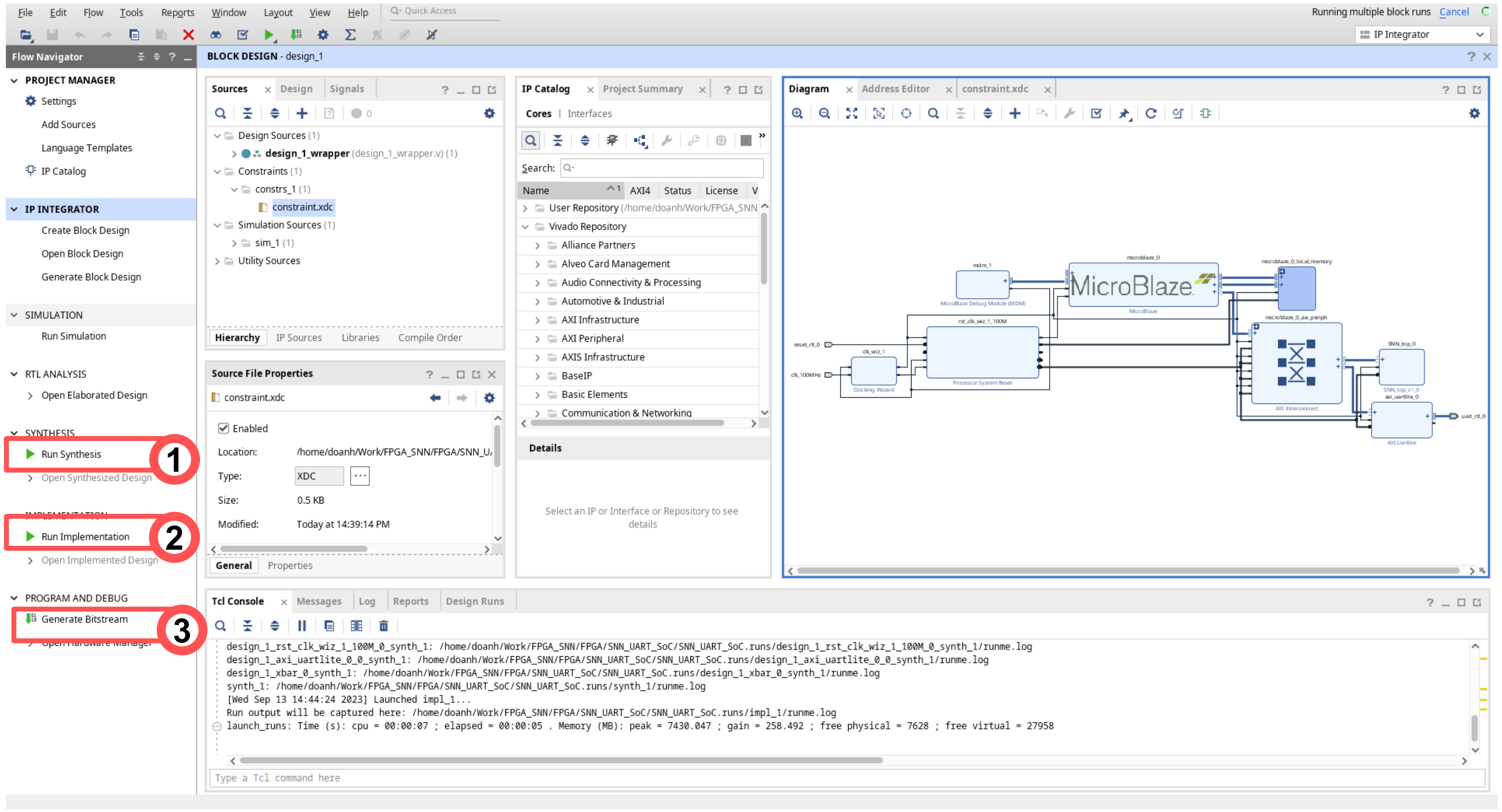
```
file mkdir /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1
file mkdir /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new
close [ open /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new/contraint.xdc w ]
add_files -fileset constrs_1 /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new/contraint.xdc
export_ip_user_files -of_objects  [get_files /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new/contraint.xdc] -no_script -reset -force -quiet
remove_files  -fileset constrs_1 /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new/contraint.xdc
add_files -fileset constrs_1 -norecurse /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.srcs/constrs_1/new/constraint.xdc
```

Source File Properties: constraint.xdc

- Enabled
- Location: /home/doanh/Work/FPGA_SNN/FPGA/SNN_U
- Type: XDC
- Size: 0.5 KB
- Modified: Today at 14:39:14 PM

SoC Integration (16)

- Synthesize + Implementation + Generate Bitstream



Source File Properties

constraint.xdc

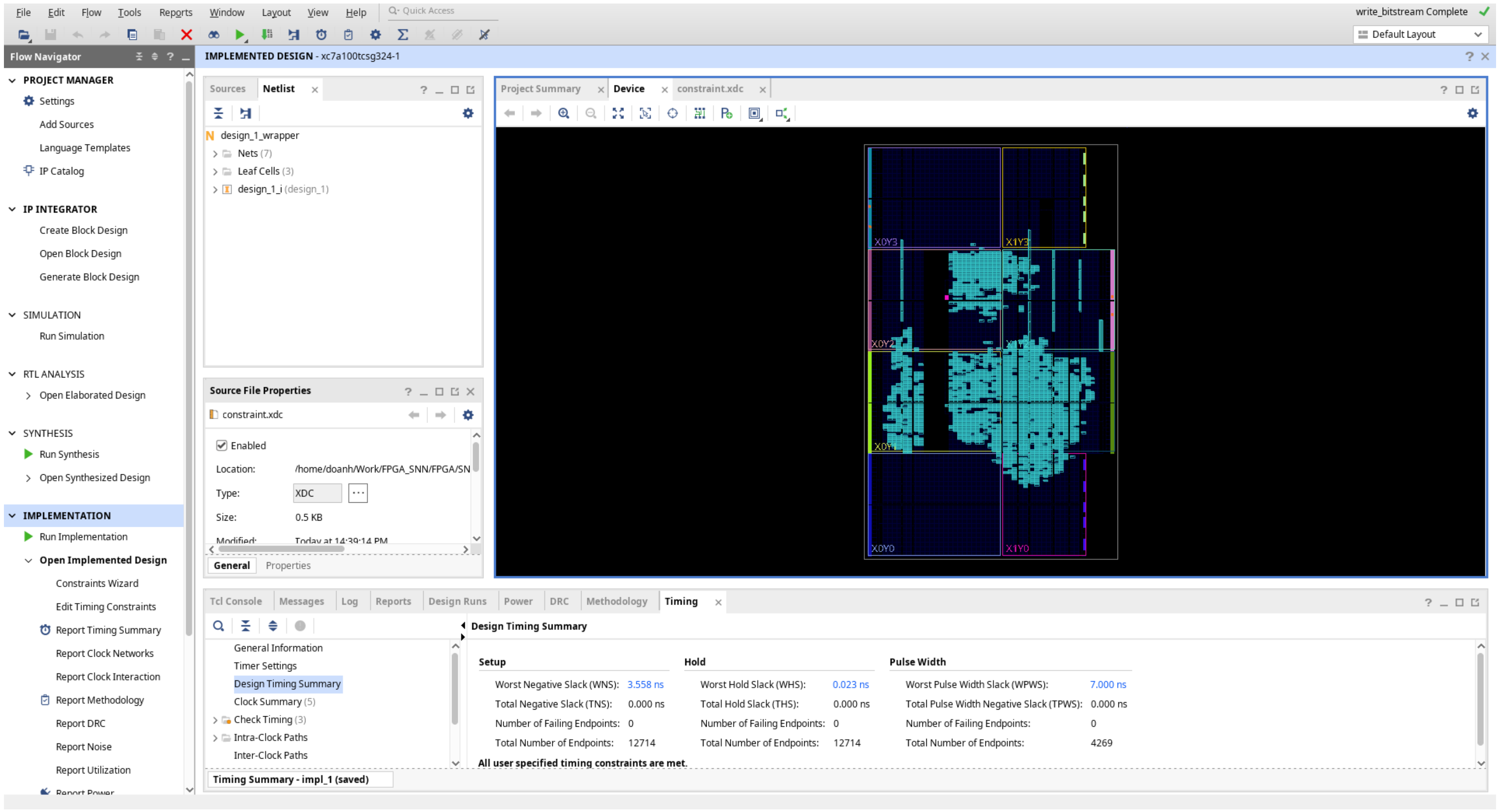
Enabled

Location:	/home/doanh/Work/FPGA_SNN/FPGA/SNN_U
Type:	XDC
Size:	0.5 KB
Modified:	Today at 14:39:14 PM

```
design_1_rst_clk_wiz_1_100M_0_synth_1: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/design_1_rst_clk_wiz_1_100M_0_synth_1/runme.log
design_1_axi_uartlite_0_0_synth_1: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/design_1_axi_uartlite_0_0_synth_1/runme.log
design_1_xbar_0_synth_1: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/design_1_xbar_0_synth_1/runme.log
synth_1: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/synth_1/runme.log
[Wed Sep 13 14:44:24 2023] Launched impl_1...
Run output will be captured here: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/impl_1/runme.log
launch_runs: Time (s): cpu = 00:00:07 ; elapsed = 00:00:05 . Memory (MB): peak = 7430.047 ; gain = 258.492 ; free physical = 7628 ; free virtual = 27958
```

SoC Integration (16)

- Successful Implementation



Enabled

Location: /home/doanh/Work/FPGA_SNN/FPGA/SN

Type: XDC

Size: 0.5 KB

Modified: Today at 14:39:14 PM

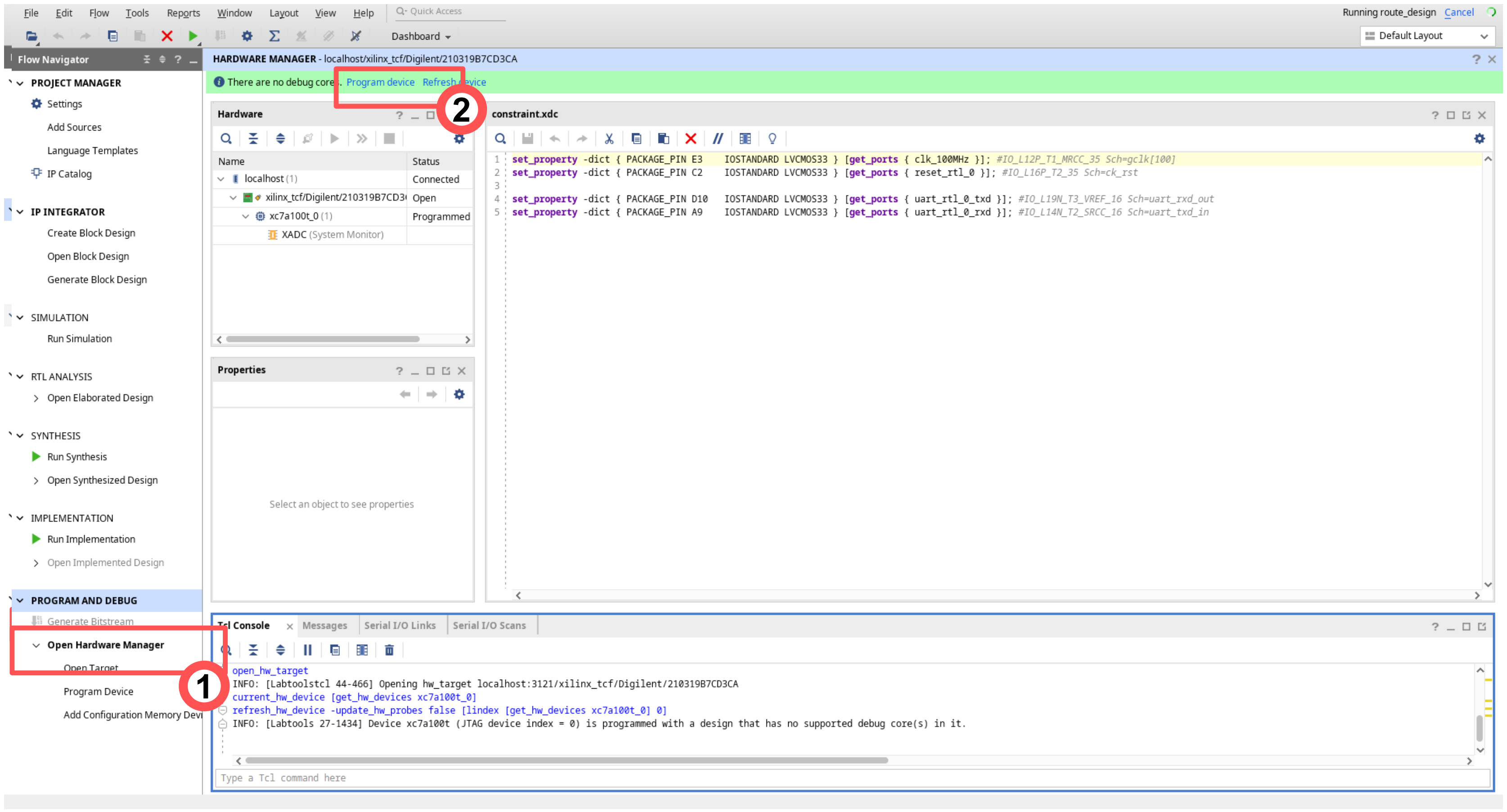
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.558 ns	Worst Hold Slack (WHS):	0.023 ns	Worst Pulse Width Slack (WPWS):	7.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	12714	Total Number of Endpoints:	12714	Total Number of Endpoints:	4269

All user specified timing constraints are met.

SoC Integration (17)

- Program FPGA



Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319B7CD3	Open
xc7a100t_0 (1)	Programmed
XADC (System Monitor)	

```
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk_100MHz }]; #IO_L12P_T1_MRCC_35 Sch=gclk[100]
set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { reset_rtl_0 }]; #IO_L16P_T2_35 Sch=ck_rst

set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_txd }]; #IO_L19N_T3_VREF_16 Sch=uart_rxd_out
set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_rxd }]; #IO_L14N_T2_SRCC_16 Sch=uart_txd_in
```

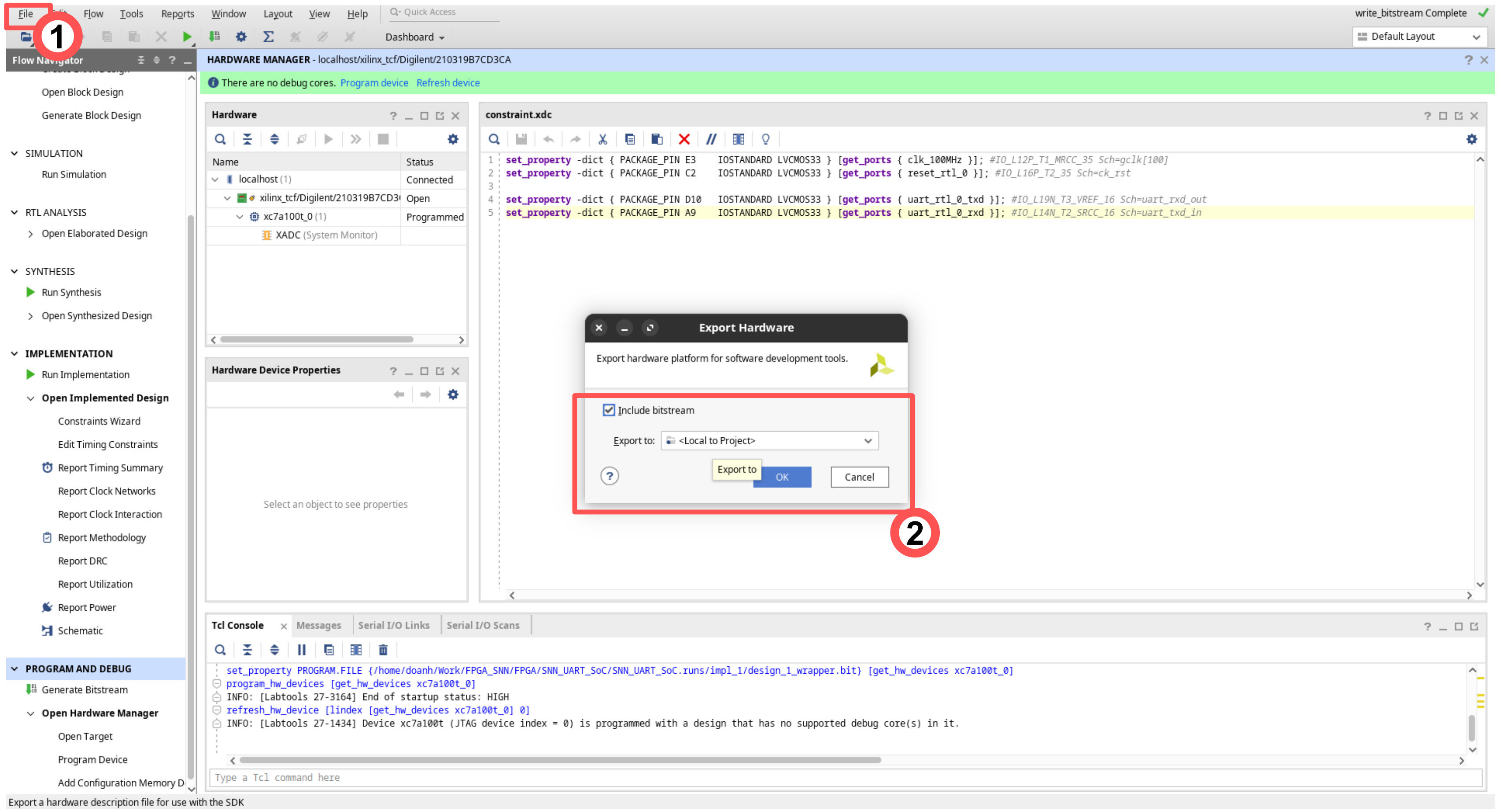
```
open_hw_target
INFO: [Labtoolstcl 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digilent/210319B7CD3CA
current_hw_device [get_hw_devices xc7a100t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a100t_0] 0]
INFO: [Labtools 27-1434] Device xc7a100t (JTAG device index = 0) is programmed with a design that has no supported debug core(s) in it.
```

Overview

1. Package SNN IP
2. SoC Integration with MicroBlaze
3. Software Implementation
4. Flash memory configuration

Software Implementation (1)

- Export Hardware platform for Software DevKit

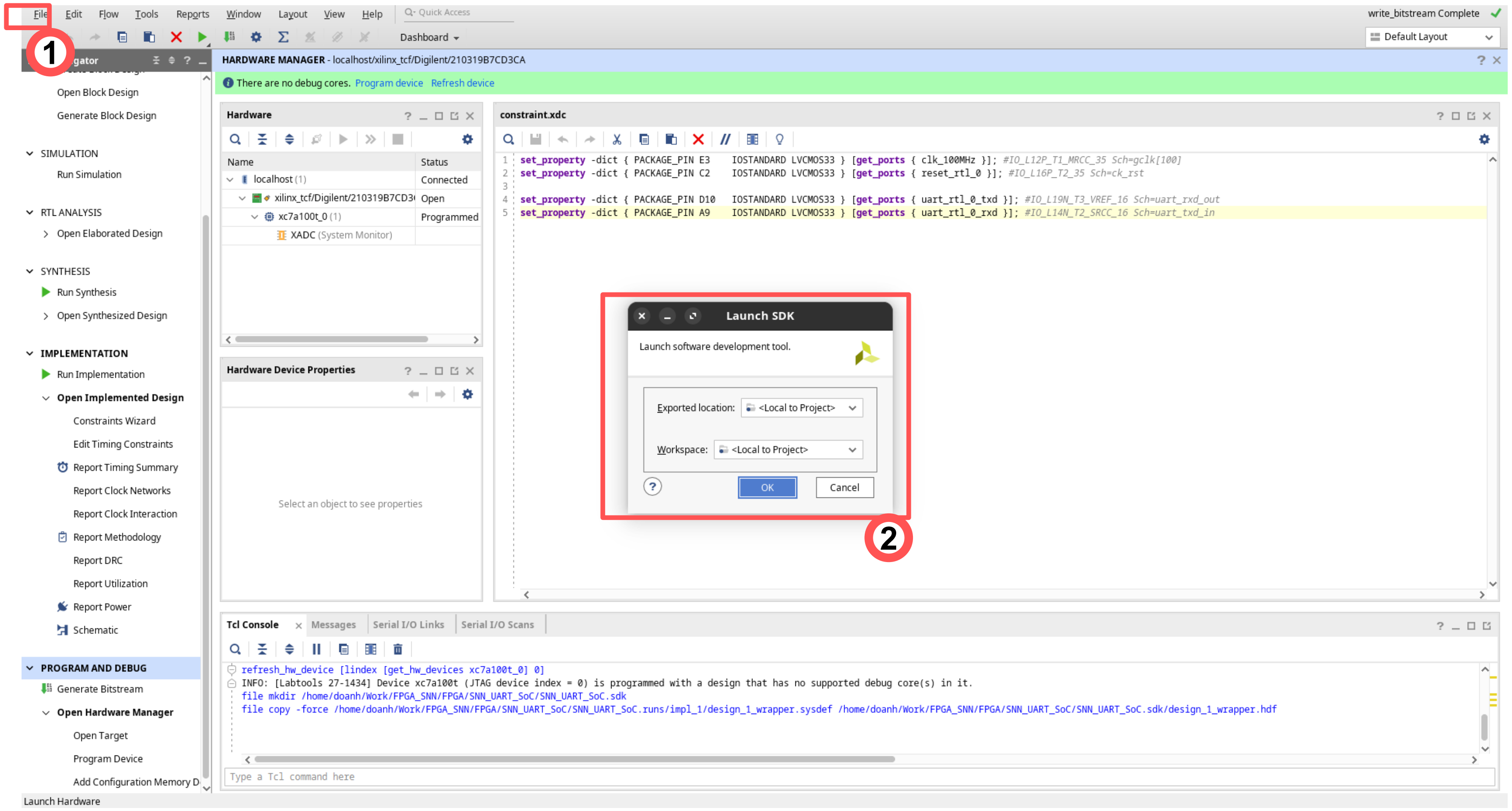


```
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk_100MHz }]; #IO_L12P_T1_MRCC_35 Sch=gclk[100]
set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { reset_rtl_0 }]; #IO_L16P_T2_35 Sch=ck_rst

set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_txd }]; #IO_L19N_T3_VREF_16 Sch=uart_rxd_out
set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_rxd }]; #IO_L14N_T2_SRCC_16 Sch=uart_txd_in
```

```
set_property PROGRAM.FILE {/home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/impl_1/design_1_wrapper.bit} [get_hw_devices xc7a100t_0]
program_hw_devices [get_hw_devices xc7a100t_0]
INFO: [Labtools 27-3164] End of startup status: HIGH
refresh_hw_device [lindex [get_hw_devices xc7a100t_0] 0]
INFO: [Labtools 27-1434] Device xc7a100t (JTAG device index = 0) is programmed with a design that has no supported debug core(s) in it.
```

Software Implementation (2)

- Launch SDK (File/Launch SDK)



Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319B7CD3	Open
xc7a100t_0 (1)	Programmed
XADC (System Monitor)	

```
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk_100MHz }]; #IO_L12P_T1_MRCC_35 Sch=gclk[100]
set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { reset_rtl_0 }]; #IO_L16P_T2_35 Sch=ck_rst

set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_txd }]; #IO_L19N_T3_VREF_16 Sch=uart_rxd_out
set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports { uart_rtl_0_rxd }]; #IO_L14N_T2_SRCC_16 Sch=uart_txd_in
```

Launch SDK

Launch software development tool.

Exported location: <Local to Project>

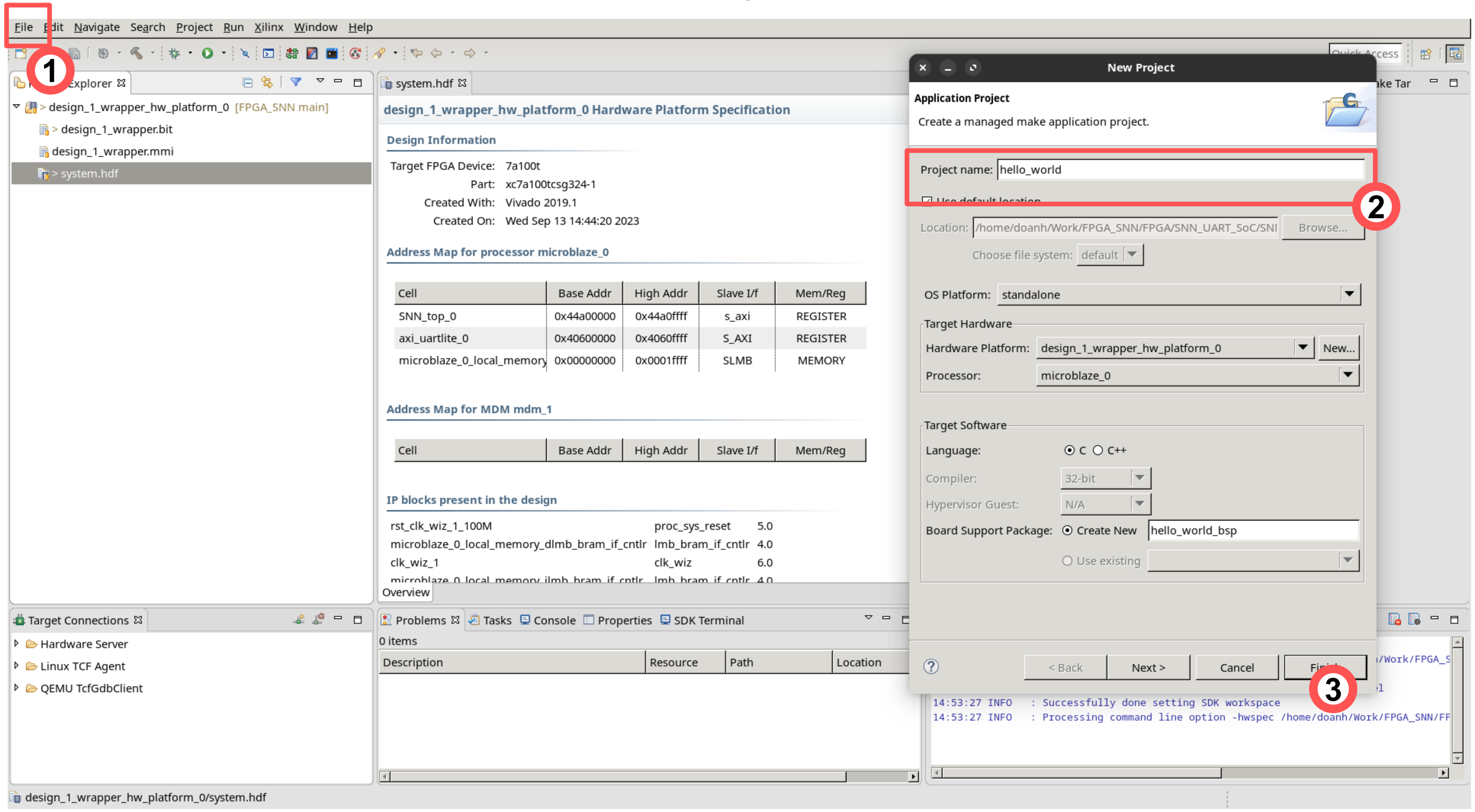
Workspace: <Local to Project>

OK Cancel

```
refresh_hw_device [lindex [get_hw_devices xc7a100t_0] 0]
INFO: [Labtools 27-1434] Device xc7a100t (JTAG device index = 0) is programmed with a design that has no supported debug core(s) in it.
file mkdir /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.sdk
file copy -force /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.runs/impl_1/design_1_wrapper.sysdef /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNN_UART_SoC.sdk/design_1_wrapper.hdf
```

Software Implementation (3)

- Create a new Application Project in SDK



File Edit Navigate Search Project Run Xilinx Window Help

design_1_wrapper_hw_platform_0 [FPGA_SNN main]
- design_1_wrapper.bit
- design_1_wrapper.mmi
- system.hdf

system.hdf

design_1_wrapper_hw_platform_0 Hardware Platform Specification

Design Information

Target FPGA Device: 7a100t
Part: xc7a100tcsg324-1
Created With: Vivado 2019.1
Created On: Wed Sep 13 14:44:20 2023

Address Map for processor microblaze_0

Cell	Base Addr	High Addr	Slave I/f	Mem/Reg
SNN_top_0	0x44a00000	0x44a0ffff	s_axi	REGISTER
axi_uartlite_0	0x40600000	0x4060ffff	S_AXI	REGISTER
microblaze_0_local_memory	0x00000000	0x0001ffff	SLMB	MEMORY

Address Map for MDM mdm_1

Cell	Base Addr	High Addr	Slave I/f	Mem/Reg

IP blocks present in the design

rst_clk_wiz_1_100M	proc_sys_reset	5.0
microblaze_0_local_memory_dlmb_bram_if_cntlr	lmb_bram_if_cntlr	4.0
clk_wiz_1	clk_wiz	6.0
microblaze_0_local_memory_ilmb_bram_if_cntlr	lmb_bram_if_cntlr	4.0

Target Connections
- Hardware Server
- Linux TCF Agent
- QEMU TcfGdbClient

Problems Tasks Console Properties SDK Terminal

0 items

Description	Resource	Path	Location

New Project

Application Project

Create a managed make application project.

Project name: hello_world

Use default location

Location: /home/doanh/Work/FPGA_SNN/FPGA/SNN_UART_SoC/SNI

Choose file system: default

OS Platform: standalone

Target Hardware

Hardware Platform: design_1_wrapper_hw_platform_0

Processor: microblaze_0

Target Software

Language: [✓ C] C++

Compiler: 32-bit

Hypervisor Guest: N/A

Board Support Package: [✓ Create New] hello_world_bsp

Use existing

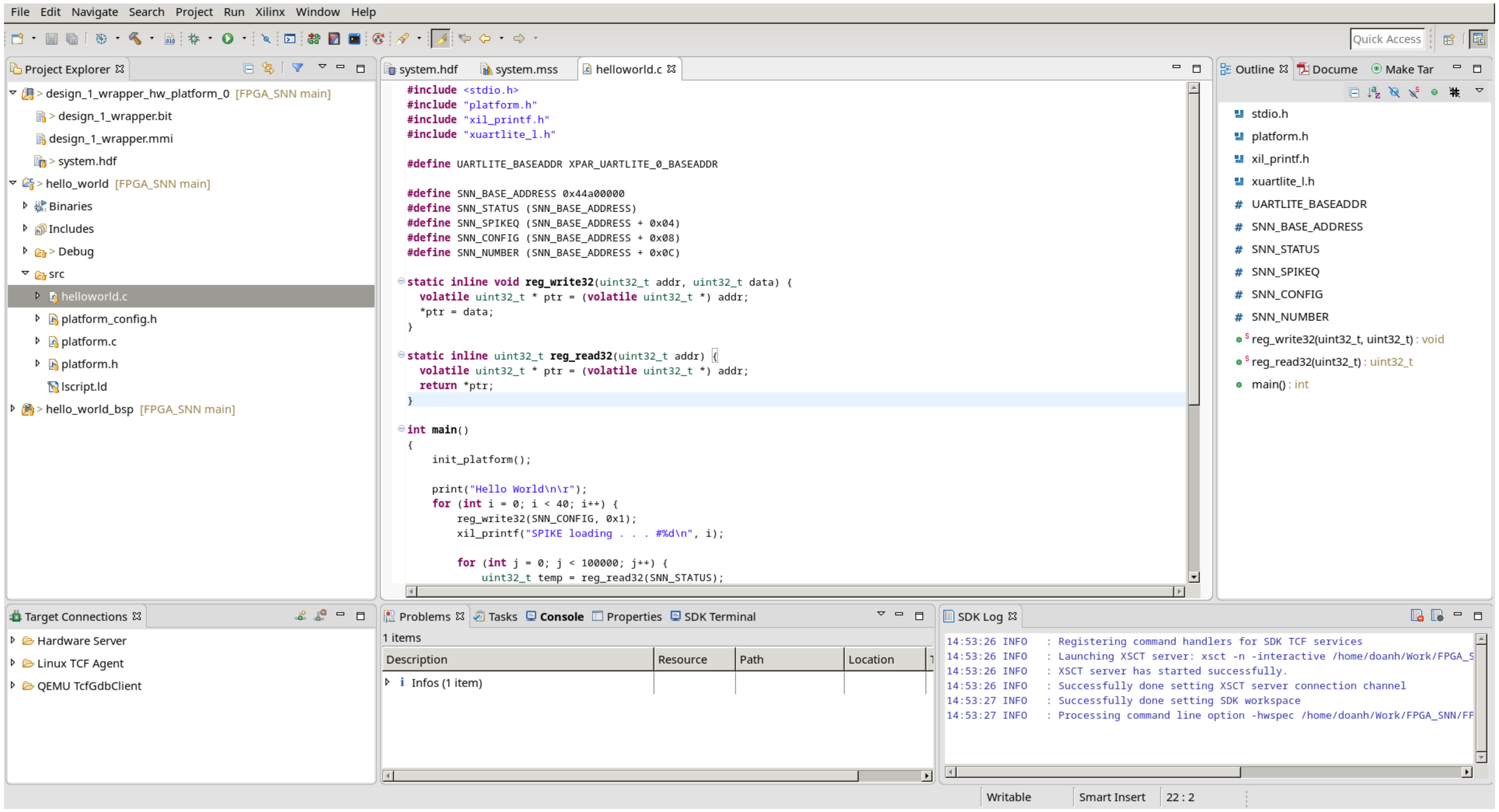
< Back Next > Cancel Finish

```
14:53:27 INFO   : Successfully done setting SDK workspace
14:53:27 INFO   : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
```

design_1_wrapper_hw_platform_0/system.hdf

Software Implementation (4)

- Make your own software



```c
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xuartlite_l.h"

#define UARTLITE_BASEADDR XPAR_UARTLITE_0_BASEADDR

#define SNN_BASE_ADDRESS 0x44a00000
#define SNN_STATUS (SNN_BASE_ADDRESS)
#define SNN_SPIKEQ (SNN_BASE_ADDRESS + 0x04)
#define SNN_CONFIG (SNN_BASE_ADDRESS + 0x08)
#define SNN_NUMBER (SNN_BASE_ADDRESS + 0x0C)

static inline void reg_write32(uint32_t addr, uint32_t data) {
  volatile uint32_t * ptr = (volatile uint32_t *) addr;
  *ptr = data;
}

static inline uint32_t reg_read32(uint32_t addr) {
  volatile uint32_t * ptr = (volatile uint32_t *) addr;
  return *ptr;
}

int main()
{
    init_platform();

    print("Hello World\n\r");
    for (int i = 0; i < 40; i++) {
        reg_write32(SNN_CONFIG, 0x1);
        xil_printf("SPIKE loading . . . #%d\n", i);

        for (int j = 0; j < 100000; j++) {
            uint32_t temp = reg_read32(SNN_STATUS);
```

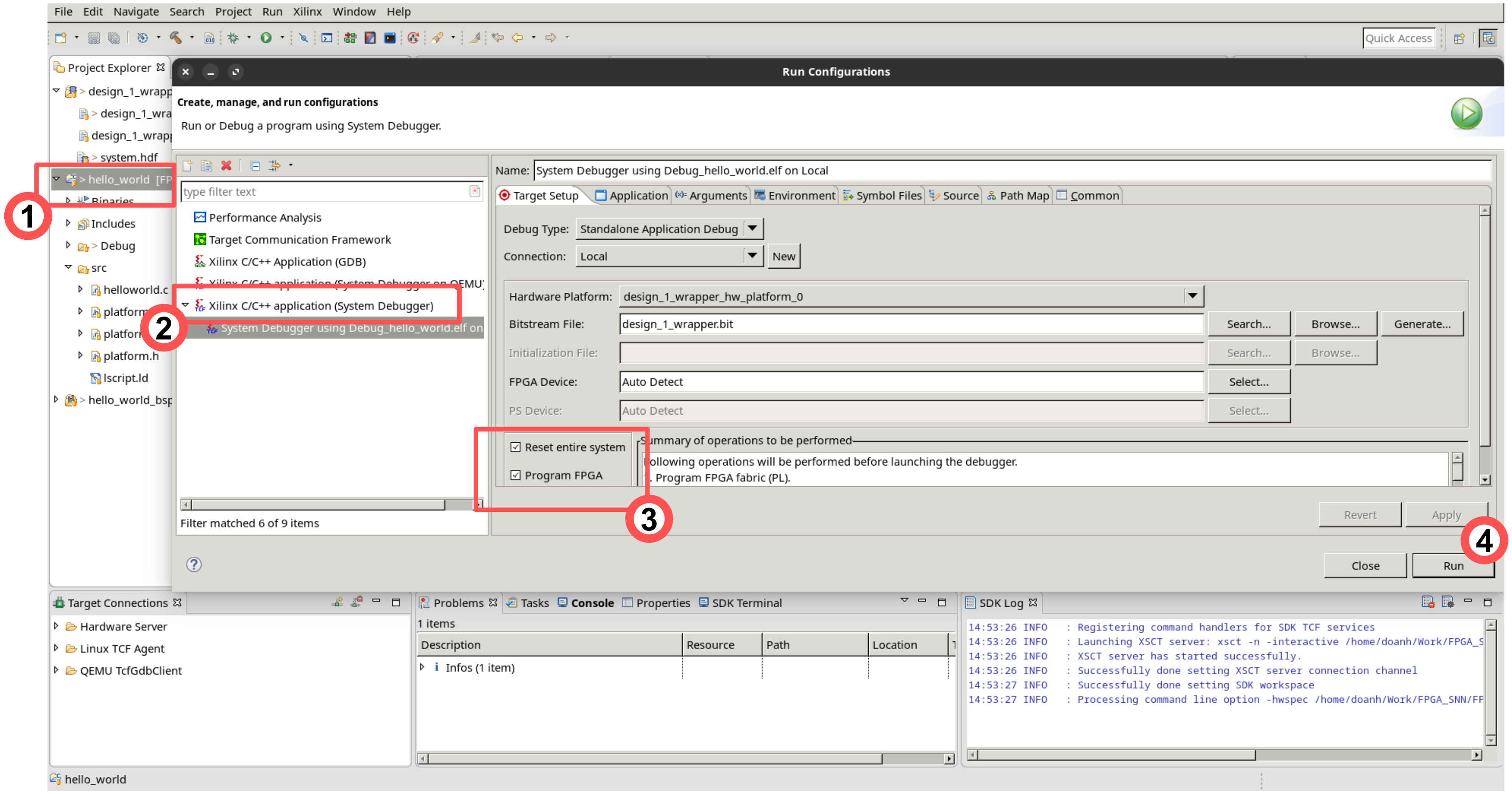
1 items

Description	Resource	Path	Location
Infos (1 item)			

```
14:53:26 INFO   : Registering command handlers for SDK TCF services
14:53:26 INFO   : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
14:53:26 INFO   : XSCT server has started successfully.
14:53:26 INFO   : Successfully done setting XSCT server connection channel
14:53:27 INFO   : Successfully done setting SDK workspace
14:53:27 INFO   : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
```

Software Implementation (5)

- Build & Run Configuration & Apply & Run



Create, manage, and run configurations

Run or Debug a program using System Debugger.

Name: System Debugger using Debug_hello_world.elf on Local

Debug Type: Standalone Application Debug

Connection: Local

Hardware Platform: design_1_wrapper_hw_platform_0

Bitstream File: design_1_wrapper.bit

Initialization File:

FPGA Device: Auto Detect

PS Device: Auto Detect

☑ Reset entire system

☑ Program FPGA

Summary of operations to be performed

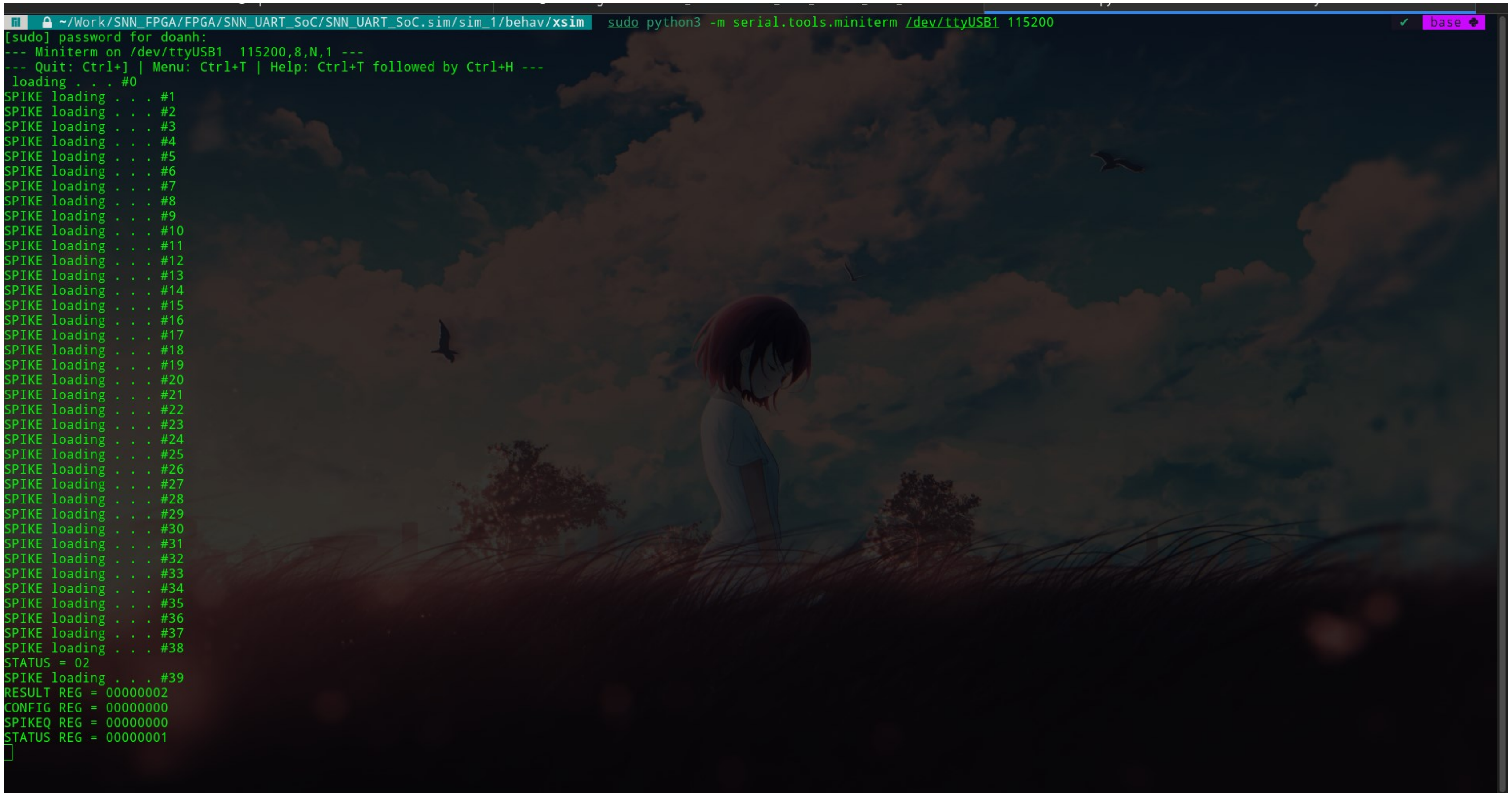
Following operations will be performed before launching the debugger.

1. Program FPGA fabric (PL).

```
14:53:26 INFO  : Registering command handlers for SDK TCF services
14:53:26 INFO  : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
14:53:26 INFO  : XSCT server has started successfully.
14:53:26 INFO  : Successfully done setting XSCT server connection channel
14:53:27 INFO  : Successfully done setting SDK workspace
14:53:27 INFO  : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
```

Software Implementation (6)

- Using UART display with baud rate of 115200



```
~/Work/SNN_FPGA/FPGA/SNN_UART_SoC/SNN_UART_SoC.sim/sim_1/behav/xsim  sudo python3 -m serial.tools.miniterm /dev/ttyUSB1 115200
[sudo] password for doanh:
--- Miniterm on /dev/ttyUSB1  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
 loading . . . #0
SPIKE loading . . . #1
SPIKE loading . . . #2
SPIKE loading . . . #3
SPIKE loading . . . #4
SPIKE loading . . . #5
SPIKE loading . . . #6
SPIKE loading . . . #7
SPIKE loading . . . #8
SPIKE loading . . . #9
SPIKE loading . . . #10
SPIKE loading . . . #11
SPIKE loading . . . #12
SPIKE loading . . . #13
SPIKE loading . . . #14
SPIKE loading . . . #15
SPIKE loading . . . #16
SPIKE loading . . . #17
SPIKE loading . . . #18
SPIKE loading . . . #19
SPIKE loading . . . #20
SPIKE loading . . . #21
SPIKE loading . . . #22
SPIKE loading . . . #23
SPIKE loading . . . #24
SPIKE loading . . . #25
SPIKE loading . . . #26
SPIKE loading . . . #27
SPIKE loading . . . #28
SPIKE loading . . . #29
SPIKE loading . . . #30
SPIKE loading . . . #31
SPIKE loading . . . #32
SPIKE loading . . . #33
SPIKE loading . . . #34
SPIKE loading . . . #35
SPIKE loading . . . #36
SPIKE loading . . . #37
SPIKE loading . . . #38
STATUS = 02
SPIKE loading . . . #39
RESULT REG = 00000002
CONFIG REG = 00000000
SPIKEQ REG = 00000000
STATUS REG = 00000001
```

Overview

1. Package SNN IP
2. SoC Integration with MicroBlaze
3. Software Implementation
4. Flash memory configuration

Flash memory configuration (1)

1. Generate a bitstream from block diagram in Vivado and export the hardware definition file (system.HDF)

2. Open SDK and define a new application project called app.ELF, based on system.HDF which was just exported from Vivado. Create it as a blank project.

3. Generate linker Script and ensure that all of the *'Section to Memory Region Mapping'* sections are loading from DDR.

4. Place all C application files in the src directory and build the .ELF (Default behaviour is automatic build).

5. Still in SDK and using the *xcst console*, enter the following command.

```
mb-objcopy -O srec app.elf app.srec
```

- This simply takes the app.ELF and converts it to app.SREC. This SREC format is a bootable format to store in flash

6. Create a new application project, using the same .HDF. This will be a *'SREC SPI Bootloader'* application project.

7. Generate linker script for this SREC Bootloader and ensure that all of the sections load from ilmb (BRAM).

8. Navigate to the blconfig.h file in the src directory for the SREC bootloader and set a value for FLASH_IMAGE_BASEADDR.

- The value which you set should be as follows.

FLASH_IMAGE_BASEADDR = Flash_base_address + OFFSET.

- The Flash_base_address is viewable in your linker script for your app.ELF.

9. Go back to Vivado and in your block diagram, associate the app.ELF file with the design. Ensure that you update your associated ELF file so that it is using your app.ELF rather than mb_bootloop_le.elf.

10. Re-generate the bitstream. This will initialise your BRAM with the app bootloader. This should produce a fresh system.bit file.

11. Generate the memory configuration files (app.mcs) for programming to your Flash device.

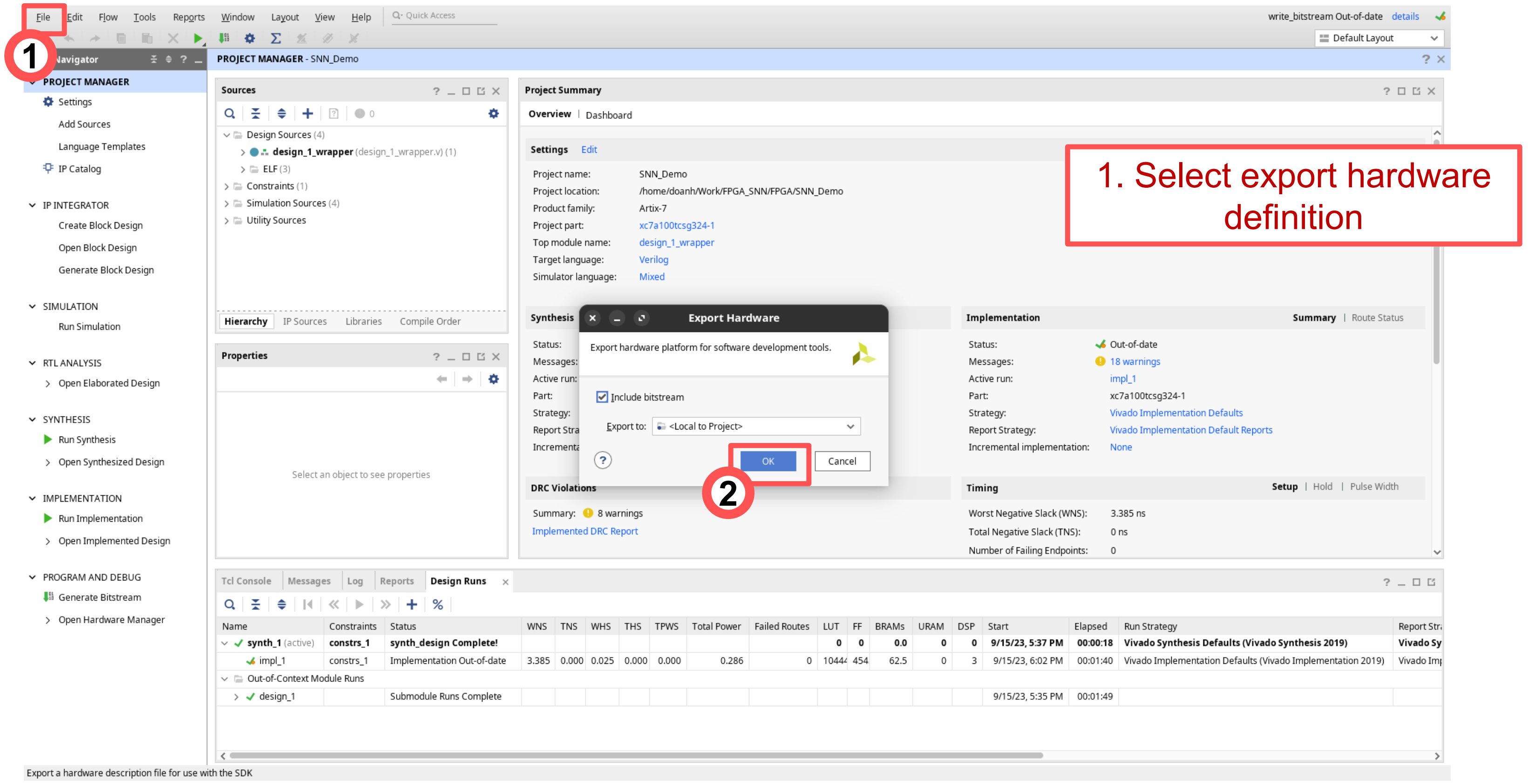
data_files = app.SREC ; bit_files = system.bit

12. Program the configuration device with app.mcs and specify the address range to be *'Entire configuration memory device'.*

13. Once this has finished, you will need to right click on your device in hardware manager and click *'Boot from configuration memory device'.*

Flash memory configuration (2)

- Generate a bitstream from block diagram in Vivado and export the hardware definition file (system.HDF)



Project name: SNN_Demo
Project location: /home/doanh/Work/FPGA_SNN/FPGA/SNN_Demo
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: design_1_wrapper
Target language: Verilog
Simulator language: Mixed

Synthesis
Status:
Messages:
Active run:
Part:
Strategy:
Report Stra
Incrementa

Implementation
Status: Out-of-date
Messages: 18 warnings
Active run: impl_1
Part: xc7a100tcsg324-1
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports
Incremental implementation: None

DRC Violations
Summary: 8 warnings
Implemented DRC Report

Timing
Worst Negative Slack (WNS): 3.385 ns
Total Negative Slack (TNS): 0 ns
Number of Failing Endpoints: 0

Export Hardware
Export hardware platform for software development tools.
[✓] Include bitstream
Export to: <Local to Project>

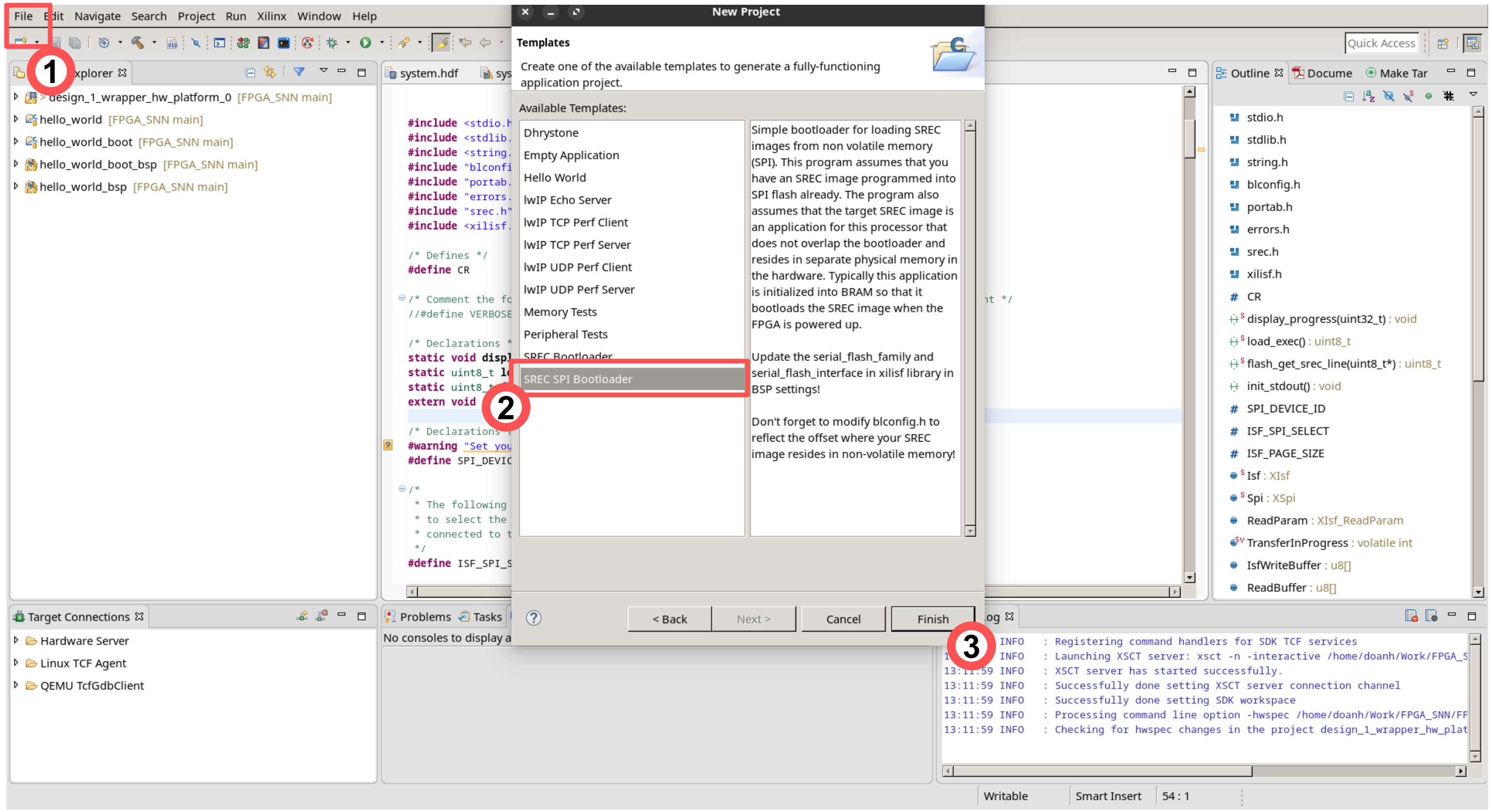
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Stra
synth_1 (active)	constrs_1	synth_design Complete!								0	0	0.0	0	0	9/15/23, 5:37 PM	00:00:18	Vivado Synthesis Defaults (Vivado Synthesis 2019)	Vivado Sy
impl_1	constrs_1	Implementation Out-of-date	3.385	0.000	0.025	0.000	0.000	0.286	0	1044	454	62.5	0	3	9/15/23, 6:02 PM	00:01:40	Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Im
Out-of-Context Module Runs																		
design_1		Submodule Runs Complete													9/15/23, 5:35 PM	00:01:49		

Export a hardware description file for use with the SDK

1. Select export hardware definition

Flash memory configuration (3)

- Launch SDK & create new SREC SPI bootloader app.



New Project

Templates

Create one of the available templates to generate a fully-functioning application project.

Available Templates:

- Dhrystone
- Empty Application
- Hello World
- lwIP Echo Server
- lwIP TCP Perf Client
- lwIP TCP Perf Server
- lwIP UDP Perf Client
- lwIP UDP Perf Server
- Memory Tests
- Peripheral Tests
- SREC Bootloader
- SREC SPI Bootloader

Simple bootloader for loading SREC images from non volatile memory (SPI). This program assumes that you have an SREC image programmed into SPI flash already. The program also assumes that the target SREC image is an application for this processor that does not overlap the bootloader and resides in separate physical memory in the hardware. Typically this application is initialized into BRAM so that it bootloads the SREC image when the FPGA is powered up.

Update the serial_flash_family and serial_flash_interface in xilisf library in BSP settings!

Don't forget to modify blconfig.h to reflect the offset where your SREC image resides in non-volatile memory!

< Back | Next > | Cancel | Finish

Project Explorer:

```
design_1_wrapper_hw_platform_0 [FPGA_SNN main]
hello_world [FPGA_SNN main]
hello_world_boot [FPGA_SNN main]
hello_world_boot_bsp [FPGA_SNN main]
hello_world_bsp [FPGA_SNN main]
```

```
#include <stdio.h
#include <stdlib.
#include <string.
#include "blconfi
#include "portab.
#include "errors.
#include "srec.h"
#include <xilisf.

/* Defines */
#define CR

/* Comment the fo
//#define VERBOSE

/* Declarations *
static void displ
static uint8_t lo
static uint8_t f
extern void

/* Declarations
#warning "Set you
#define SPI_DEVIC

/*
 * The following
 * to select the
 * connected to t
 */
#define ISF_SPI_S
```

Outline:

```
stdio.h
stdlib.h
string.h
blconfig.h
portab.h
errors.h
srec.h
xilisf.h
CR
display_progress(uint32_t) : void
load_exec() : uint8_t
flash_get_srec_line(uint8_t*) : uint8_t
init_stdout() : void
SPI_DEVICE_ID
ISF_SPI_SELECT
ISF_PAGE_SIZE
Isf : XIsf
Spi : XSpi
ReadParam : XIsf_ReadParam
TransferInProgress : volatile int
IsfWriteBuffer : u8[]
ReadBuffer : u8[]
```

Target Connections:

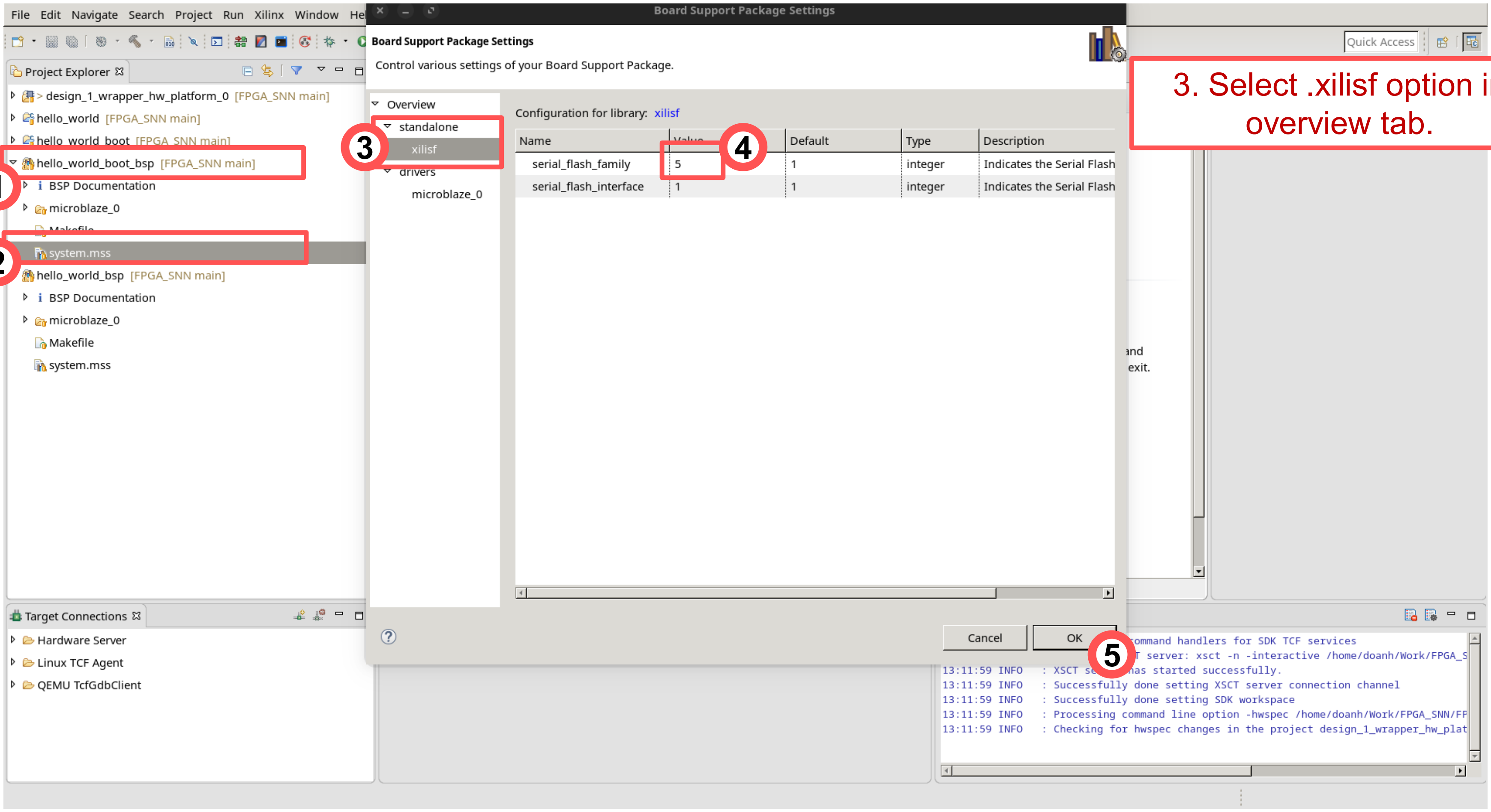
```
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
```

No consoles to display a

```
         INFO  : Registering command handlers for SDK TCF services
13:      INFO  : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
13:11:59 INFO  : XSCT server has started successfully.
13:11:59 INFO  : Successfully done setting XSCT server connection channel
13:11:59 INFO  : Successfully done setting SDK workspace
13:11:59 INFO  : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
13:11:59 INFO  : Checking for hwspec changes in the project design_1_wrapper_hw_plat
```

Flash memory configuration (4)

- Modify BSP settings of SREC SPI bootloader app.



Board Support Package Settings

Control various settings of your Board Support Package.

Configuration for library: xilisf

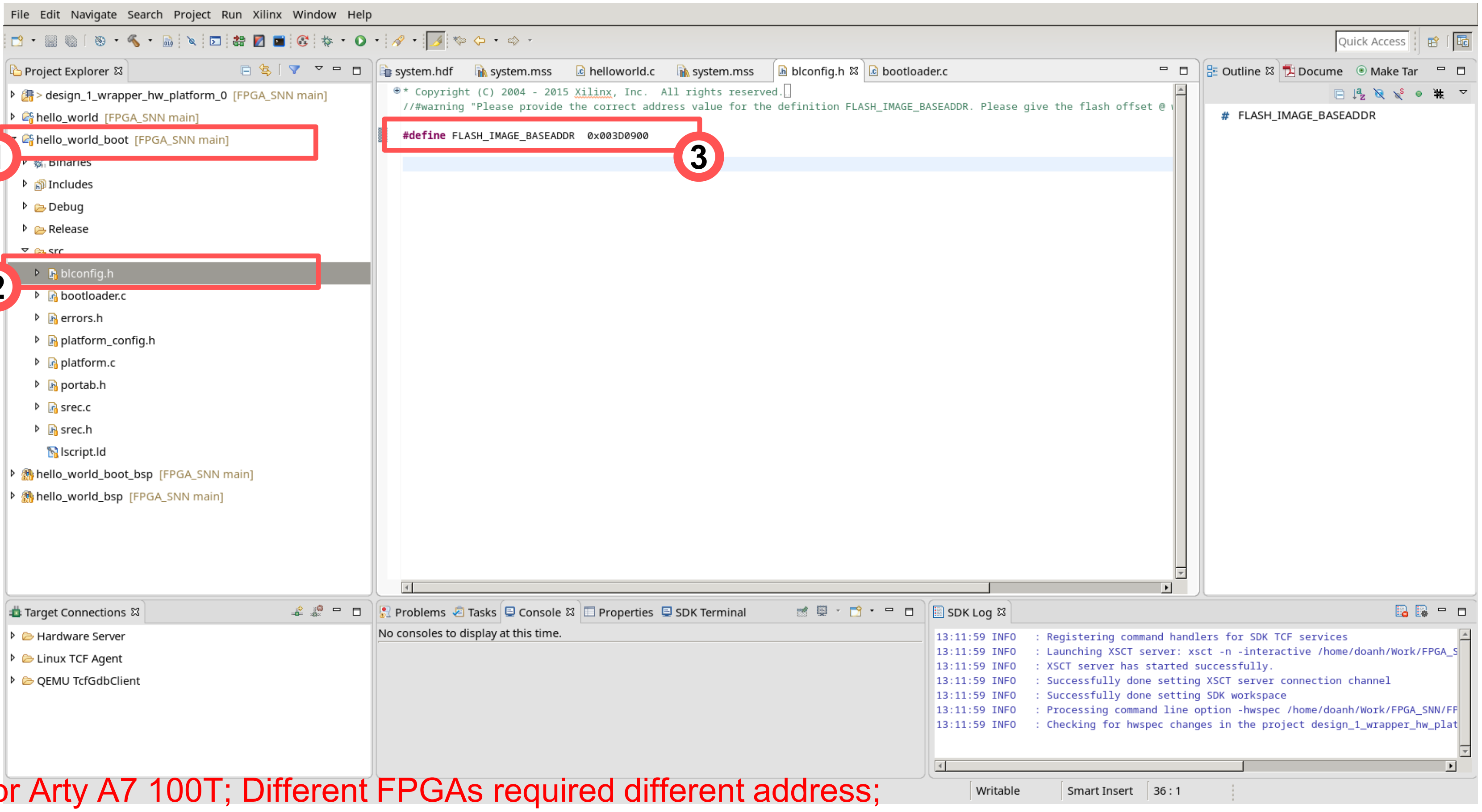
Name	Value	Default	Type	Description
serial_flash_family	5	1	integer	Indicates the Serial Flash
serial_flash_interface	1	1	integer	Indicates the Serial Flash

```
13:11:59 INFO  : XSCT server has started successfully.
13:11:59 INFO  : Successfully done setting XSCT server connection channel
13:11:59 INFO  : Successfully done setting SDK workspace
13:11:59 INFO  : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
13:11:59 INFO  : Checking for hwspec changes in the project design_1_wrapper_hw_plat
```

1
2
3
4
5
3. Select .xilisf option in overview tab.
Project Explorer
design_1_wrapper_hw_platform_0 [FPGA_SNN main]
hello_world [FPGA_SNN main]
hello_world_boot [FPGA_SNN main]
hello_world_boot_bsp [FPGA_SNN main]
BSP Documentation
microblaze_0
Makefile
system.mss
hello_world_bsp [FPGA_SNN main]
Overview
standalone
xilisf
drivers
microblaze_0
Target Connections
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
Cancel
OK

Flash memory configuration (5)

- Change FLASH_IMAGE_BASEADDR = 0x003D0900 *



```
* Copyright (C) 2004 - 2015 Xilinx, Inc.  All rights reserved.
//#warning "Please provide the correct address value for the definition FLASH_IMAGE_BASEADDR. Please give the flash offset @ 

#define FLASH_IMAGE_BASEADDR  0x003D0900
```

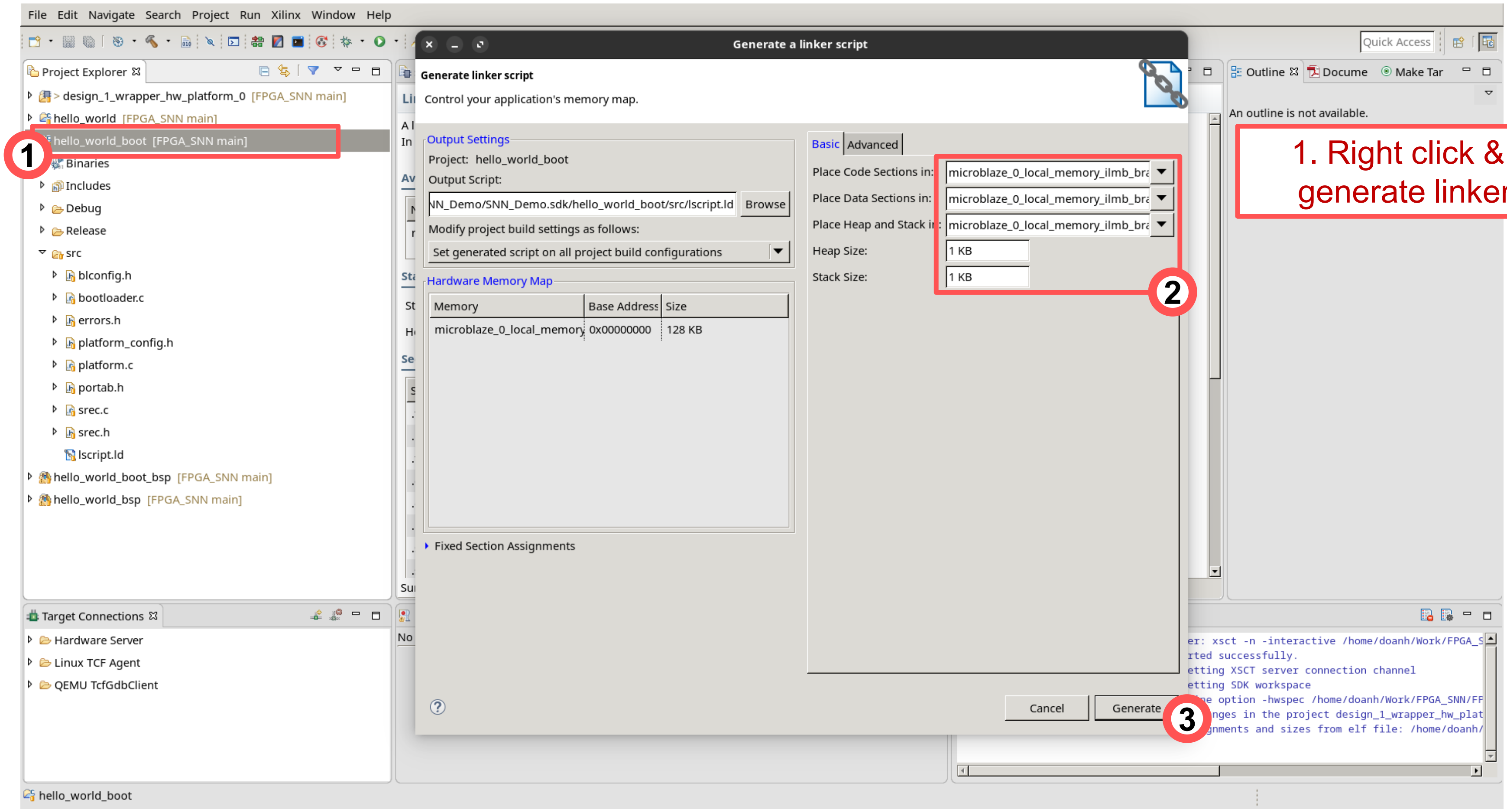
```
No consoles to display at this time.
```

```
13:11:59 INFO   : Registering command handlers for SDK TCF services
13:11:59 INFO   : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
13:11:59 INFO   : XSCT server has started successfully.
13:11:59 INFO   : Successfully done setting XSCT server connection channel
13:11:59 INFO   : Successfully done setting SDK workspace
13:11:59 INFO   : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
13:11:59 INFO   : Checking for hwspec changes in the project design_1_wrapper_hw_plat
```

* Apply for Arty A7 100T; Different FPGAs required different address;

Flash memory configuration (6)

- Generate linker script & assign to ilmb_bram



Generate linker script
Control your application's memory map.

Output Settings
Project: hello_world_boot
Output Script: NN_Demo/SNN_Demo.sdk/hello_world_boot/src/lscript.ld Browse
Modify project build settings as follows: Set generated script on all project build configurations

Hardware Memory Map

Memory	Base Address	Size
microblaze_0_local_memory	0x00000000	128 KB

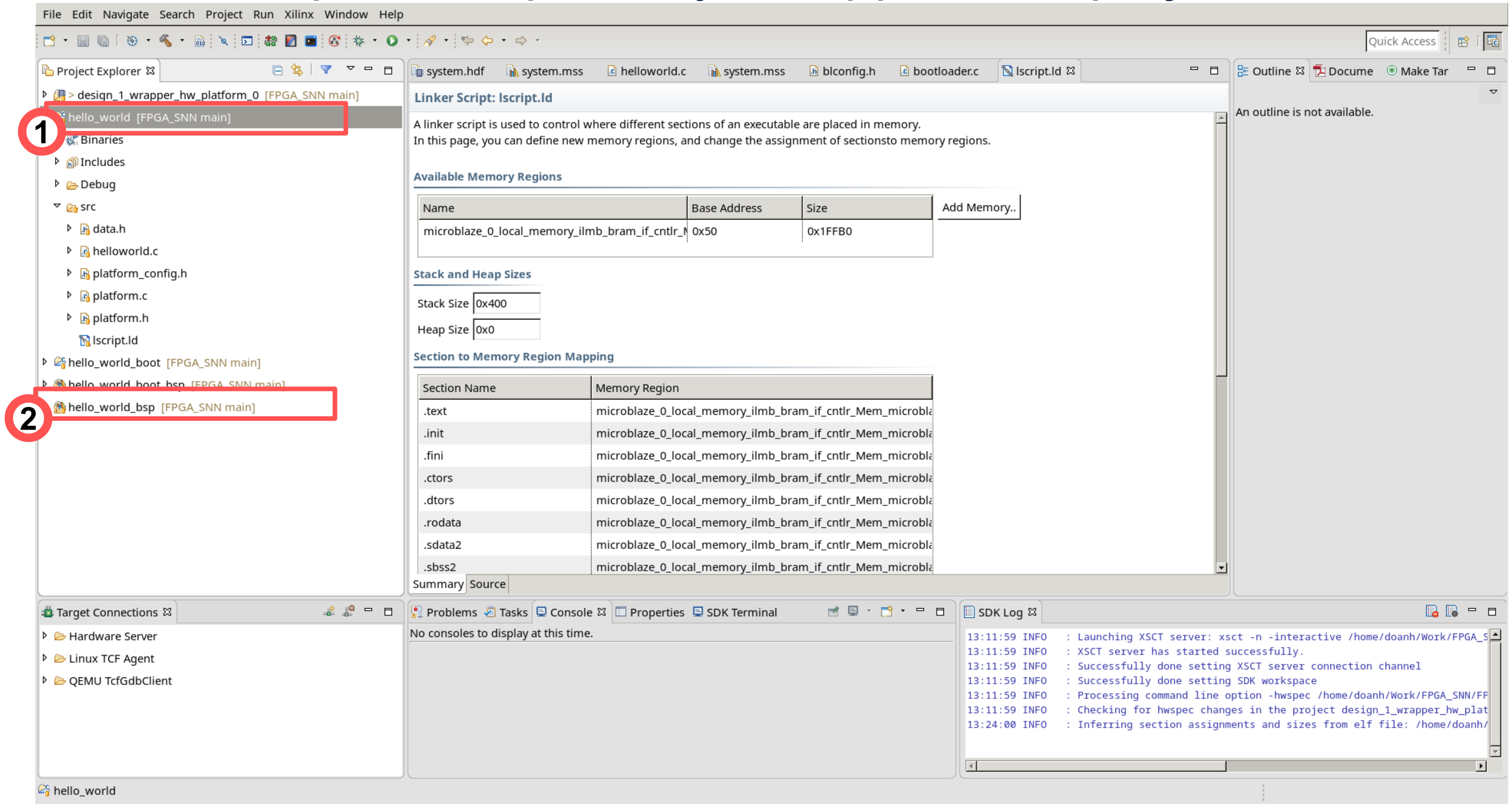
Place Code Sections in: microblaze_0_local_memory_ilmb_bra
Place Data Sections in: microblaze_0_local_memory_ilmb_bra
Place Heap and Stack in: microblaze_0_local_memory_ilmb_bra
Heap Size: 1 KB
Stack Size: 1 KB

```
er: xsct -n -interactive /home/doanh/Work/FPGA_S
rted successfully.
etting XSCT server connection channel
etting SDK workspace
e option -hwspec /home/doanh/Work/FPGA_SNN/FF
nges in the project design_1_wrapper_hw_plat
gnments and sizes from elf file: /home/doanh/
```

1. Right click & Select generate linker script

Flash memory configuration (7)

- Re-do step 4 & step 6 for your application project.



Linker Script: lscript.ld

A linker script is used to control where different sections of an executable are placed in memory.
In this page, you can define new memory regions, and change the assignment of sectionsto memory regions.

Available Memory Regions

Name	Base Address	Size
microblaze_0_local_memory_ilmb_bram_if_cntlr_M	0x50	0x1FFB0

Add Memory..

Stack and Heap Sizes

Stack Size	0x400
Heap Size	0x0

Section to Memory Region Mapping

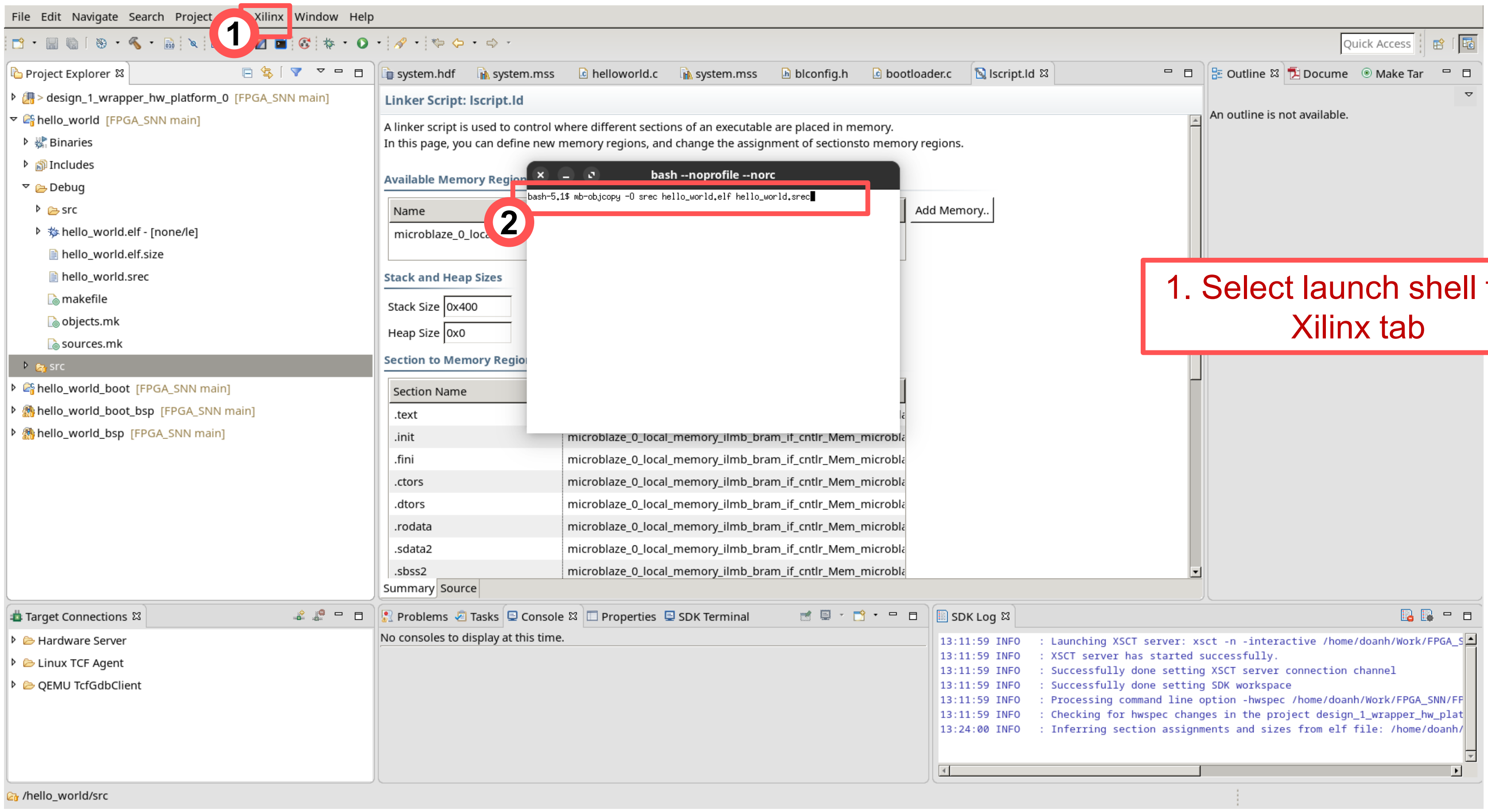
Section Name	Memory Region
.text	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.init	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.fini	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.ctors	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.dtors	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.rodata	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.sdata2	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.sbss2	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla

No consoles to display at this time.

```
13:11:59 INFO  : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
13:11:59 INFO  : XSCT server has started successfully.
13:11:59 INFO  : Successfully done setting XSCT server connection channel
13:11:59 INFO  : Successfully done setting SDK workspace
13:11:59 INFO  : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
13:11:59 INFO  : Checking for hwspec changes in the project design_1_wrapper_hw_plat
13:24:00 INFO  : Inferring section assignments and sizes from elf file: /home/doanh/
```

Flash memory configuration (8)

- Generate .SREC file from .ELF file of your application



Linker Script: lscript.ld

A linker script is used to control where different sections of an executable are placed in memory.
In this page, you can define new memory regions, and change the assignment of sectionsto memory regions.

An outline is not available.

bash --noprofile --norc

```
bash-5.1$ mb-objcopy -O srec hello_world.elf hello_world.srec
```

Section Name	
.text	
.init	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.fini	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.ctors	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.dtors	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.rodata	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.sdata2	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla
.sbss2	microblaze_0_local_memory_ilmb_bram_if_cntlr_Mem_microbla

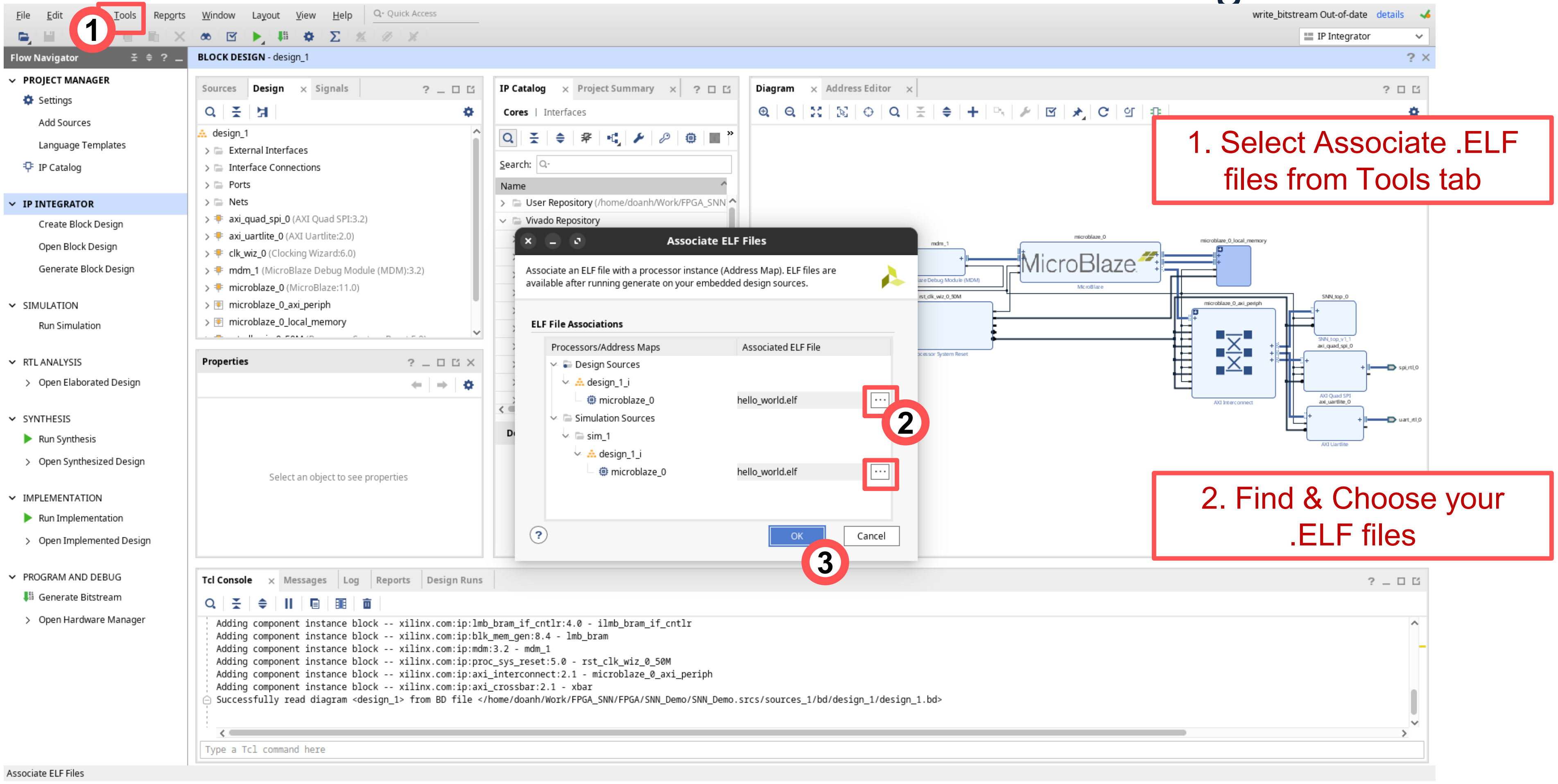
No consoles to display at this time.

```
13:11:59 INFO   : Launching XSCT server: xsct -n -interactive /home/doanh/Work/FPGA_S
13:11:59 INFO   : XSCT server has started successfully.
13:11:59 INFO   : Successfully done setting XSCT server connection channel
13:11:59 INFO   : Successfully done setting SDK workspace
13:11:59 INFO   : Processing command line option -hwspec /home/doanh/Work/FPGA_SNN/FF
13:11:59 INFO   : Checking for hwspec changes in the project design_1_wrapper_hw_plat
13:24:00 INFO   : Inferring section assignments and sizes from elf file: /home/doanh/
```

1. Select launch shell from Xilinx tab

Flash memory configuration (9)

- Go back Vivado & associate .ELF file to block design



1. Select Associate .ELF files from Tools tab

2. Find & Choose your .ELF files

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.

ELF File Associations

Processors/Address Maps	Associated ELF File
Design Sources	
design_1_i	
microblaze_0	hello_world.elf
Simulation Sources	
sim_1	
design_1_i	
microblaze_0	hello_world.elf

OK Cancel

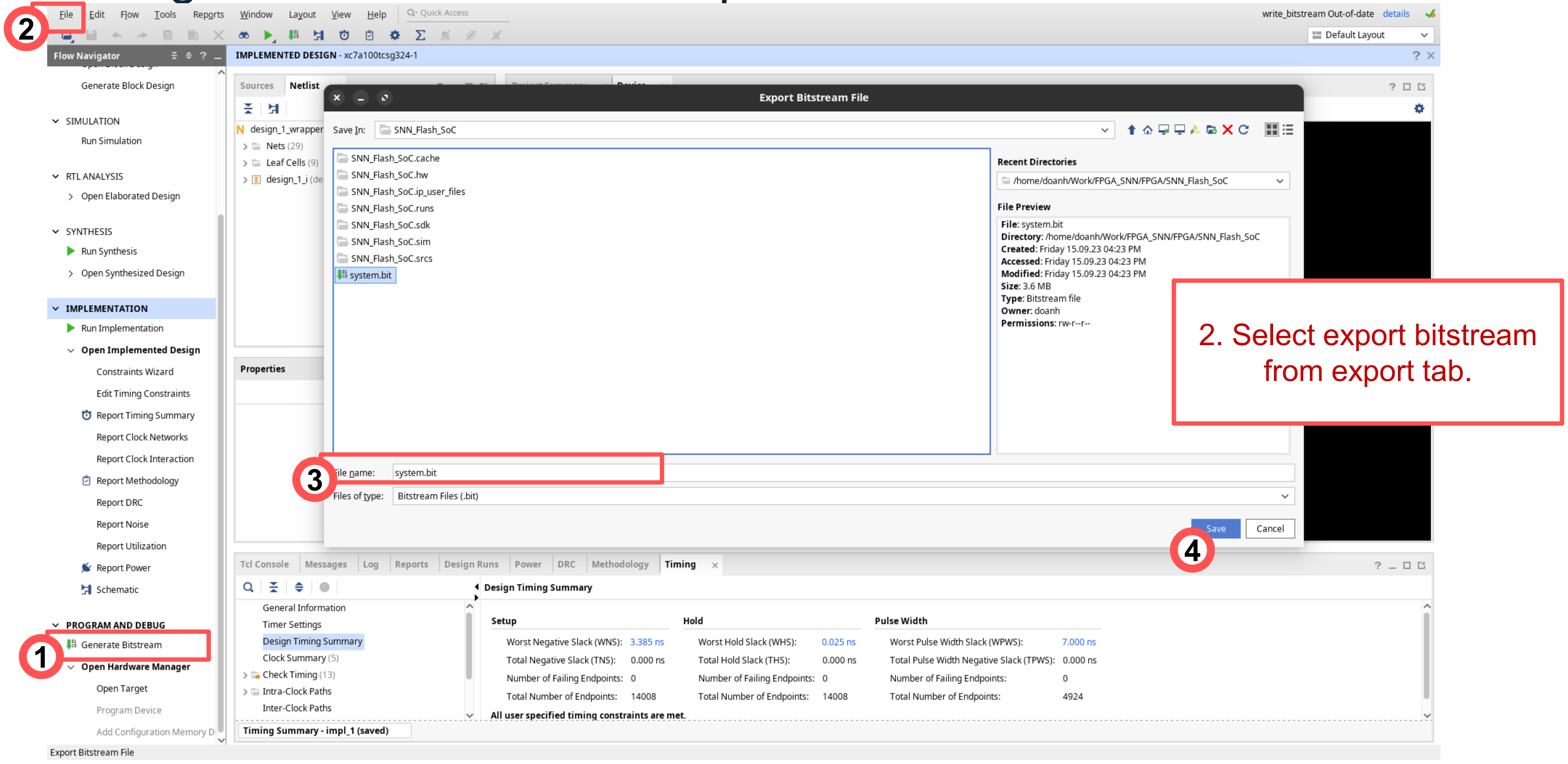
Tcl Console | Messages | Log | Reports | Design Runs

```
Adding component instance block -- xilinx.com:ip:lmb_bram_if_cntlr:4.0 - ilmb_bram_if_cntlr
Adding component instance block -- xilinx.com:ip:blk_mem_gen:8.4 - lmb_bram
Adding component instance block -- xilinx.com:ip:mdm:3.2 - mdm_1
Adding component instance block -- xilinx.com:ip:proc_sys_reset:5.0 - rst_clk_wiz_0_50M
Adding component instance block -- xilinx.com:ip:axi_interconnect:2.1 - microblaze_0_axi_periph
Adding component instance block -- xilinx.com:ip:axi_crossbar:2.1 - xbar
Successfully read diagram <design_1> from BD file </home/doanh/Work/FPGA_SNN/FPGA/SNN_Demo/SNN_Demo.srcs/sources_1/bd/design_1/design_1.bd>
```

Type a Tcl command here

Flash memory configuration (10)

- Re-generate Bitstream & export it



File Preview
File: system.bit
Directory: /home/doanh/Work/FPGA_SNN/FPGA/SNN_Flash_SoC
Created: Friday 15.09.23 04:23 PM
Accessed: Friday 15.09.23 04:23 PM
Modified: Friday 15.09.23 04:23 PM
Size: 3.6 MB
Type: Bitstream file
Owner: doanh
Permissions: rw-r--r--

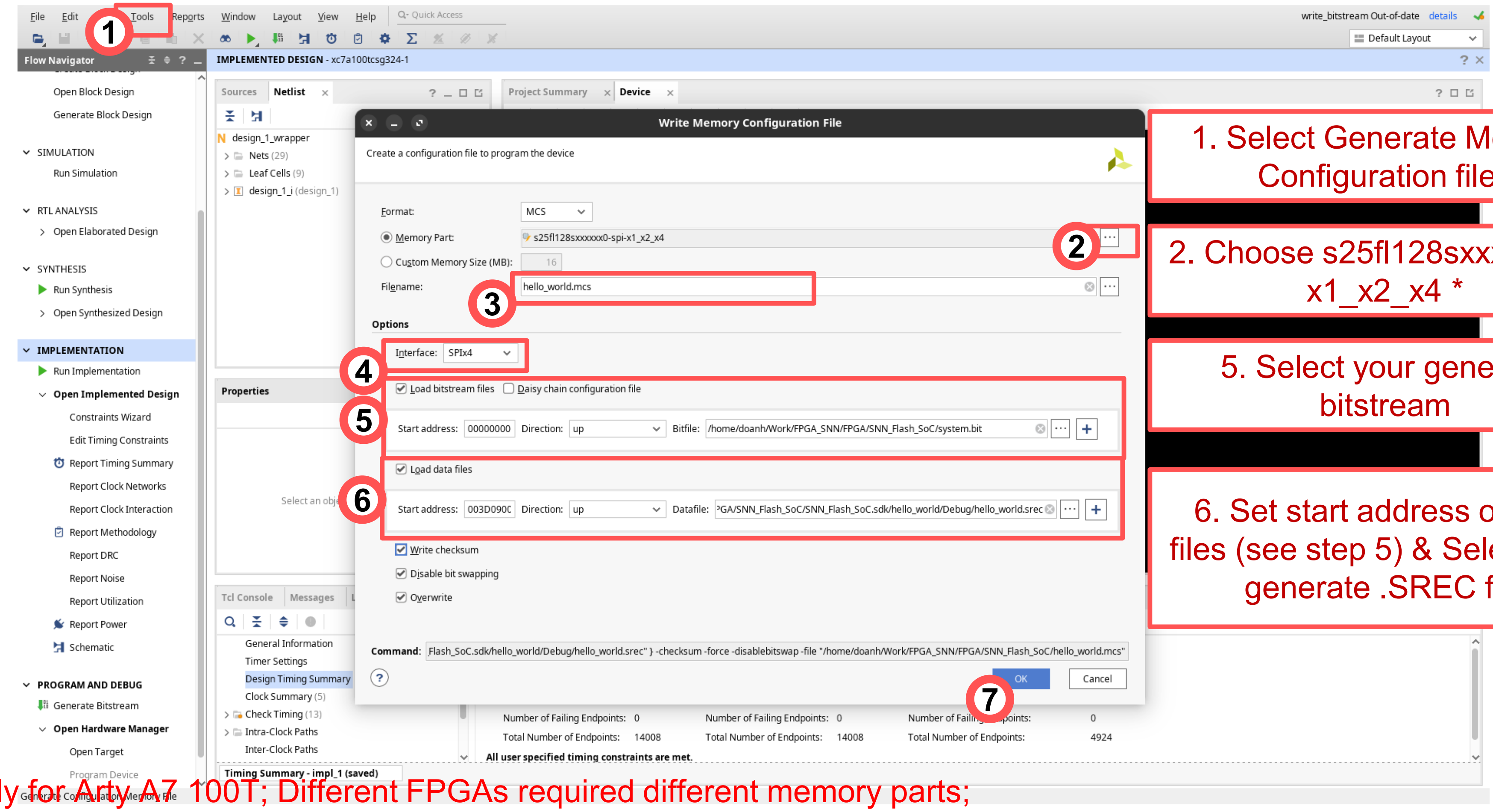
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.385 ns	Worst Hold Slack (WHS):	0.025 ns	Worst Pulse Width Slack (WPWS):	7.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	14008	Total Number of Endpoints:	14008	Total Number of Endpoints:	4924

All user specified timing constraints are met.

Flash memory configuration (11)

- Generate memory configuration file



Write Memory Configuration File

Create a configuration file to program the device

Format: MCS

(●) Memory Part: s25fl128sxxxxxx0-spi-x1_x2_x4

( ) Custom Memory Size (MB): 16

Filename: hello_world.mcs

Options

Interface: SPIx4

[✓] Load bitstream files [ ] Daisy chain configuration file

Start address: 00000000 Direction: up Bitfile: /home/doanh/Work/FPGA_SNN/FPGA/SNN_Flash_SoC/system.bit

[✓] Load data files

Start address: 003D090C Direction: up Datafile: ?GA/SNN_Flash_SoC/SNN_Flash_SoC.sdk/hello_world/Debug/hello_world.srec

[✓] Write checksum

[✓] Disable bit swapping

[✓] Overwrite

```
Command: Flash_SoC.sdk/hello_world/Debug/hello_world.srec" } -checksum -force -disablebitswap -file "/home/doanh/Work/FPGA_SNN/FPGA/SNN_Flash_SoC/hello_world.mcs"
```

1. Select Generate Memory Configuration files

2. Choose s25fl128sxxxx0-spi-x1_x2_x4 *

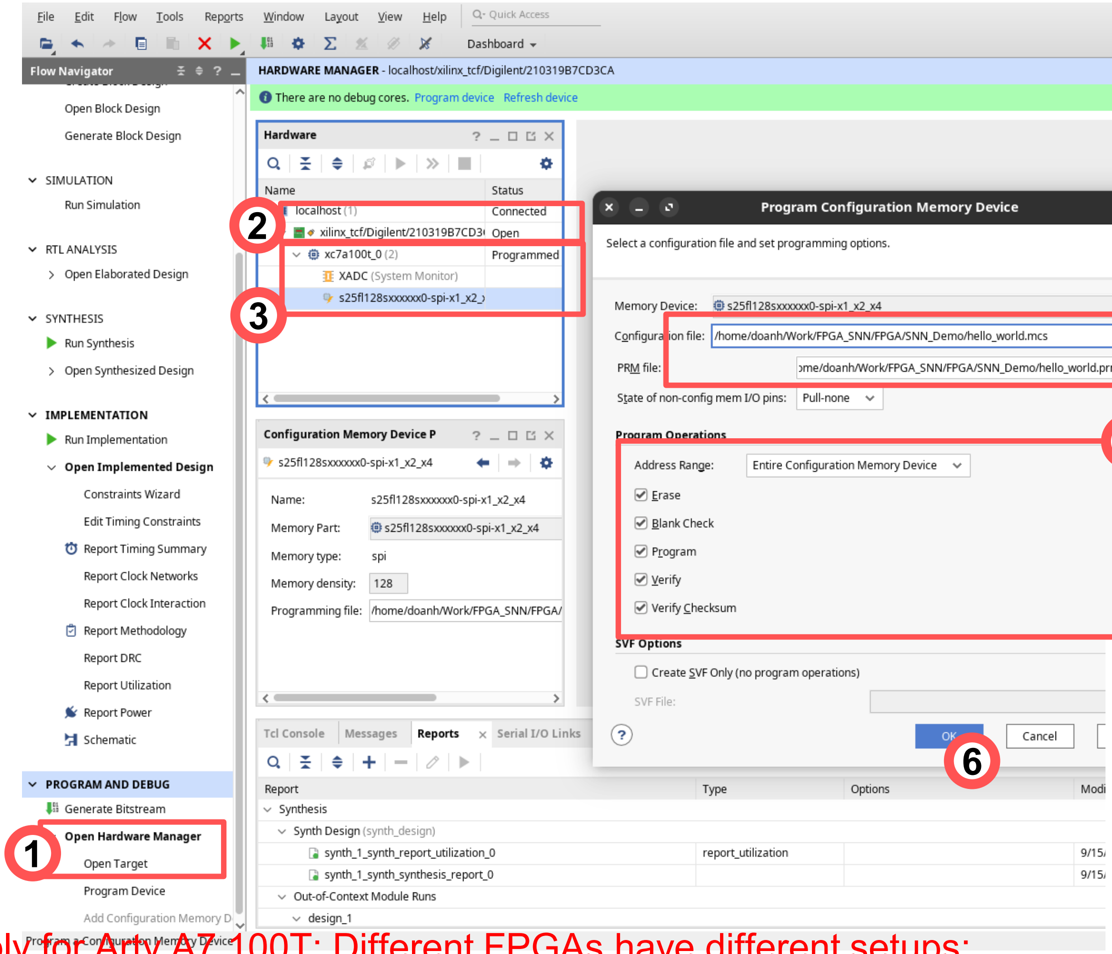
5. Select your generate bitstream

6. Set start address of data files (see step 5) & Select your generate .SREC file

* Apply for Arty A7 100T; Different FPGAs required different memory parts;

Flash memory configuration (12)

- Add configuration memory device & program it



HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210319B7CD3CA

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319B7CD3	Open
xc7a100t_0 (2)	Programmed
XADC (System Monitor)	
s25fl128sxxxxxx0-spi-x1_x2_x4	

Configuration Memory Device P

Name:	s25fl128sxxxxxx0-spi-x1_x2_x4
Memory Part:	s25fl128sxxxxxx0-spi-x1_x2_x4
Memory type:	spi
Memory density:	128
Programming file:	/home/doanh/Work/FPGA_SNN/FPGA/

Program Configuration Memory Device

Select a configuration file and set programming options.

Memory Device: s25fl128sxxxxxx0-spi-x1_x2_x4

Configuration file: /home/doanh/Work/FPGA_SNN/FPGA/SNN_Demo/hello_world.mcs

PRM file: ome/doanh/Work/FPGA_SNN/FPGA/SNN_Demo/hello_world.prm

State of non-config mem I/O pins: Pull-none

Program Operations

Address Range: Entire Configuration Memory Device

- [x] Erase
- [x] Blank Check
- [x] Program
- [x] Verify
- [x] Verify Checksum

SVF Options

- [ ] Create SVF Only (no program operations)

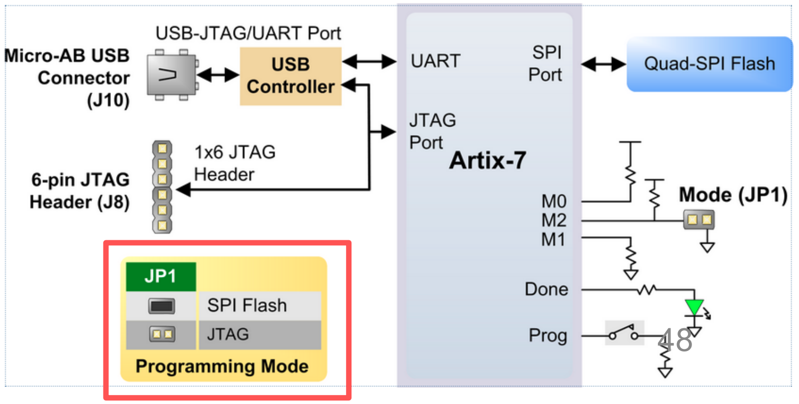
SVF File:

2. New target with JTAG frequency of 3MHz

3. Right click & Add configuration memory device & program

4. Select your generated memory configuration file

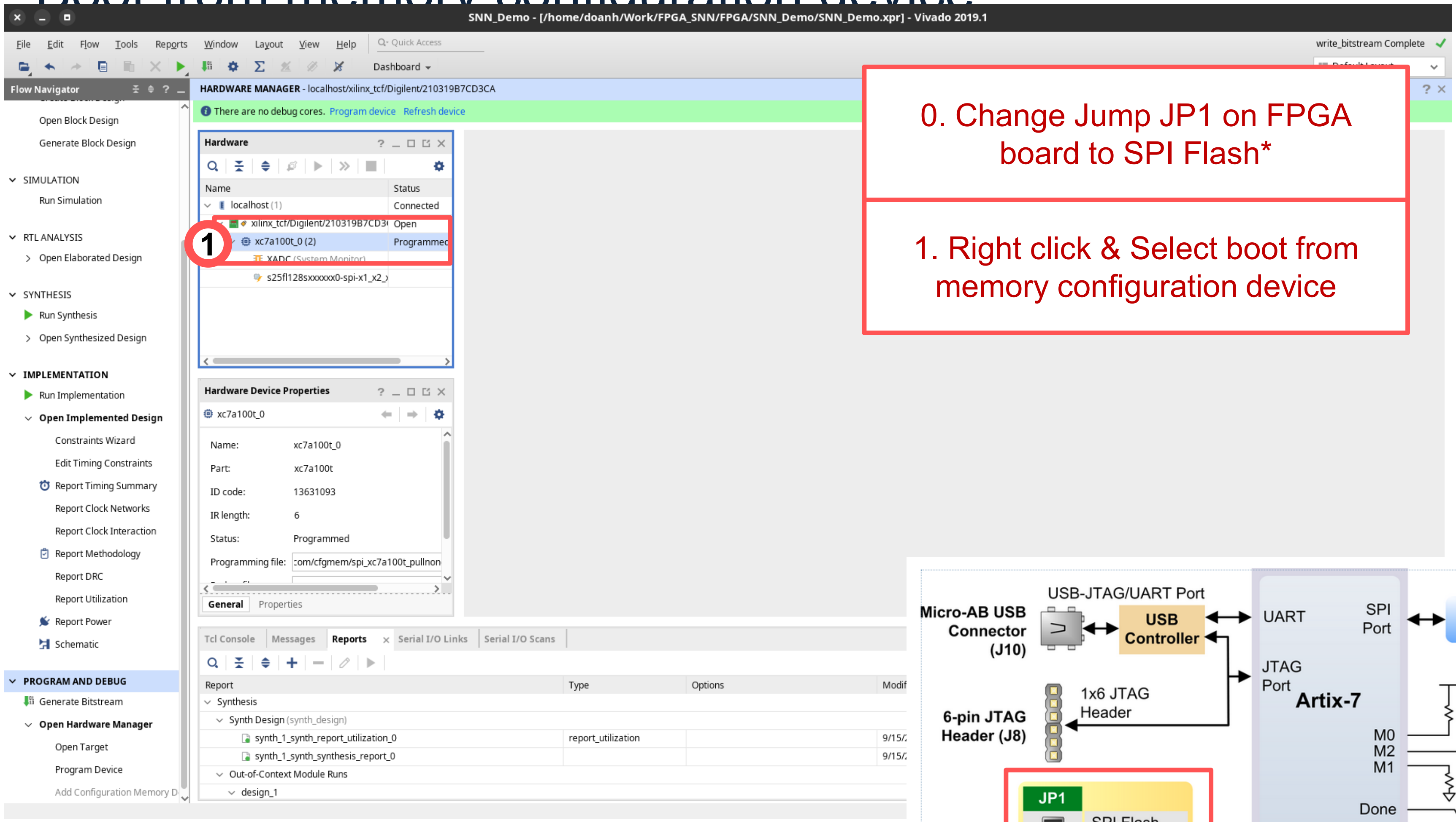
6. NOTE: Before OK. Need to change Jump JP1 on FPGA board to JTAG*



* Apply for Arty A7 100T; Different FPGAs have different setups;

Flash memory configuration (13)

- Boot from memory configuration device



SNN_Demo - [/home/doanh/Work/FPGA_SNN/FPGA/SNN_Demo/SNN_Demo.xpr] - Vivado 2019.1

write_bitstream Complete ✓

HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210319B7CD3CA

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319B7CD3	Open
xc7a100t_0 (2)	Programmed
XADC (System Monitor)	
s25fl128sxxxxxx0-spi-x1_x2_x	

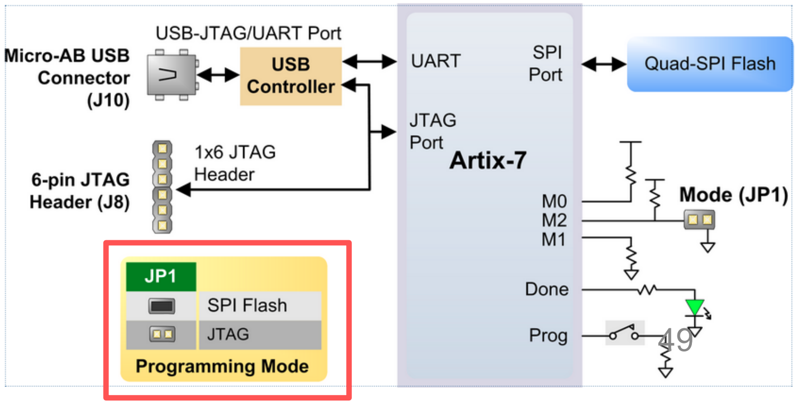
Hardware Device Properties: xc7a100t_0

Name:	xc7a100t_0
Part:	xc7a100t
ID code:	13631093
IR length:	6
Status:	Programmed
Programming file:	:om/cfgmem/spi_xc7a100t_pullnon

Report	Type	Options	Modif
Synthesis			
Synth Design (synth_design)			
synth_1_synth_report_utilization_0	report_utilization		9/15/
synth_1_synth_synthesis_report_0			9/15/
Out-of-Context Module Runs			
design_1			

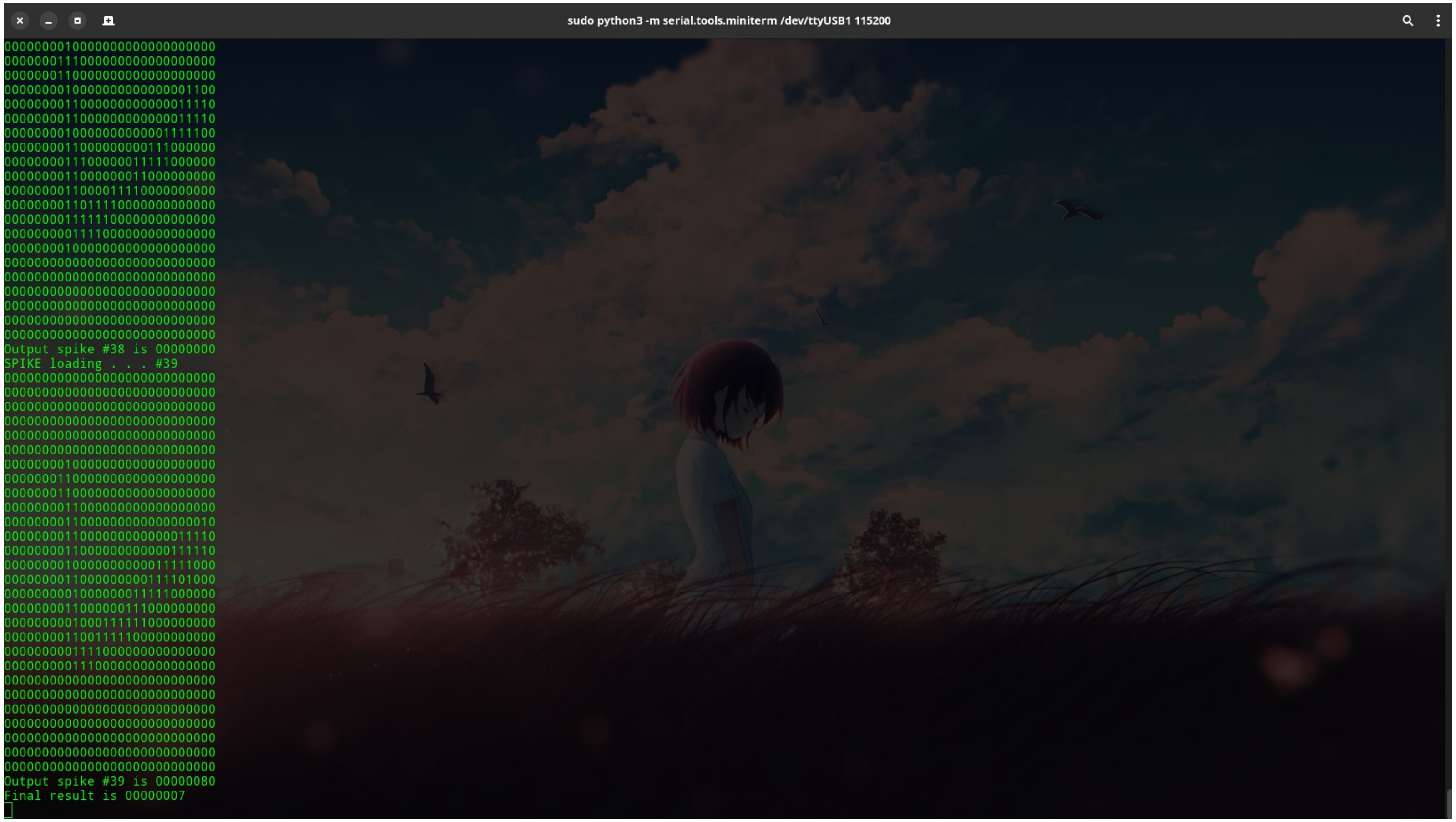
0. Change Jump JP1 on FPGA board to SPI Flash*

1. Right click & Select boot from memory configuration device



Flash memory configuration (14)

- Connect to UART display & programs automatically start.



sudo python3 -m serial.tools.miniterm /dev/ttyUSB1 115200

```
0000000010000000000000000000
0000000111000000000000000000
0000000110000000000000000000
0000000010000000000000001100
0000000011000000000000011110
0000000011000000000000011110
0000000010000000000001111100
0000000011000000000111000000
0000000011100000011111000000
0000000011000000011000000000
0000000011000011110000000000
0000000011011111000000000000
0000000011111100000000000000
0000000001111000000000000000
0000000010000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
Output spike #38 is 00000000
SPIKE loading . . . #39
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000010000000000000000000
0000000110000000000000000000
0000000110000000000000000000
0000000011000000000000000000
0000000011000000000000000010
0000000011000000000000011110
0000000011000000000000111110
0000000010000000000011111000
0000000011000000000111101000
0000000001000000011111000000
0000000011000000111000000000
0000000001000111111000000000
0000000011001111110000000000
0000000001111000000000000000
0000000001110000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
0000000000000000000000000000
Output spike #39 is 00000080
Final result is 00000007
```

The University of Aizu

Thank you

for your attention.