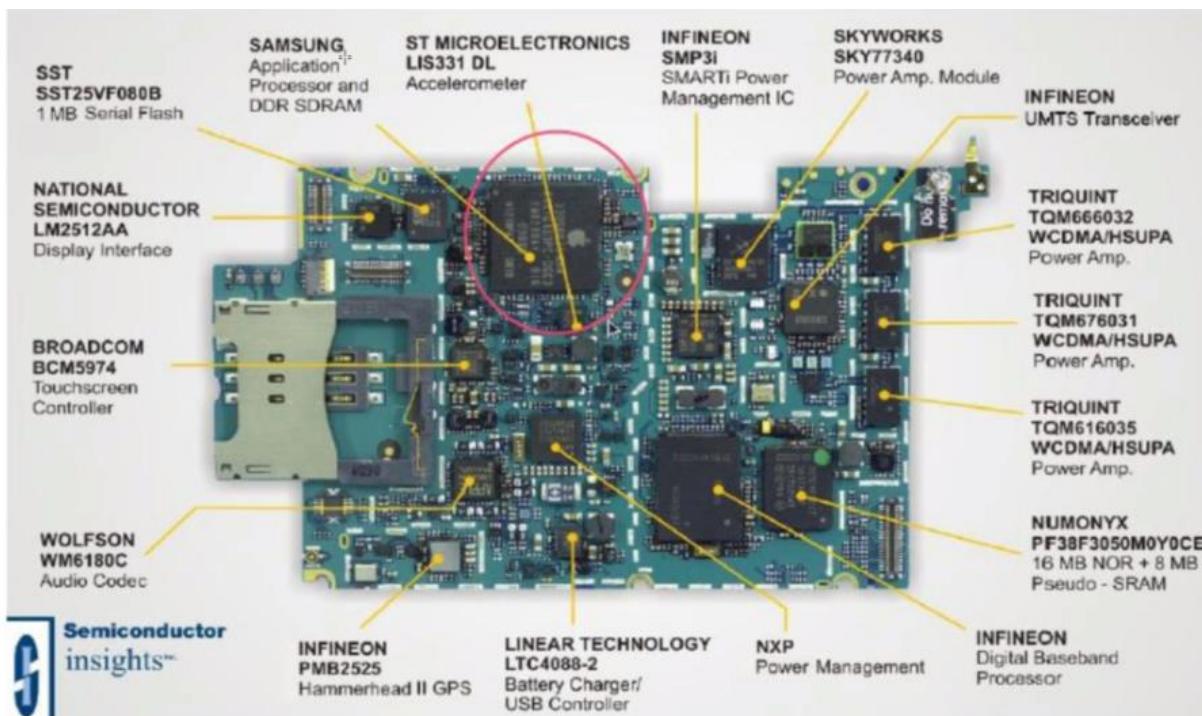


A SoC is a complete system on a chip. A 'system' can include microprocessors, FPGAs, memory and peripherals.

The processor may be a custom or standard microprocessor, or it could be a specialized media processor for sound, modem or video applications. There may be multiple processors.

Processors are interconnected using a variety of mechanisms.

SoCs are found in every consumer product, from modems, mobile phones, DVD players, televisions and iPods.



One of the two main PCBs of an Apple iPhone. Main SoC is top, left-centre.

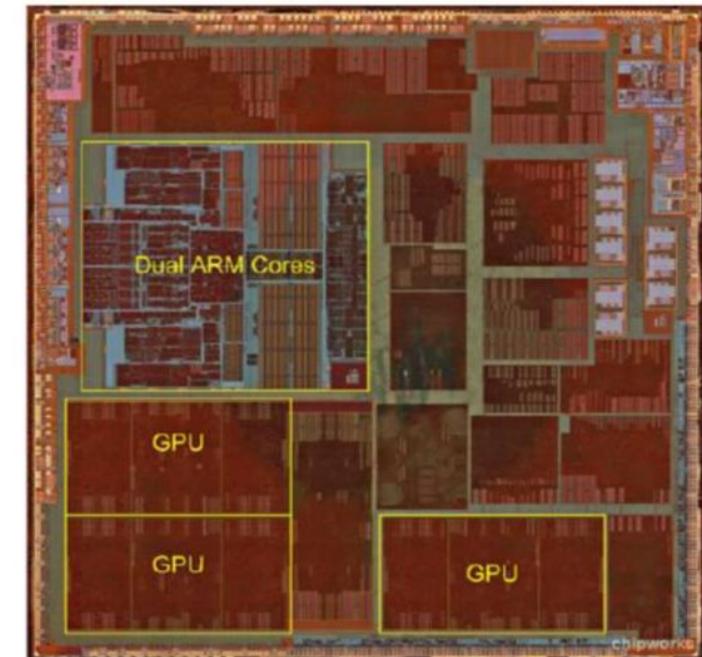


Fig. 3

An Apple SoC - Two ARM and 3 GPU cores. Made by arch rival Samsung.

AT91SAM: AT91SAM is an ARM-based SOC

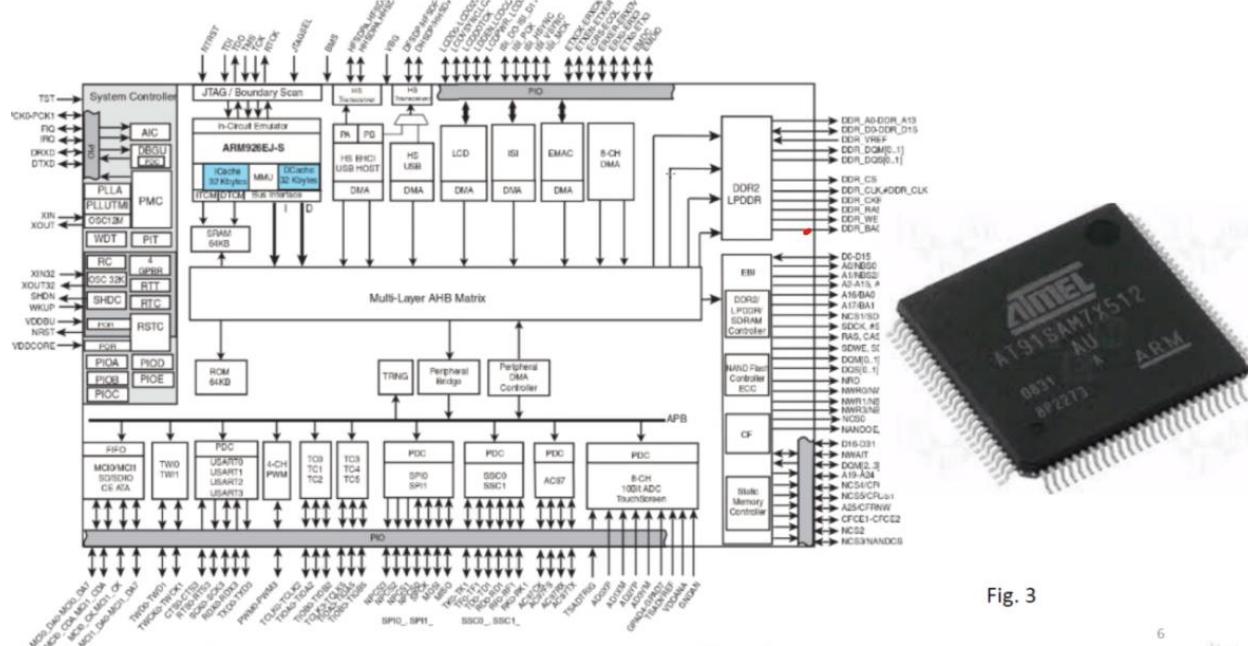
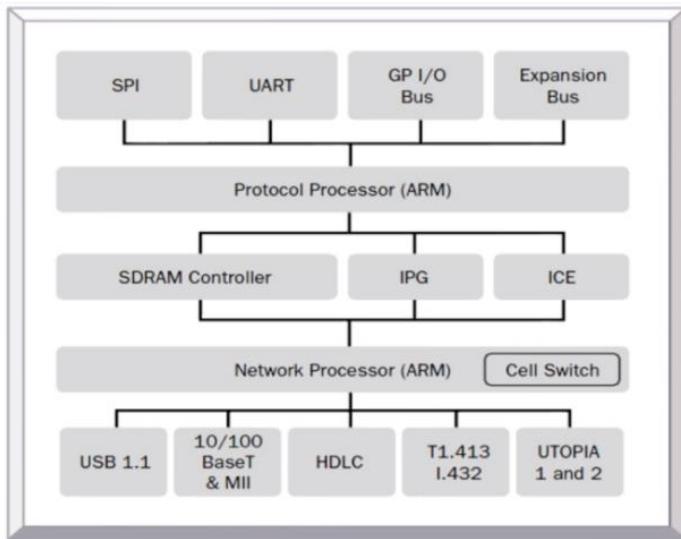


Fig. 3

6

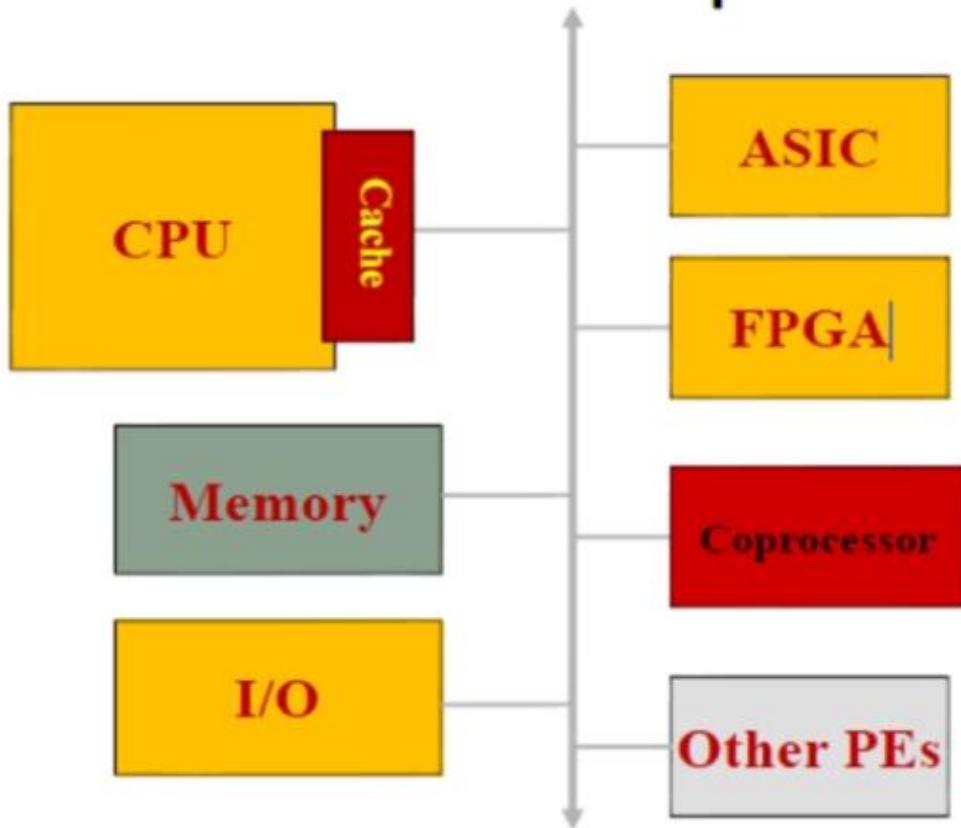
SOC Example: Helium 210

The Helium 210 is a single chip Communications Processor (SOC). Used in many different ADSL and home networking products



SoC Hardware Structure

Various Hardware Options:



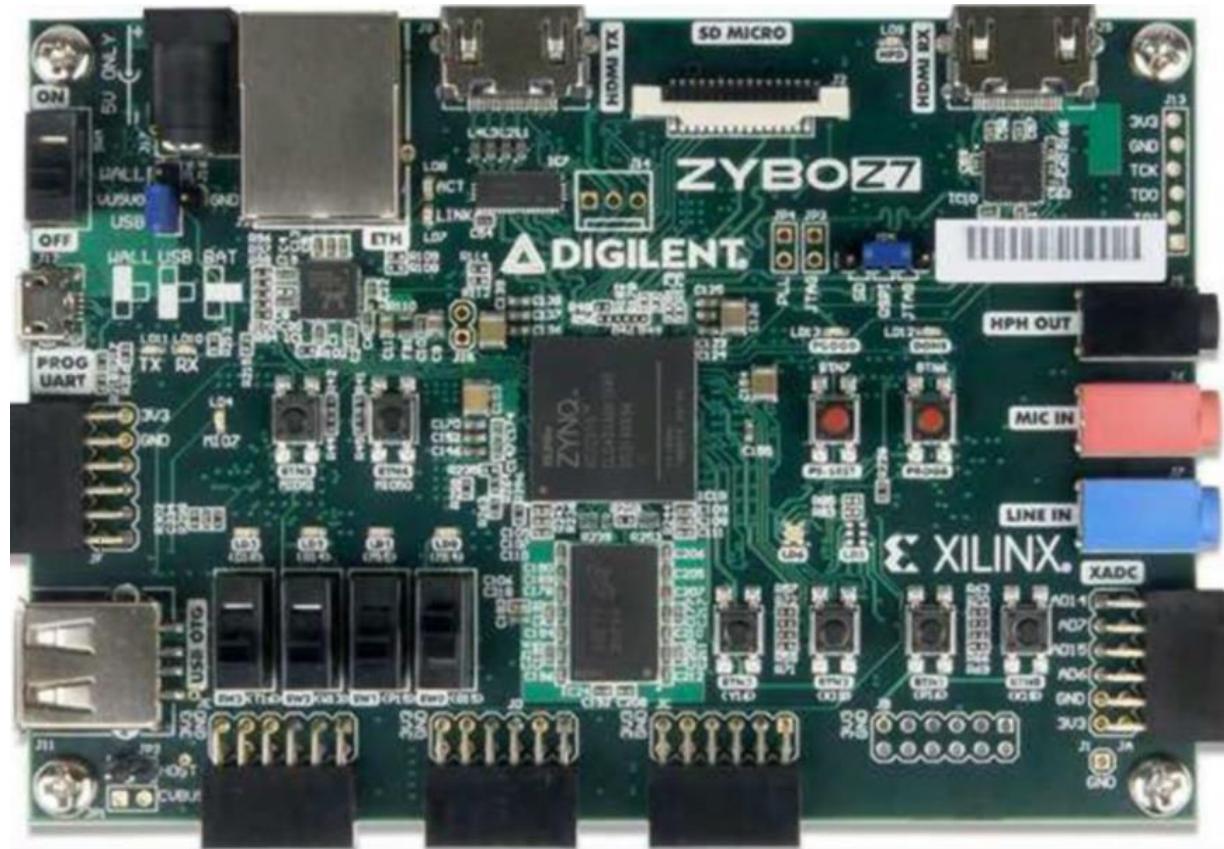
SOC Developments Boards:

Mainly two types of SOC Developments Boards are available to work for the SOC development studies, and these boards are produced by the Altera (INTEL) and XILINX companies.

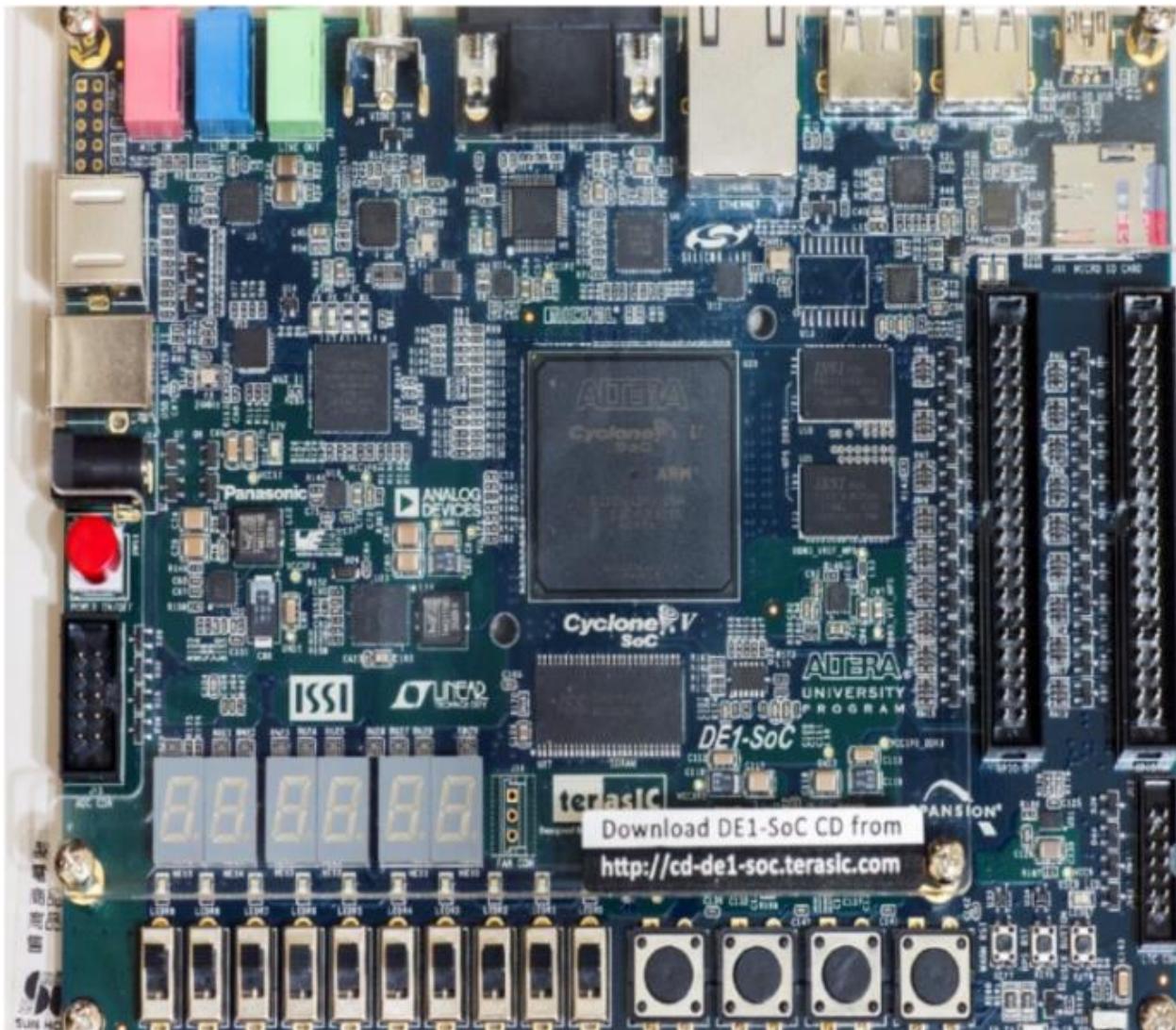
XILINX provides Zynq SOC development boards.

The Altera's SOC development board is named as **DE1-SOC, DE2-SOC**

→the board we'll use

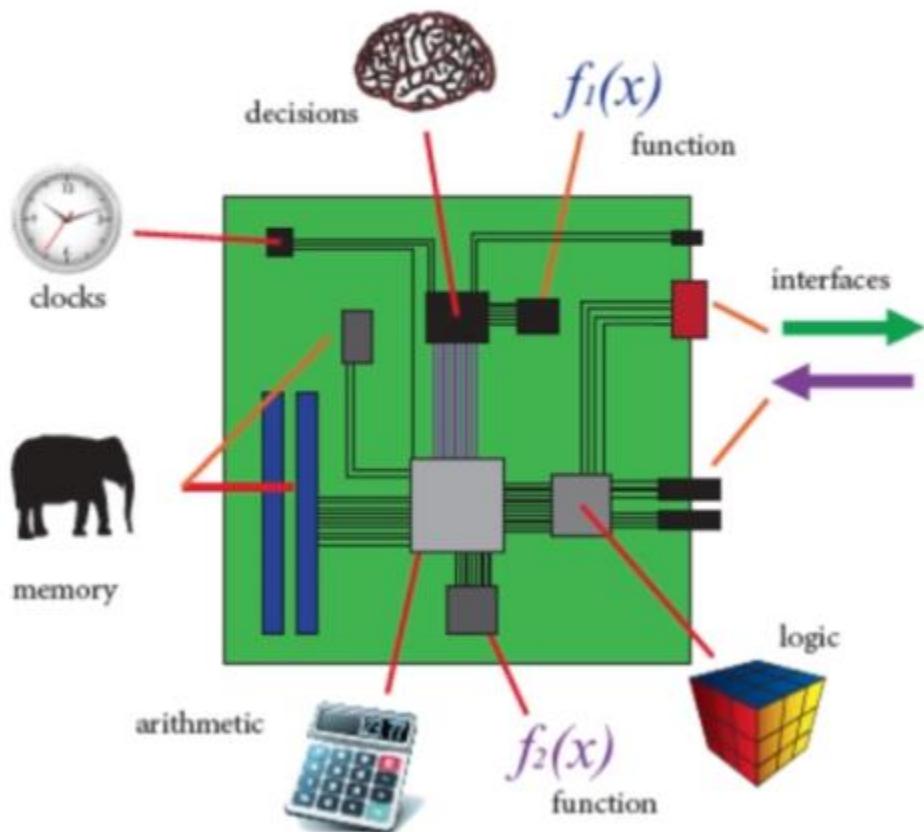


→Altera(Intel) board



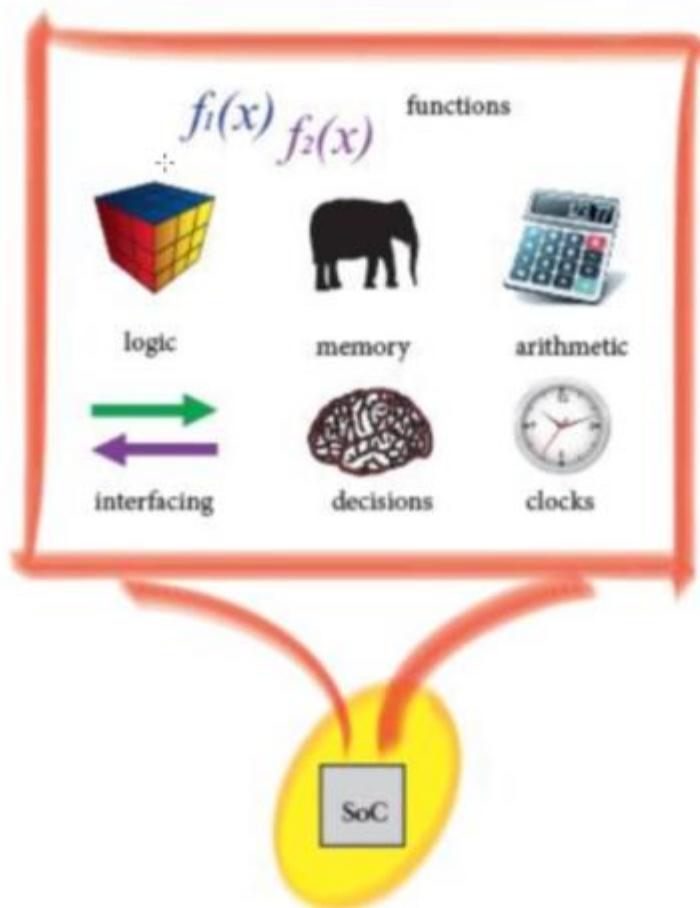
→ PCBs

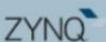
System-on-a-Board



→ SOCs:

System-on-Chip (SoC)



Cost-Optimized	Mid-Range	High-End	High-End
 <p>Zynq-7000 SoC Artix Devices</p> <p>Dual Arm® Cortex®-A9</p>  <p>Zynq UltraScale+ MPSoC CG Devices</p> <p>Dual Arm Cortex-A53 Dual Arm Cortex-R5</p>  <p>Zynq-7000 SoC Kintex Devices</p> <p>Dual Arm Cortex-A9</p>	 <p>Zynq UltraScale+ MPSoC EV Devices</p> <p>Quad Arm Cortex-A53 Dual Cortex-R5 + GPU + Video Codec</p>  <p>Zynq UltraScale+ MPSoC EG Devices</p> <p>Quad Arm Cortex-A53 Dual Cortex-R5 + GPU</p>	 <p>Zynq UltraScale+ RFSoC with RF Data Converters</p> <p>Quad Arm Cortex-A53 Dual Cortex-R5</p>	 <p>Zynq UltraScale+ RFSoC with SD-FEC Cores</p> <p>Quad Arm Cortex-A53 Dual Arm Cortex-R5</p>  <p>Zynq UltraScale+ RFSoC with RF Data Converters & SD-FEC Cores</p> <p>Quad Arm Cortex-A53</p>

Zynq-7000 SOC comprises two main parts:

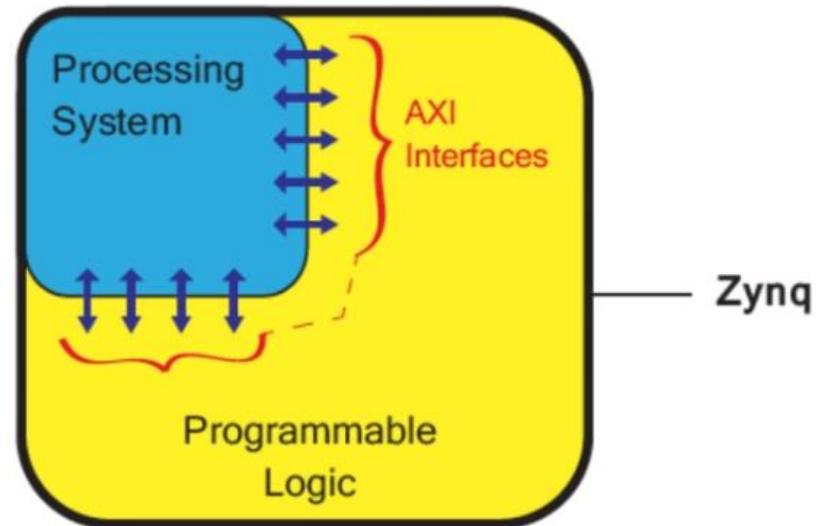
a Processing System (PS) formed around a dual-core ARM Cortex-A9 processor, and Programmable Logic (PL), which is equivalent to that of an FPGA.

It also features integrated memory, a variety of peripherals, and high-speed communications interfaces.

The PL section is ideal for implementing high-speed logic, arithmetic and data flow subsystems, while the PS supports software routines and/or operating systems, meaning that the overall functionality of any designed system can be appropriately partitioned between hardware and software.

Links between the PL and PS are made using industry standard Advanced eXtensible Interface (AXI) connections.

A Simplified Model of the Zynq Architecture



Simplified Hardware Architecture of an Embedded SoC

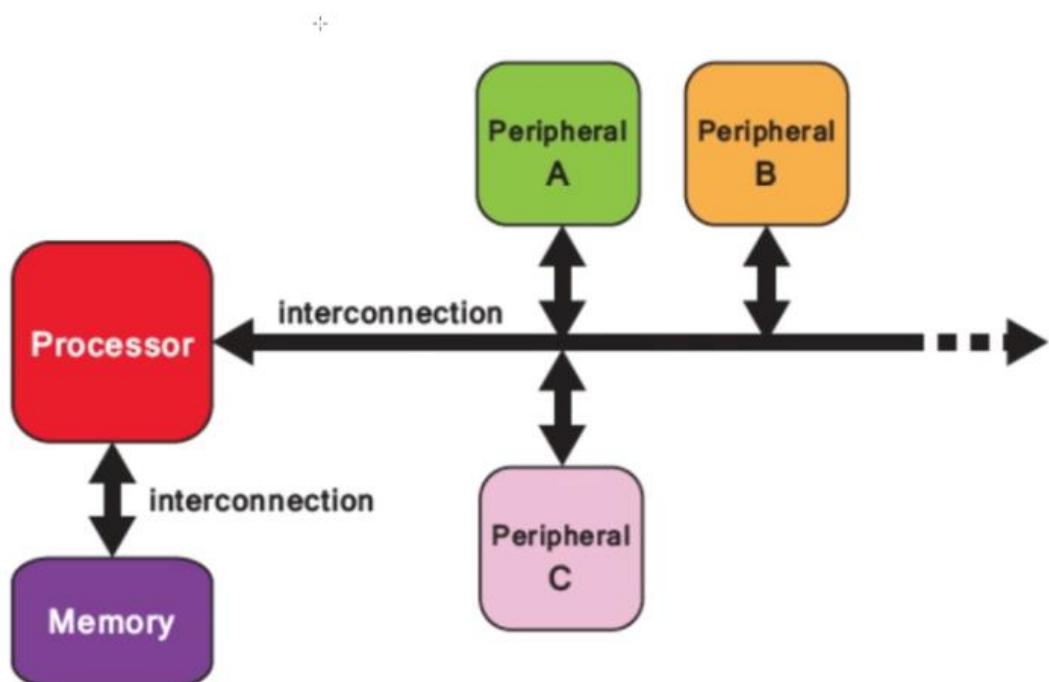
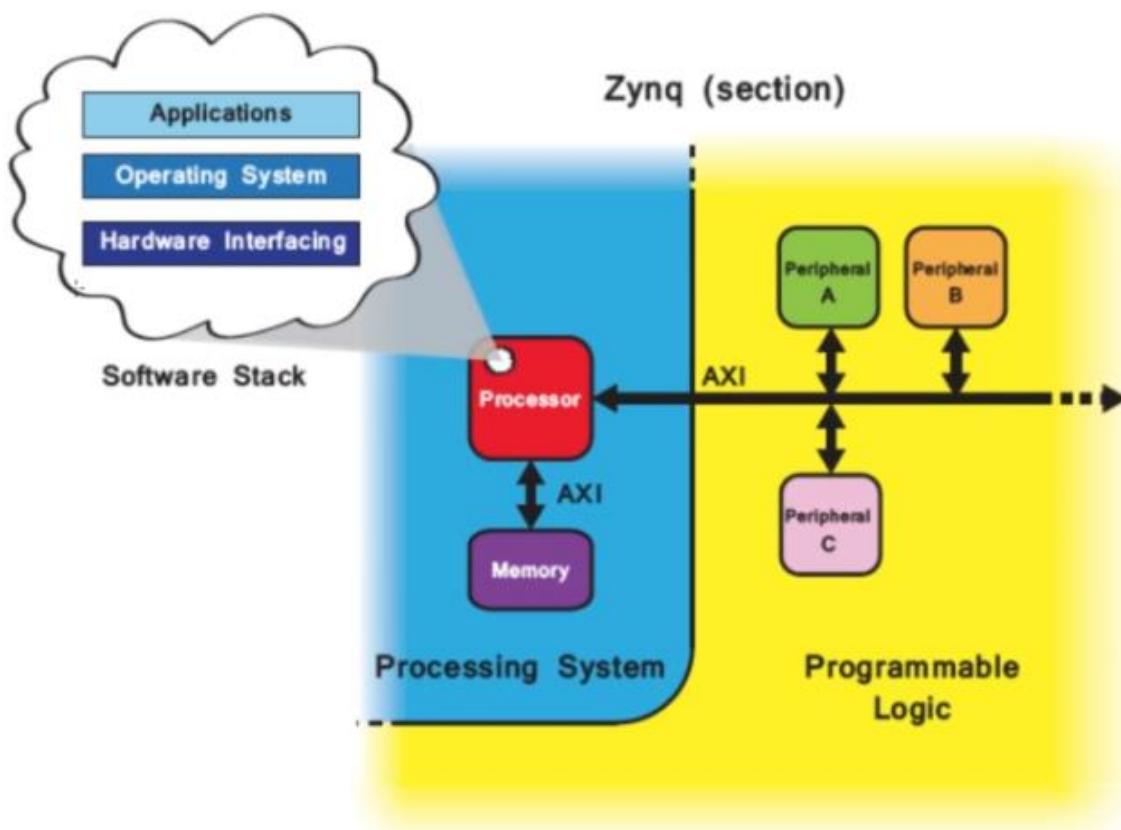


Figure 2-3: Hardware architecture of SOC

Mapping of an Embedded SoC Hardware Architecture to Zynq



The processor can be regarded as the central element of the hardware system.

The software system (a software ‘stack’) is run on the processor, comprising applications (usually based on an Operating System (OS)), and with a lower layer of software functionality for interfacing with the hardware system.

Communication between system elements takes place via interconnections. These may be in the style of direct, point-to-point links, or buses serving multiple components.

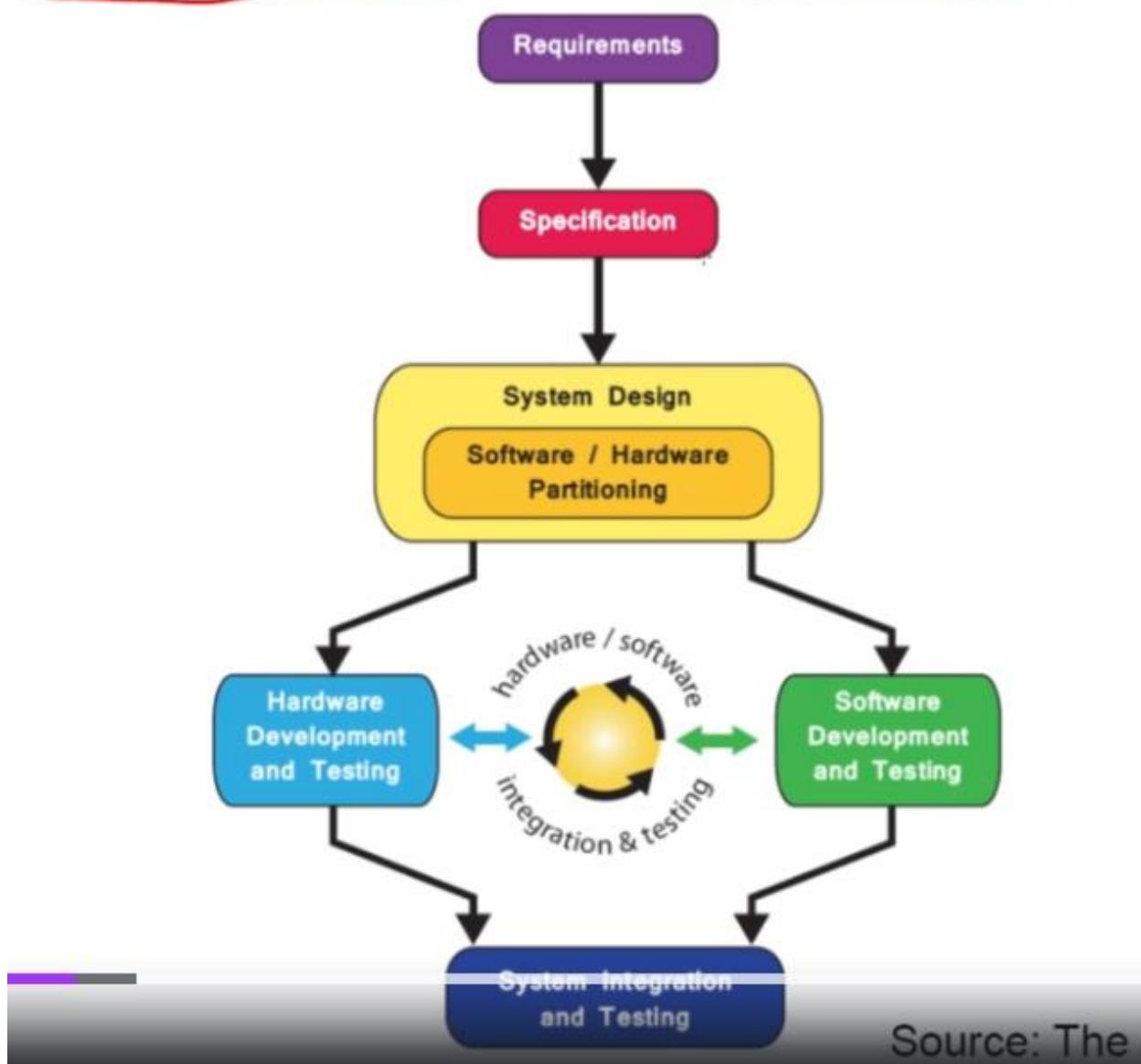
In the latter case, a protocol is required to manage access to the bus.

Note that, although a single bus with connected peripherals is shown in Figure 2-3, a processor may serve several connected buses.

Peripherals are functional components residing away from the processor, and in general these perform one of three functions:

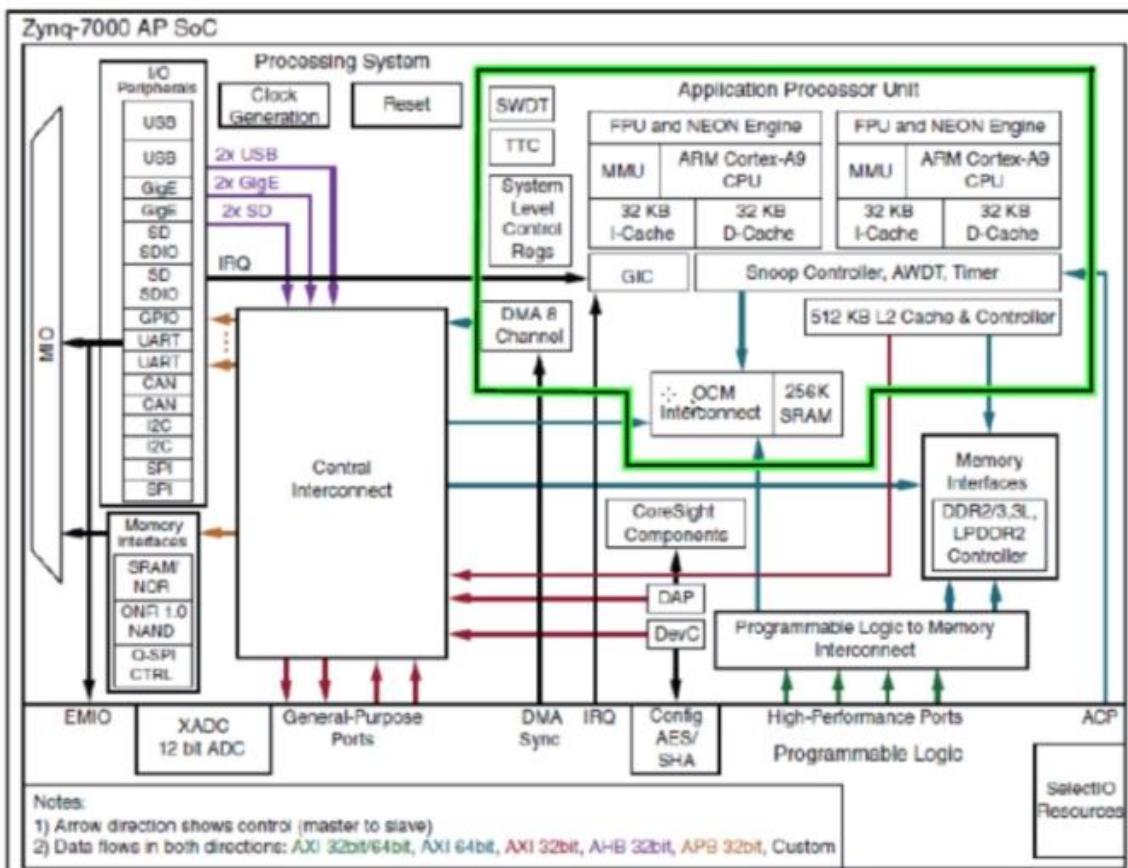
- (i) coprocessors — elements that supplement the primary processor, usually optimized for a certain task;
- (ii) cores for interacting with external interfaces, e.g. connecting to LEDs and switches, codecs, etc.; and
- (iii) additional memory elements.

Basic Design Flow for Zynq SoC

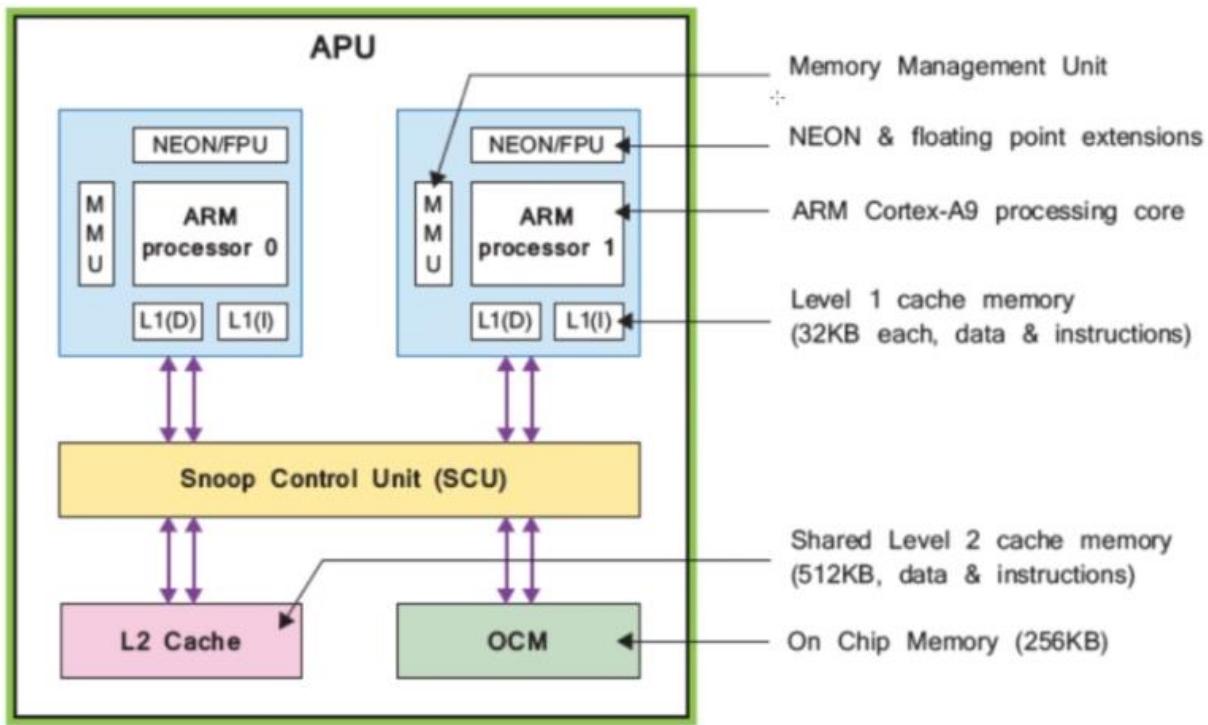


Source: The

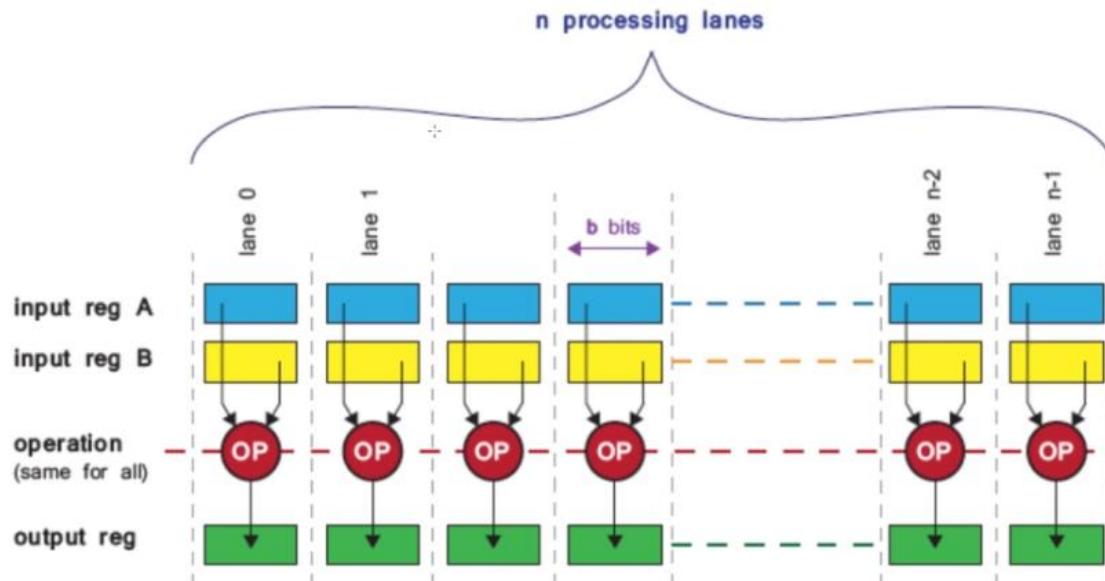
The Zynq Processing System



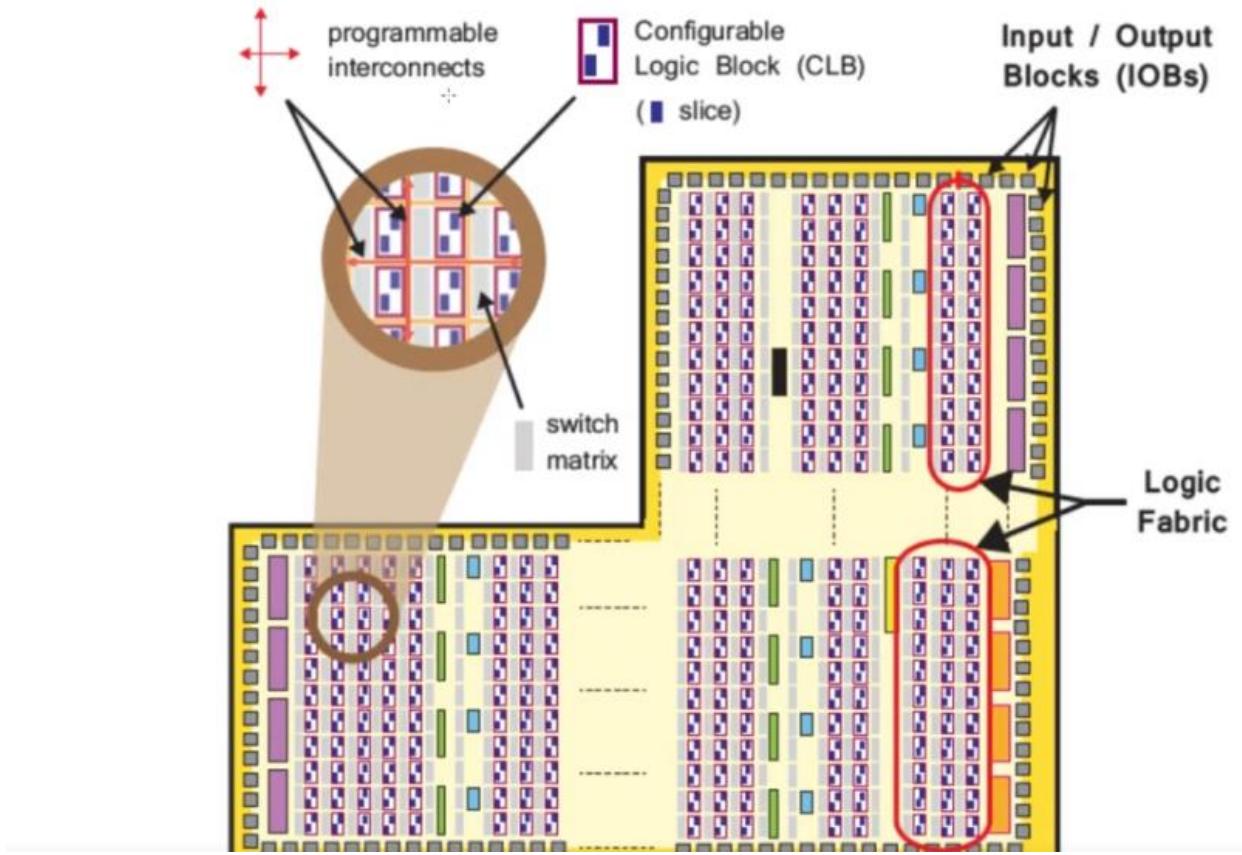
Simplified Block Diagram of the Application Processing Unit (APU)



SIMD (Single Instruction Multiple Data) Processing in the NEON Media Processing Engine (MPE)

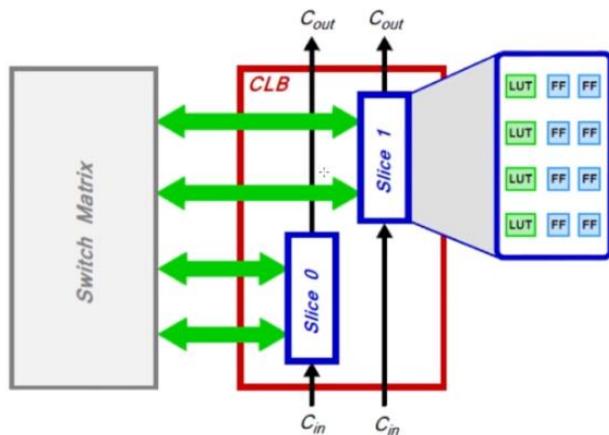


Programmable Logic (PL) CLBs and IOBs



Configurable Logic Block

Configurable Logic Blocks (CLB) are small, regular groupings of logic elements that are laid out in a two-dimensional array on the PL, and connected to other similar resources via programmable interconnects. Each CLB is positioned next to a switch matrix and contains two logic slices

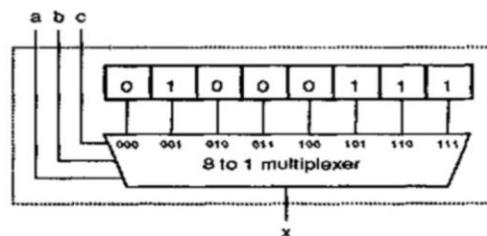


→look Up Table(LUT):

A flexible resource capable of implementing (i) a logic function of up to six inputs; (ii) a small Read Only Memory (ROM); (iii) a small Random Access Memory (RAM); or (iv) a shift register. LUTs can be combined together to form larger logic functions, memories, or shift registers, as required.

a	b	c	$x = ab + \bar{b}c$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

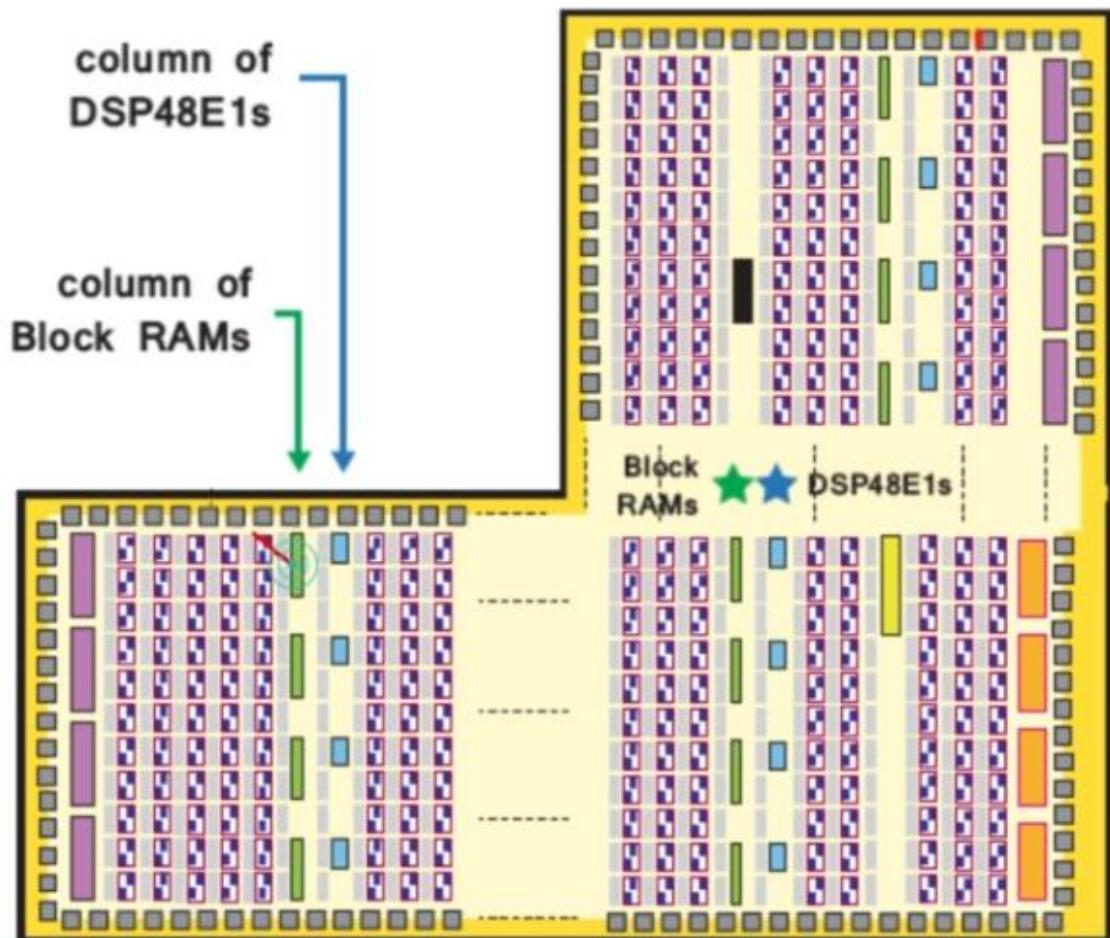
a) Truth Table



Special Resources: DSP48E1s and Block RAMs: In addition to the general fabric, there are two special purpose components: Block RAMs for dense memory requirements; and DSP48E1 slices for high-speed arithmetic.

Both of these resources are integrated into the logic array in a column arrangement, embedded into the fabric logic and normally in proximity to each other

Programmable Logic (PL) BRAMs and DSP units



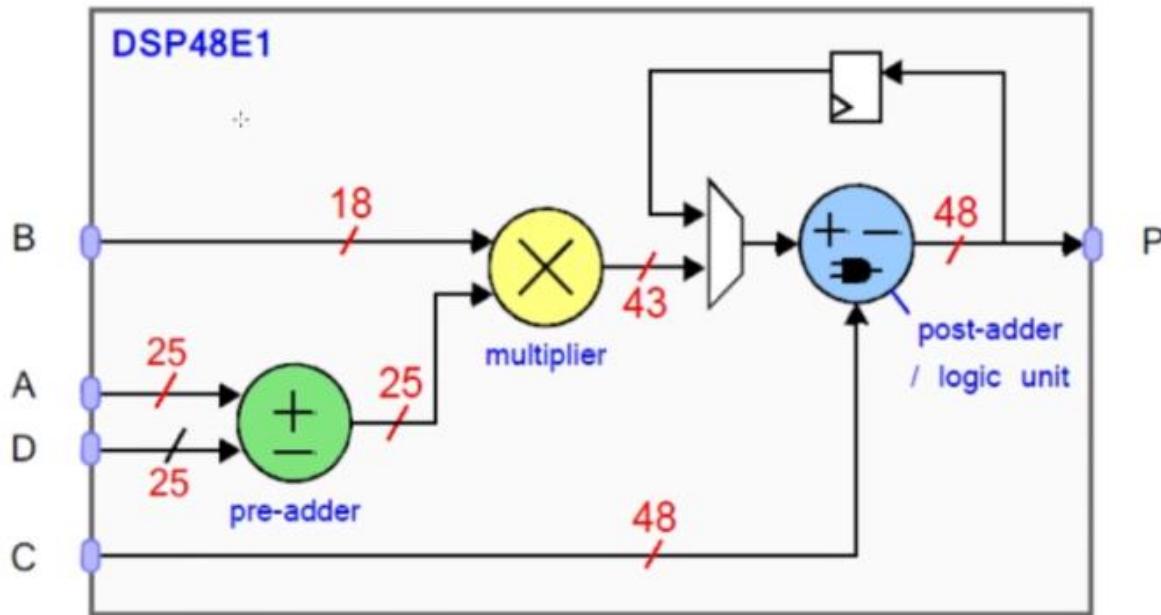


Figure 2-12: Arithmetic capabilities of the DSP48E1 slice

Communications Interfaces

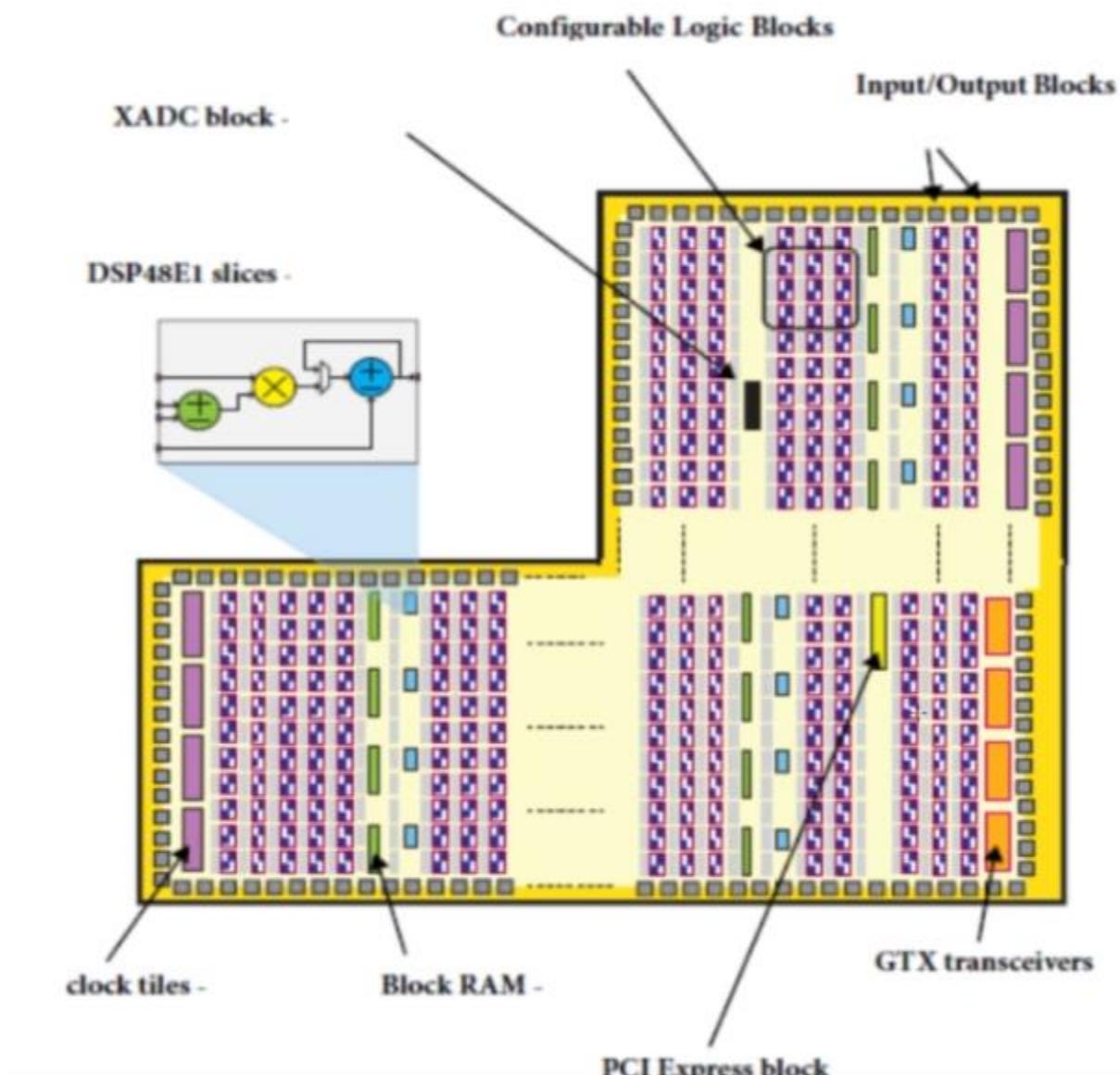
The more highly specified Zynq devices include GTX Transceivers, high-speed communications interface blocks which are embedded into the logic fabric.

These are dedicated silicon blocks (“Hard IP” blocks), and they are capable of supporting a number of standard interfaces including PCI Express, Serial RapidIO, SCSI and SATA.

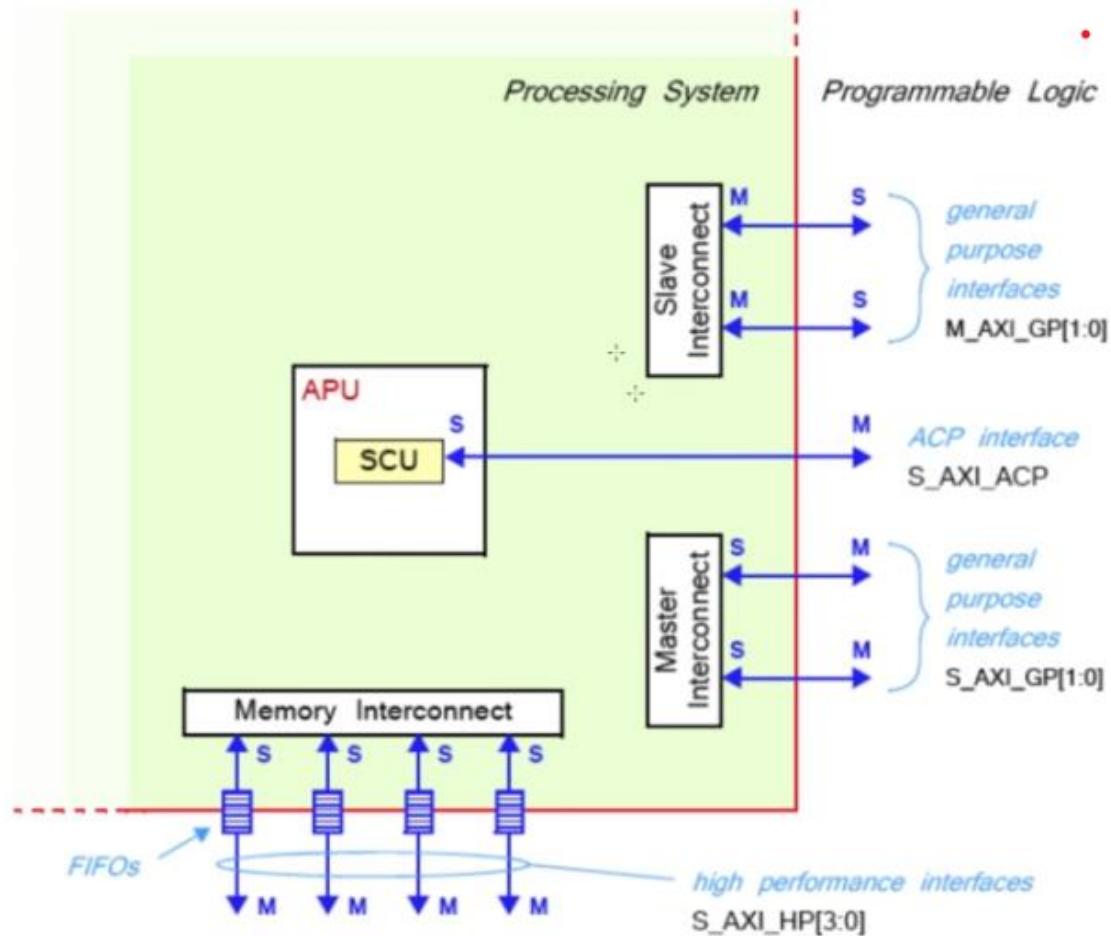
GTX Transceivers are implemented as ‘quads’, i.e. groups of 4 individual channels, each of which comprises a dedicated Phase Locked Loop (PLL) for that channel, a transmitter, and a receiver.

Depending on the Zynq device and package chosen, rates of up to 12.5Gbps are supported.

The interfaces can be used to create connections to independent external devices such as networking equipment, hard disks, and further FPGA or Zynq devices.



AXI Interconnects and Interfaces



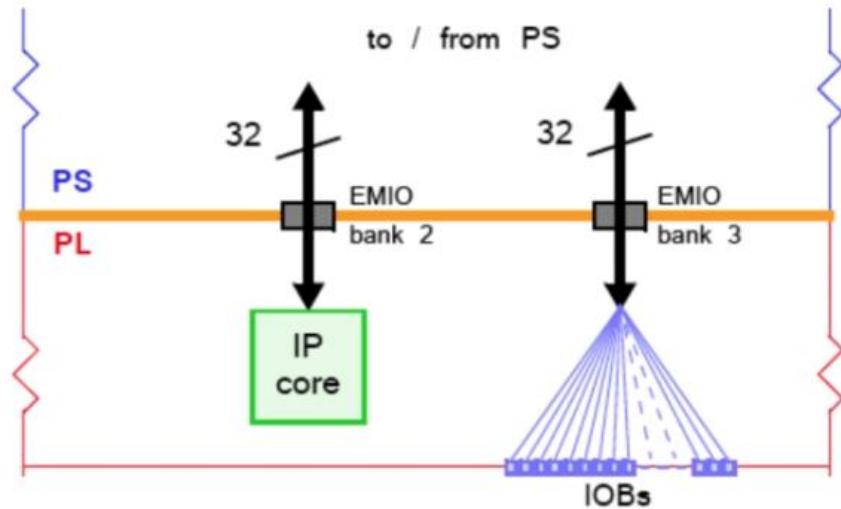
ACP-Accelerator Coherency Port

SCU-Snoop Control Unit

S→Slave

M→Master

Using Extended Multiplexed Input/Output (EMIO) to Interface Between PS and PL



3. ZYNN SOC Hardware Design Using VIVADO

VIVADO
HLS Editions

Quick Start

- [Create Project >](#)
- [Open Project >](#)
- [Open Example Project >](#)

Recent Project

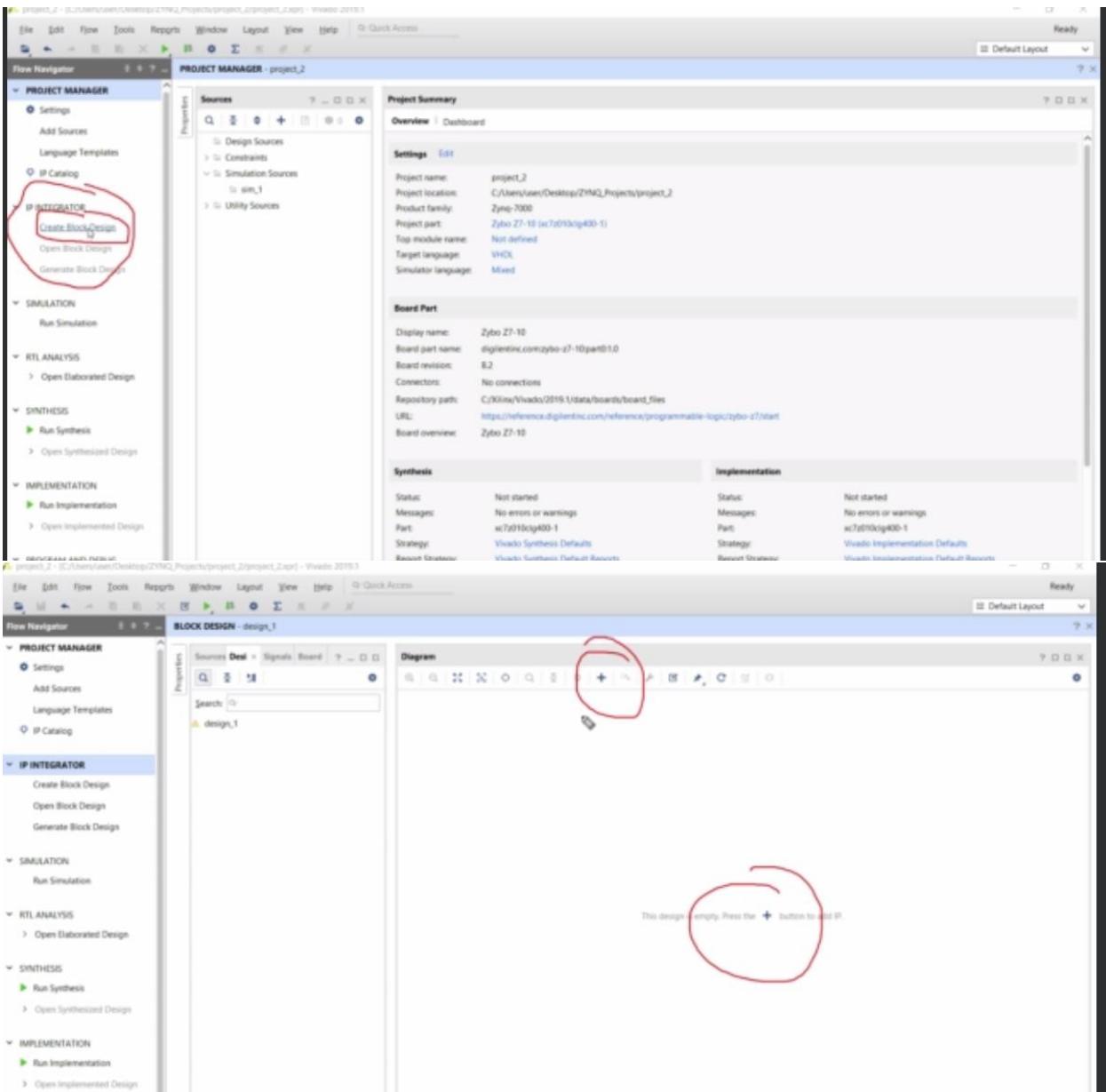
- project_1
C:/Users/User/0
- project_1
C:/Users/User/0

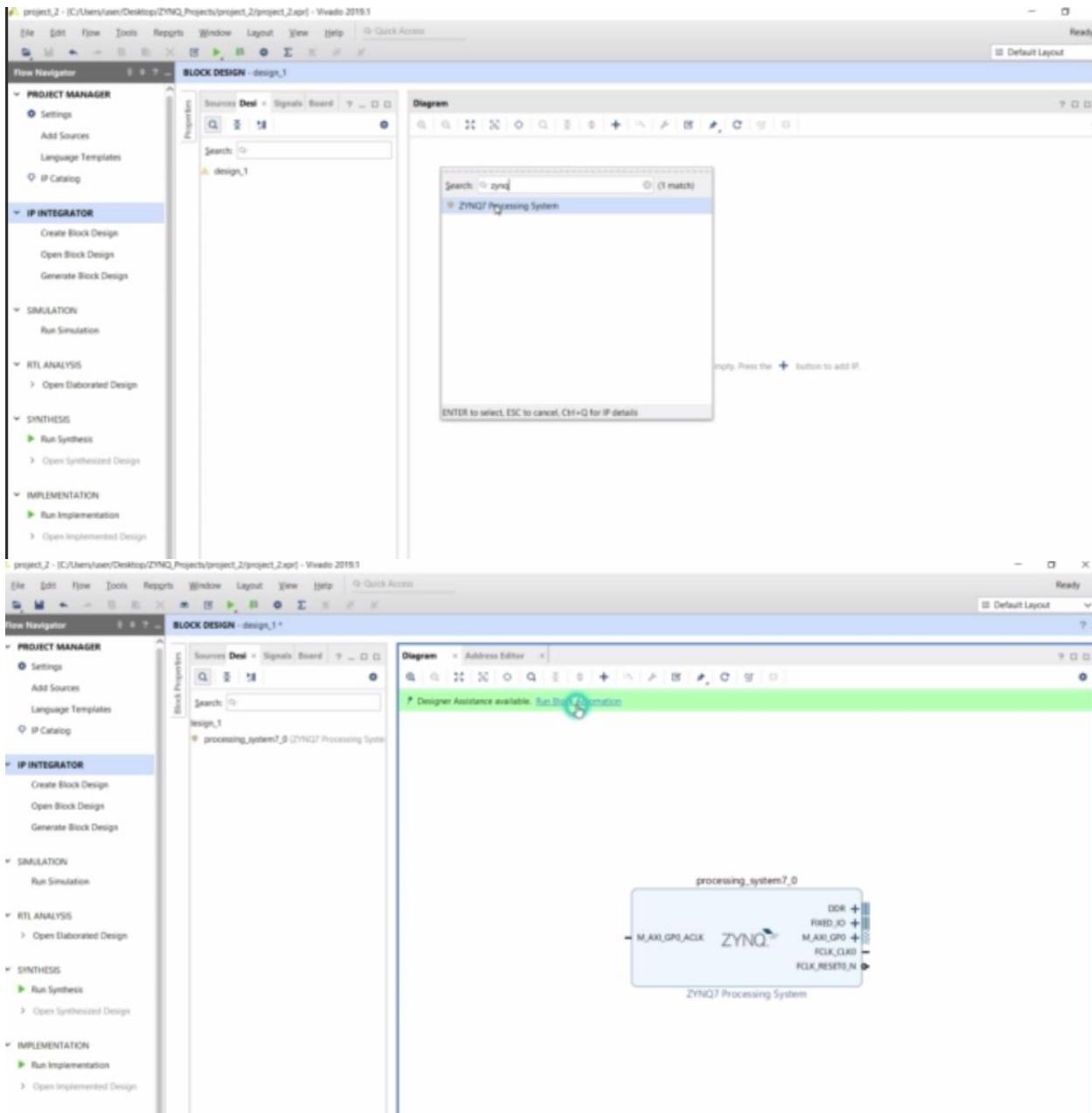
Tasks

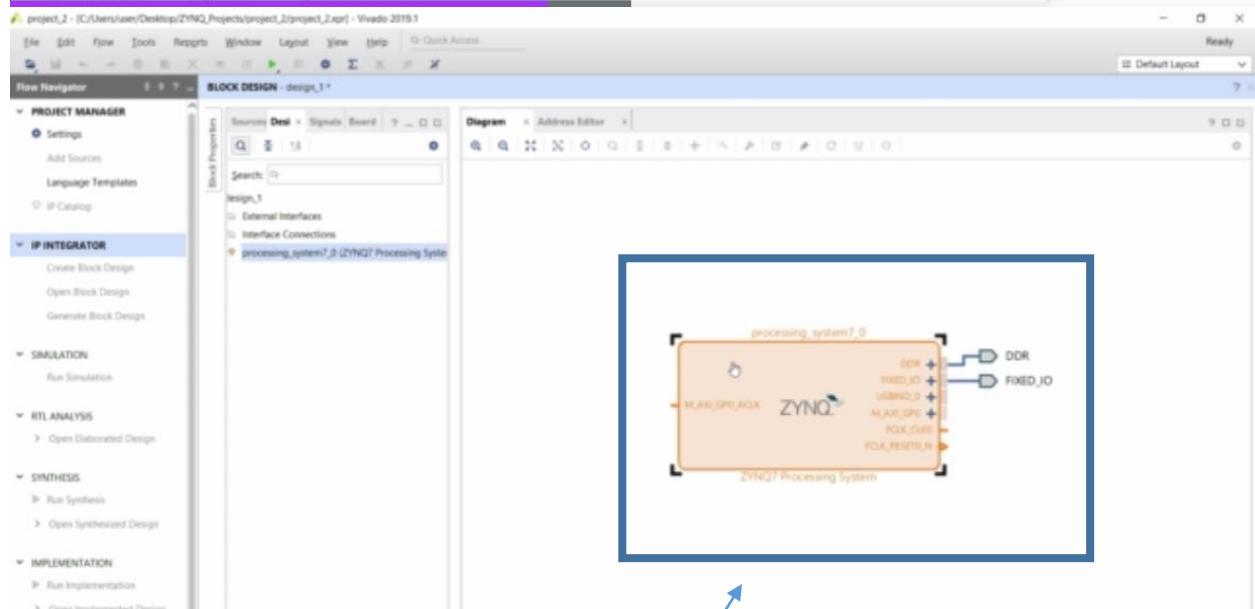
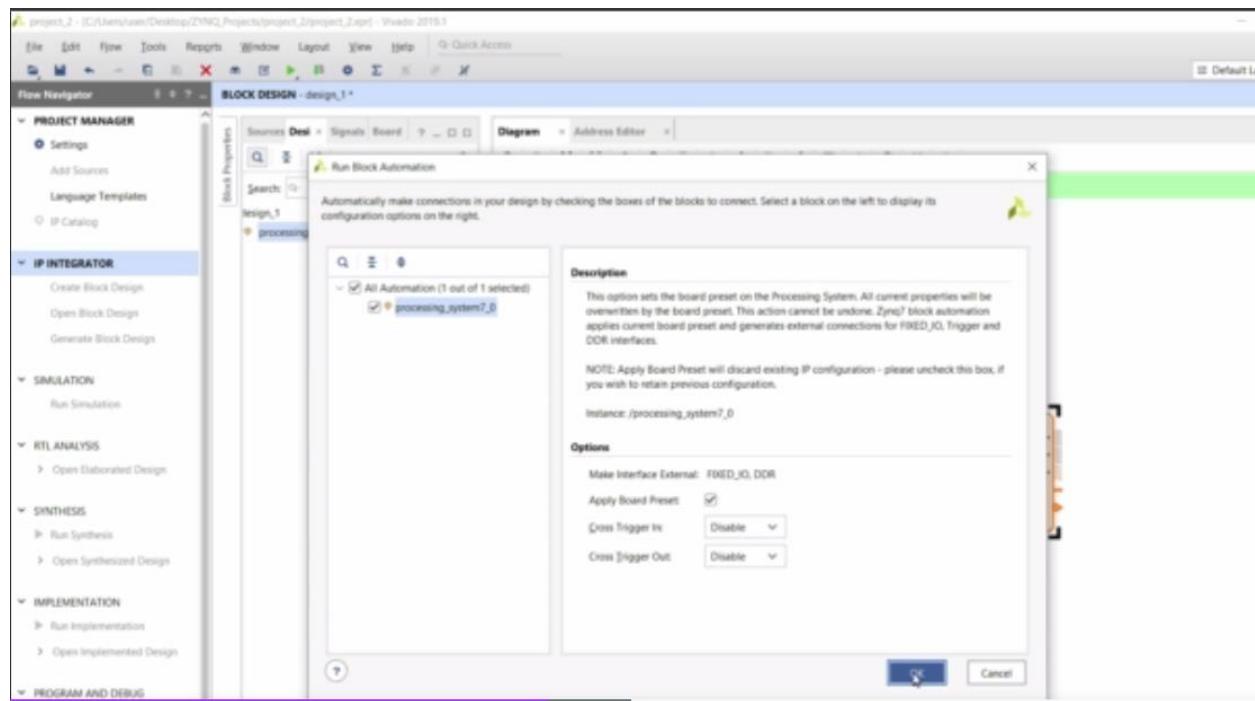
- [Manage IP >](#)
- [Open Hardware Manager >](#)
- [Xilinx Tcl Store >](#)

Learning Center

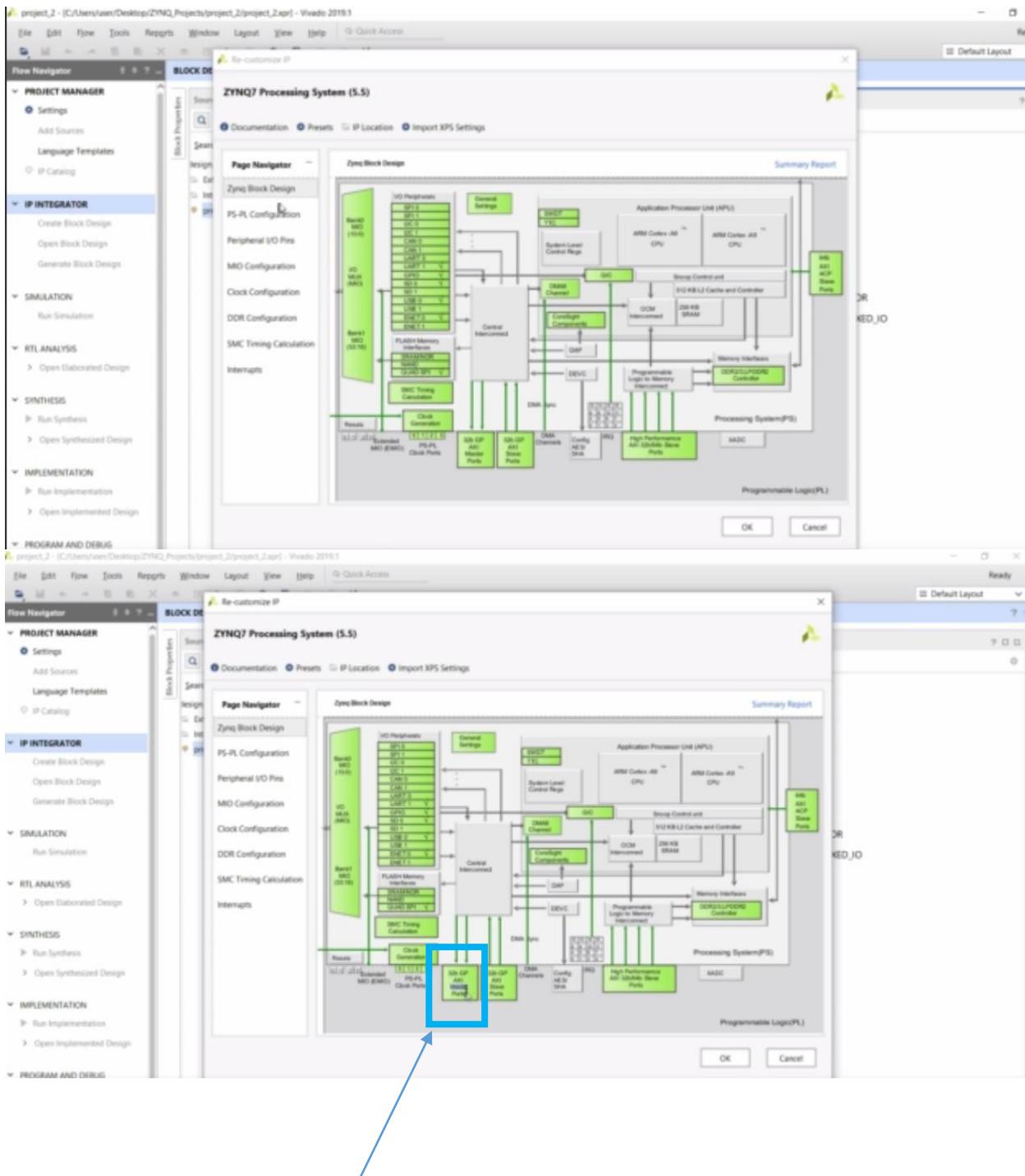
- [Documentation and Tutorials >](#)
- [Quick Take Videos >](#)
- [Release Notes Guide >](#)



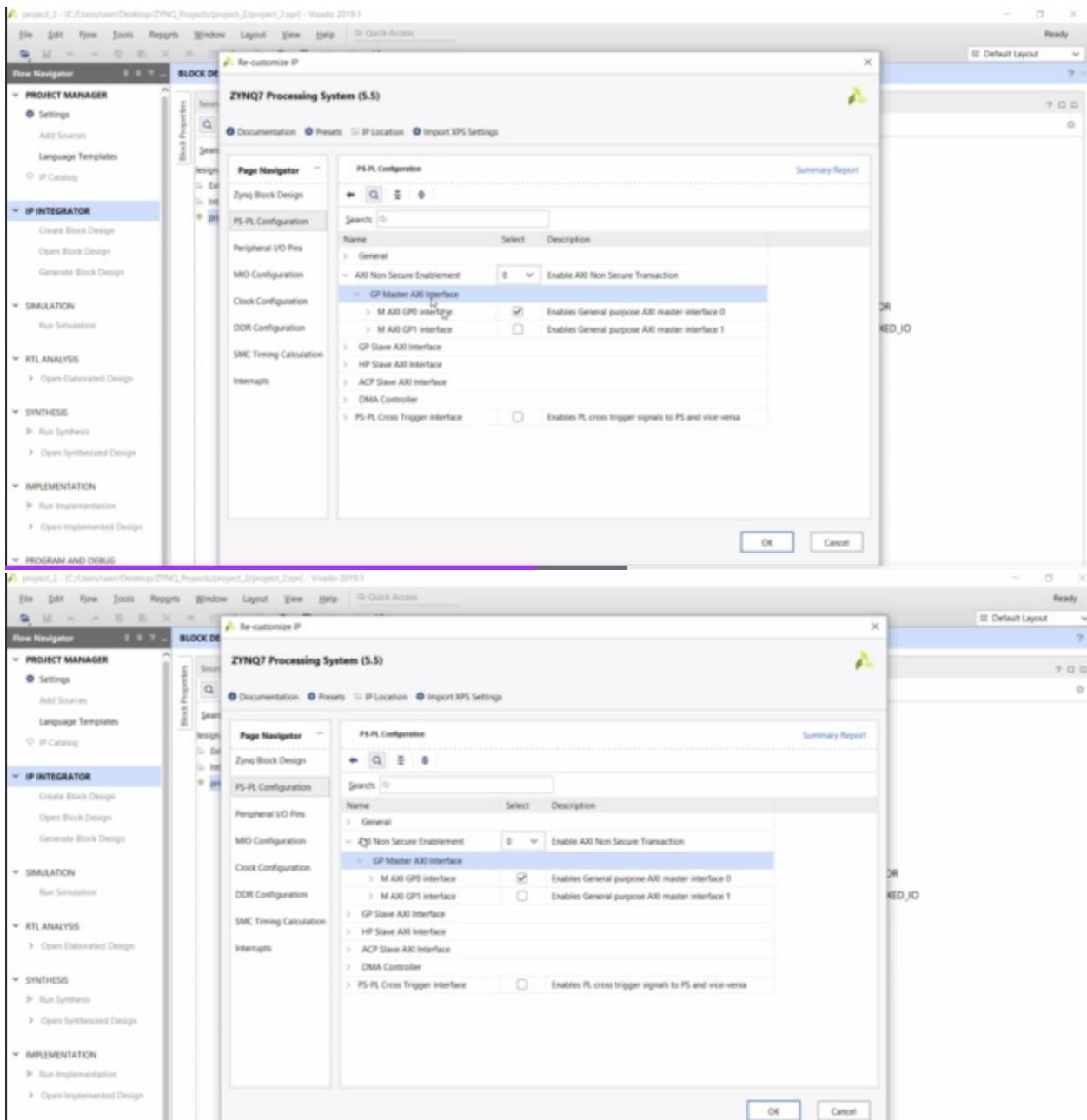


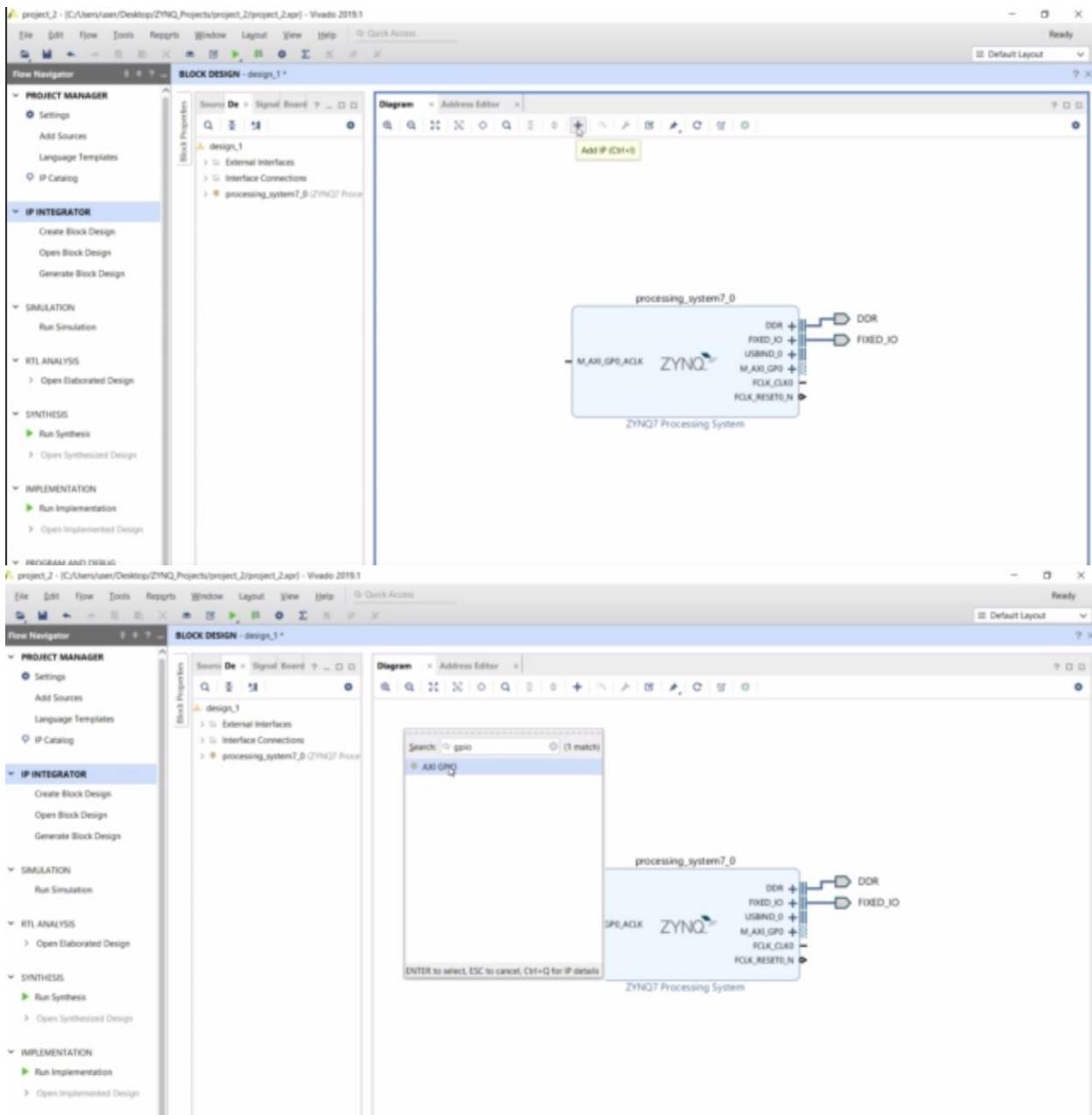


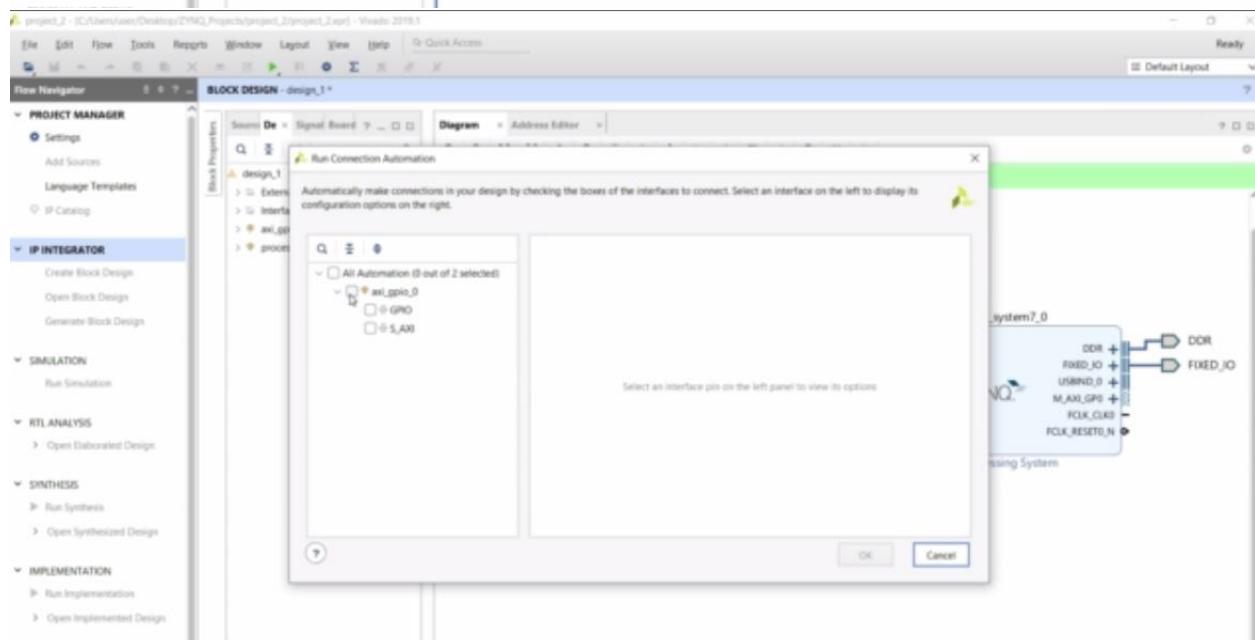
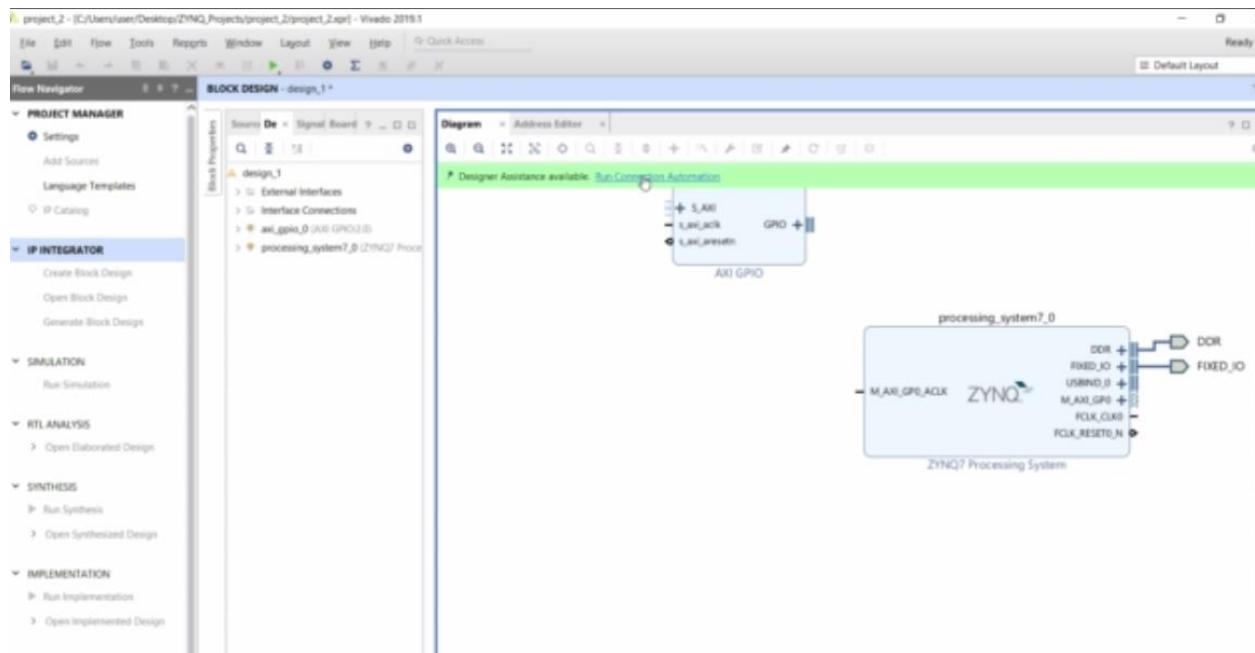
Double click

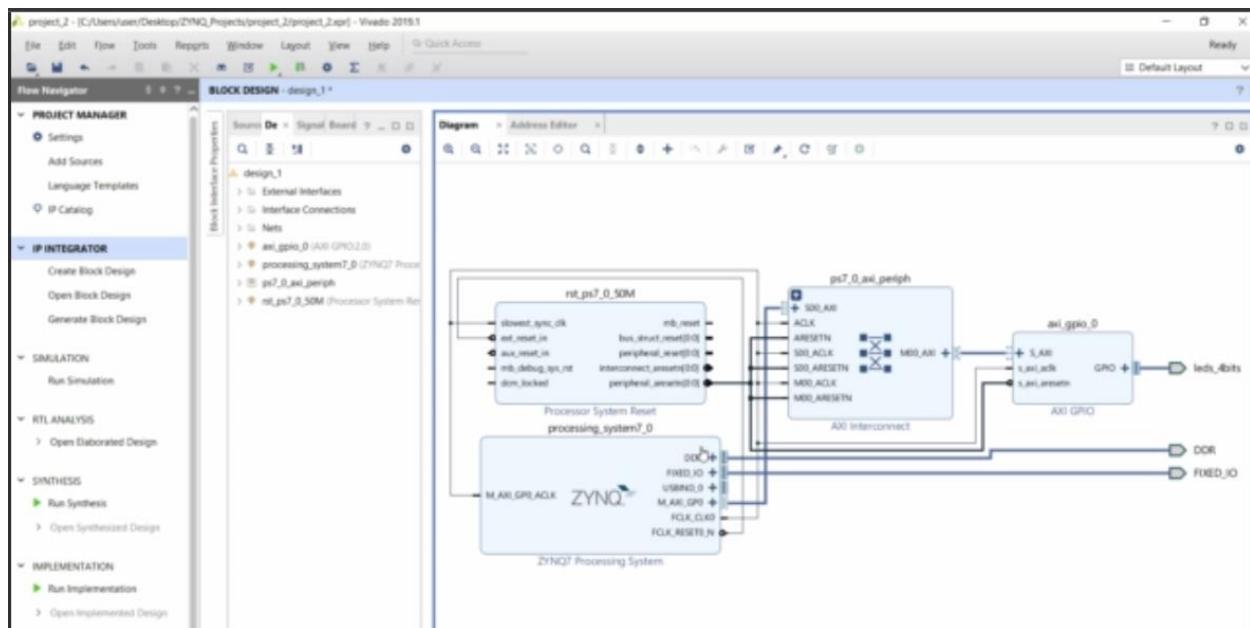
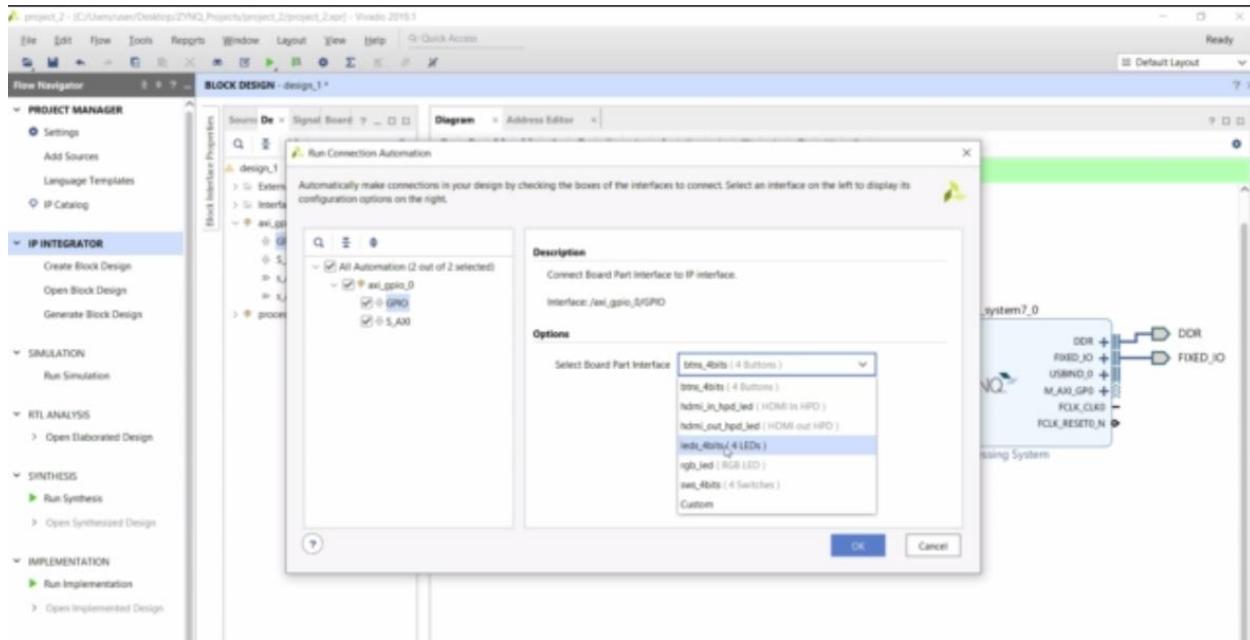


Double Click on green areas

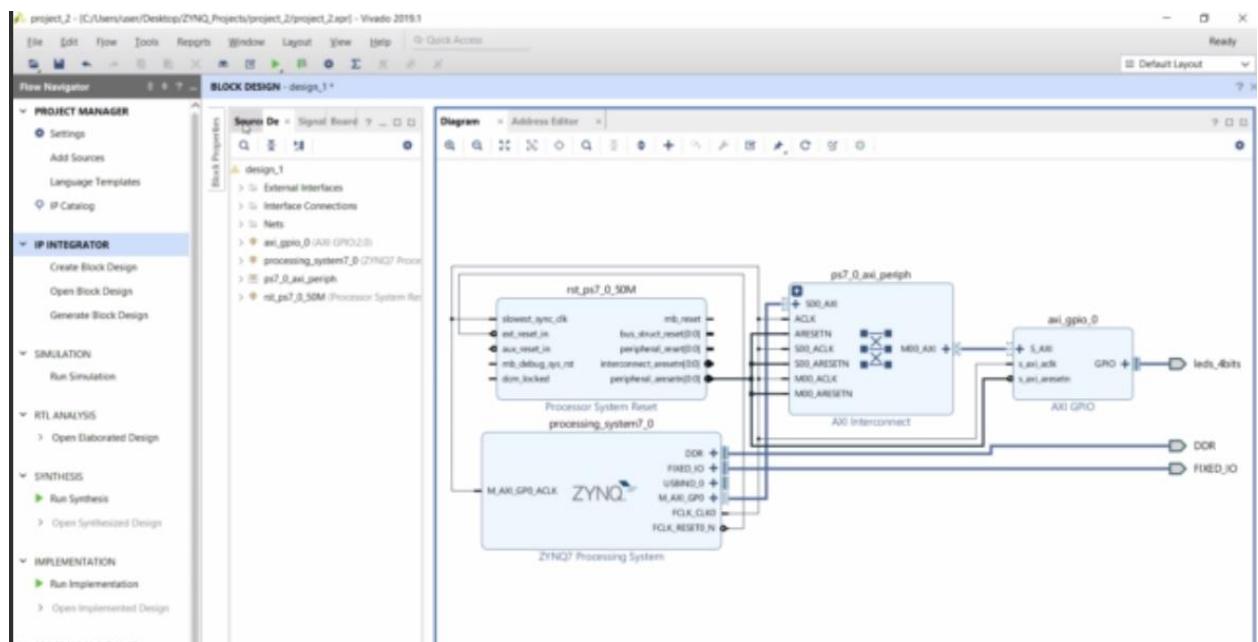


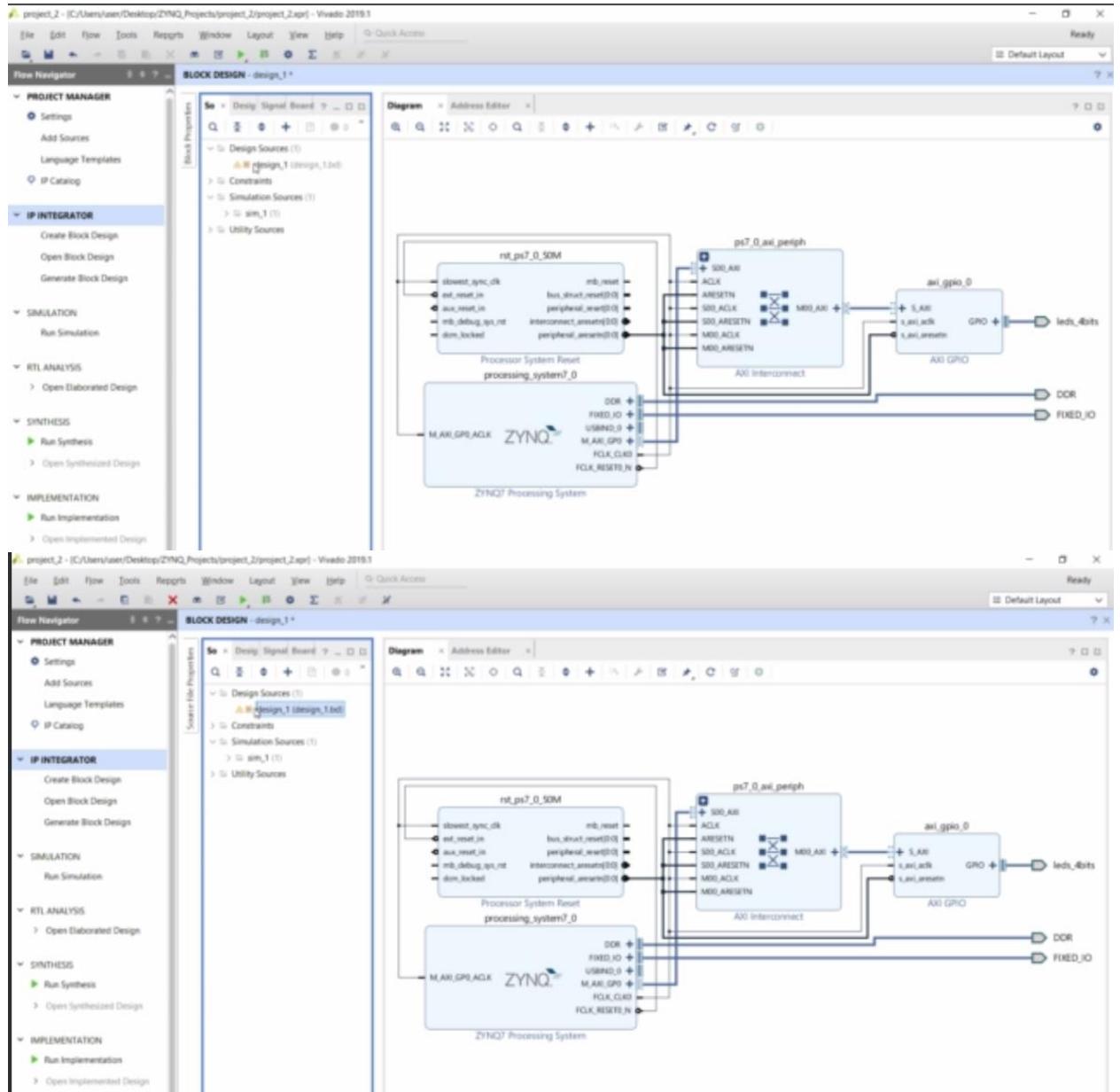


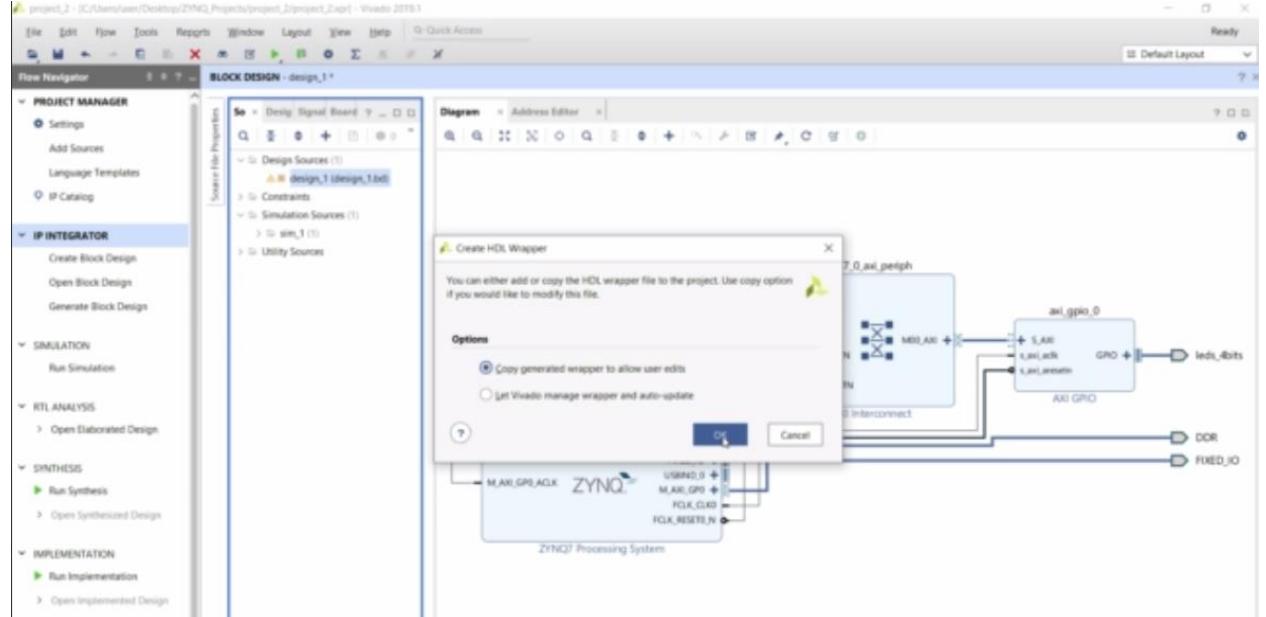
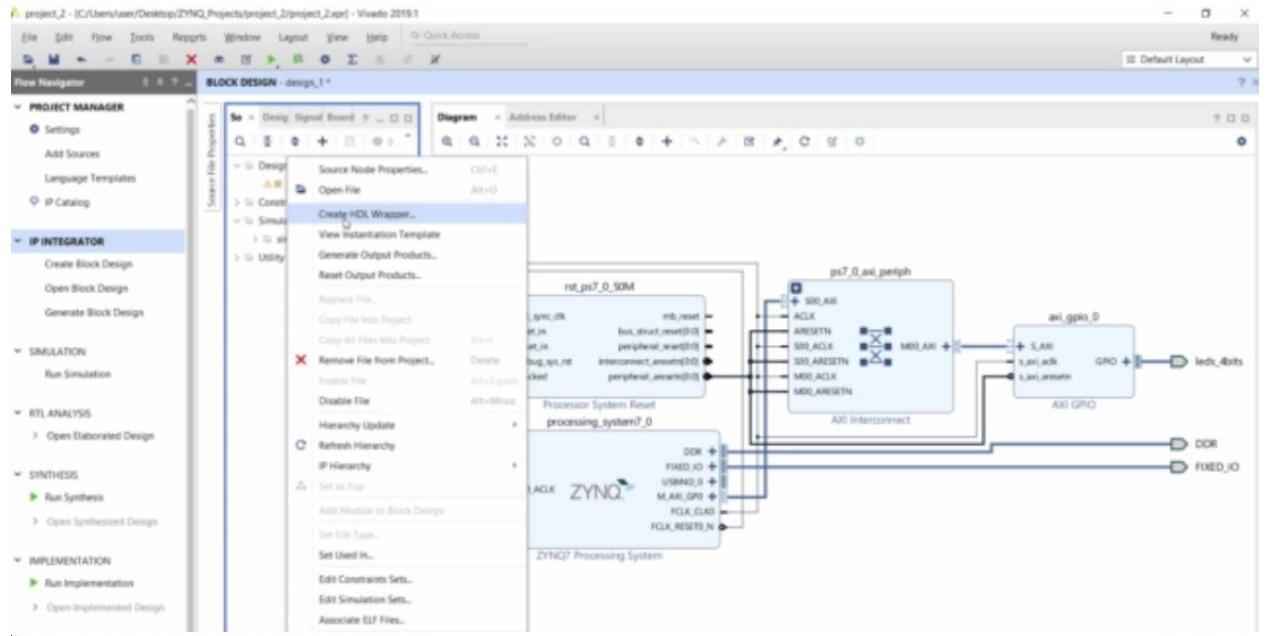


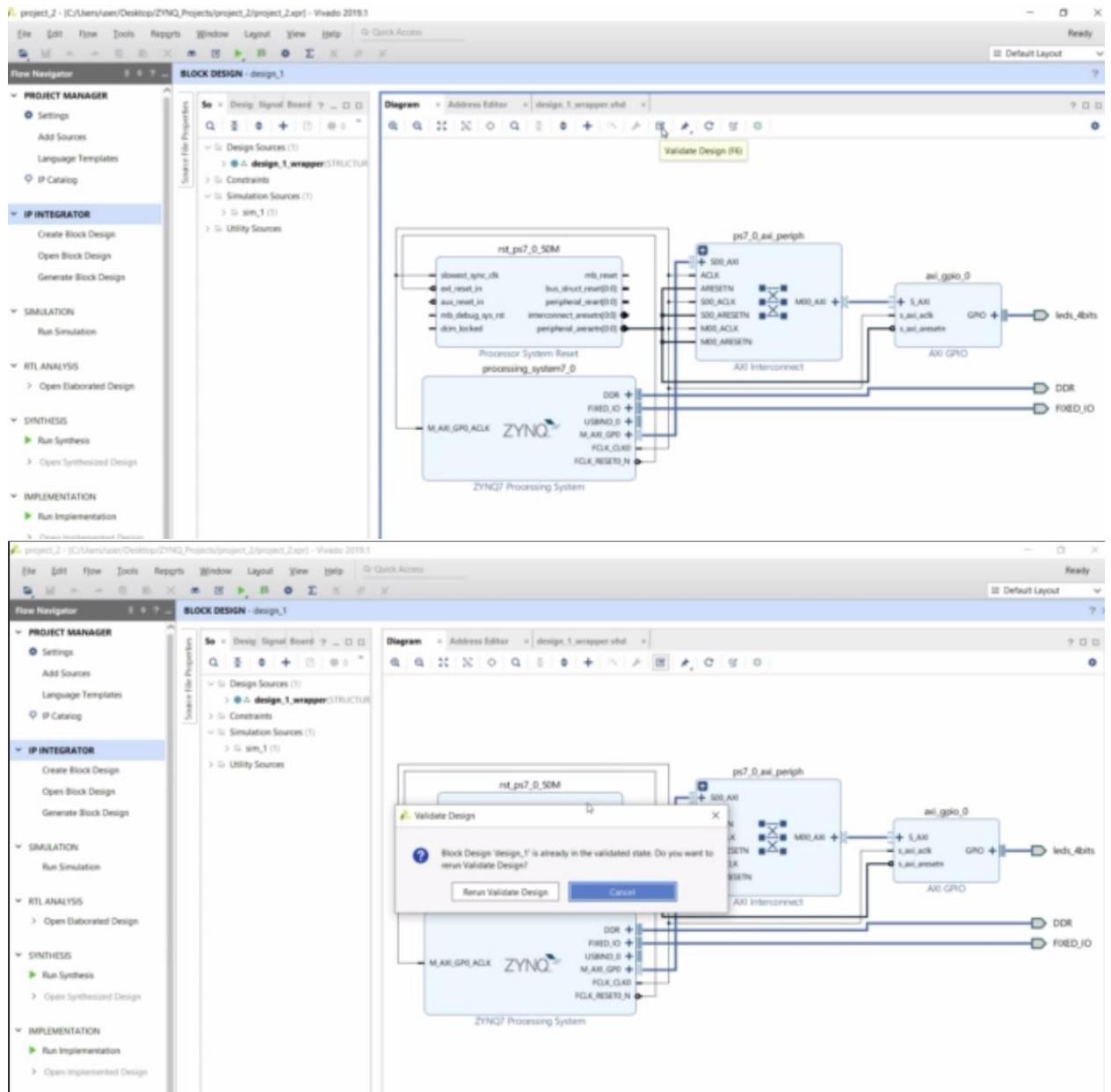


→ Sources:

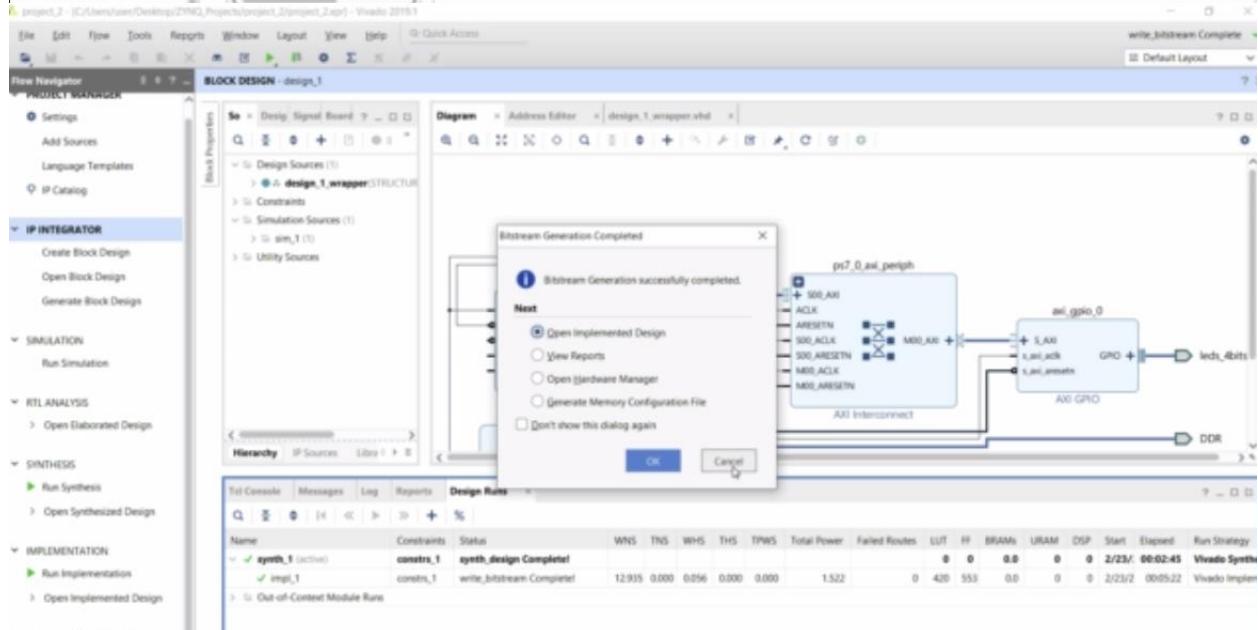
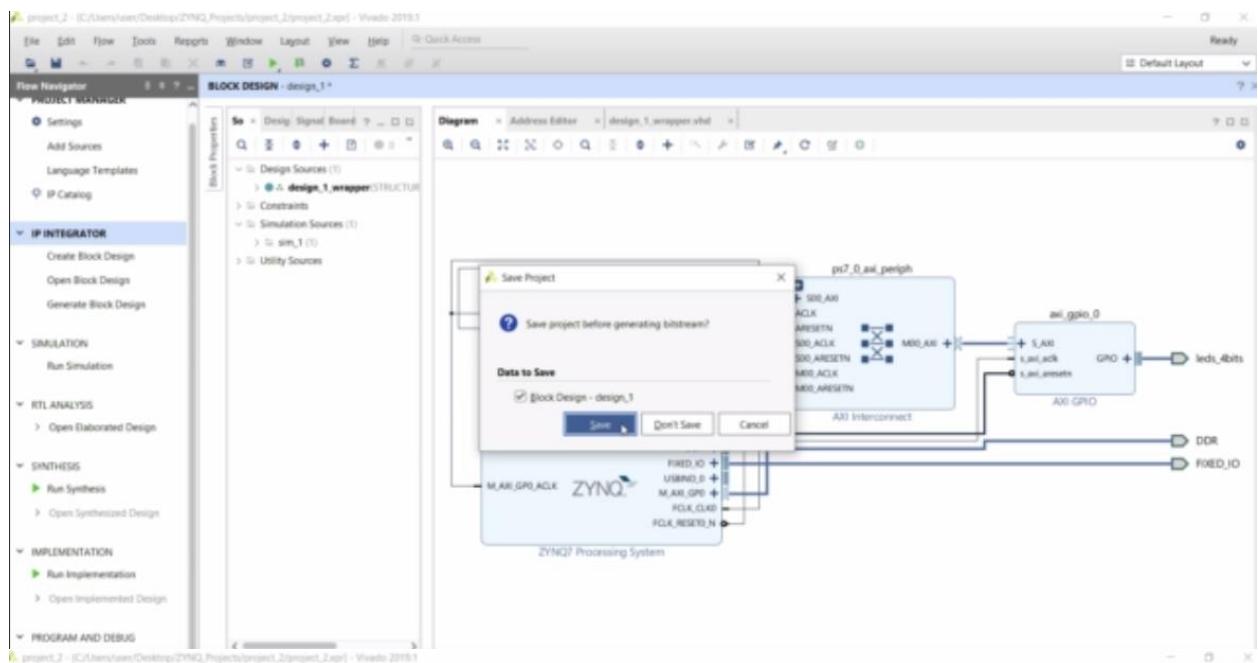


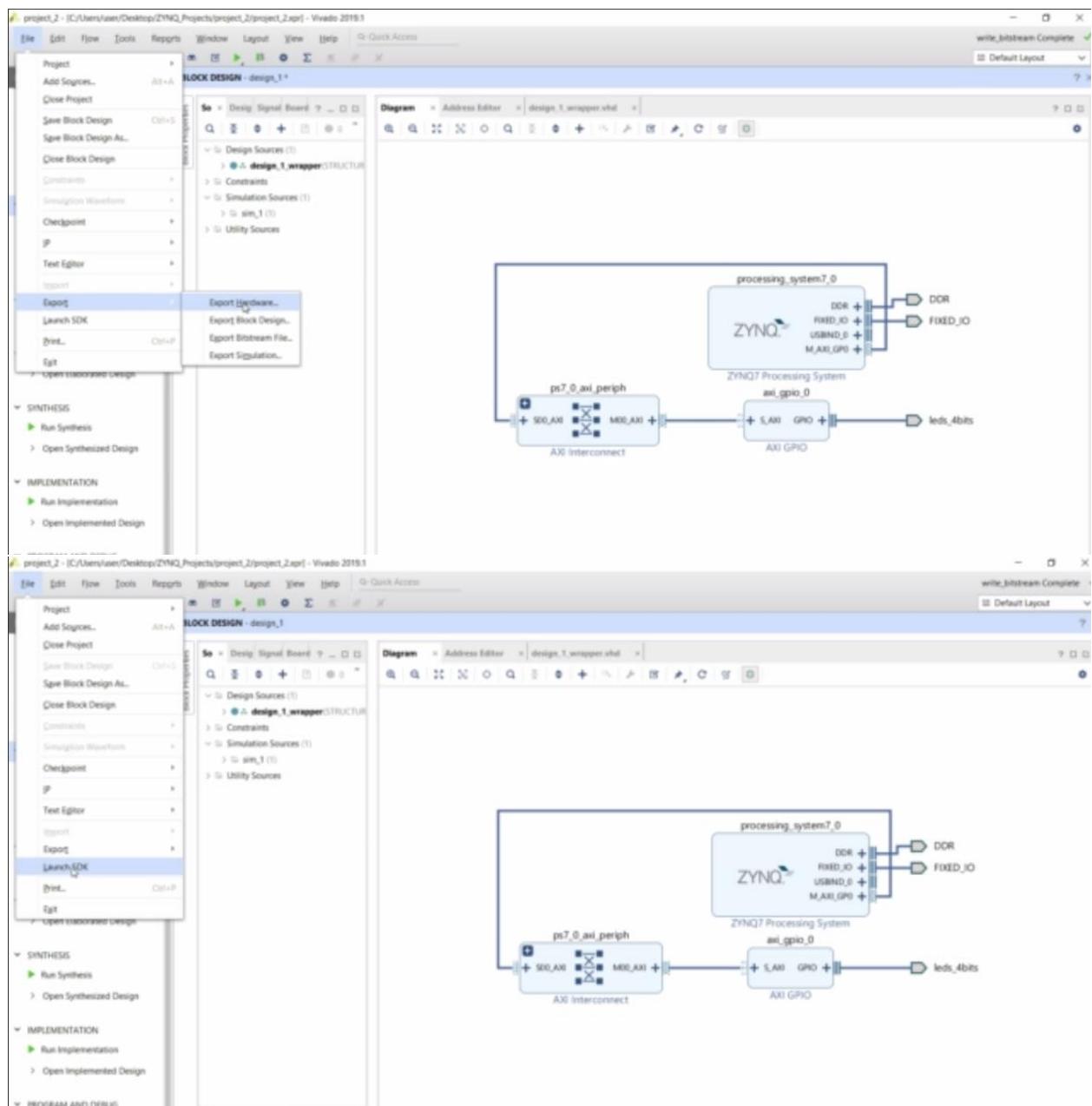


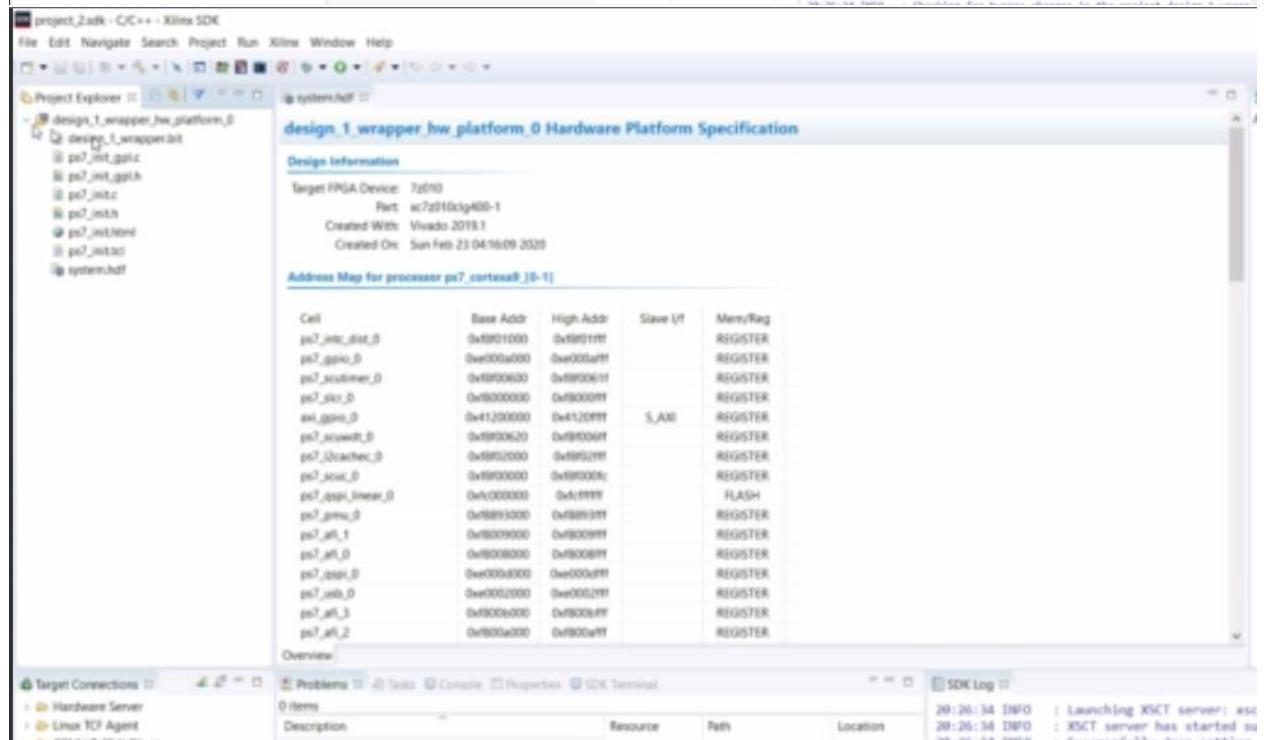
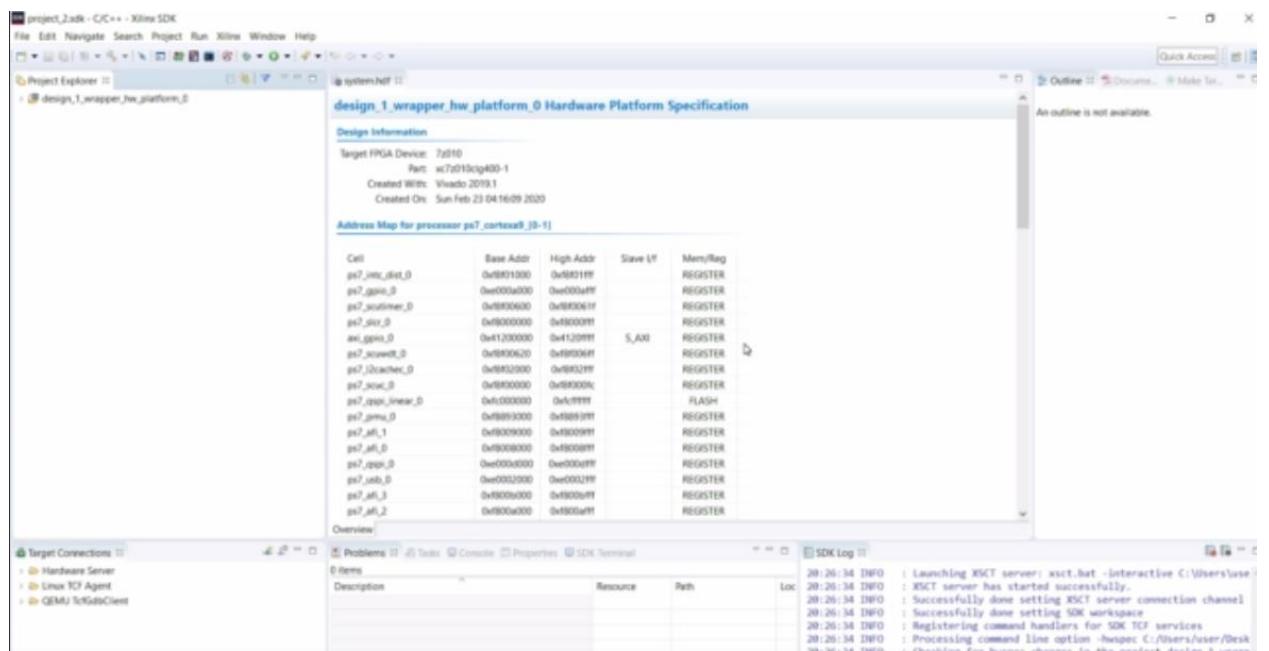


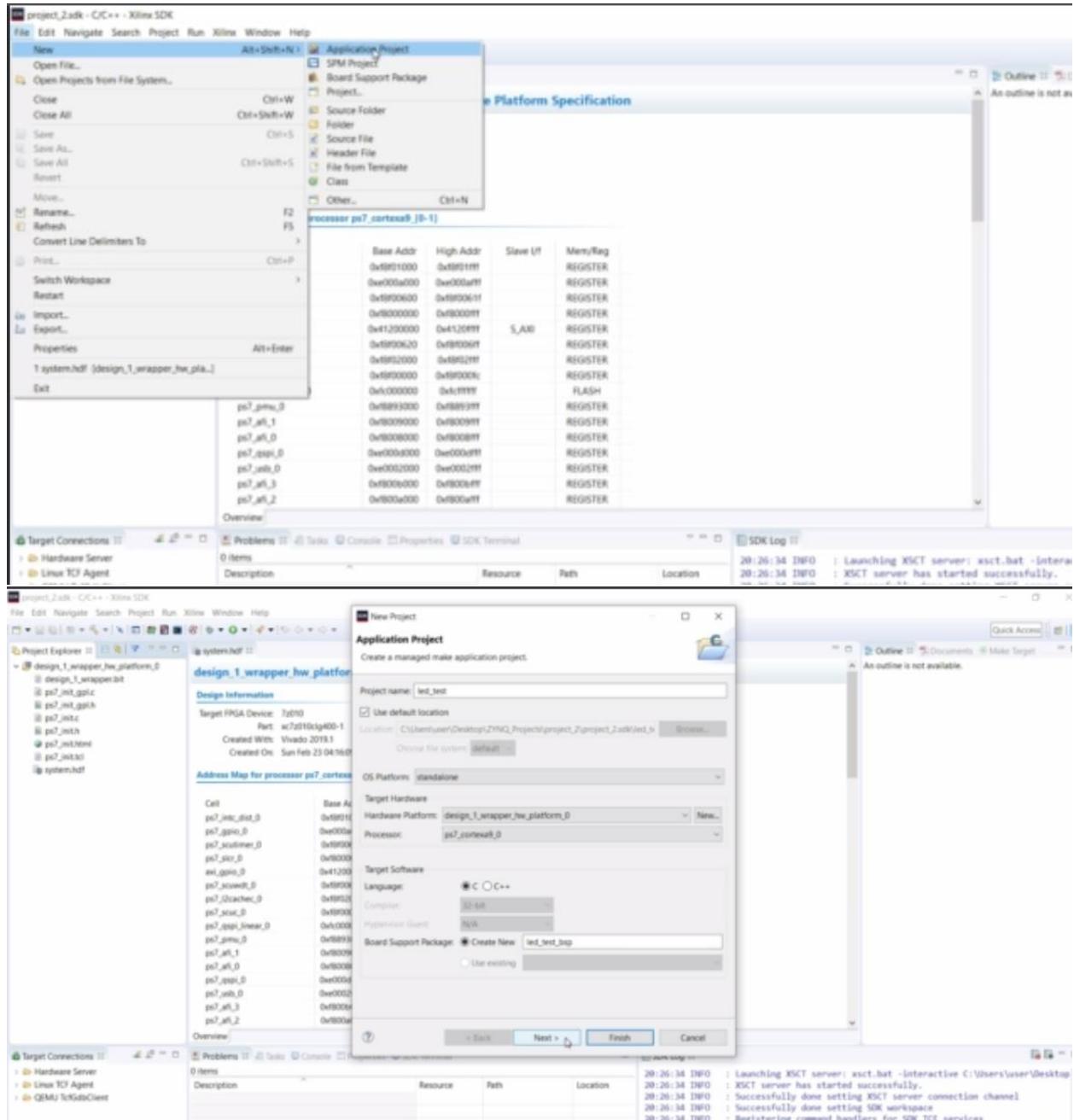


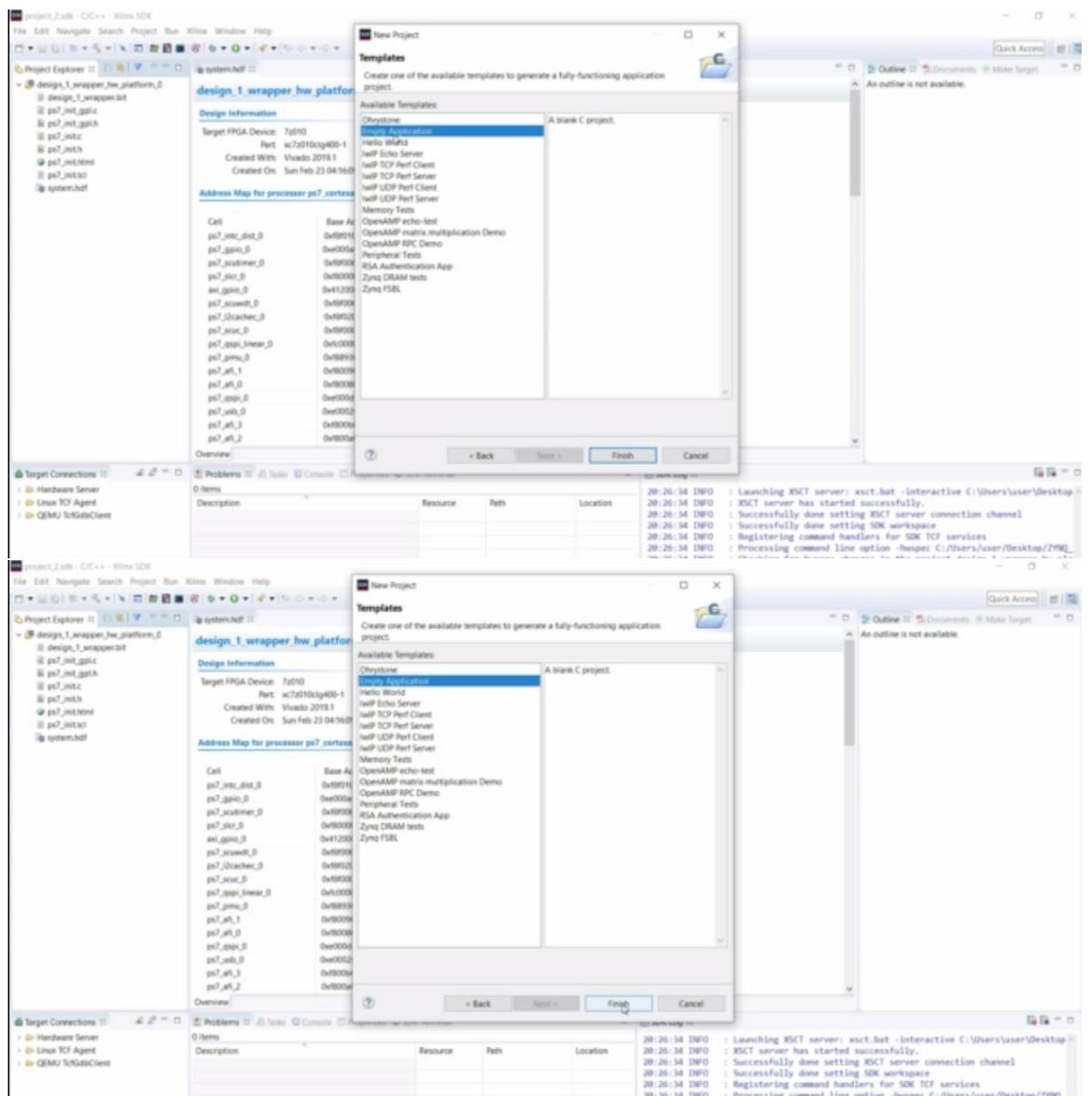
→ Now click on generate bitsream

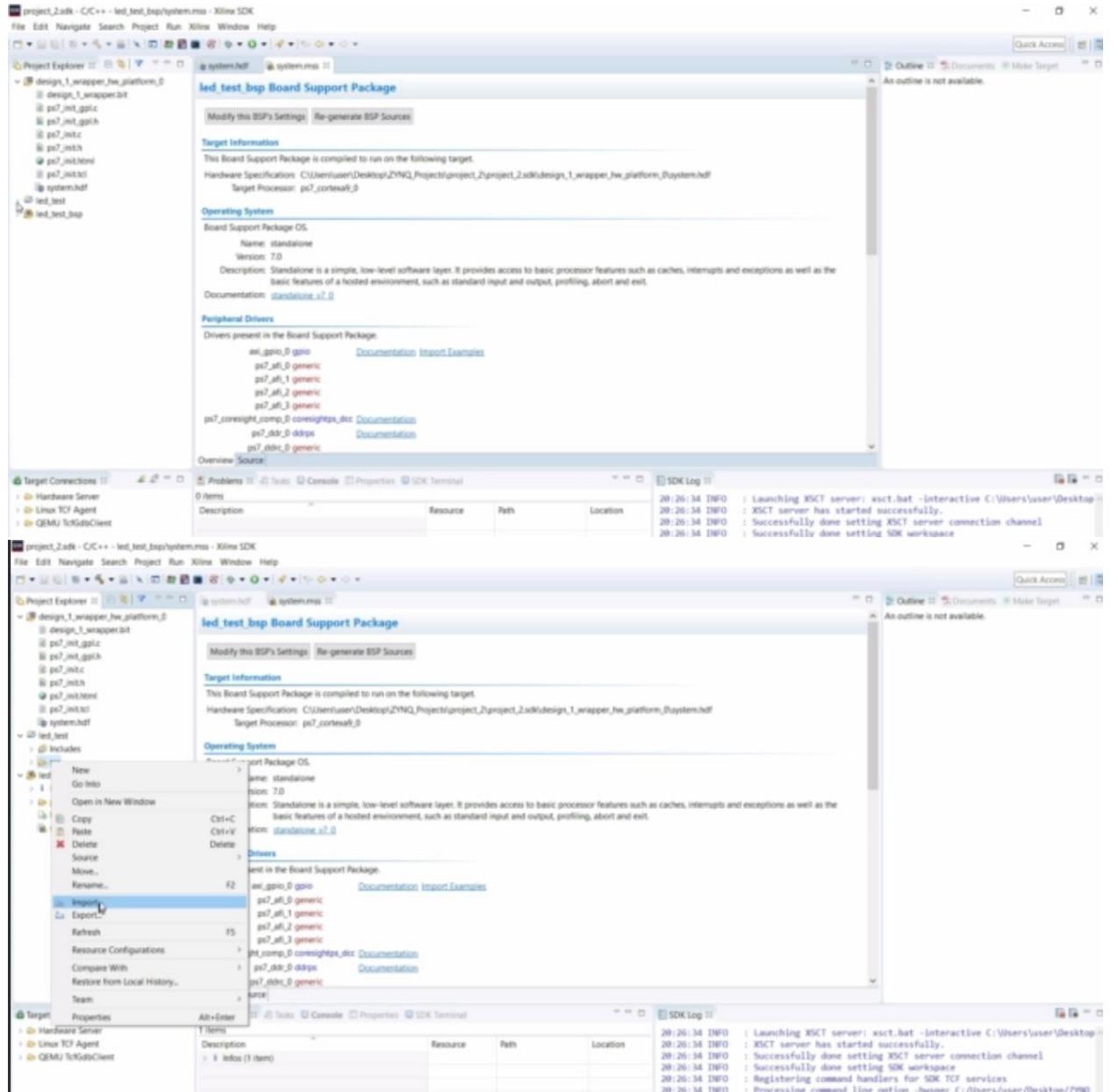


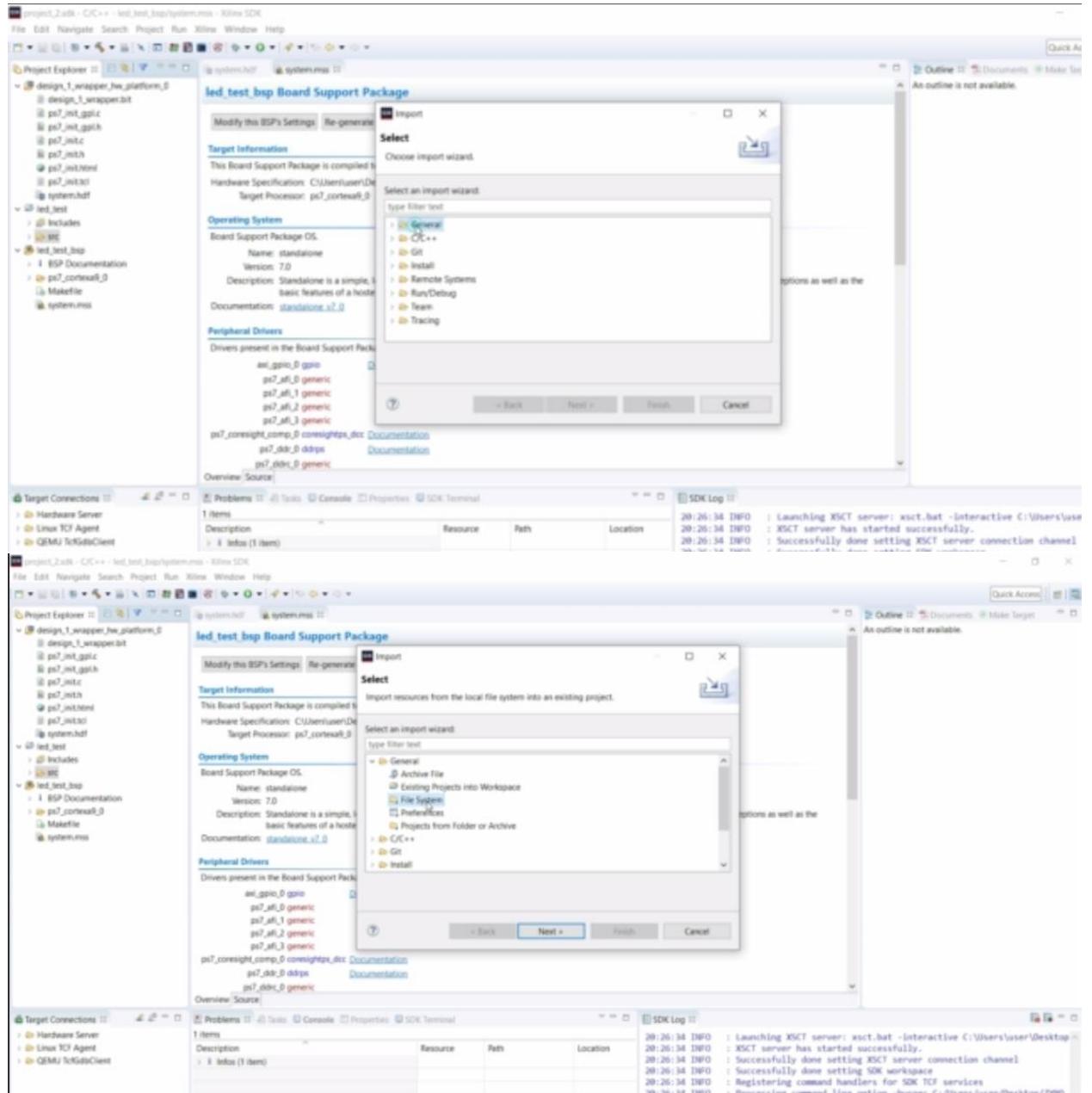


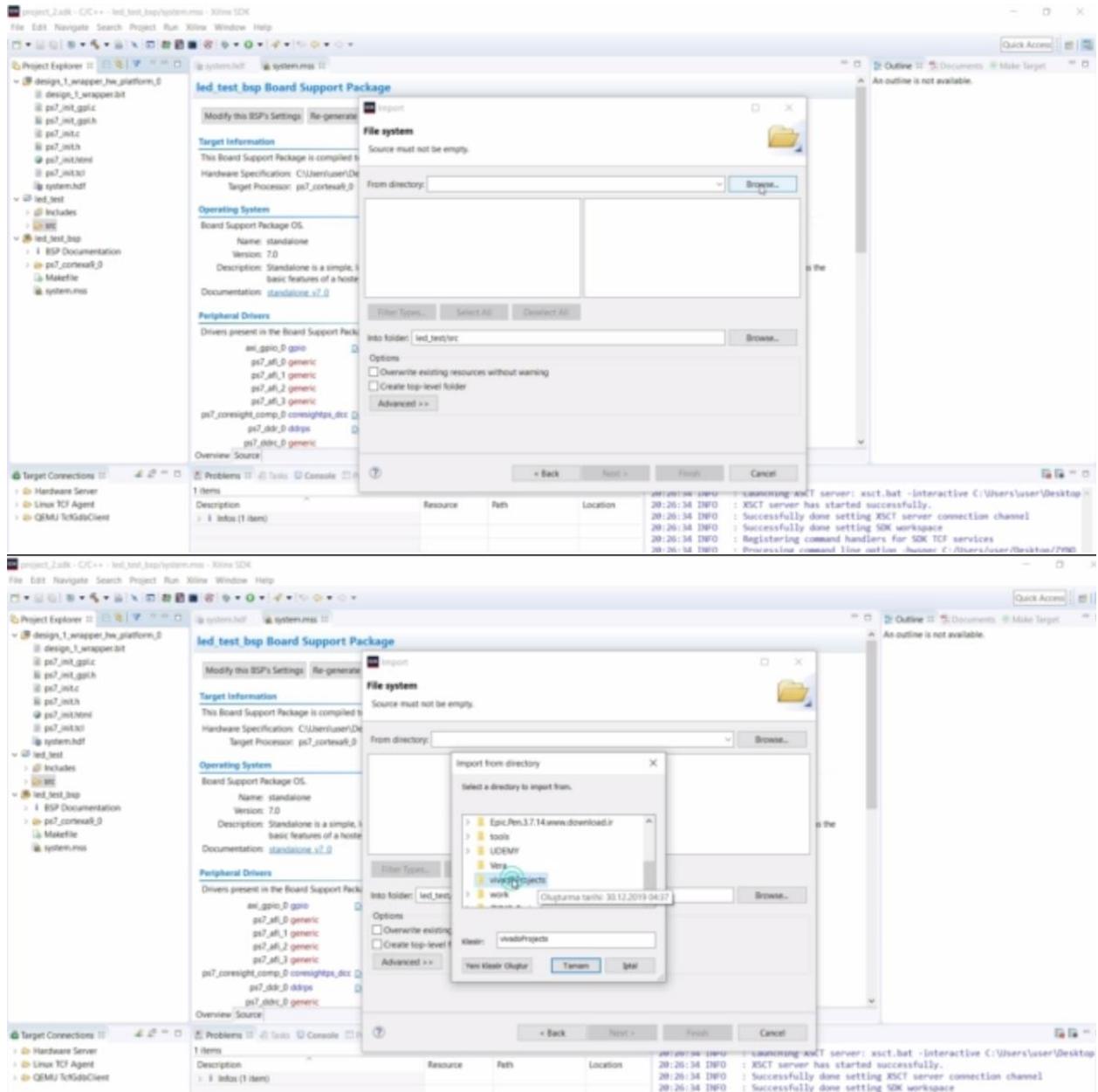


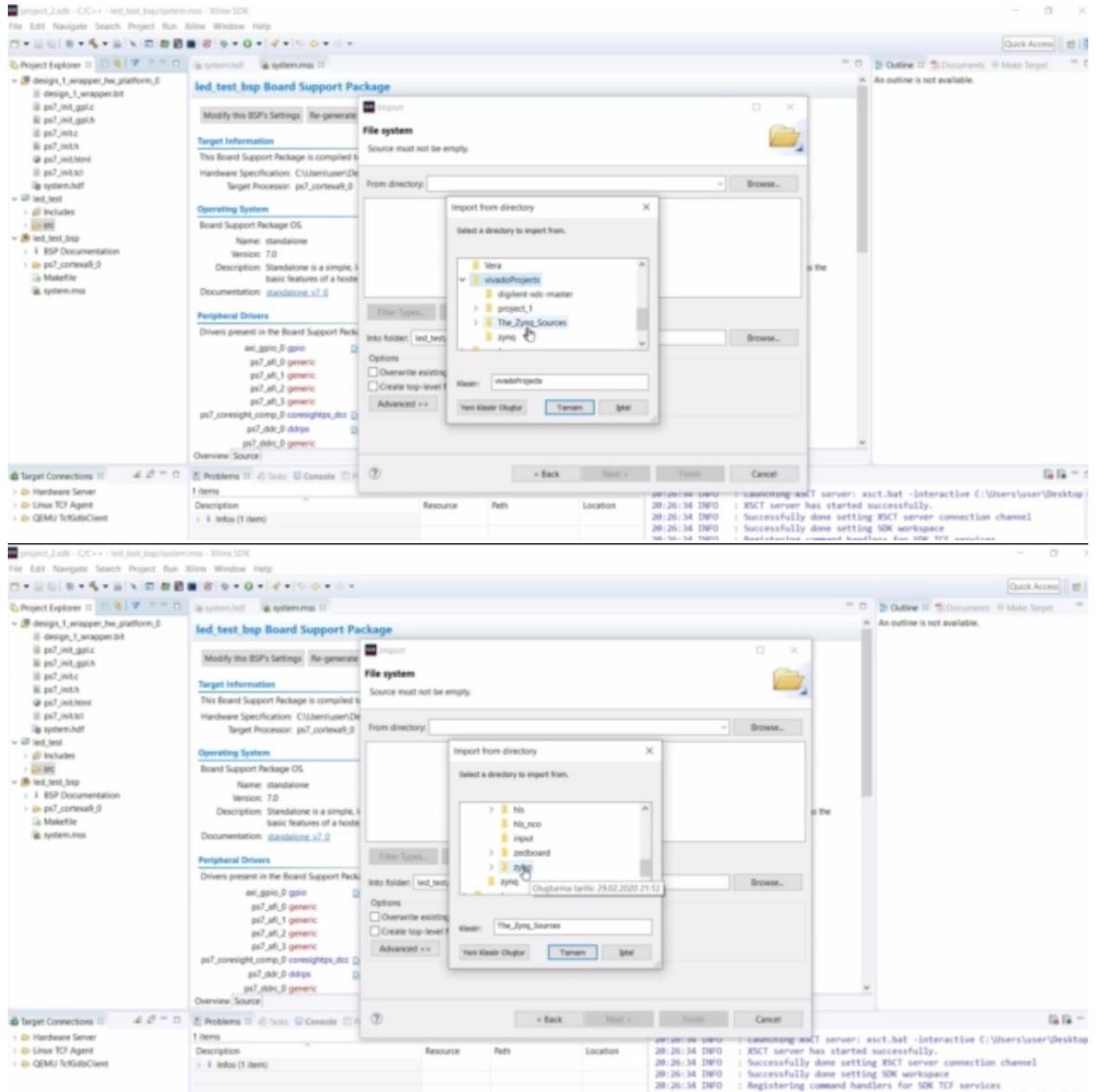


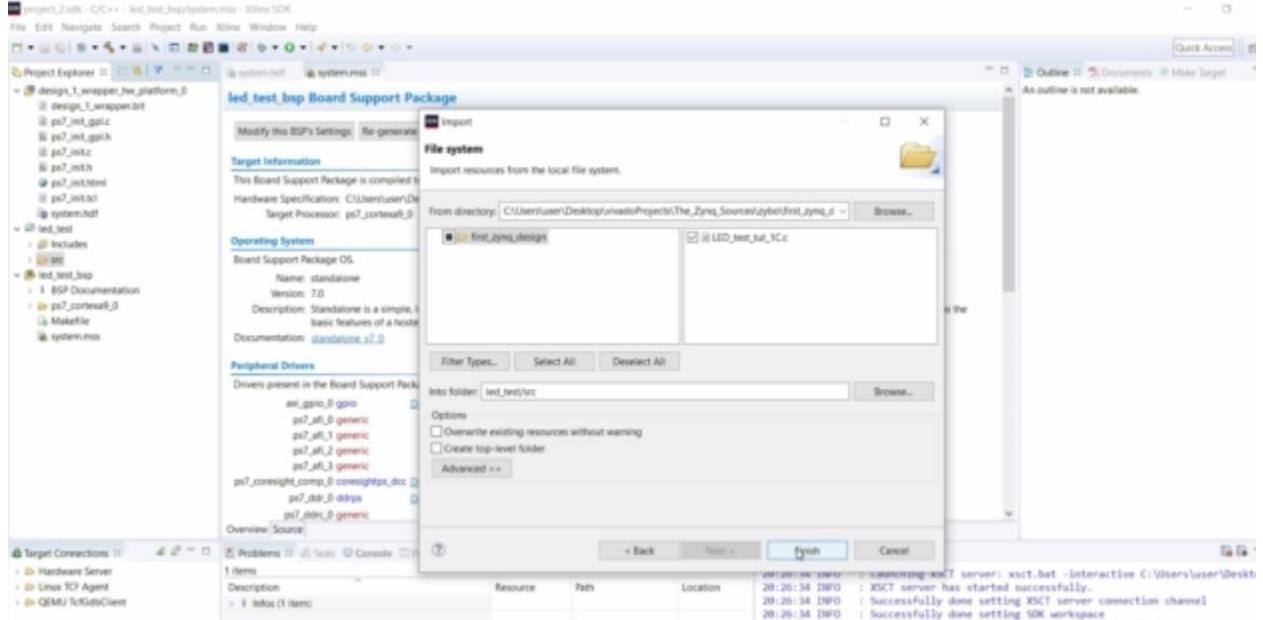
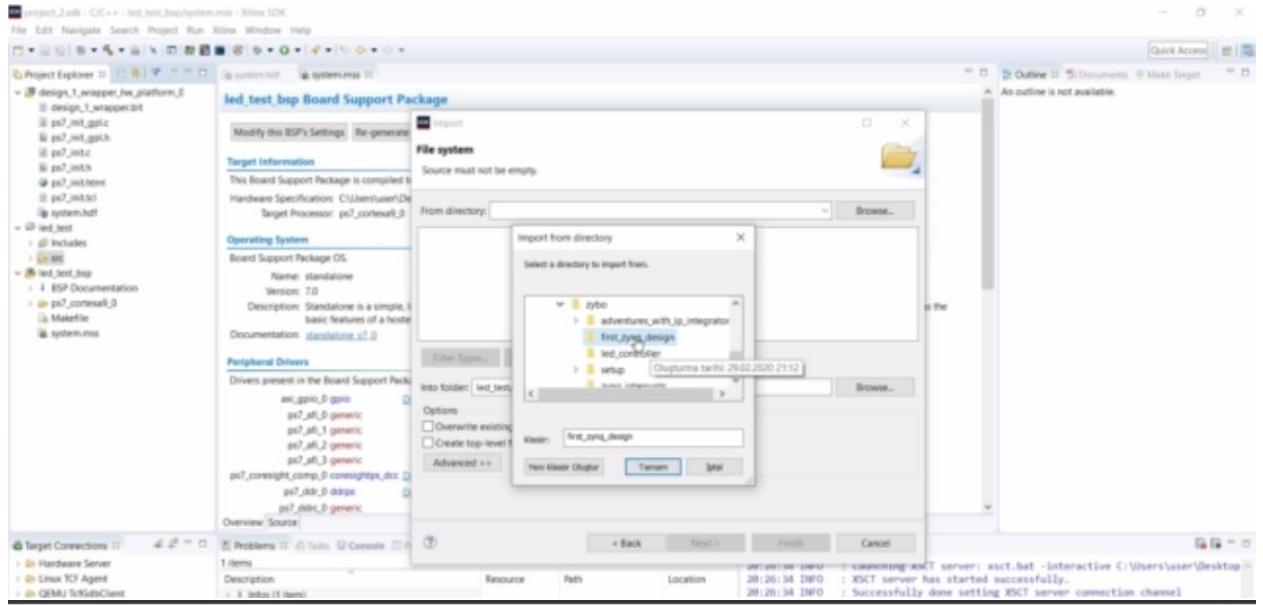












Project Explorer

- design_1_wrapper_hw_platform_0
 - design_1_wrapper.sit
 - ps7_init_gplc.c
 - ps7_init_gplc.h
 - ps7_initc.c
 - ps7_initc.h
 - ps7_initd.c
 - ps7_initd.h
 - system.hdf
- led_test
 - Binaries
 - Includes
 - Debug
 - HW
 - LED_test_hw_NC.c
 - scriptid
 - README.txt
 - XilinxSpec
- led_test_bsp
 - ESP Documentation
 - ps7_conserv0
 - Makefile
 - system.mss

led_test_bsp Board Support Package

Modify this BSP's Settings | Re-generate BSP Sources

Target Information

This Board Support Package is compiled to run on the following target.

Hardware Specification: C:\Users\Desktop\ZYNQ\Projects\project_2\zdk\design_1_wrapper_hw_platform\bsp\system.mss

Target Processor: ps7_cortexa9_0

Operating System

Board Support Package OS.

Name: standalone
Version: 7.0

Description: Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Documentation: [standalone.v7.0](#)

Peripheral Drivers

Drivers present in the Board Support Package:

axi_gpio_0_gplc	Documentation	Import Examples
ps7_af1_0_generic		
ps7_af1_1_generic		
ps7_af1_2_generic		
ps7_af1_3_generic		
ps7_consight_comp_0_consight_axi_dcc	Documentation	
ps7_ddr_0_ddrps	Documentation	
ps7_ddr_0_generic		

Overview | **Source**

Target Connections

- Hardware Server
- Linux TCF Agent
- QEMU TcfGdbClient

SDK Log

```

20-26:34 INFO : Launching xsct server: xsct.bat -interactive C:\1
20-26:34 INFO : xsct server has started successfully.
20-26:34 INFO : Successfully done setting xsct server connection
20-26:34 INFO : Successfully done setting SDK workspace

```

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer

- design_1_wrapper_hw_platform_0
 - design_1_wrapper.sit
 - ps7_init_gplc.c
 - ps7_init_gplc.h
 - ps7_initc.c
 - ps7_initc.h
 - ps7_initd.c
 - ps7_initd.h
 - system.hdf
- led_test
 - Binaries
 - Includes
 - Debug
 - HW
 - LED_test_hw_NC.c
 - scriptid
 - README.txt
 - XilinxSpec
- led_test_bsp
 - ESP Documentation
 - ps7_conserv0
 - Makefile
 - system.mss

Show View

- C/C++ Projects
- Console
- Documents
- Include Browser
- Main Target
- Navigator
- Outline **(selected)**
- Problem Details
- Properties
- Project Explorer
- Search
- Tasks
- Tools
- Other...

Alt+Shift+Q, C

Alt+Shift+Q, X

Alt+Shift+Q, S

Alt+Shift+Q, Q

Outline

new naming conventions in Vivado 2013.3

over to provide communication between lock implemented in the Zynq Programmable logic (PL). The A

The provided code

GPIO block, which

driver to write to the memory mapped AXI

Code

```

/* Include Files */
#include "parameters.h"
#include "apcie.h"
#include "status.h"
#include "xil_printf.h"

/* Definitions */
#define GPIO_DEVICE_ID XPAR_AXI_GPIO_DEVICE_ID /* GPIO device that LEDs are connected to */
#define ID_dv9 /* Initial LED value 3000 */
#define ID_DELAY 10000000 /* Software delay length */
#define ID_CHANNEL 1 /* GPIO port for LEDs */
#define printf xil_printf /* smaller, optimised printf */

Xgpio Gpio; /* GPIO Device driver instance */

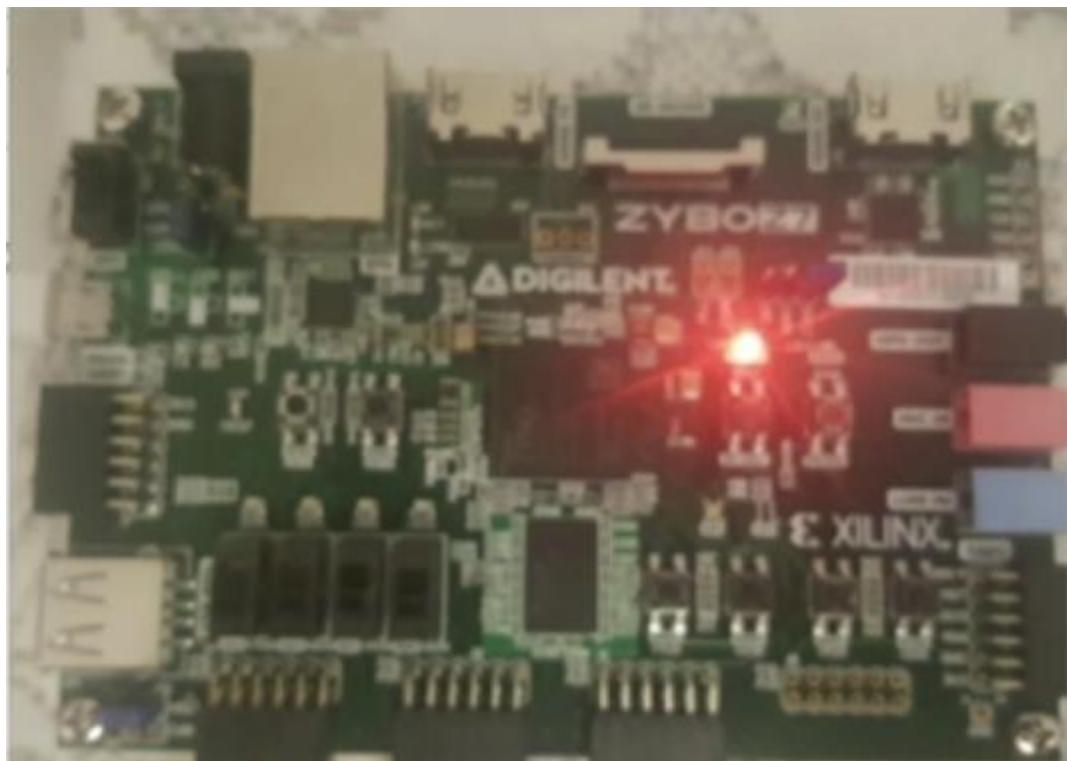
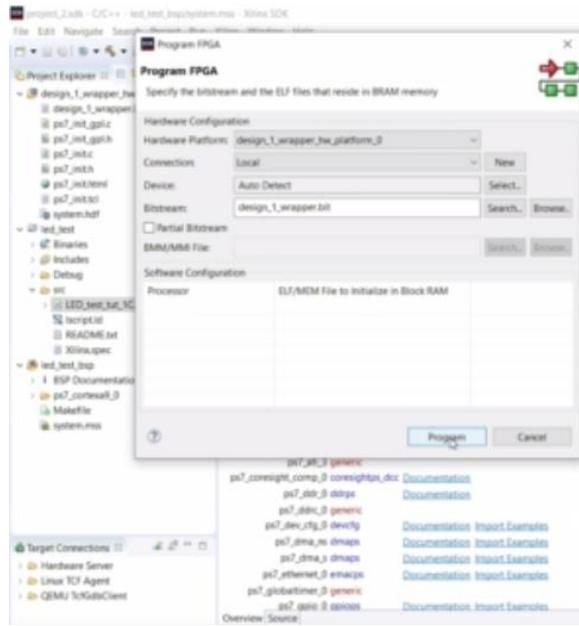
/* LEDOutputExample(void)
{
    volatile int Delay;
    status;
    int led = LED0; /* Hold current LED value. Initialize to LED definition */
}

```

Target Connections

- Hardware Server
- Linux TCF Agent
- QEMU TcfGdbClient

Tasks



→when the program is complete we'll get green light which shows that our bitsream file is being uploaded:



→next we'll program the ARM microprocessor in the SOC:



