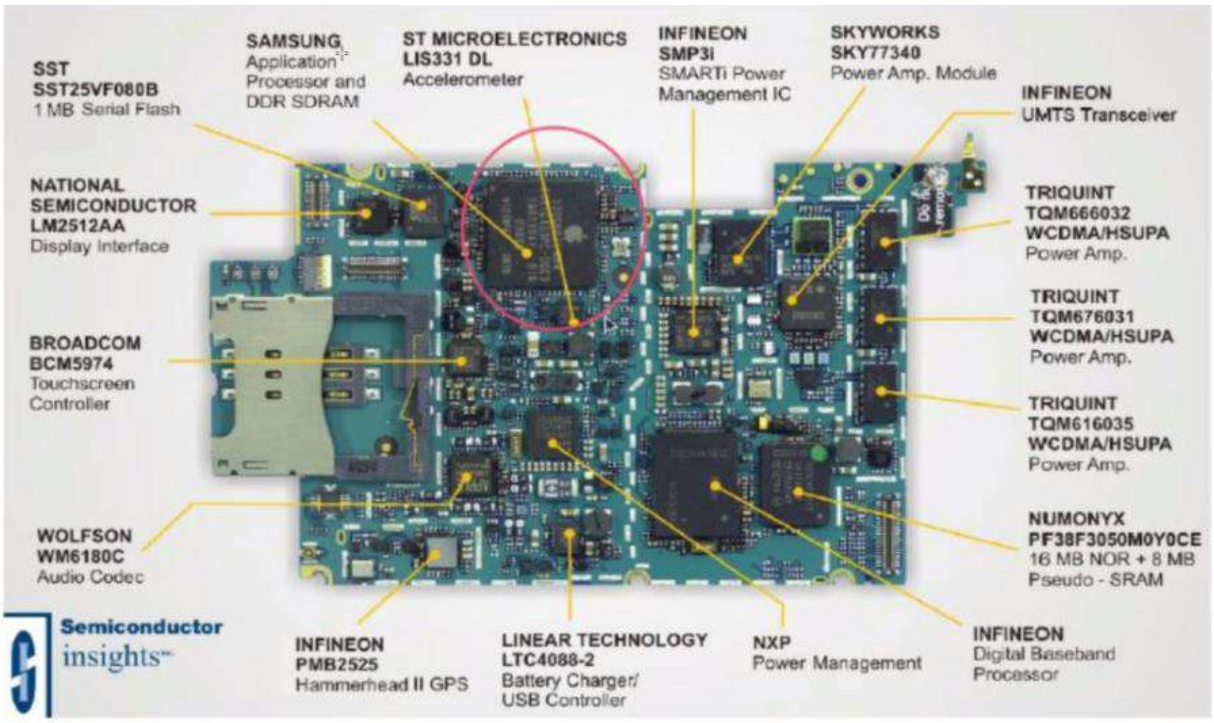
A SoC is a complete system on a chip. A 'system' can include microprocessors, FPGAs, memory and peripherals.

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.

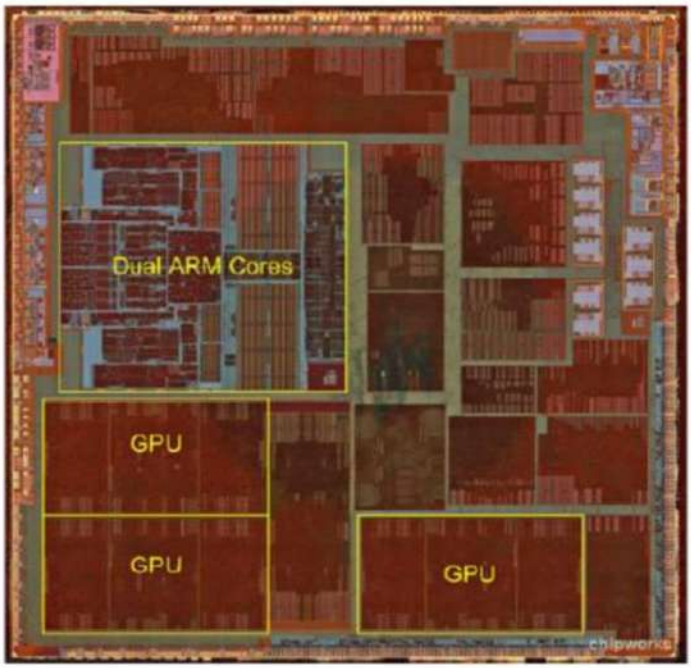


Fig. 3

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

AT91SAM: AT91SAM is an ARM-based SOC

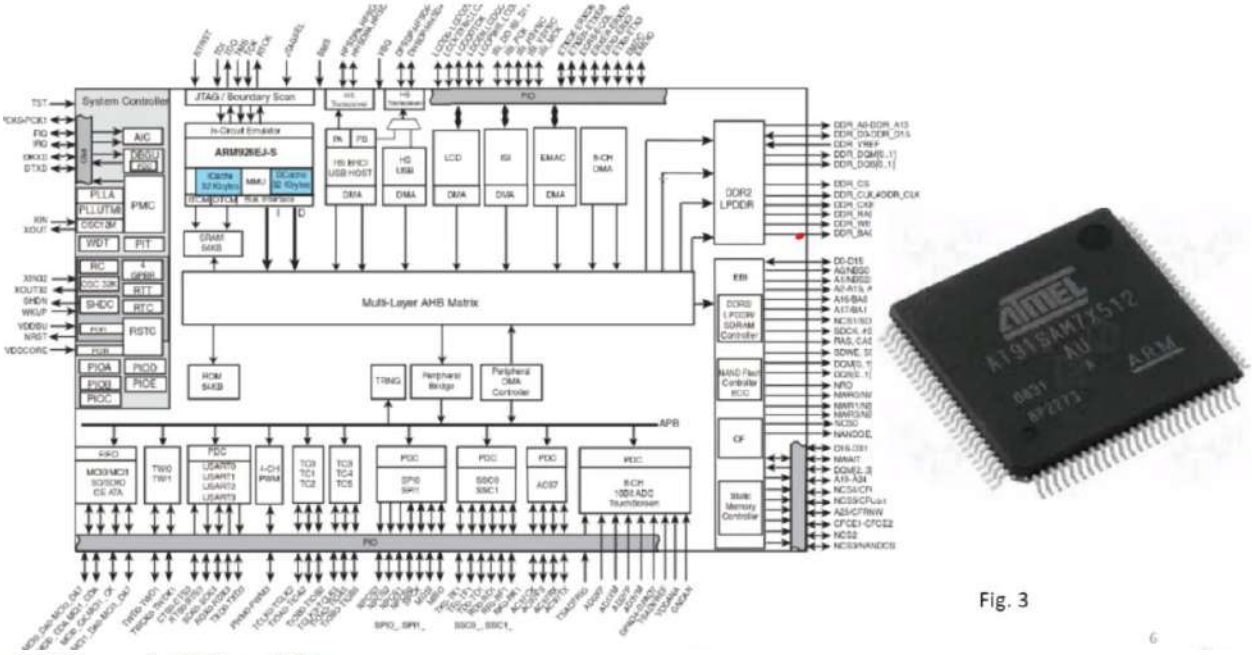
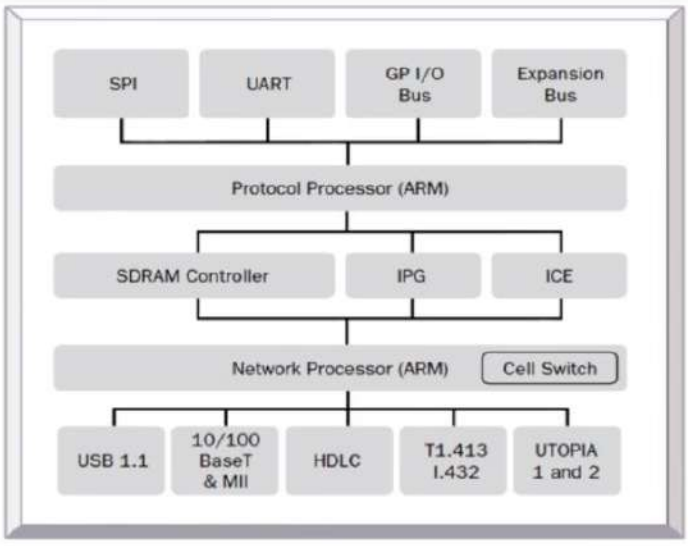


Fig. 3

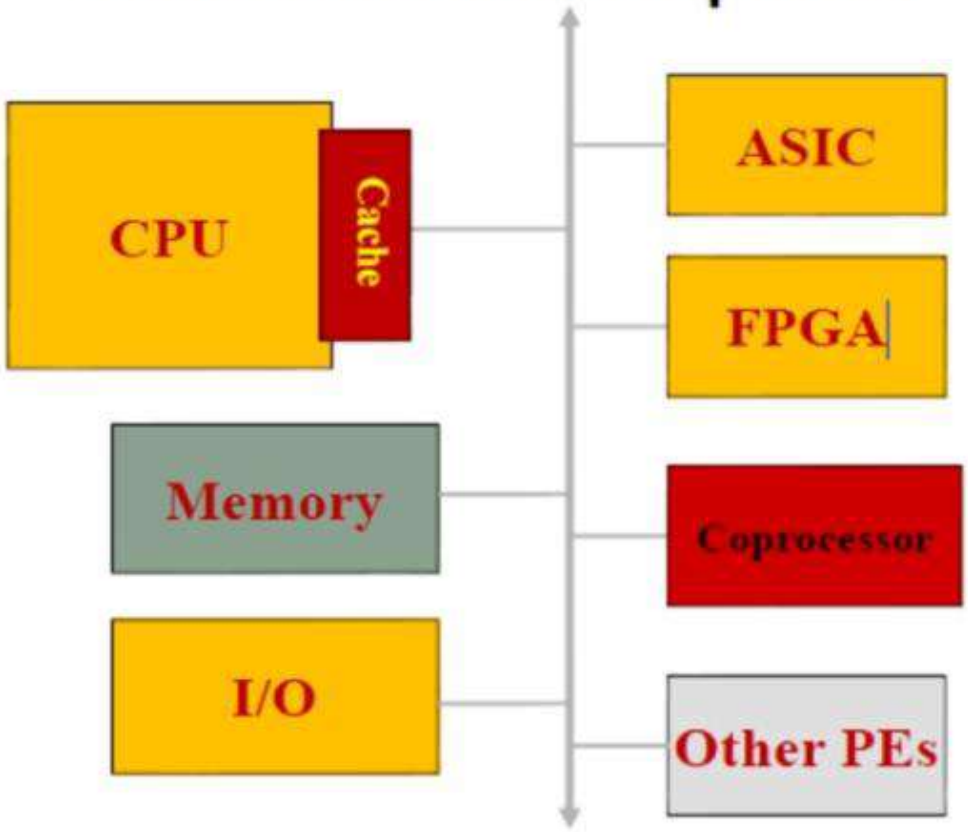
SOC Example: Helium 210

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:



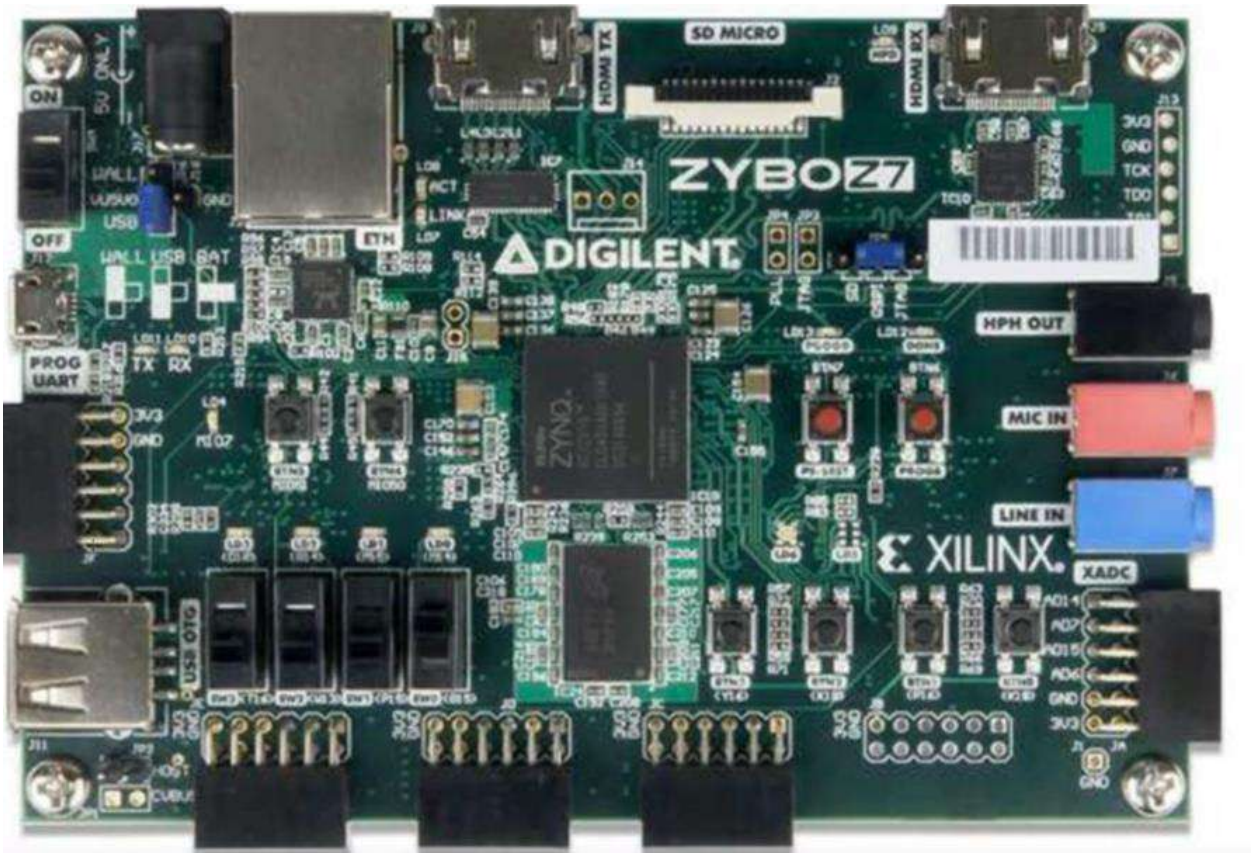
SOC Developments Boards:

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.

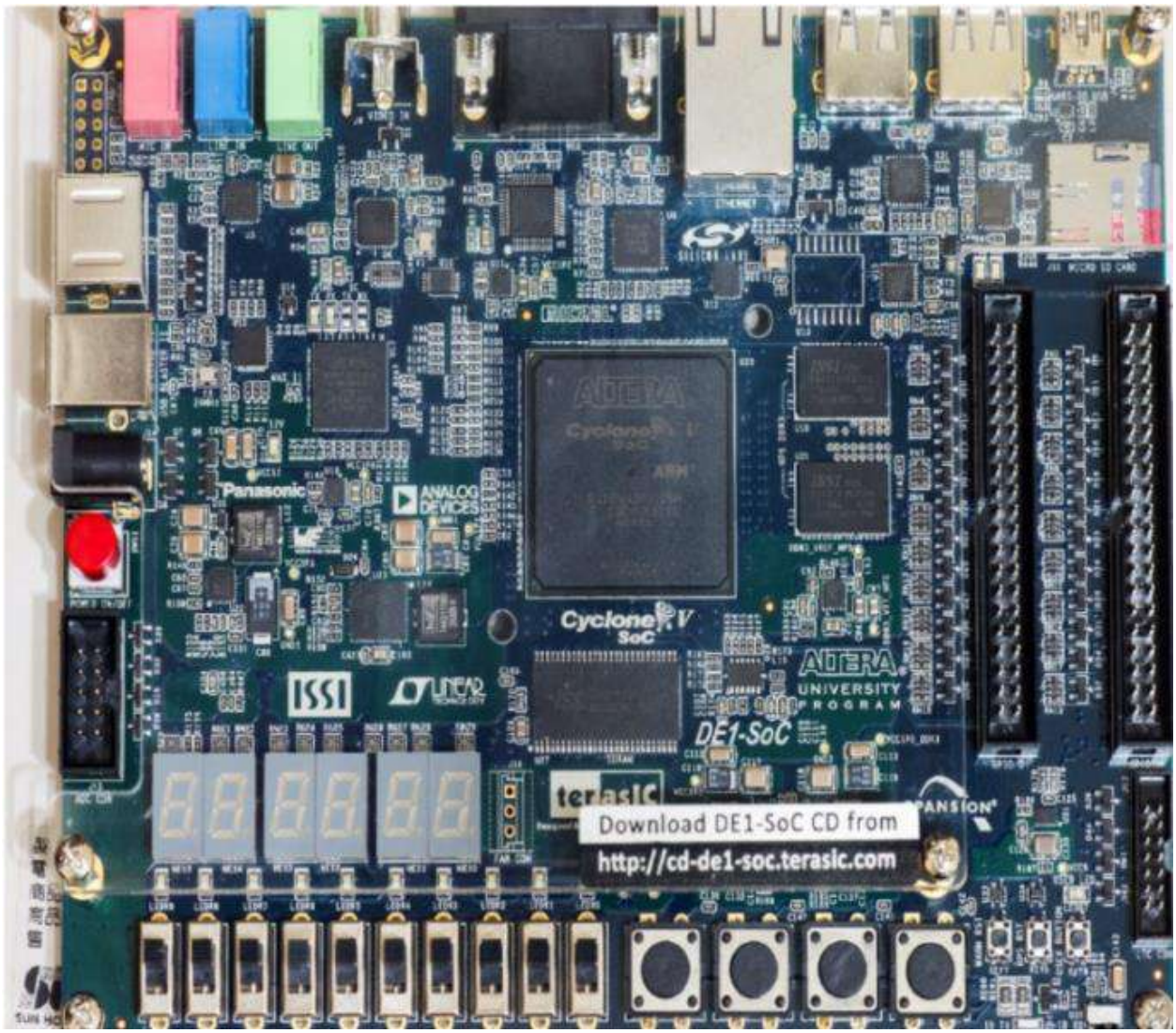
XILINX provides Zynq SOC development boards.

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

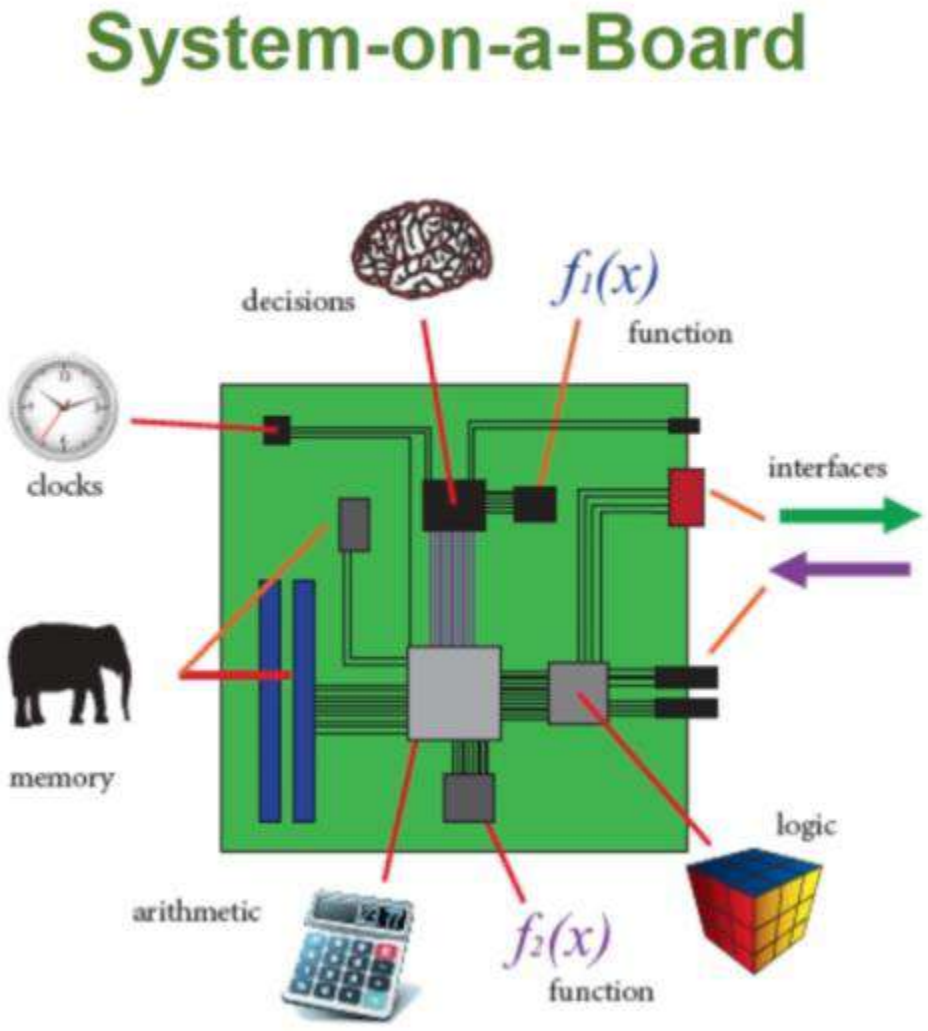
→the board we'll use



→Altera(Intel) board

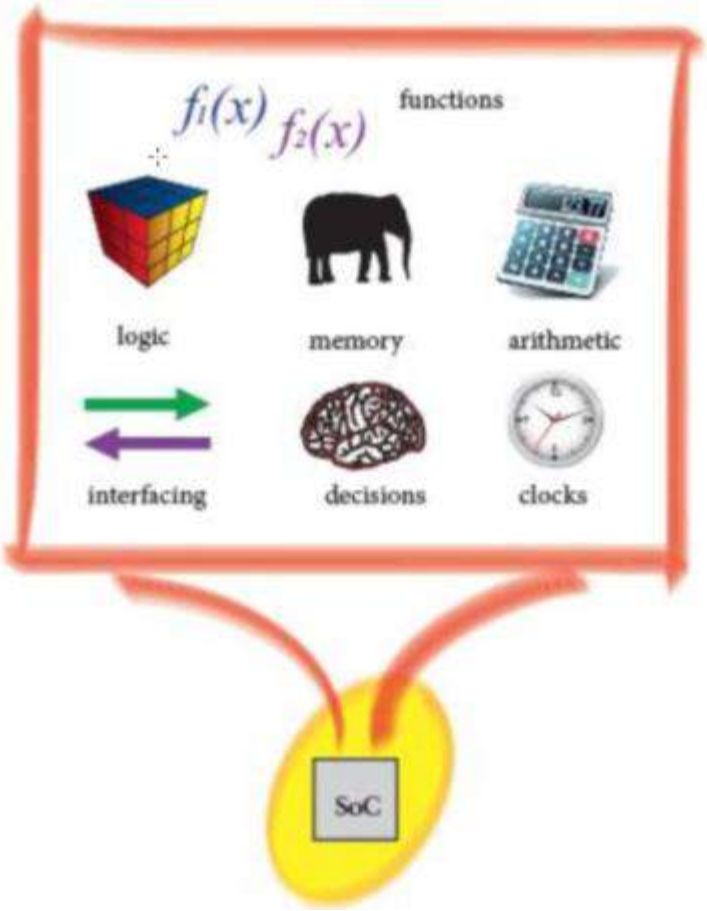

Panasonic
ANALOG DEVICES
Cyclone V SoC
ALTERA
UNIVERSITY PROGRAM
DE1-SoC
terasIC
Download DE1-SoC CD from
http://cd-de1-soc.terasic.com

→PCBs


System-on-a-Board
decisions
$f_1(x)$
function
clocks
interfaces
memory
logic
arithmetic
$f_2(x)$
function

➔ SOCs:

System-on-Chip (SoC)



Zynq-7000 SOC comprises two main parts:

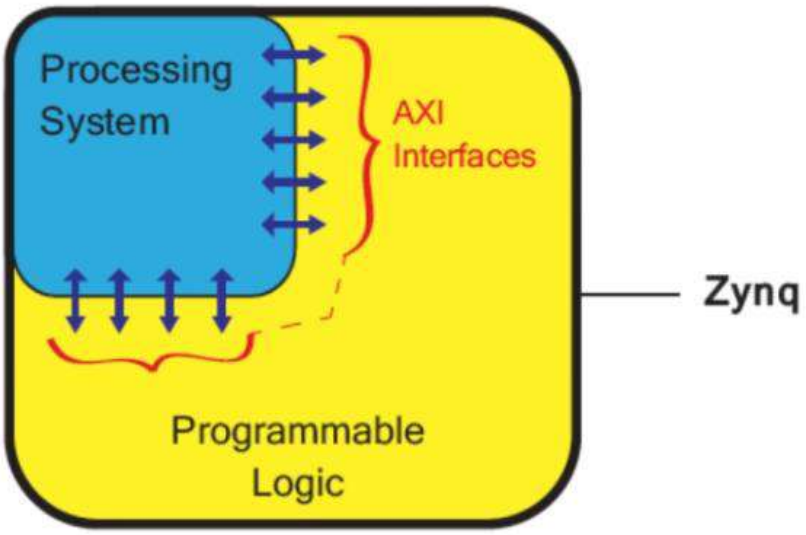
a Processing System (PS) formed around a dual-core ARM Cortex-A9 processor, and Programmable Logic (PL), which is equivalent to that of an FPGA.

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

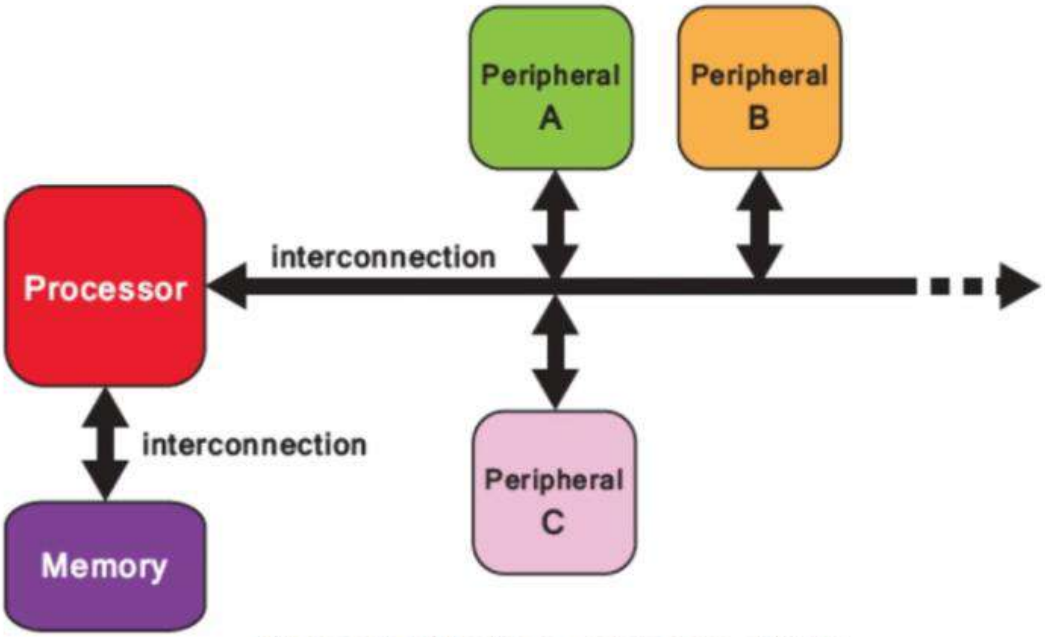
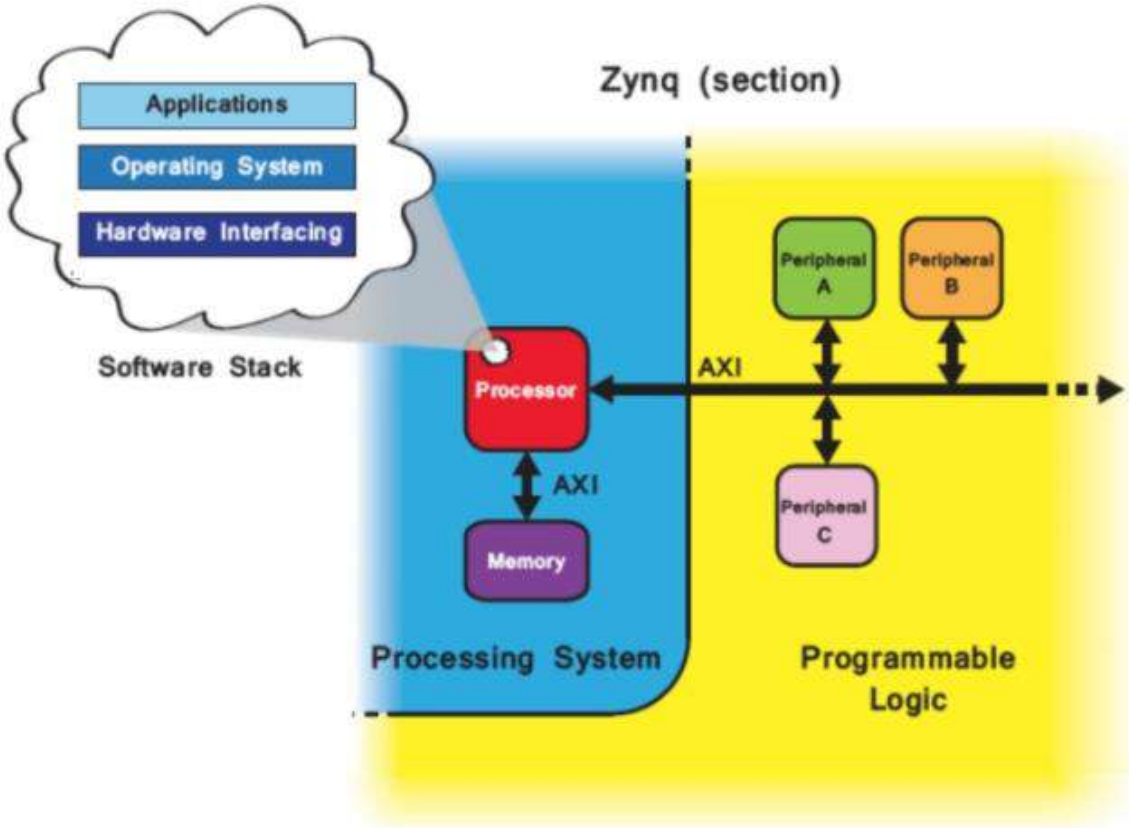


Figure 2-3: Hardware architecture of SOC

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.

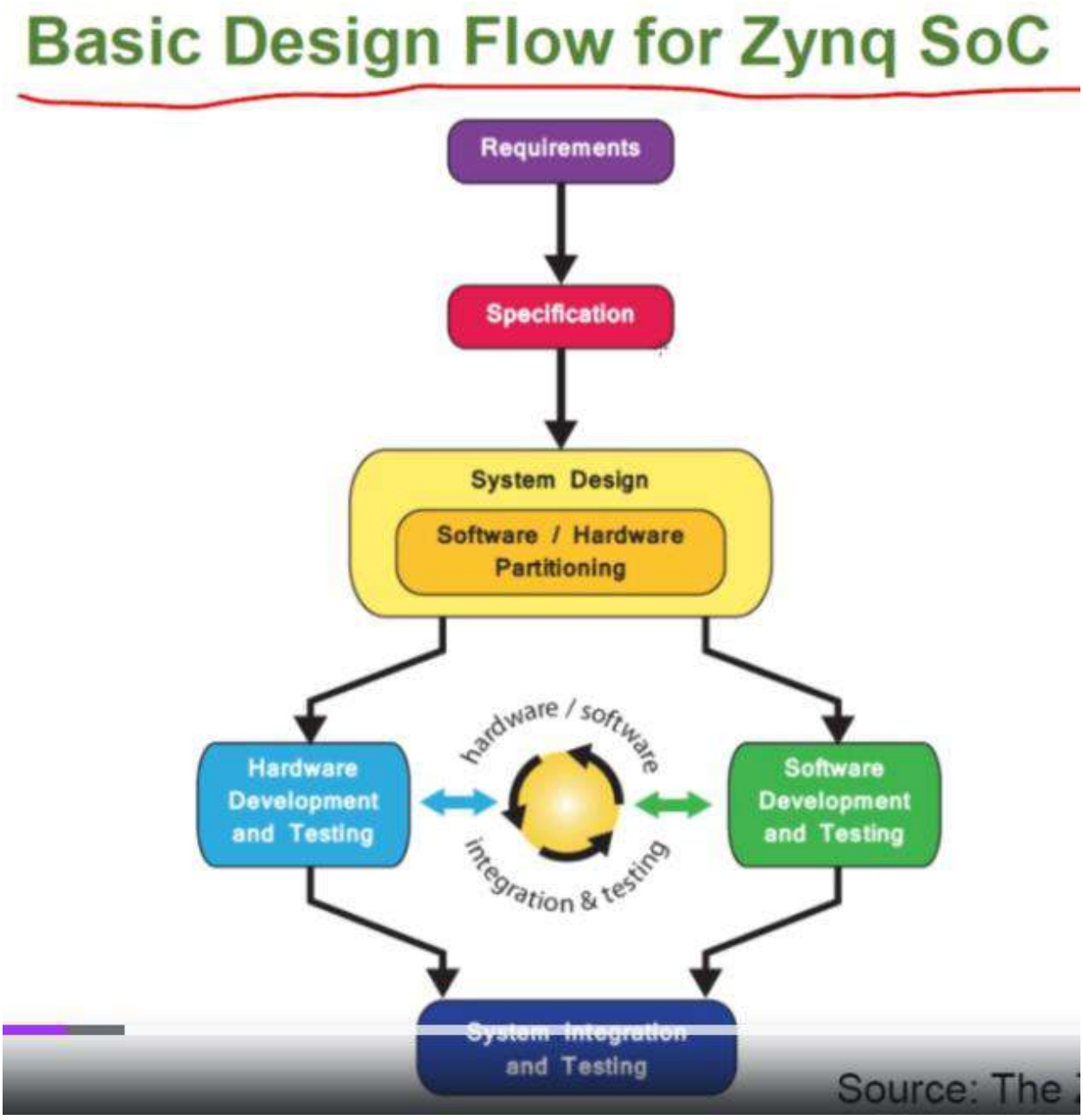
Note that, although a single bus with connected peripherals is shown in Figure 2-3, a processor may severe several connected buses.

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

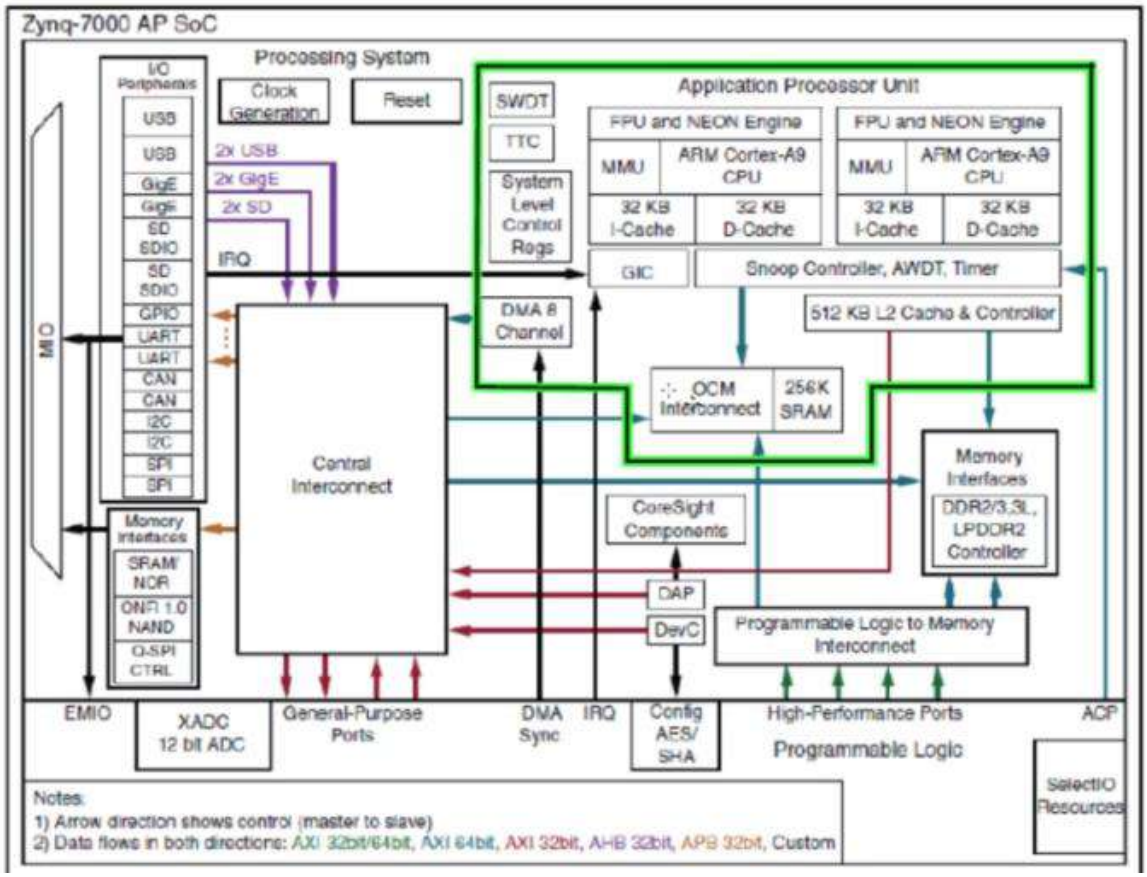
(i) coprocessors — elements that supplement the primary processor, usually optimized for a certain task;

(ii) cores for interacting with external interfaces, e.g. connecting to LEDs and switches, codecs, etc.; and

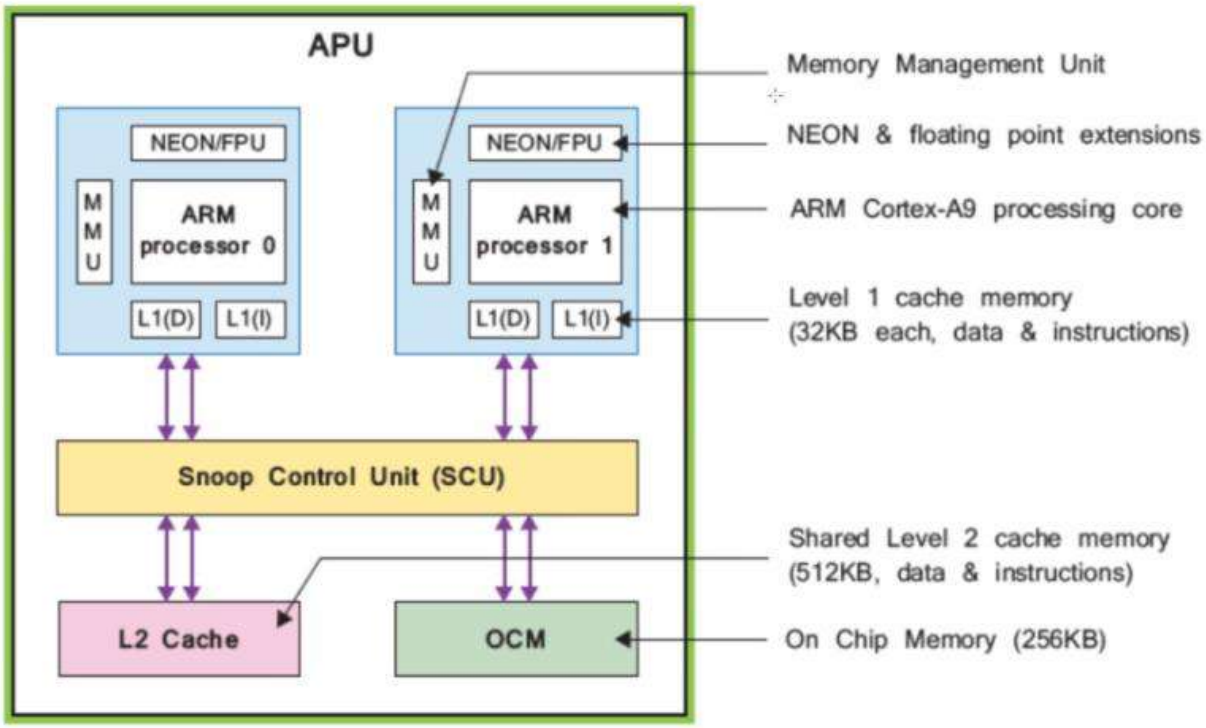
(iii) additional memory elements.



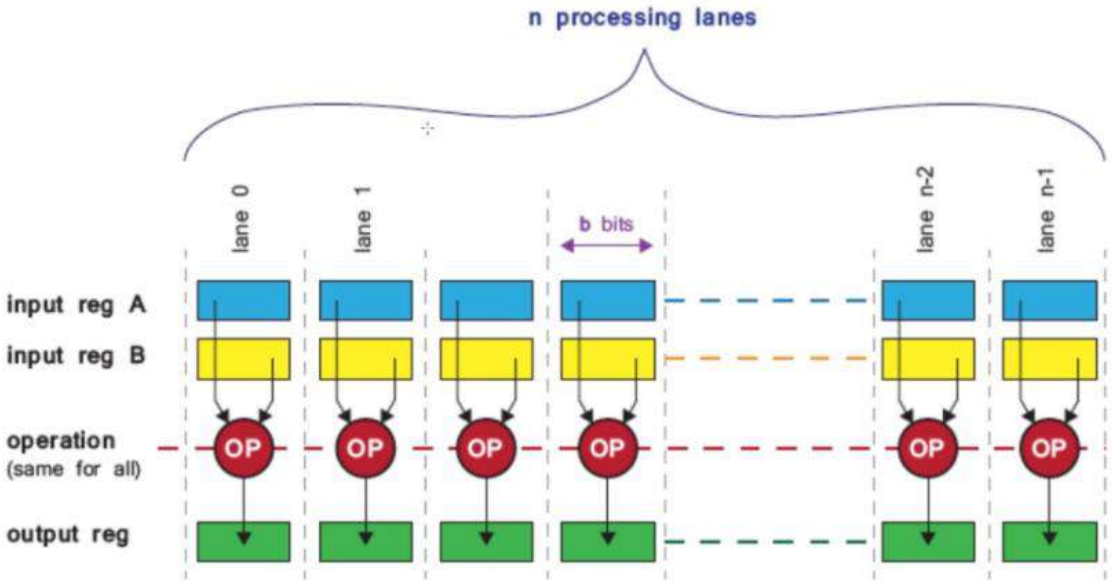
The Zynq Processing System

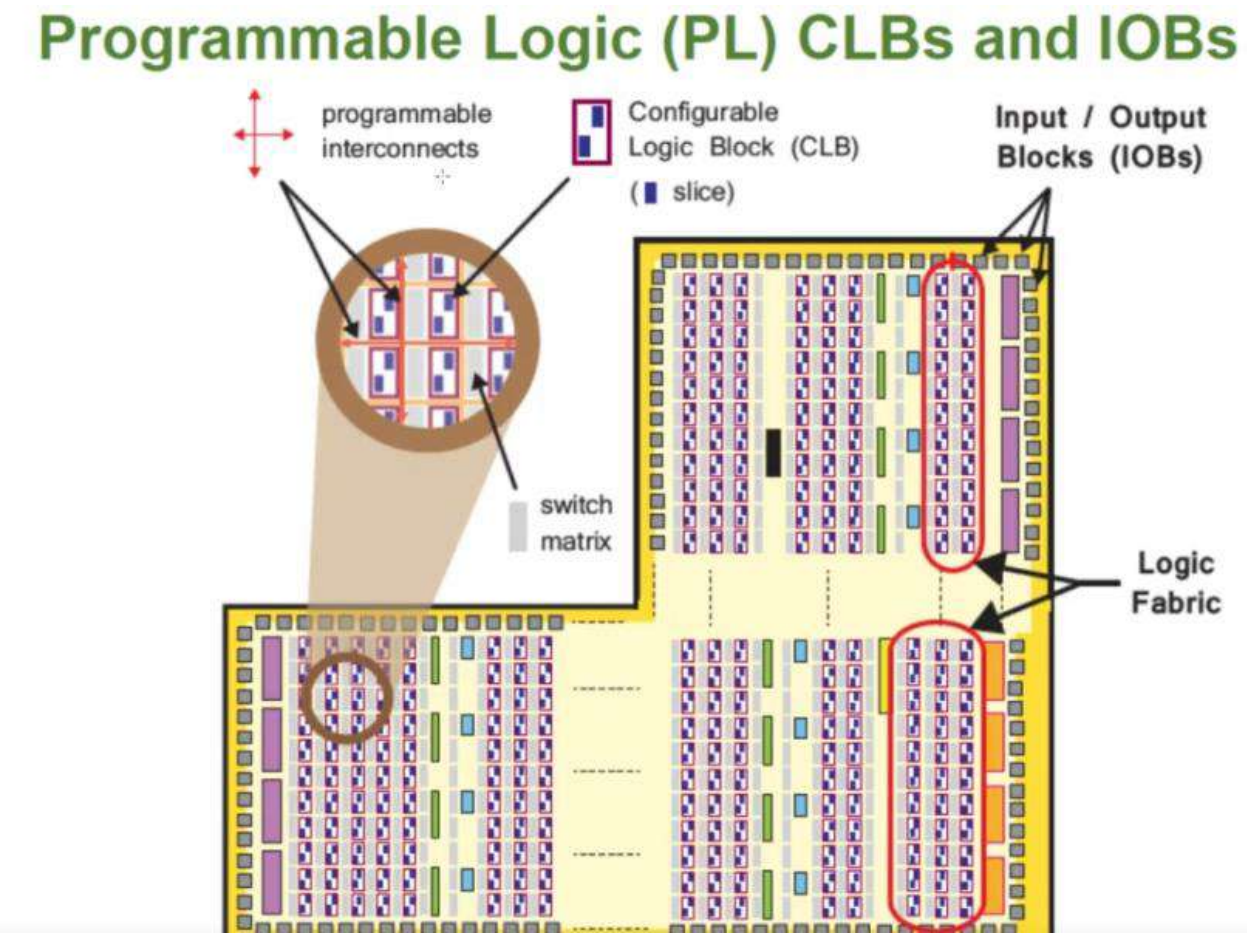


Simplified Block Diagram of the Application Processing Unit (APU)



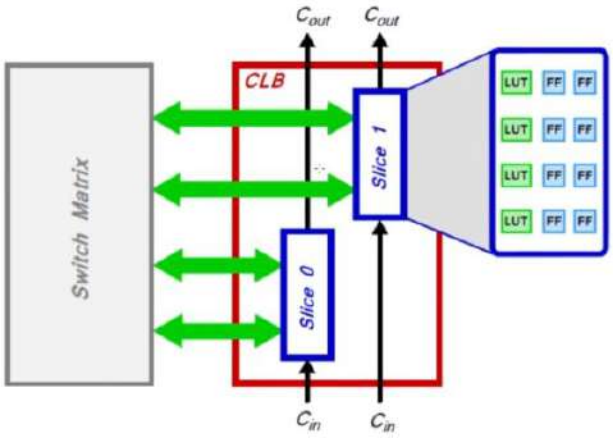
SIMD (Single Instruction Multiple Data) Processing in the NEON Media Processing Engine (MPE)





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

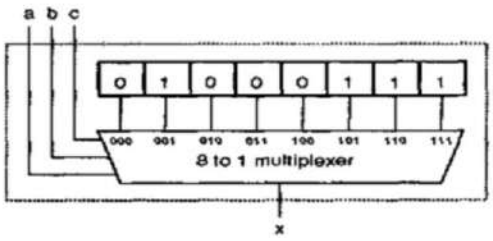


→look Up Table(LUT):

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.

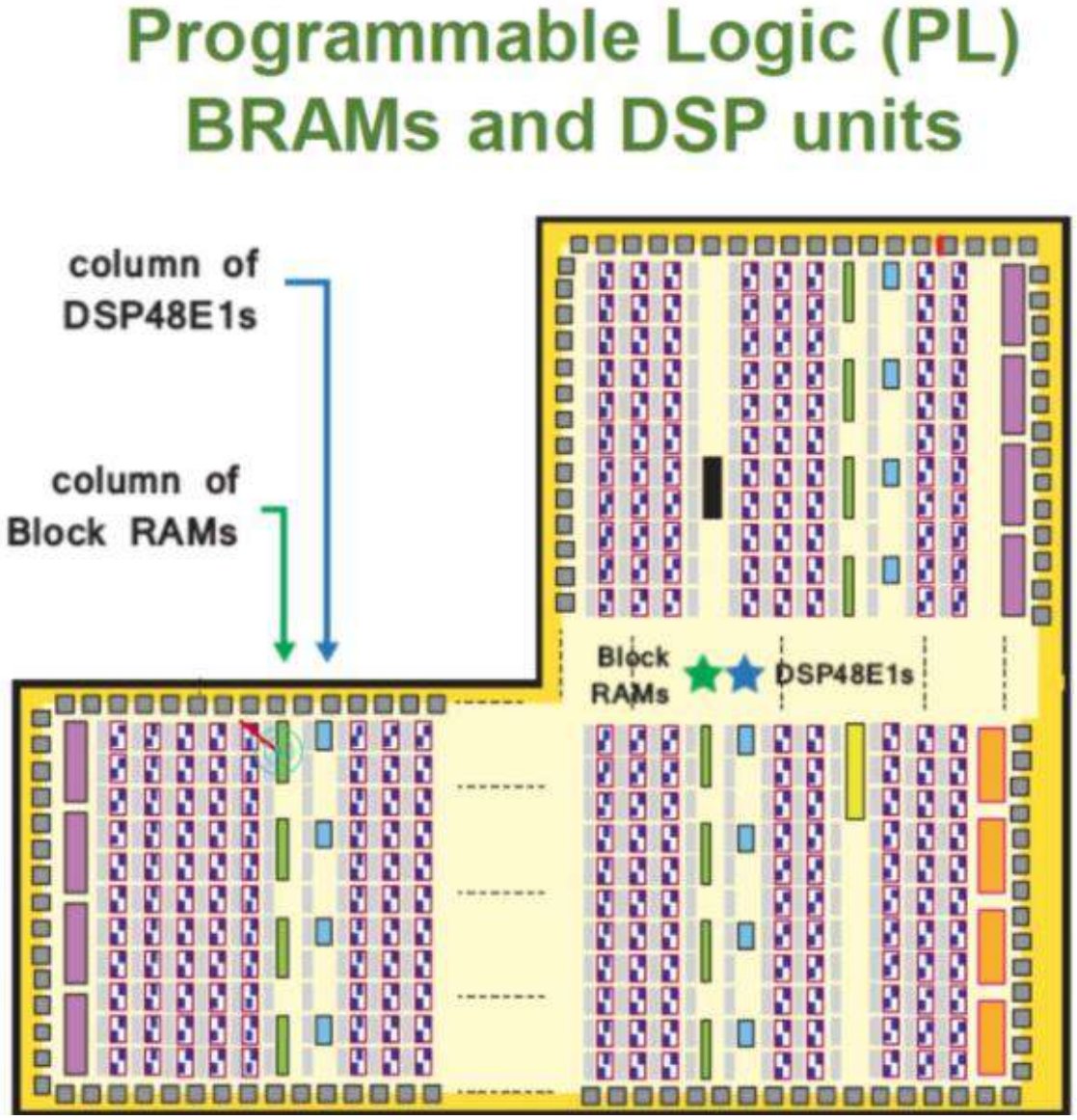
a	b	c	$x = ab + \bar{b}c$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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



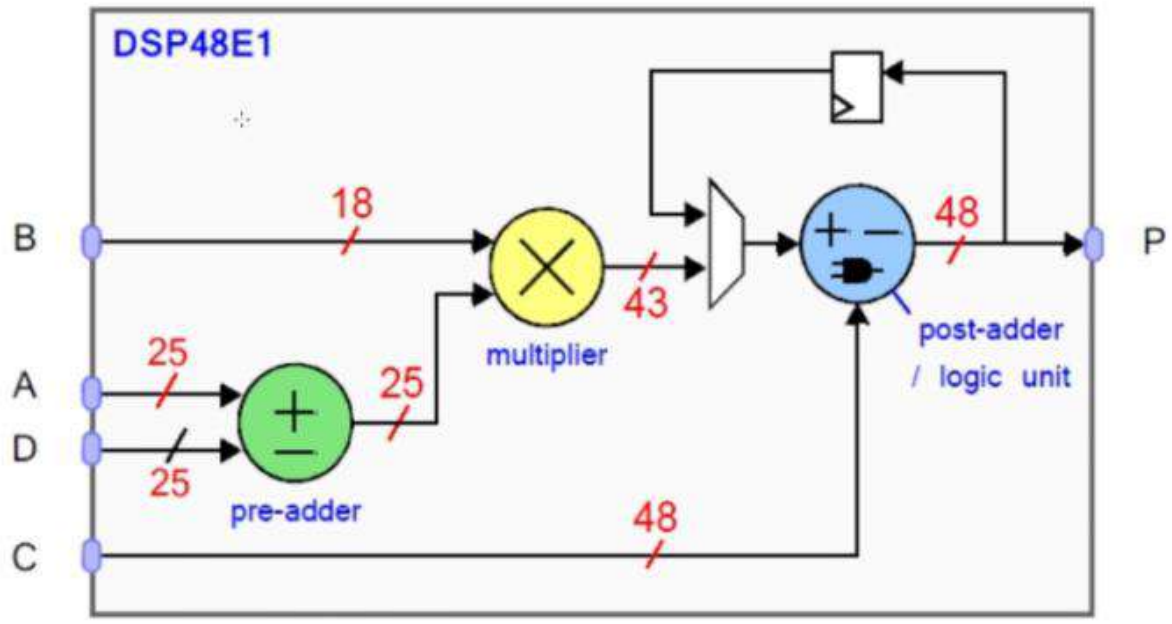


Figure 2-12: Arithmetic capabilities of the DSP48E1 slice

Communications Interfaces

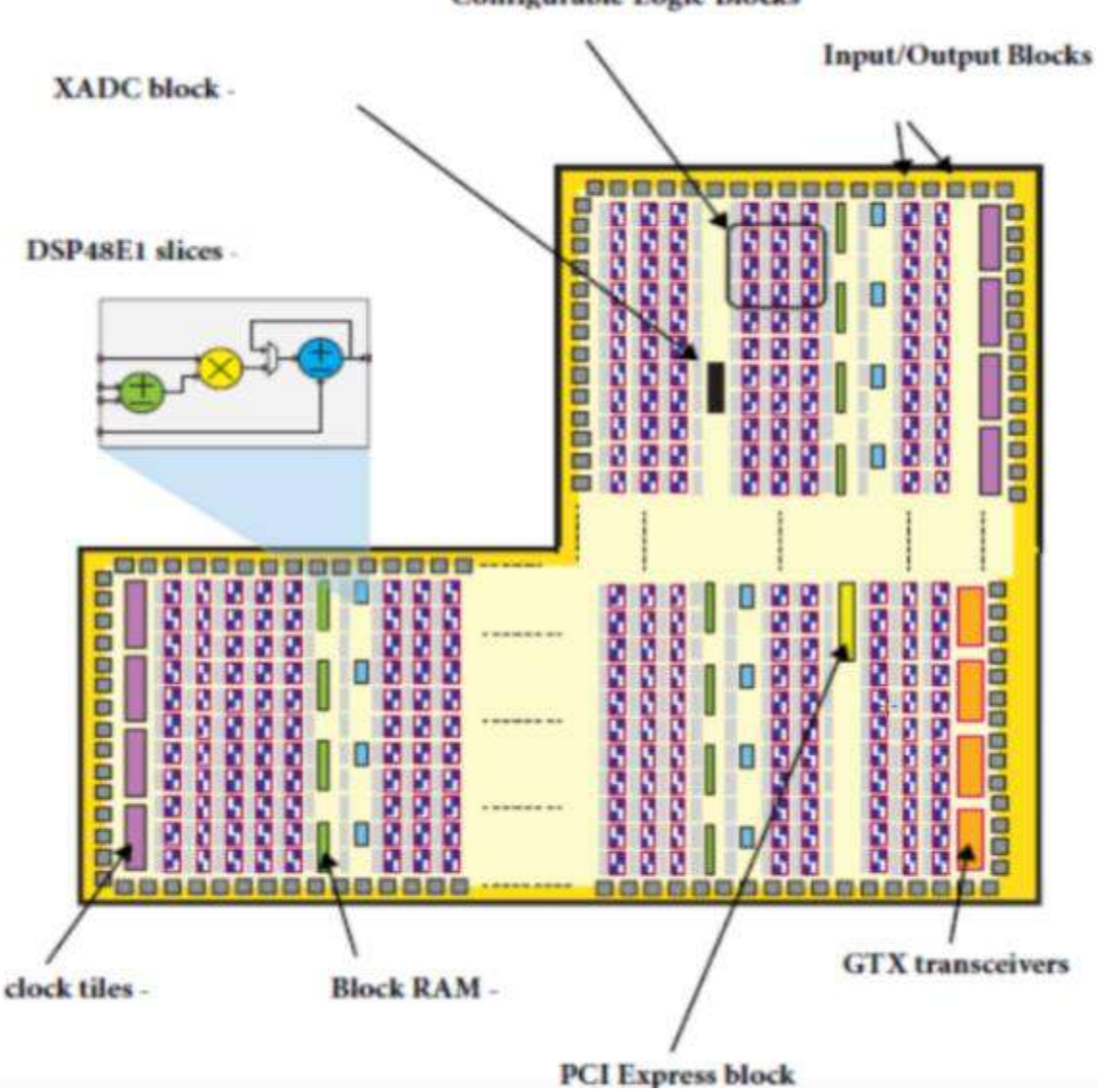
The more highly specified Zynq devices include GTX Transceivers, high-speed communications interface blocks which are embedded into the logic fabric.

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.

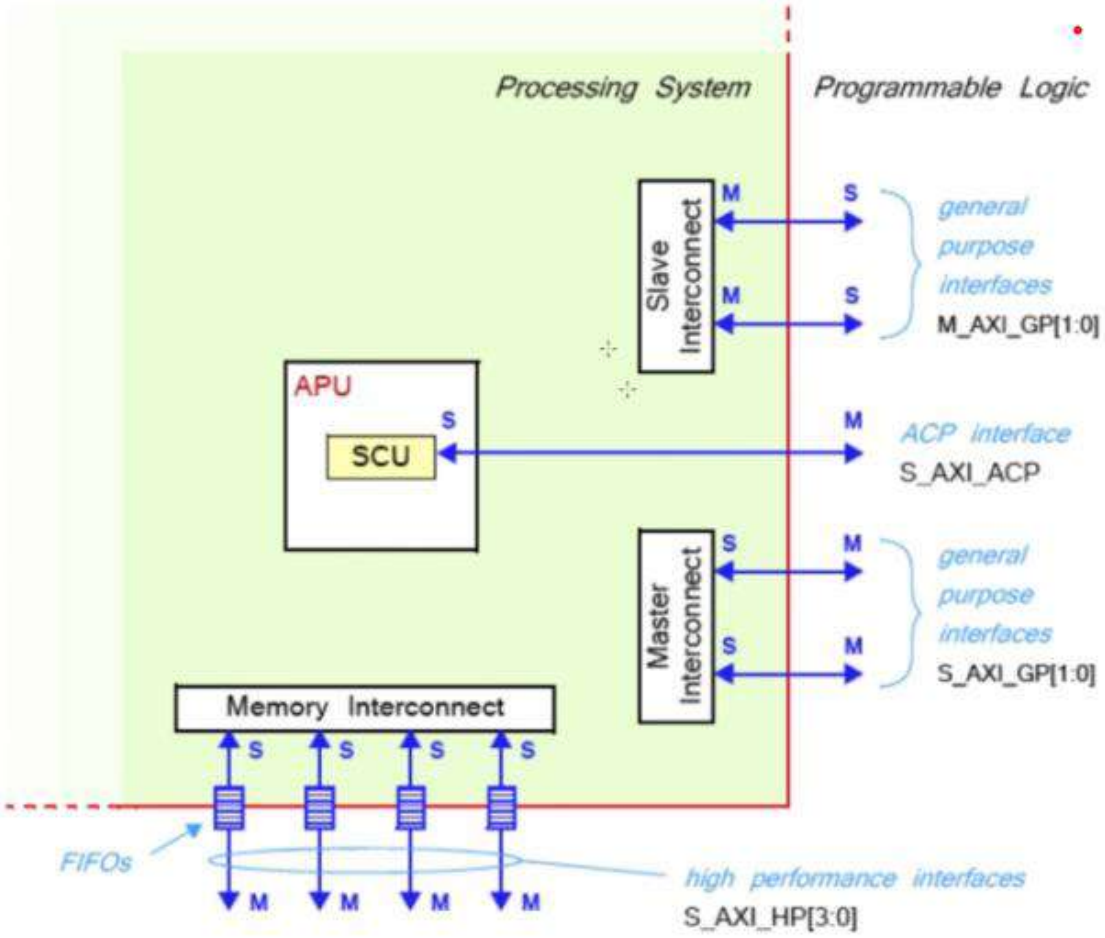
GTX Transceivers are implemented as 'quads', i.e. groups of 4 individual channels, each of which comprises a dedicated Phase Locked Loop (PLL) for that channel, a transmitter, and a receiver.

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.


Configurable Logic Blocks
Input/Output Blocks
XADC block -
DSP48E1 slices -
clock tiles -
Block RAM -
PCI Express block
GTX transceivers

AXI Interconnects and Interfaces



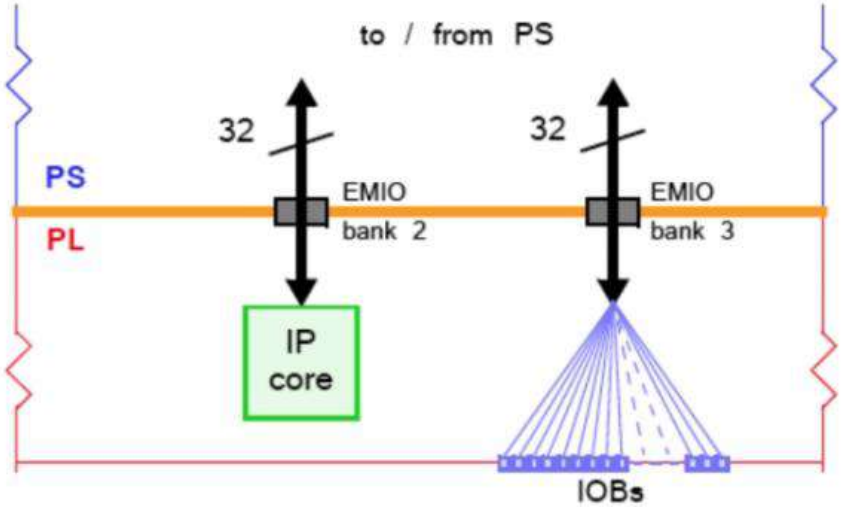
ACP-Accelerator Coherency Port

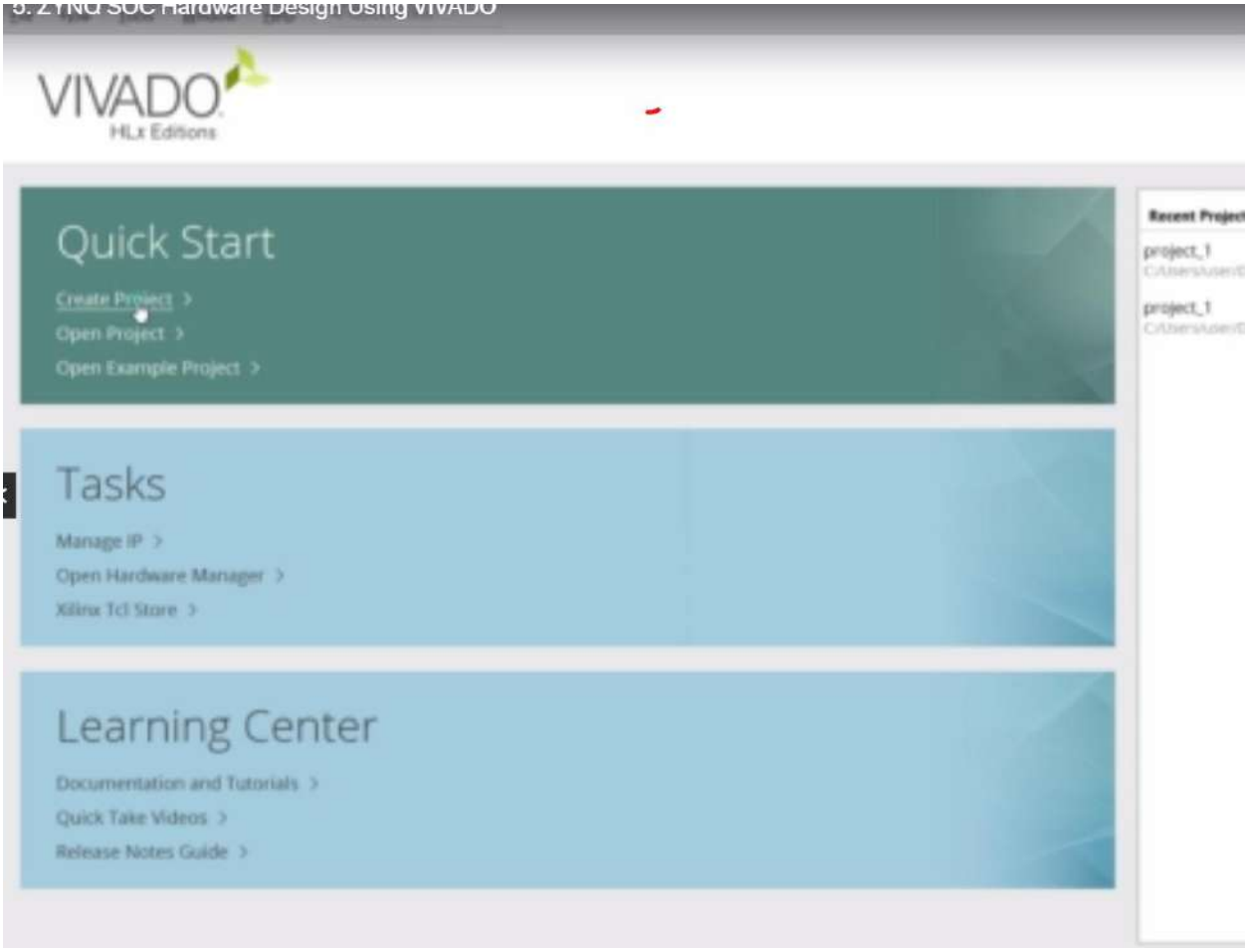
SCU-Snoop Control Unit

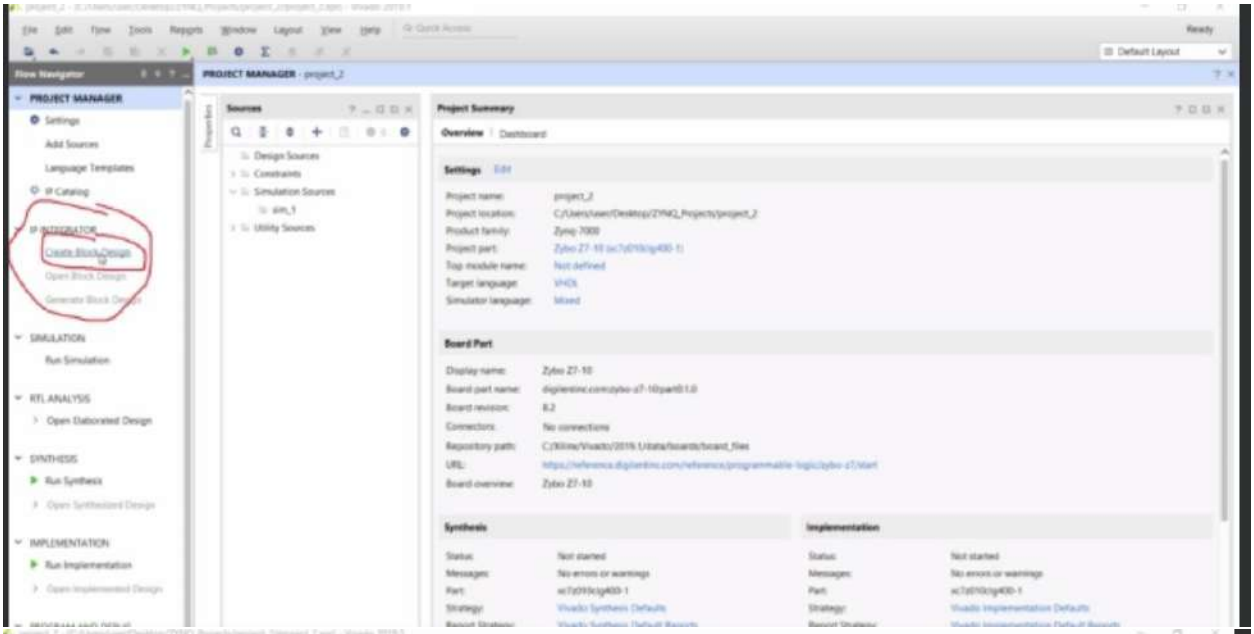
S→Slave

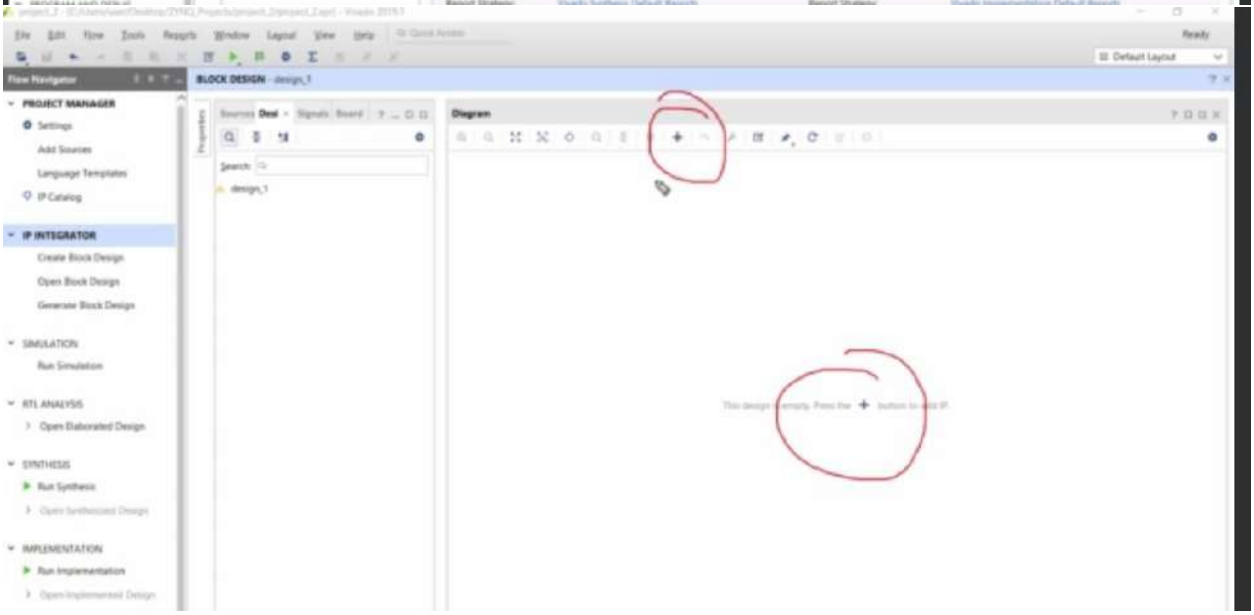
M→Master

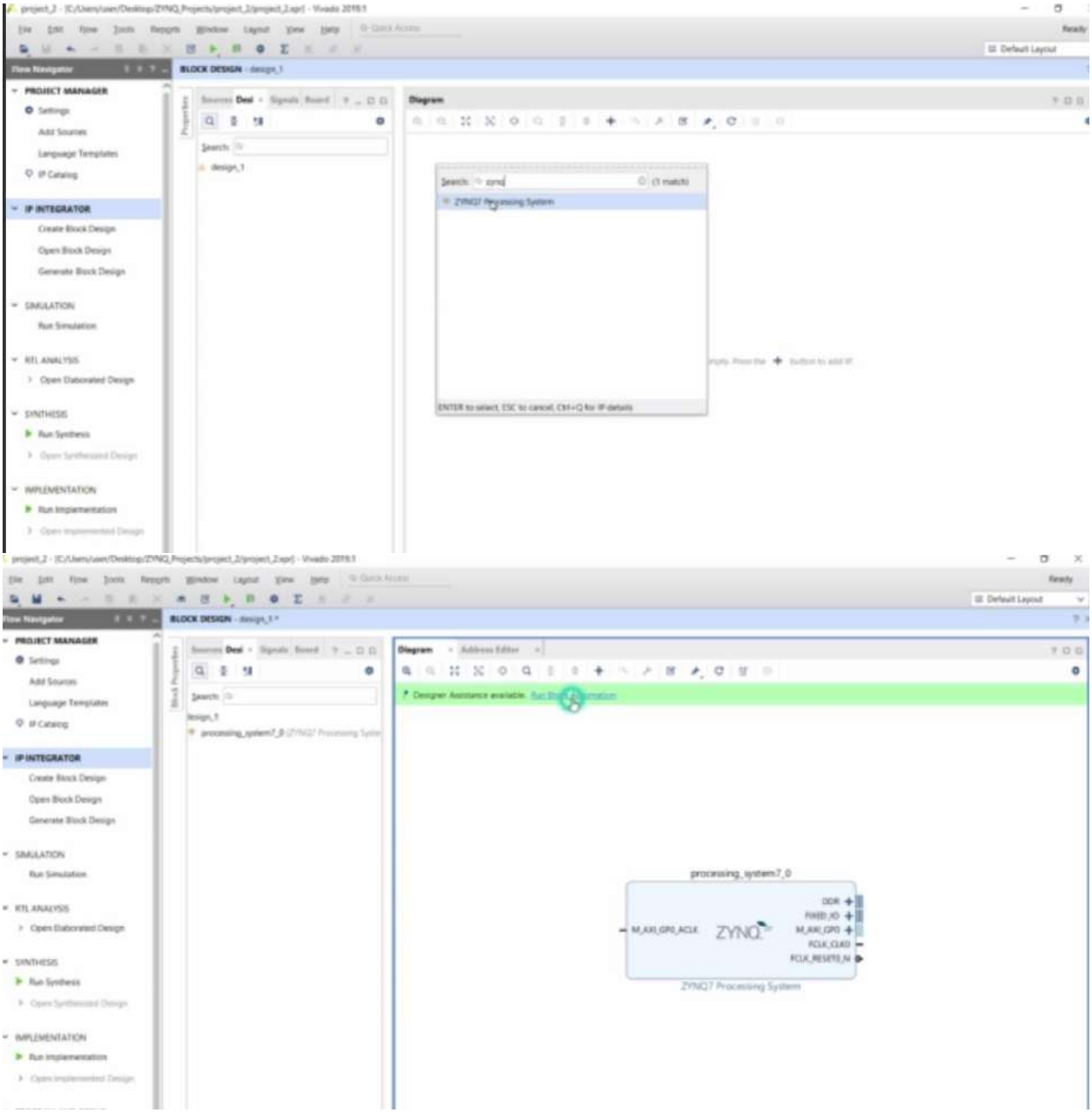
Using Extended Multiplexed Input/Output (EMIO) to Interface Between PS and PL

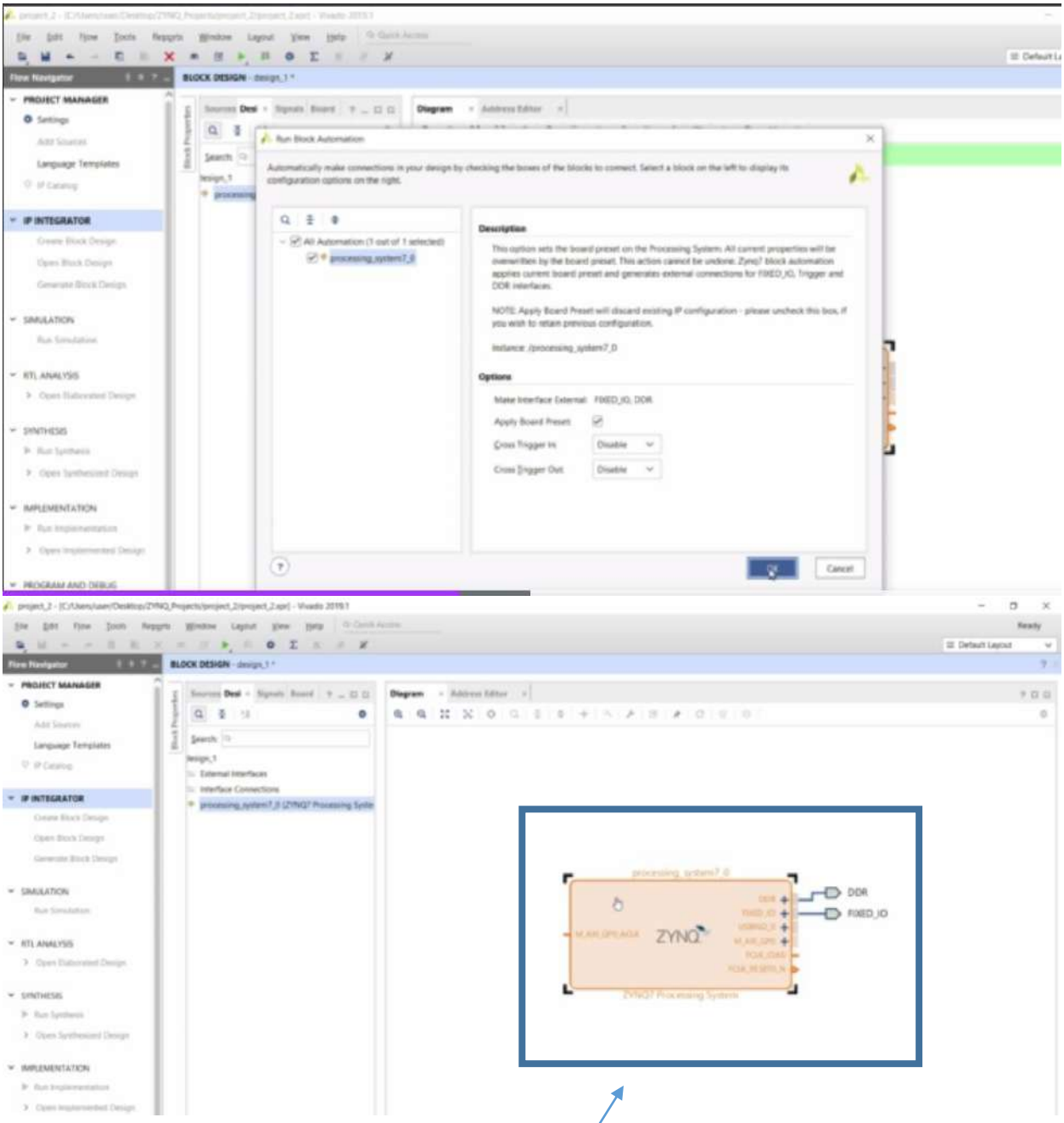




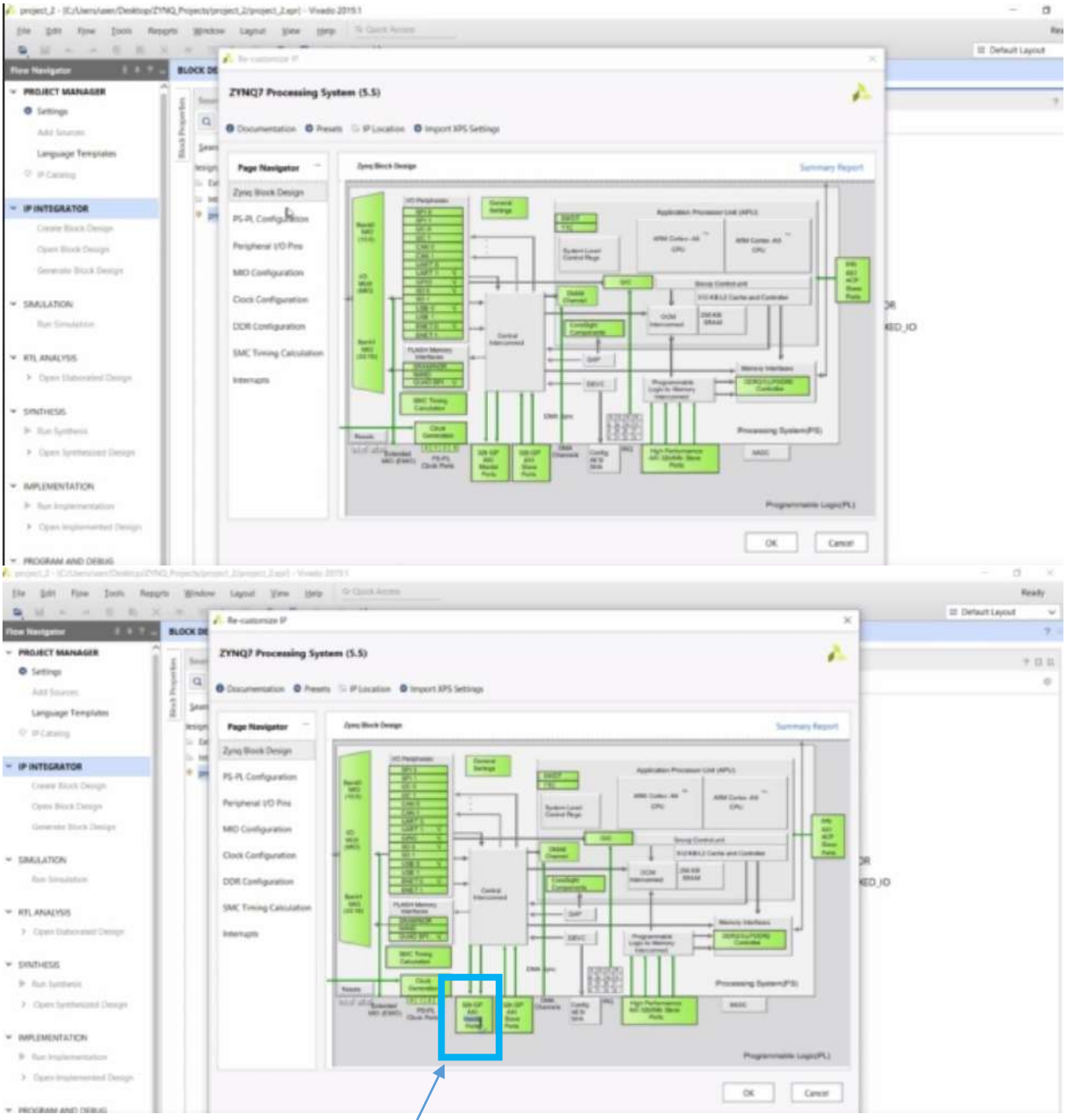

PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
IMPLEMENTATION
Run Implementation


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
IMPLEMENTATION
Run Implementation

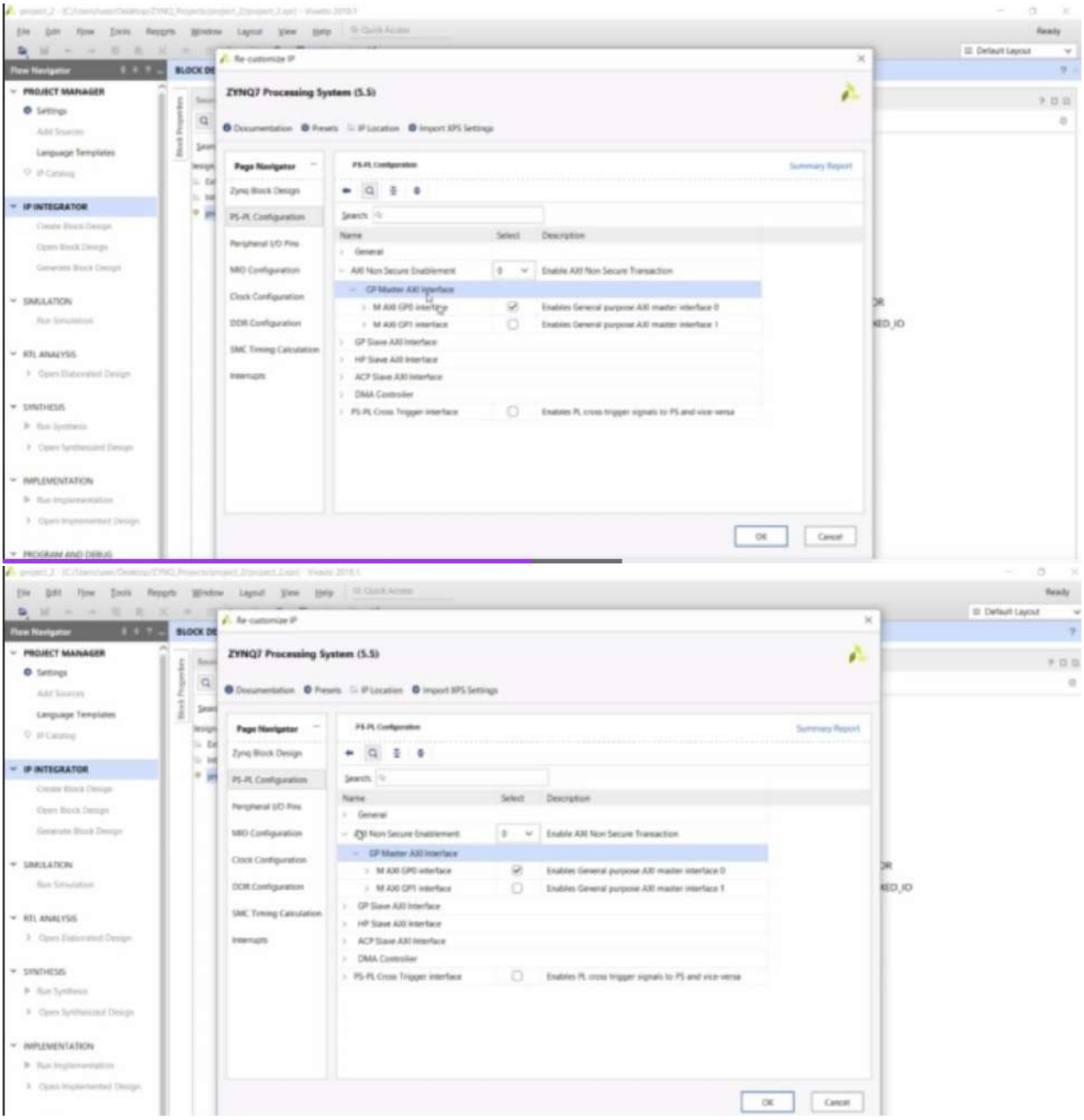

PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
IMPLEMENTATION
Run Implementation
BLOCK DESIGN - design_1
Diagram
Search: zynq
(1 match)
ZYNQ7 Processing System
ENTER to select, ESC to cancel, Ctrl+Q for IP details
Designer Assistance available.
processing_system7_0
M_AXI_GP0_ACLK
ZYNQ7
DDR
FIXED_IO
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System

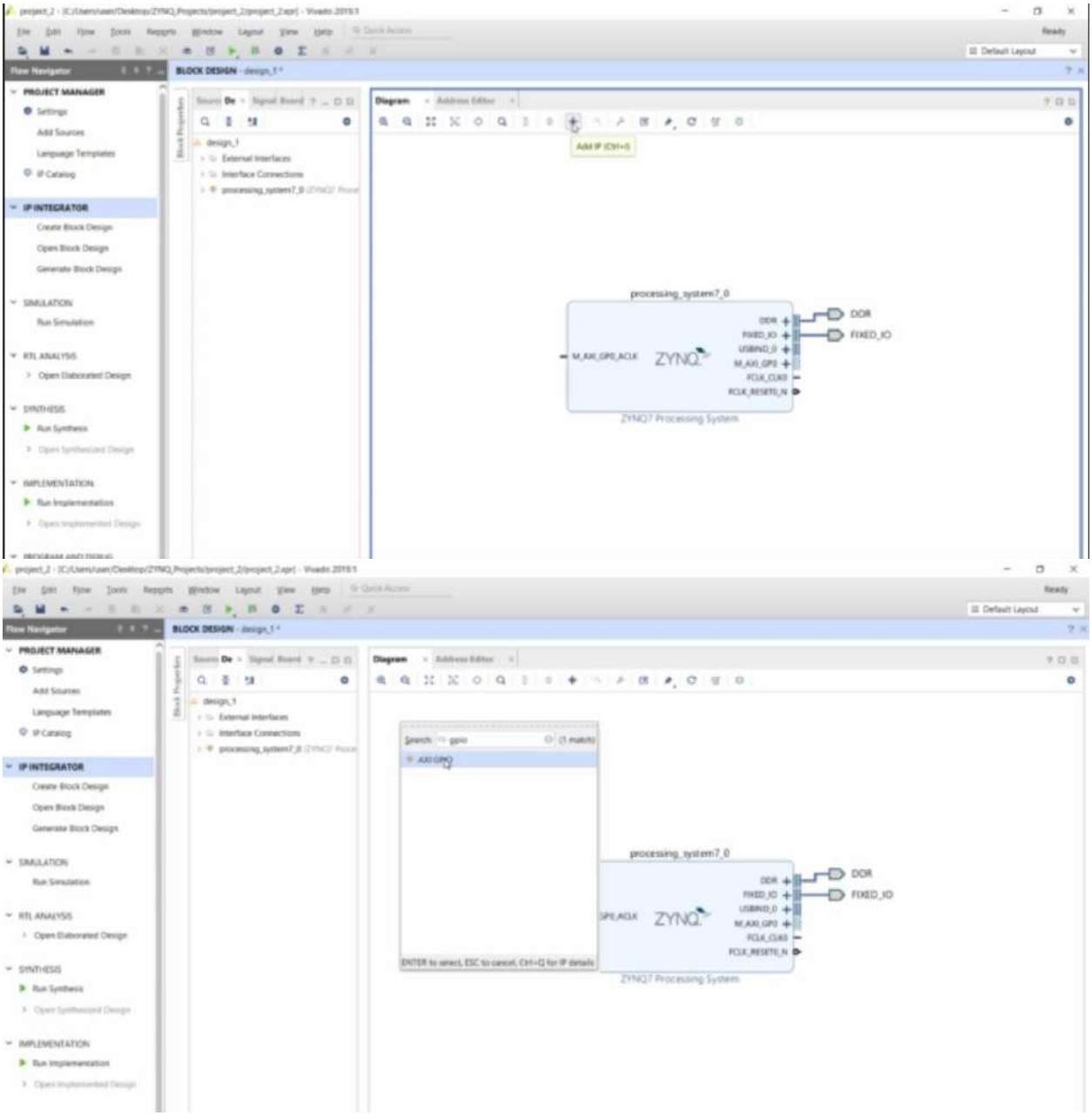


Double click

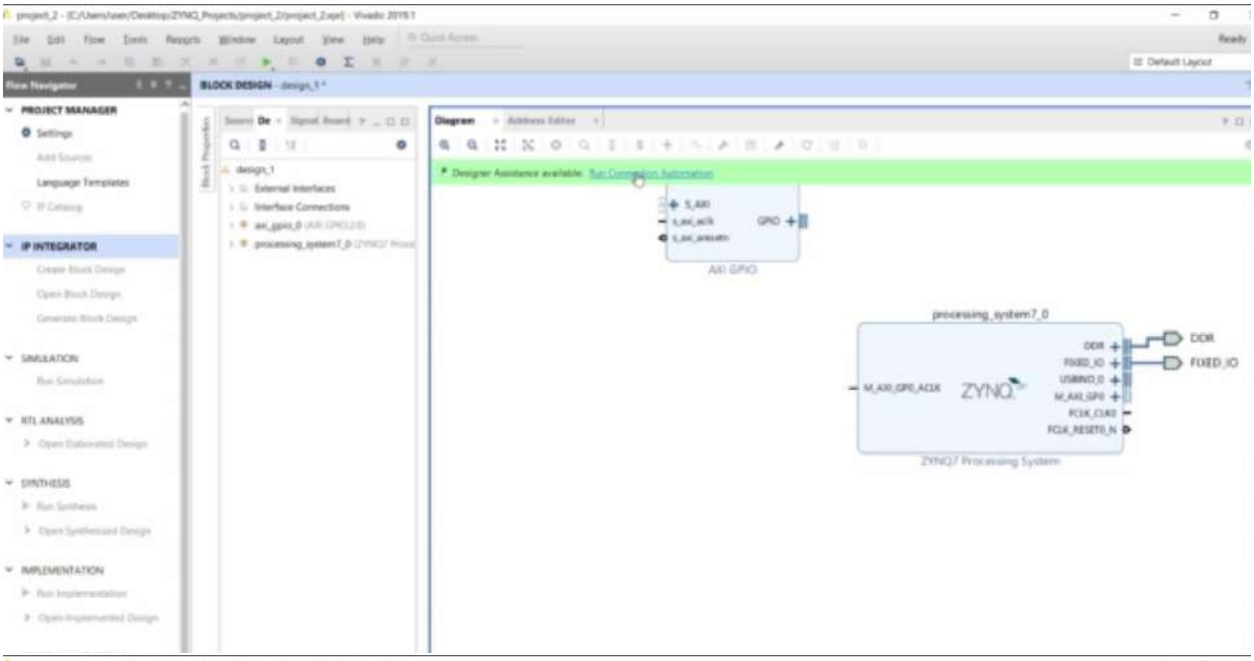


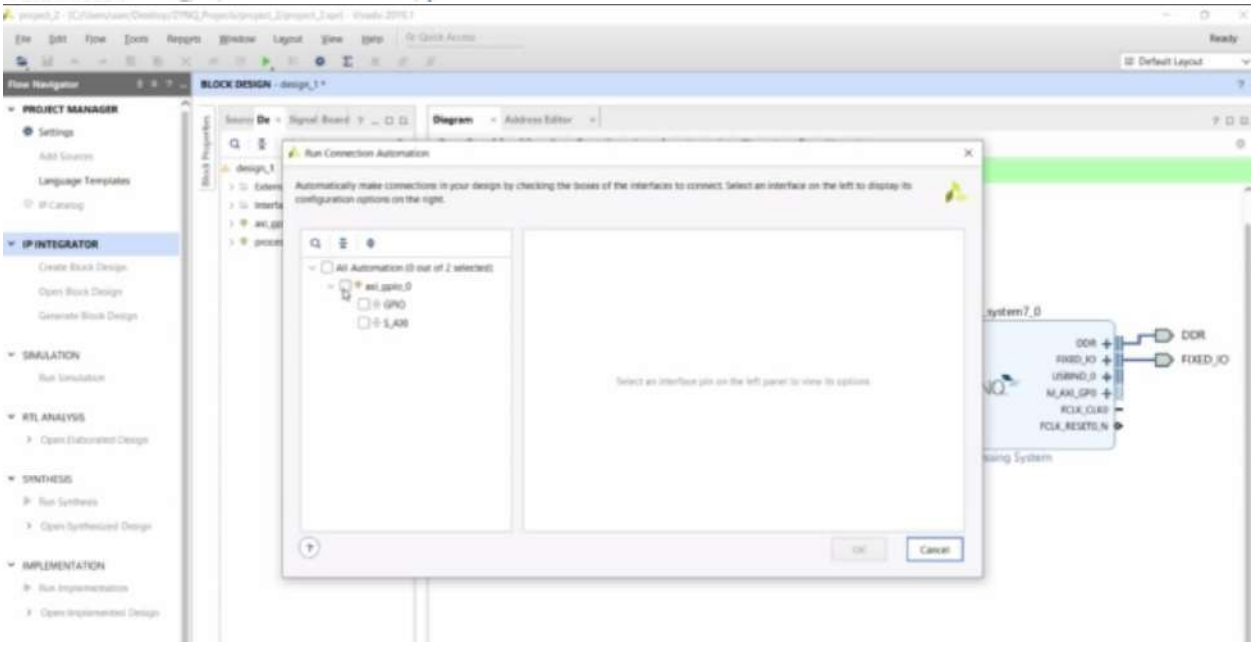
Double Click on green areas

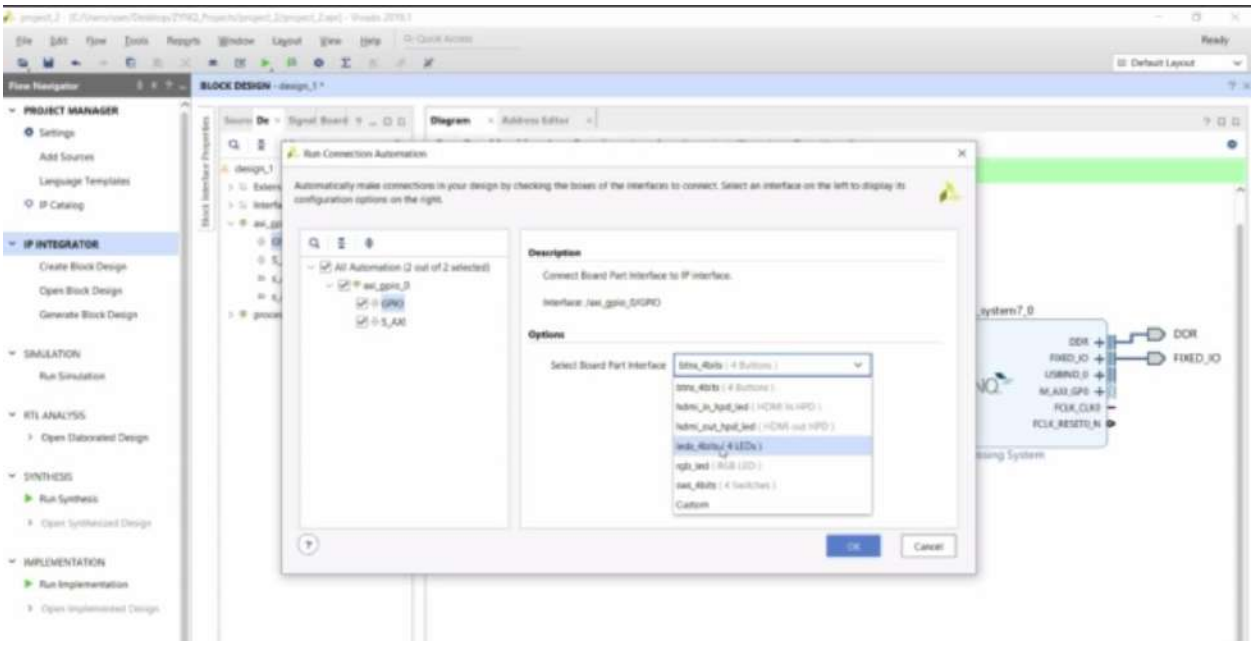

Re-customize IP
ZYNQ7 Processing System (5.5)
Documentation
Presets
IP Location
Import XPS Settings
Page Navigator
Zynq Block Design
PS-PL Configuration
Peripheral I/O Pins
MIO Configuration
Clock Configuration
DDR Configuration
SMC Timing Calculation
Interrupts
PS-PL Configuration
Summary Report
Search:
Name
Select
Description
General
AXI Non Secure Enablement
Enable AXI Non Secure Transaction
GP Master AXI Interface
M AXI GP0 interface
Enables General purpose AXI master interface 0
M AXI GP1 interface
Enables General purpose AXI master interface 1
GP Slave AXI Interface
HP Slave AXI Interface
ACP Slave AXI Interface
DMA Controller
PS-PL Cross Trigger interface
Enables PL cross trigger signals to PS and vice-versa
OK
Cancel

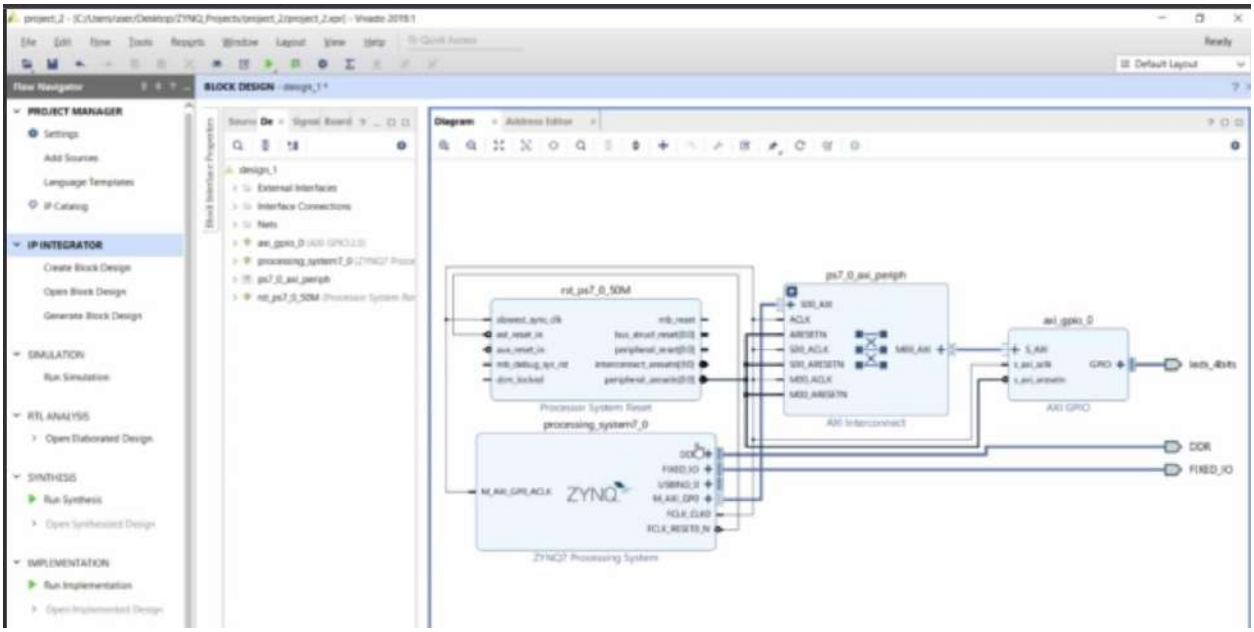

BLOCK DESIGN - design_1 *
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
IMPLEMENTATION
Run Implementation
Add IP (Ctrl+I)
processing_system7_0
DDR
FIXED_IO
M_AXI_GP0_ACLK
ZYNQ7 Processing System

BLOCK DESIGN - design_1 *
PROJECT MANAGER
IP INTEGRATOR
SIMULATION
RTL ANALYSIS
SYNTHESIS
IMPLEMENTATION
Search: gpio
AXI GPIO
processing_system7_0
DDR
FIXED_IO
ZYNQ7 Processing System

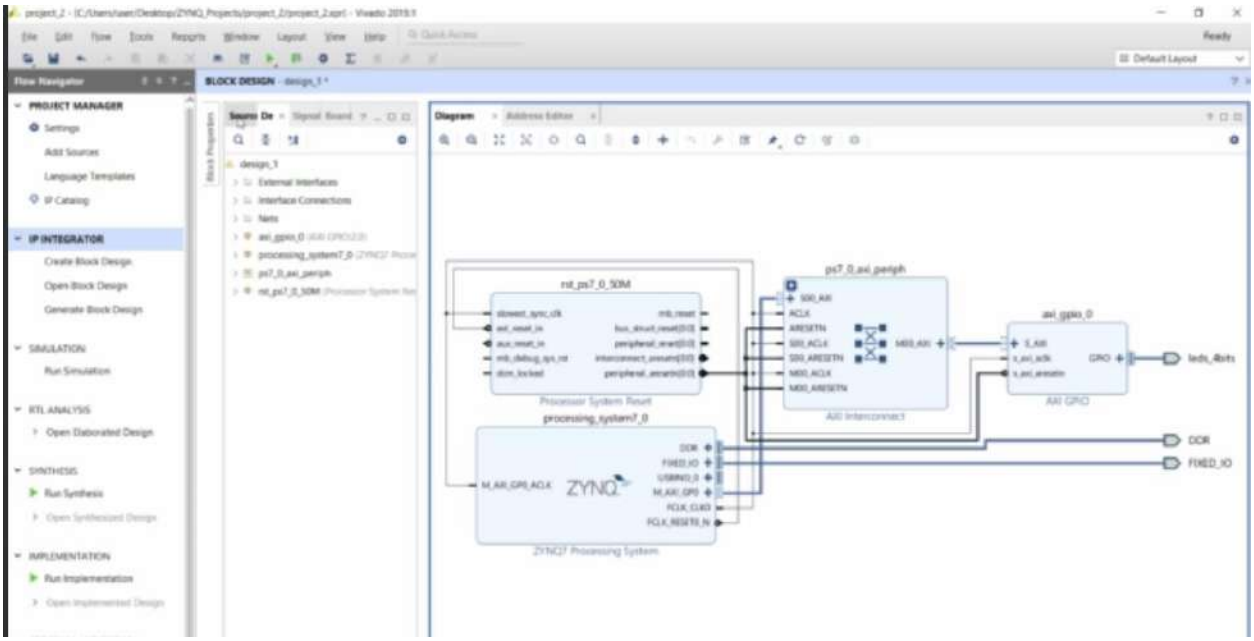

BLOCK DESIGN - design_1 *
PROJECT MANAGER
IP INTEGRATOR
SIMULATION
RTL ANALYSIS
SYNTHESIS
IMPLEMENTATION
Designer Assistance available. Run Connection Automation
S_AXI
GPIO
AXI GPIO
processing_system7_0
DDR
FIXED_IO
M_AXI_GP0_ACLK
ZYNQ7 Processing System

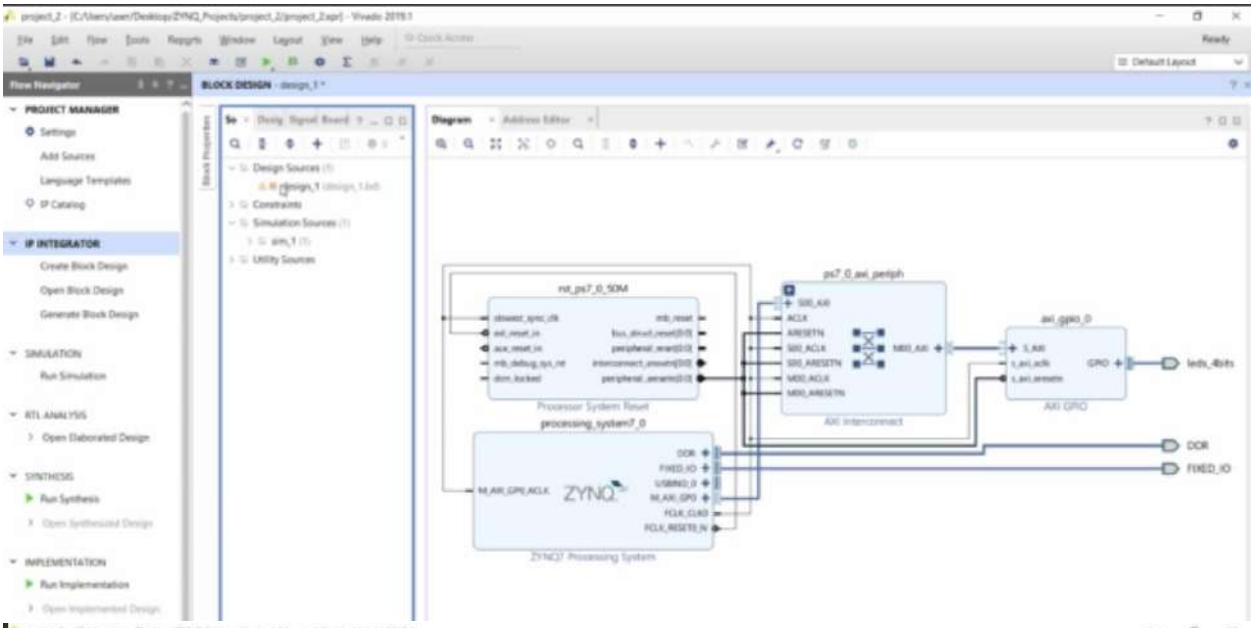

BLOCK DESIGN - design_1 *
Run Connection Automation
Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.
All Automation (0 out of 2 selected)
axi_gpio_0
GPIO
S_AXI
Cancel
DDR
FIXED_IO

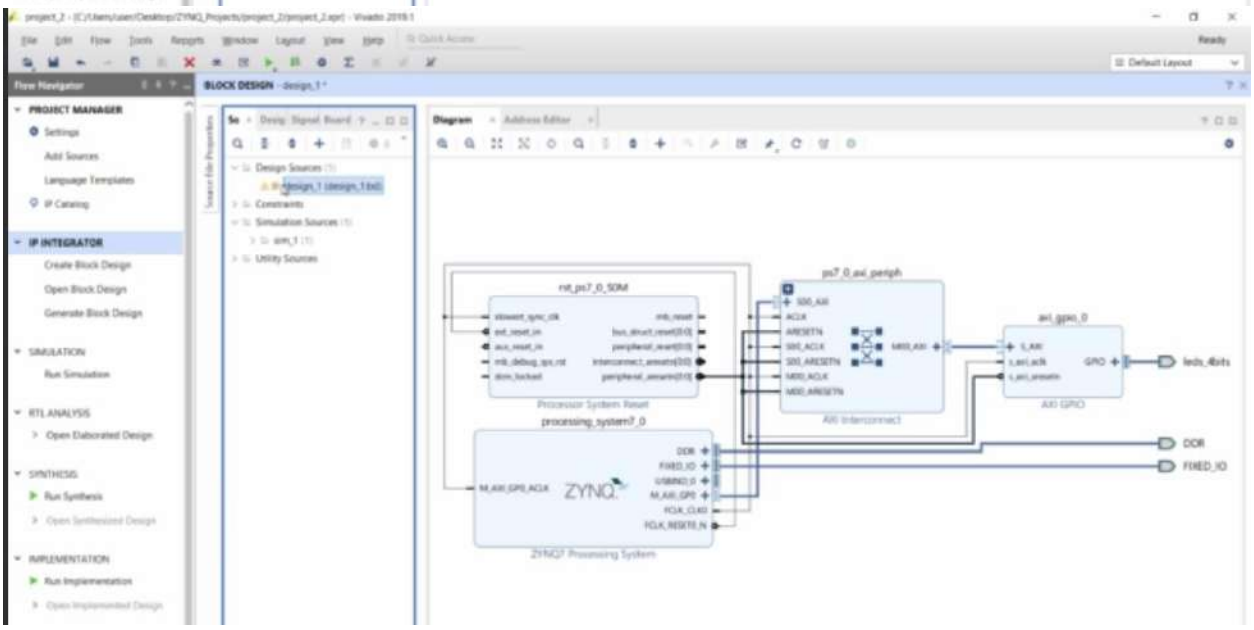

Run Connection Automation
Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.
Description
Connect Board Part Interface to IP interface.
Options
Select Board Part Interface
OK
Cancel
DDR
FIXED_IO

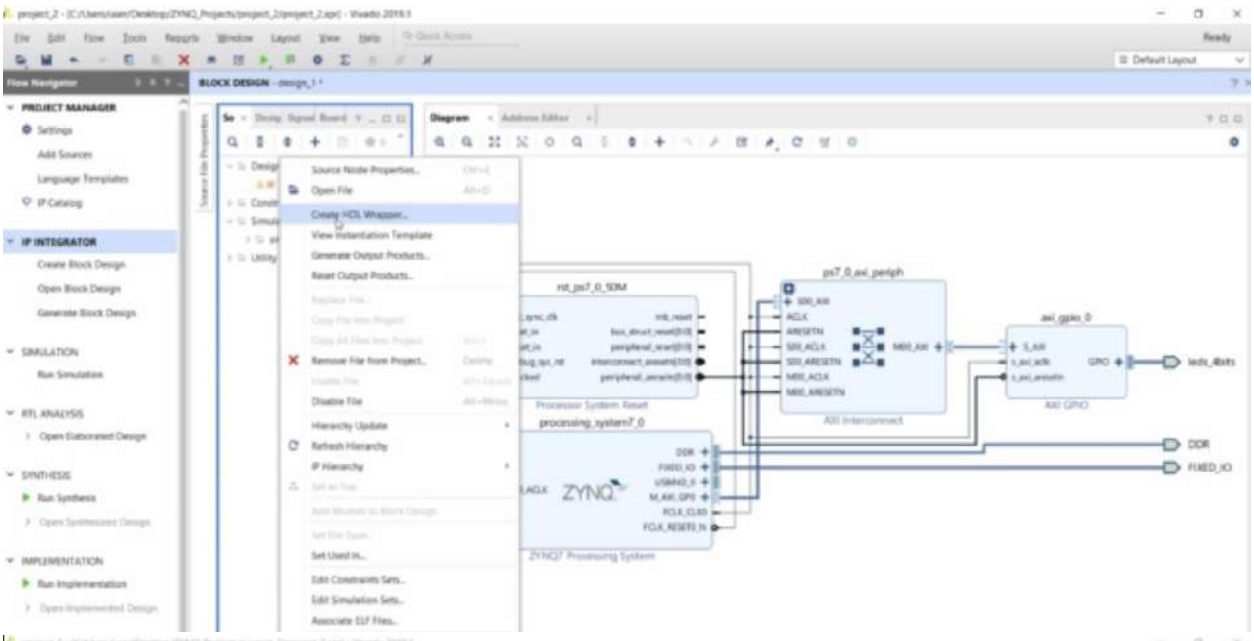

BLOCK DESIGN
Diagram
rst_ps7_0_50M
ps7_0_axi_periph
axi_gpio_0
processing_system7_0
ZYNQ
DDR
FIXED_IO

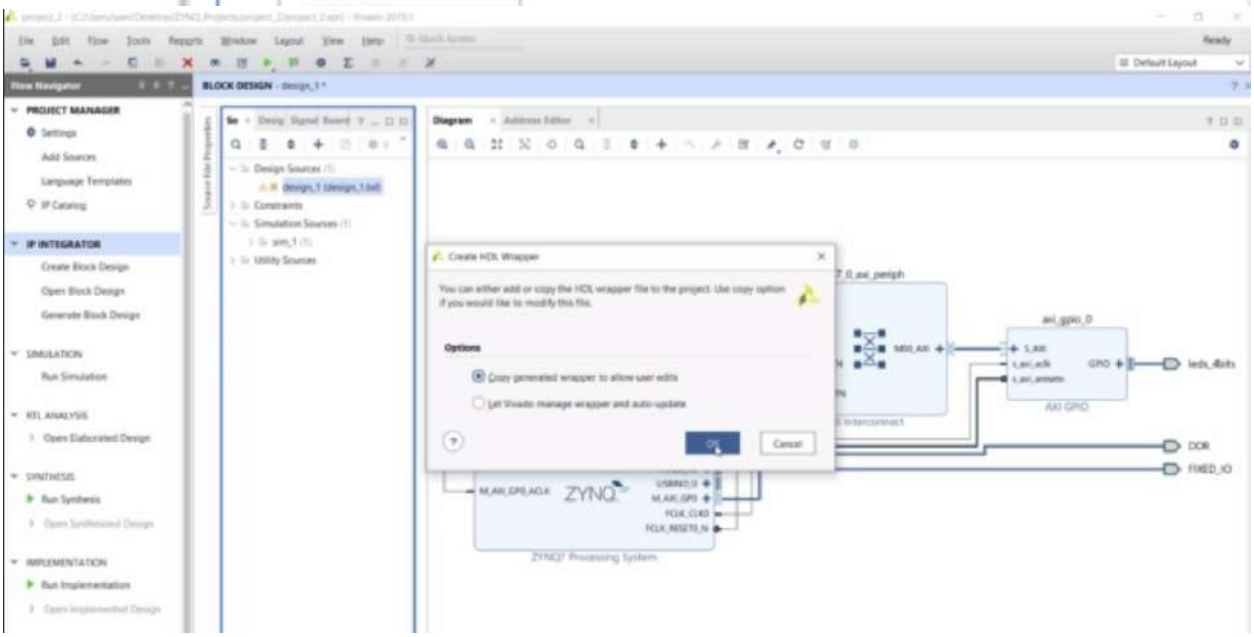
➔ Sources:

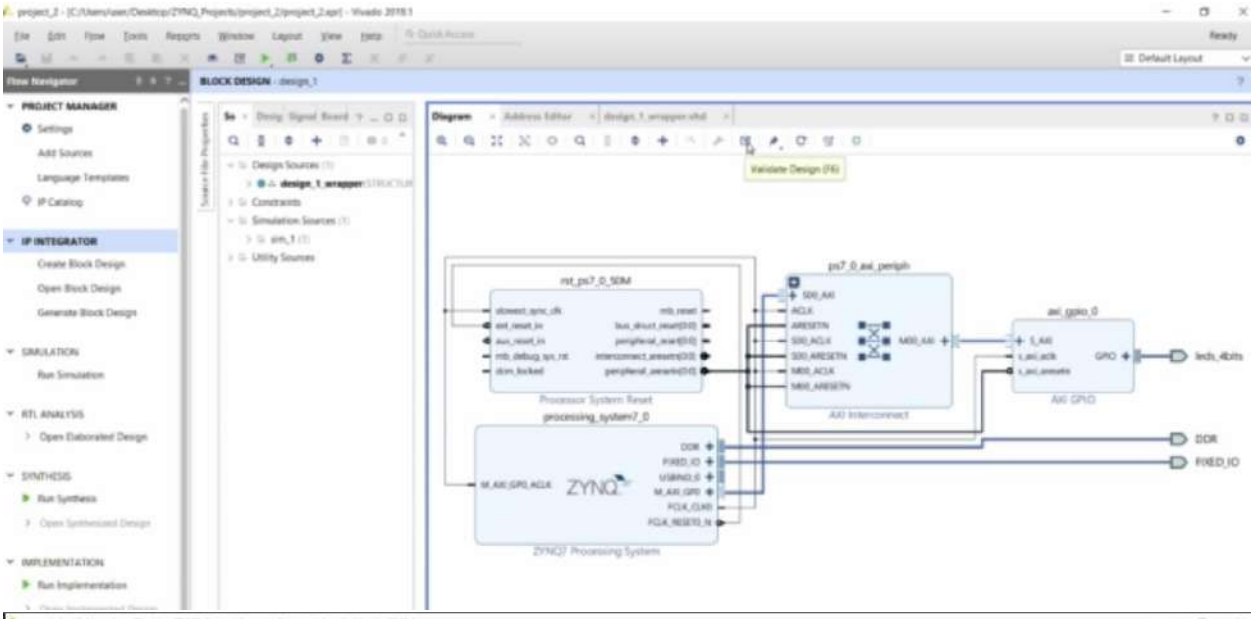

PROJECT MANAGER
IP INTEGRATOR
SIMULATION
RTL ANALYSIS
SYNTHESIS
IMPLEMENTATION
BLOCK DESIGN
rst_ps7_0_50M
ps7_0_axi_periph
axi_gpio_0
processing_system7_0
ZYNQ
leds_4bits
DDR
FIXED_IO

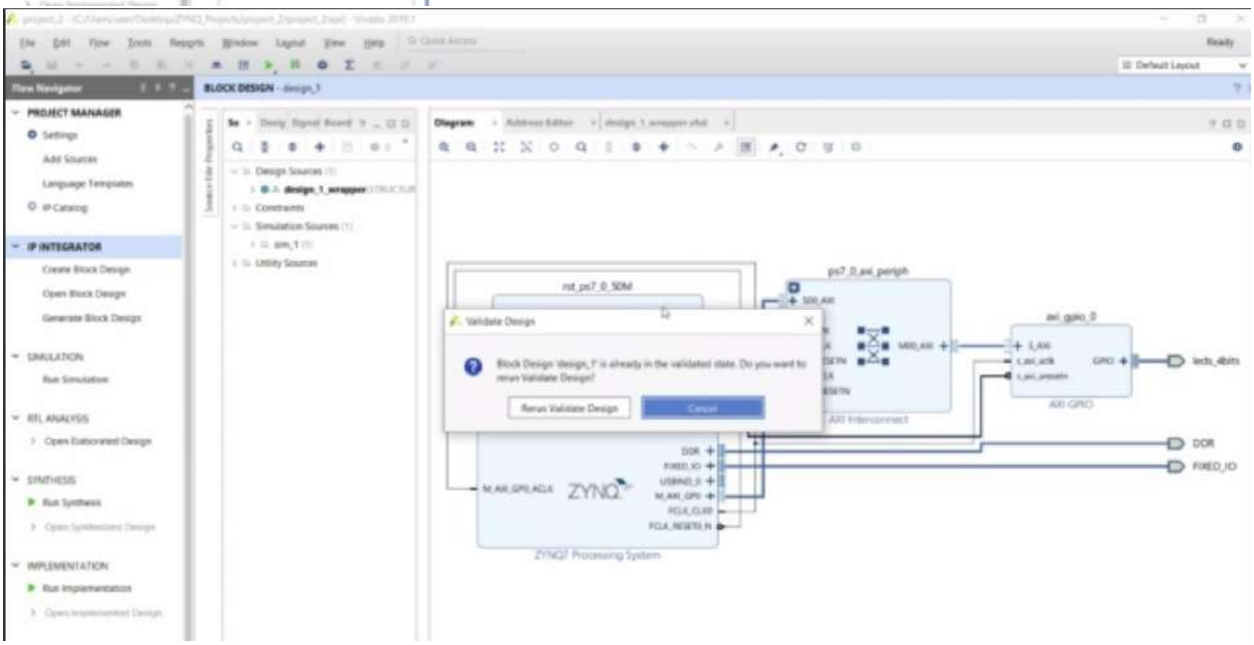

PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
IMPLEMENTATION
Run Implementation
BLOCK DESIGN - design_1 *
Design Sources
Constraints
Simulation Sources
Utility Sources
Diagram
Address Editor
rst_ps7_0_50M
Processor System Reset
ps7_0_axi_periph
S00_AXI
M00_AXI
Axi Interconnect
axi_gpio_0
GPIO
AXI GPIO
leds_4bits
processing_system7_0
ZYNQ
ZYNQ7 Processing System
DDR
FIXED_IO


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
IMPLEMENTATION
Run Implementation
BLOCK DESIGN - design_1 *
Design Sources
design_1 (design_1.bd)
Constraints
Simulation Sources
Utility Sources
Diagram
Address Editor
rst_ps7_0_50M
Processor System Reset
ps7_0_axi_periph
S00_AXI
M00_AXI
Axi Interconnect
axi_gpio_0
GPIO
AXI GPIO
leds_4bits
processing_system7_0
ZYNQ
ZYNQ7 Processing System
DDR
FIXED_IO

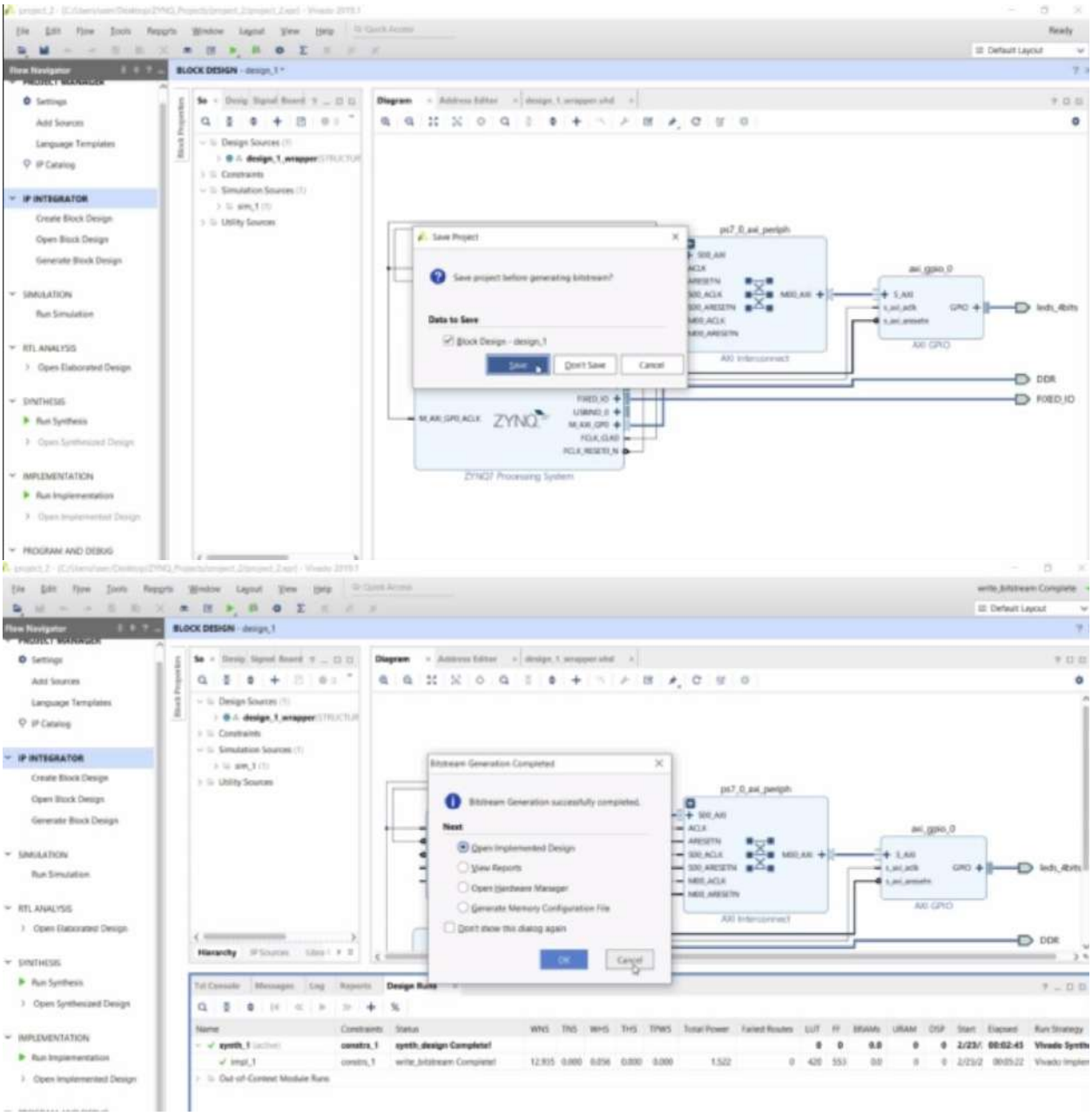

Create HDL Wrapper...
ps7_0_axi_periph
axi_gpio_0
processing_system7_0
ZYNQ7 Processing System
DDR
FIXED_IO


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
ZYNQ7 Processing System
DDR
FIXED_IO

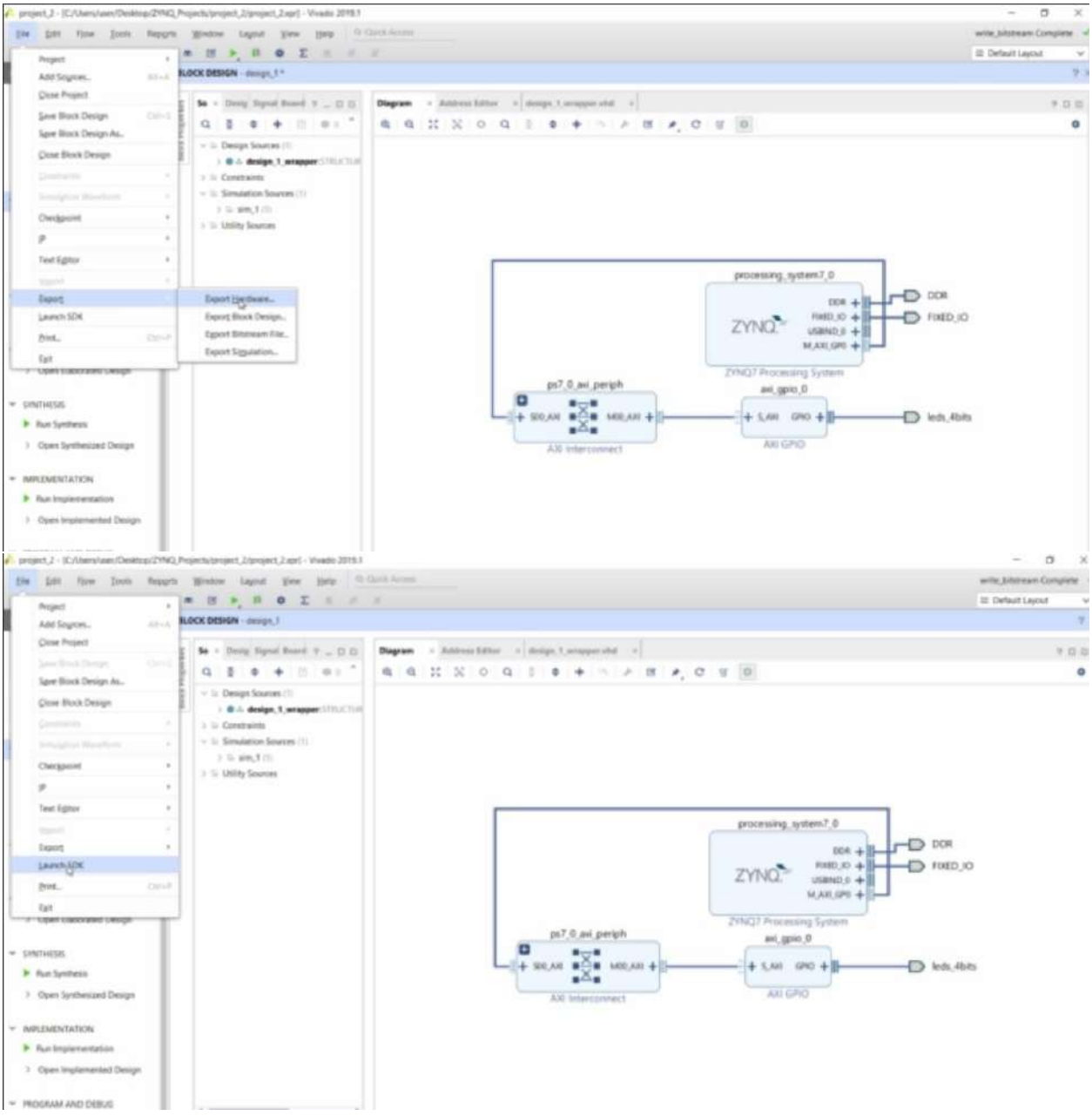


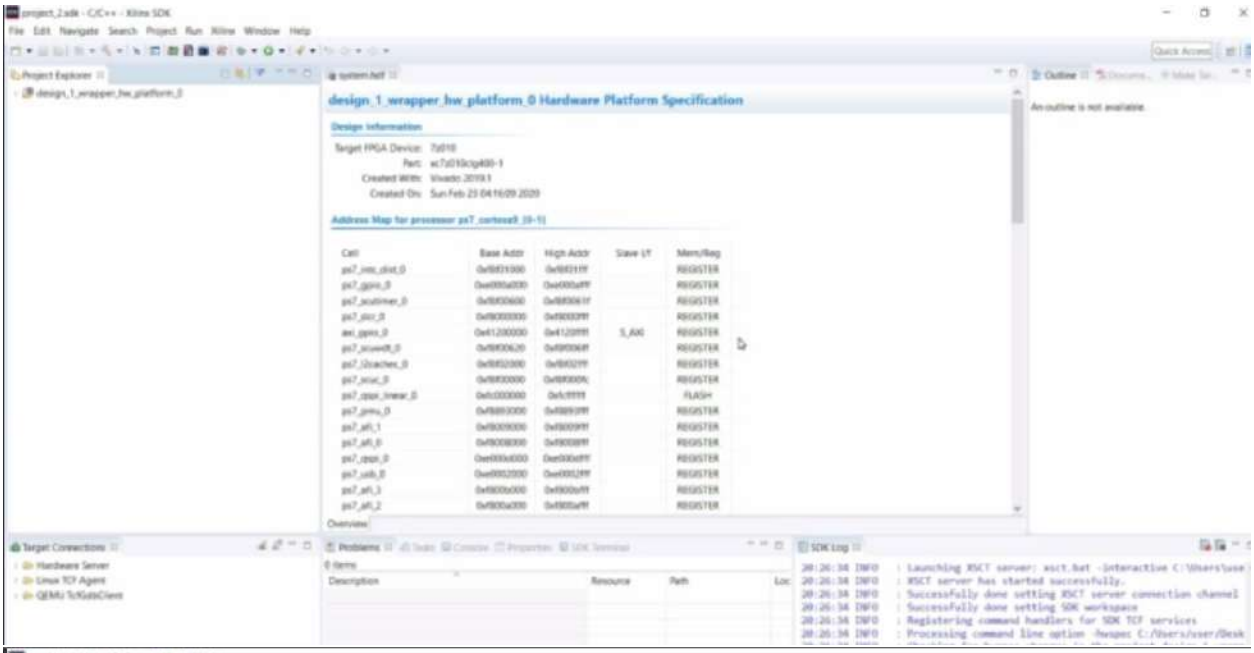


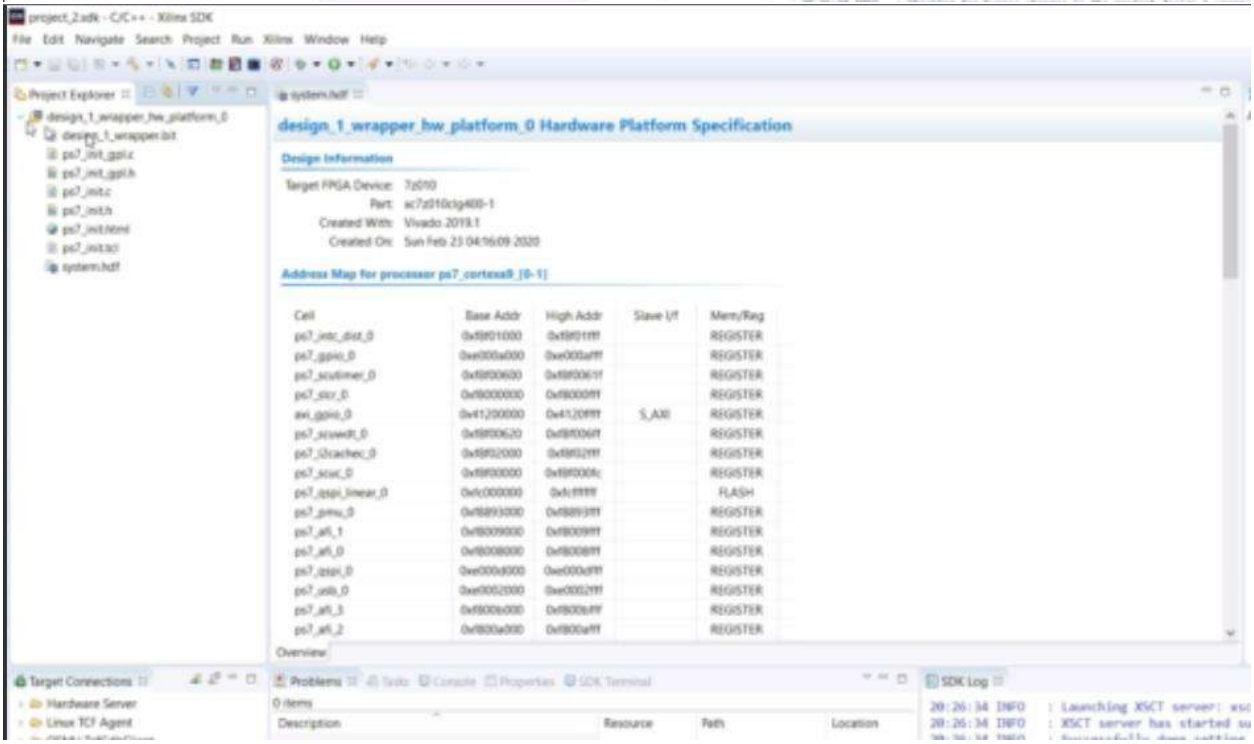
➔ Now click on generate bitsream

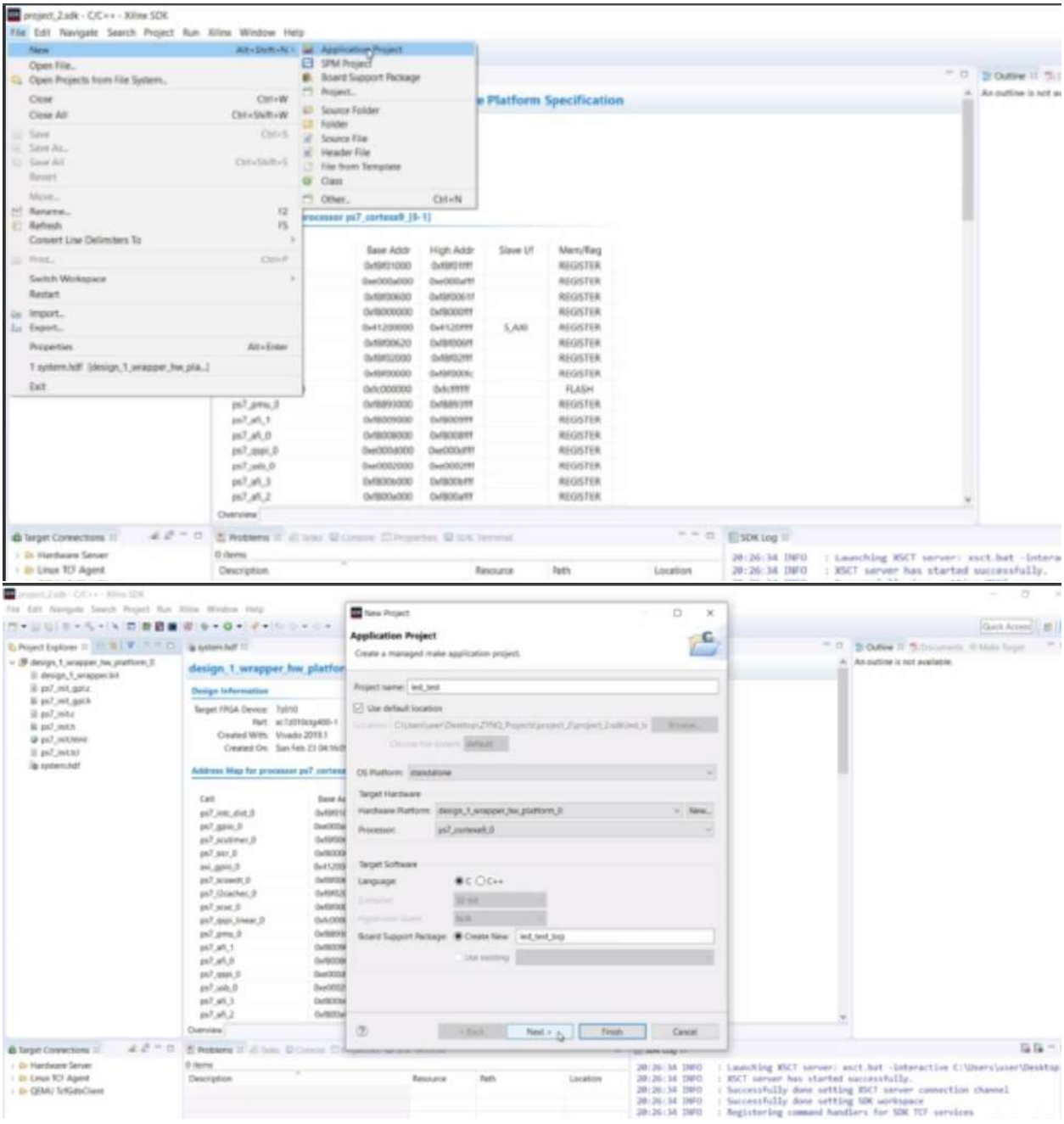

Save Project
Save project before generating bitstream?
Data to Save
Block Design - design_1
Save
Don't Save
Cancel
Bitstream Generation Completed
Bitstream Generation successfully completed.
Next
Open Implemented Design
View Reports
Open Hardware Manager
Generate Memory Configuration File
Don't show this dialog again
OK
Cancel
BLOCK DESIGN - design_1
ZYNQ7 Processing System
AXI Interconnect
AXI GPIO
DDR
FIXED_IO

\

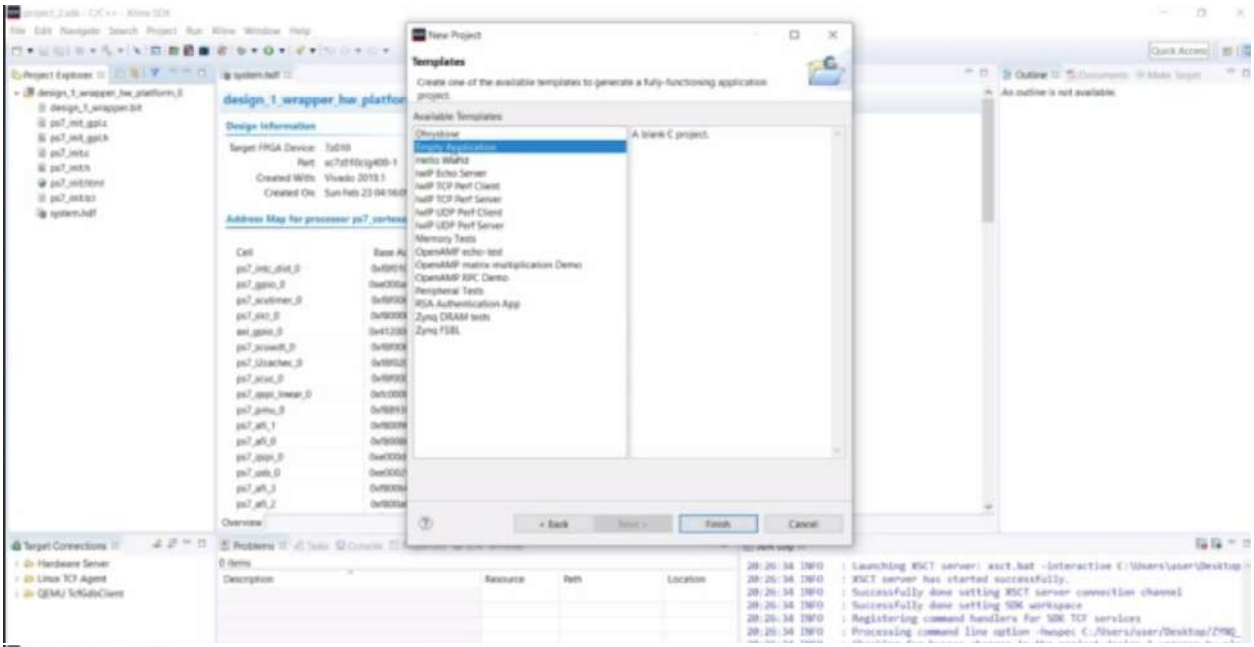

Export Hardware...
Launch SDK
processing_system7_0
ZYNQ7 Processing System
ps7_0_axi_periph
AXI Interconnect
axi_gpio_0
AXI GPIO
leds_4bits
DDR
FIXED_IO

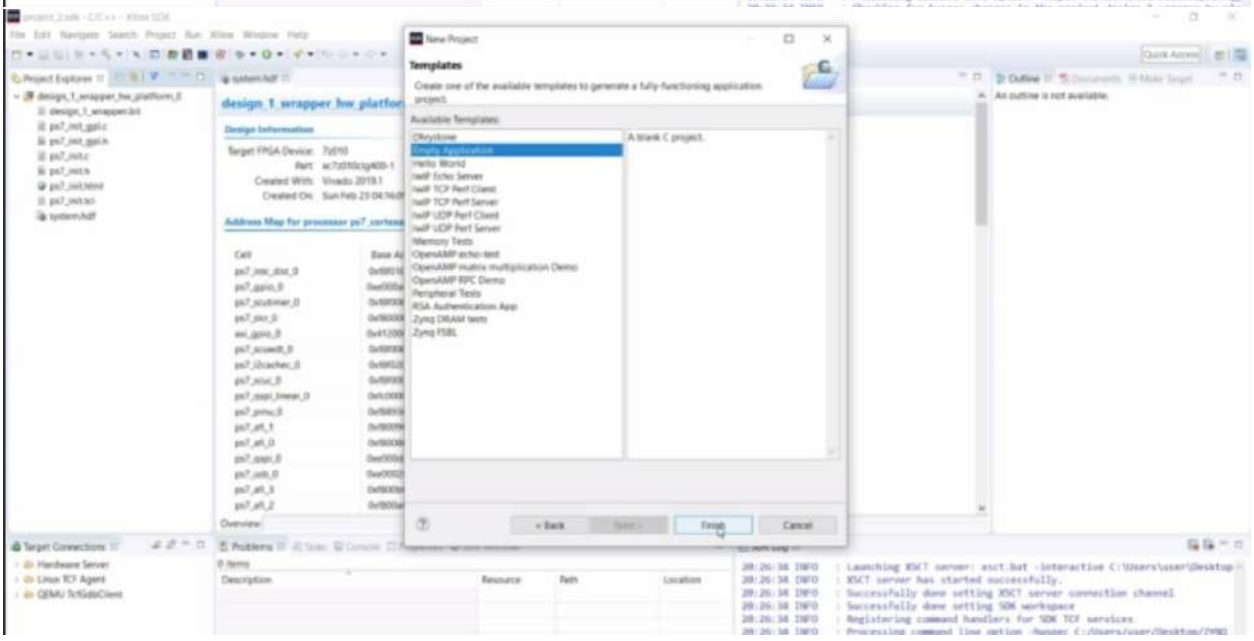

project_2.sdk - C/C++ - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Project Explorer
design_1_wrapper_hw_platform_0
system.hdf
design_1_wrapper_hw_platform_0 Hardware Platform Specification
Design Information
Target FPGA Device: 7z010
Part: xc7z010clg400-1
Created With: Vivado 2019.1
Created On: Sun Feb 23 04:16:09 2020
Address Map for processor ps7_cortexa9_0 (0-1)
Cell
Base Addr
High Addr
Slave I/f
Mem/Reg
Target Connections
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
Problems
0 items
Description
Resource
Path
SDK Log
Outline
An outline is not available.

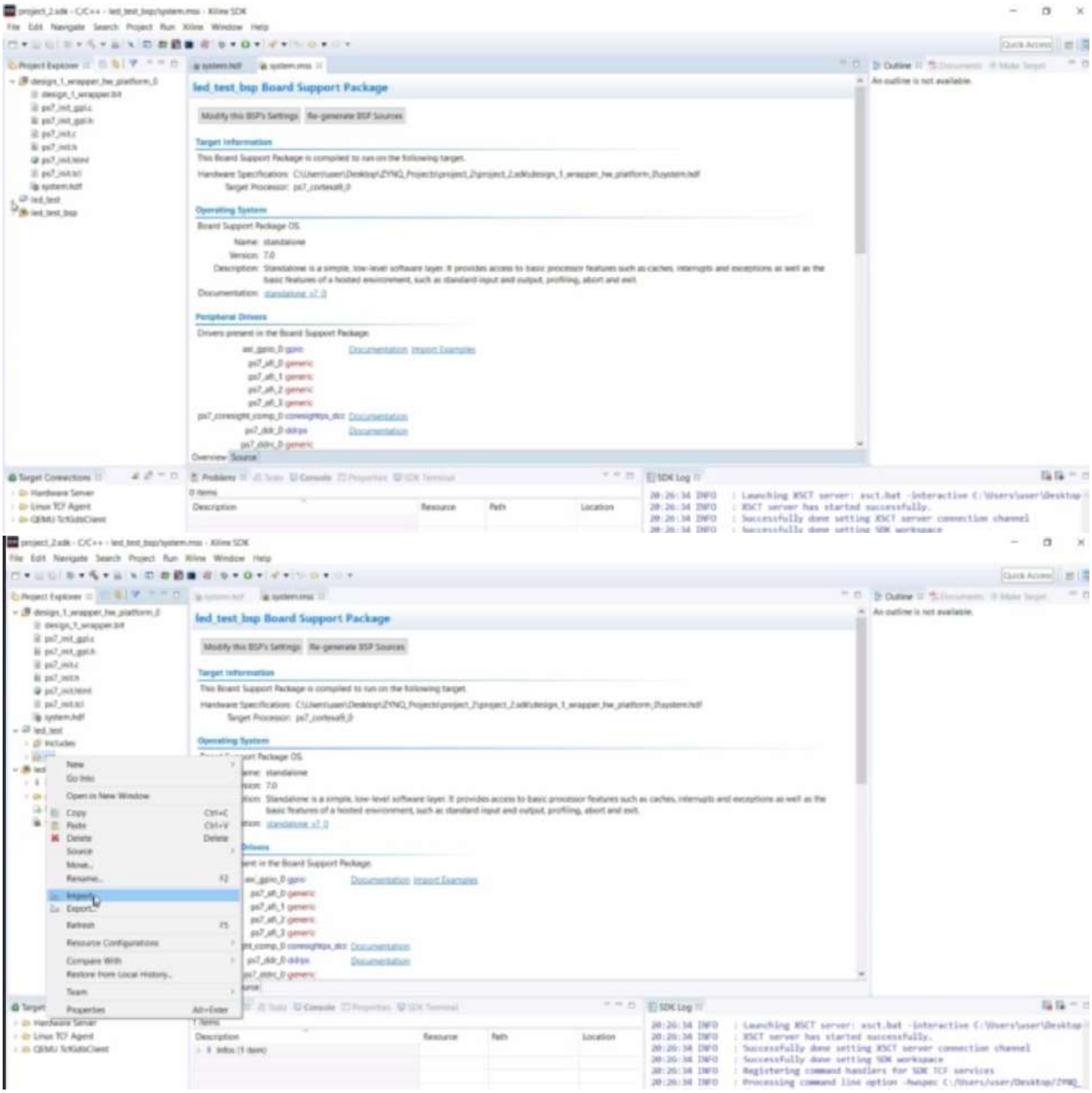

project_2.sdk - C/C++ - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Project Explorer
design_1_wrapper_hw_platform_0
design_1_wrapper.bit
ps7_init_gpl.c
ps7_init_gpl.h
ps7_init.c
ps7_init.h
ps7_init.html
ps7_init.tcl
system.hdf
system.hdf
design_1_wrapper_hw_platform_0 Hardware Platform Specification
Design Information
Target FPGA Device: 7z010
Part: xc7z010clg400-1
Created With: Vivado 2019.1
Created On: Sun Feb 23 04:16:09 2020
Address Map for processor ps7_cortexa9_0 (0-1)
Cell
Base Addr
High Addr
Slave I/f
Mem/Reg
Overview
Target Connections
Hardware Server
Linux TCF Agent
Problems
0 items
Description
Resource
Path
Location
SDK Log

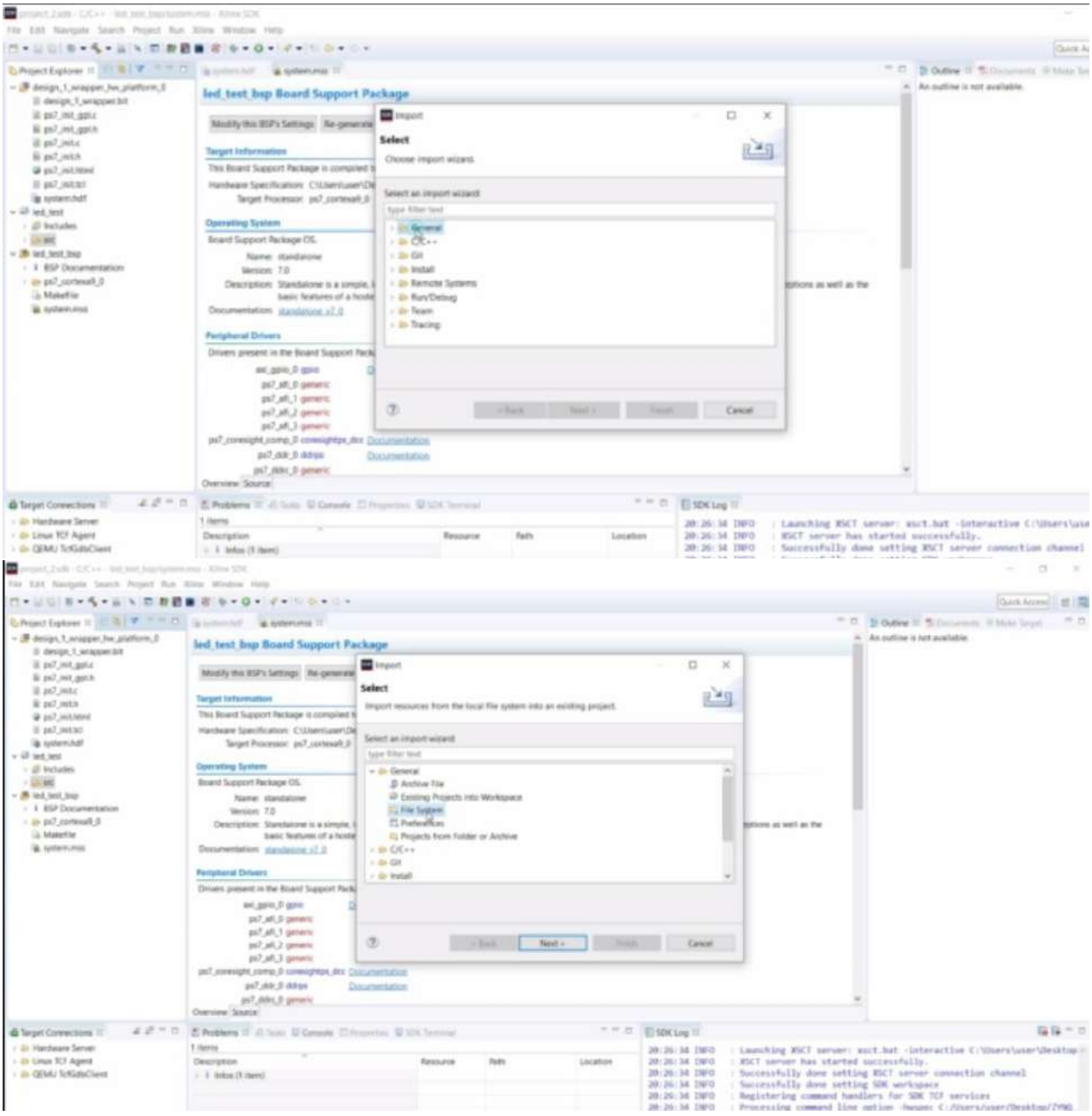

project_2.sdk - C/C++ - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
New
Open File...
Open Projects from File System...
Close
Close All
Save
Save As...
Save All
Revert
Move...
Rename...
Refresh
Convert Line Delimiters To
Print...
Switch Workspace
Restart
Import...
Export...
Properties
1 system.hdf [design_1_wrapper_hw_pla...]
Exit
Application Project
SPM Project
Board Support Package
Project...
Source Folder
Folder
Source File
Header File
File from Template
Class
Other...
Platform Specification
Base Addr
High Addr
Slave I/f
Mem/Reg
REGISTER
FLASH
S_AXI
Overview
Target Connections
Hardware Server
Linux TCF Agent
Problems
0 items
Description
Resource
Path
Location
SDK Log
20:26:34 INFO : Launching XSCT server: xsct.bat -interactive
20:26:34 INFO : XSCT server has started successfully.

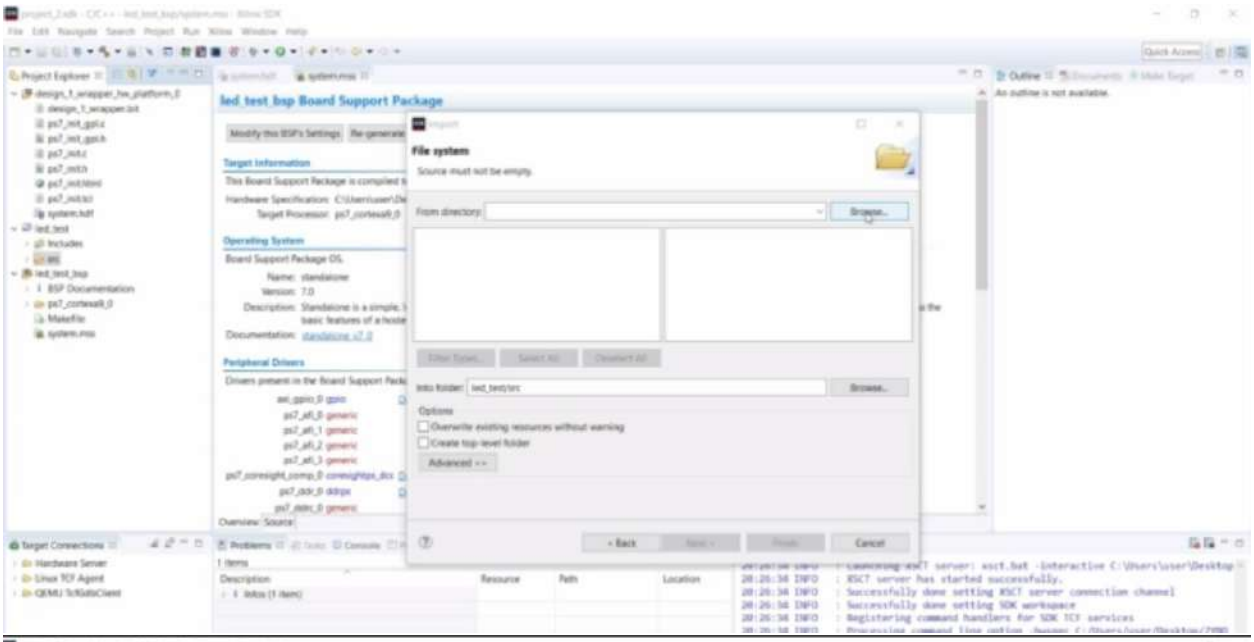
New Project
Application Project
Create a managed make application project.
Project name:
Use default location
OS Platform:
standalone
Target Hardware
Hardware Platform:
Processor:
Target Software
Language:
Board Support Package:
Create New
Use existing
< Back
Next >
Finish
Cancel
Project Explorer
Outline
An outline is not available.
Target Connections
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
Problems
0 items
Description
Resource
Path
Location
20:26:34 INFO : Launching XSCT server: xsct.bat -interactive
20:26:34 INFO : XSCT server has started successfully.
20:26:34 INFO : Successfully done setting XSCT server connection channel
20:26:34 INFO : Successfully done setting SDK workspace
20:26:34 INFO : Registering command handlers for SDK TCF services

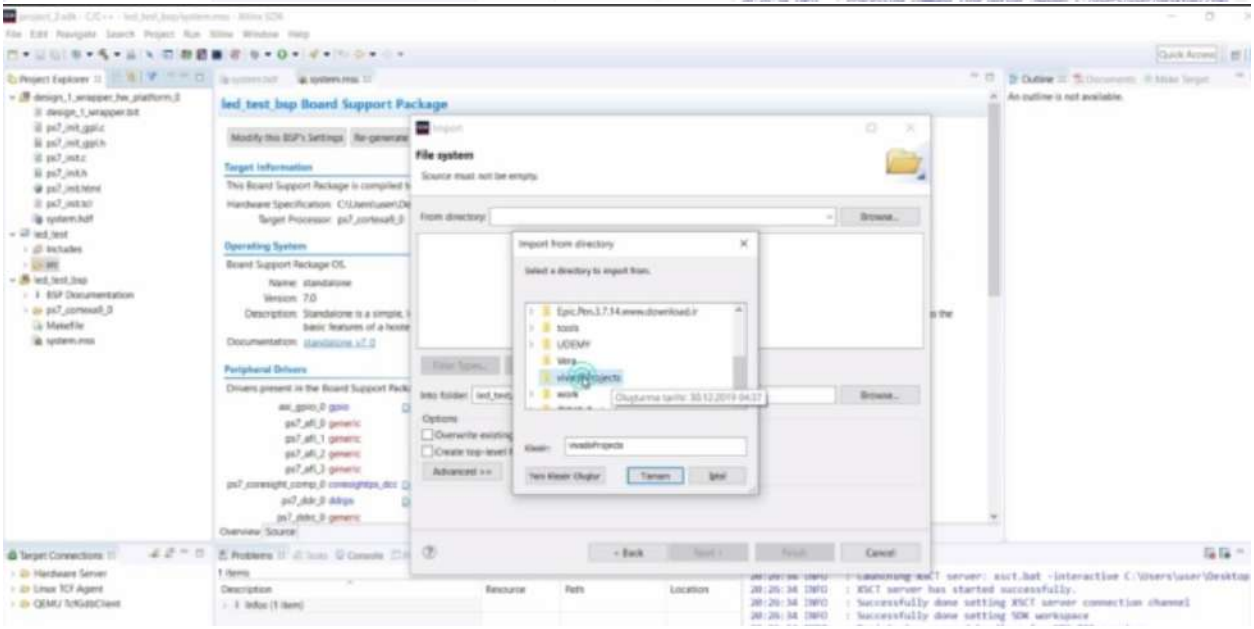

New Project
Templates
Create one of the available templates to generate a fully-functioning application project.
Available Templates:
Empty Application
Hello World
A blank C project.
< Back
Next >
Finish
Cancel

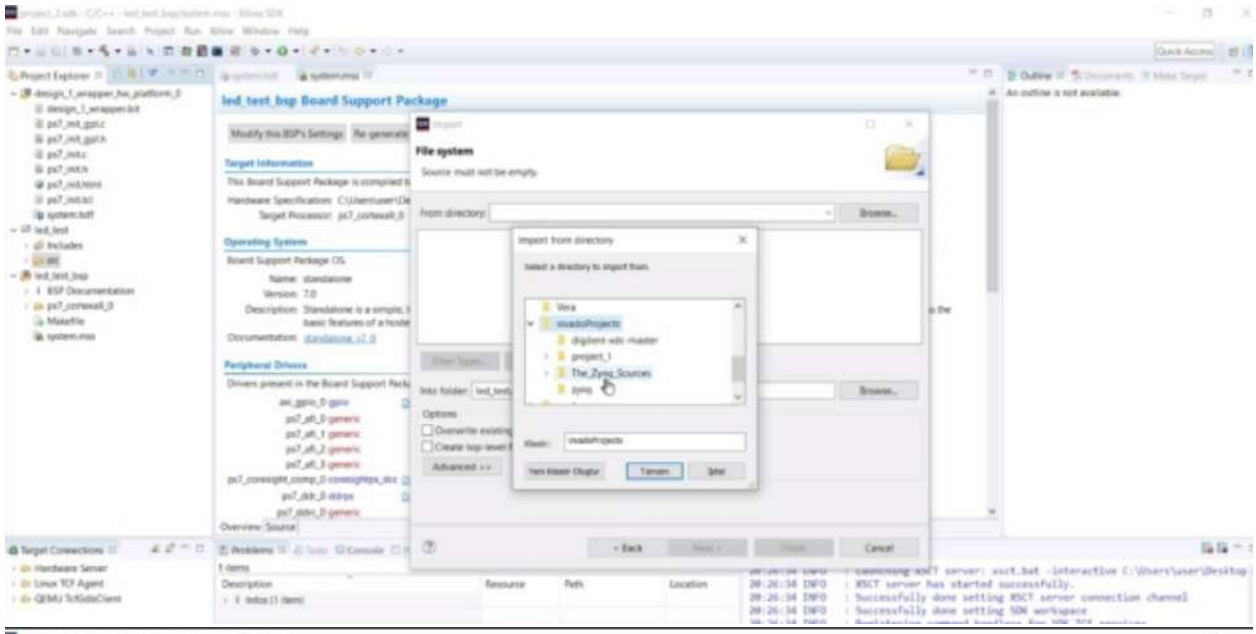

New Project
Templates
Create one of the available templates to generate a fully-functioning application project.
Available Templates:
Empty Application
Hello World
A blank C project.
< Back
Next >
Finish
Cancel

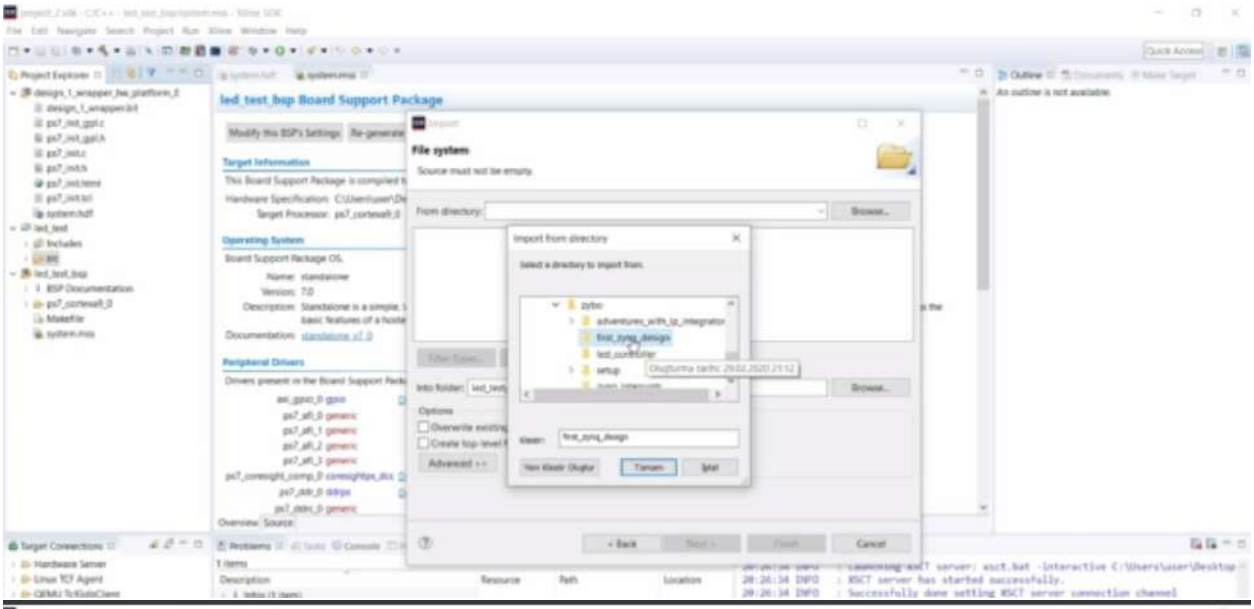

led_test_bsp Board Support Package
Modify this BSP's Settings
Re-generate BSP Sources
Target Information
Operating System
Peripheral Drivers
led_test_bsp Board Support Package
Modify this BSP's Settings
Re-generate BSP Sources
Target Information
Operating System
Peripheral Drivers
New
Go Into
Open in New Window
Copy
Paste
Delete
Source
Move...
Rename...
Import...
Export...
Refresh
Resource Configurations
Compare With
Restore from Local History...
Team
Properties

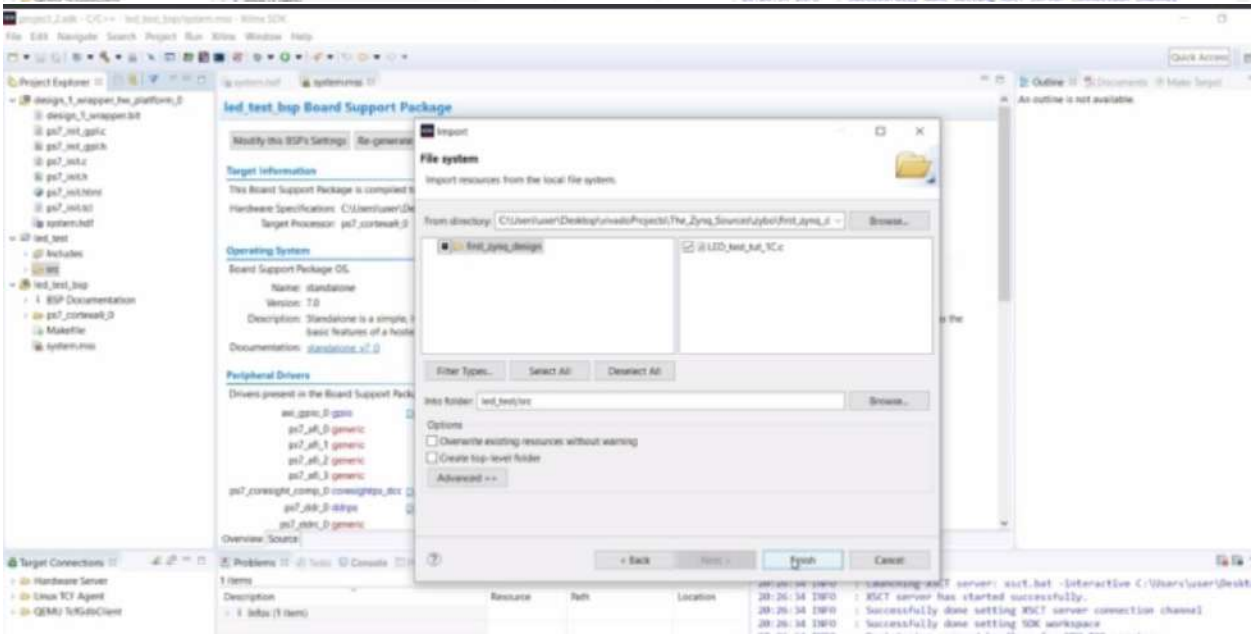

led_test_bsp Board Support Package
Import
Select
Choose import wizards.
Select an import wizard:
General
C/C++
Git
Install
Remote Systems
Run/Debug
Team
Tracing
Cancel
led_test_bsp Board Support Package
Import
Select
Import resources from the local file system into an existing project.
Select an import wizard:
General
Archive File
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
C/C++
Git
Install
Next >
Cancel

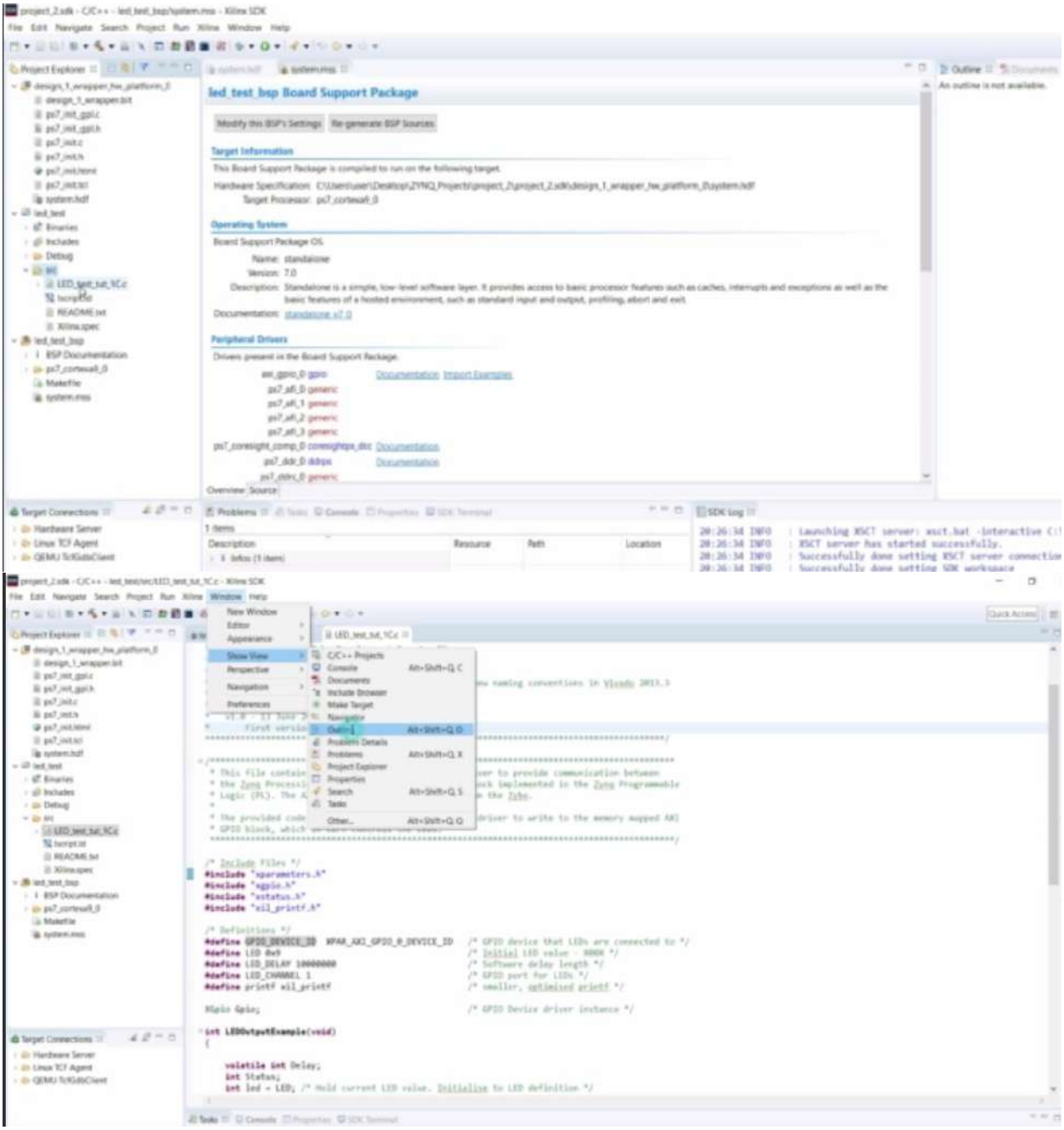

led_test_bsp Board Support Package
File system
Source must not be empty.
From directory:
Browse...
Into folder:
Options
Overwrite existing resources without warning
Create top-level folder
Advanced >>
< Back
Next >
Finish
Cancel

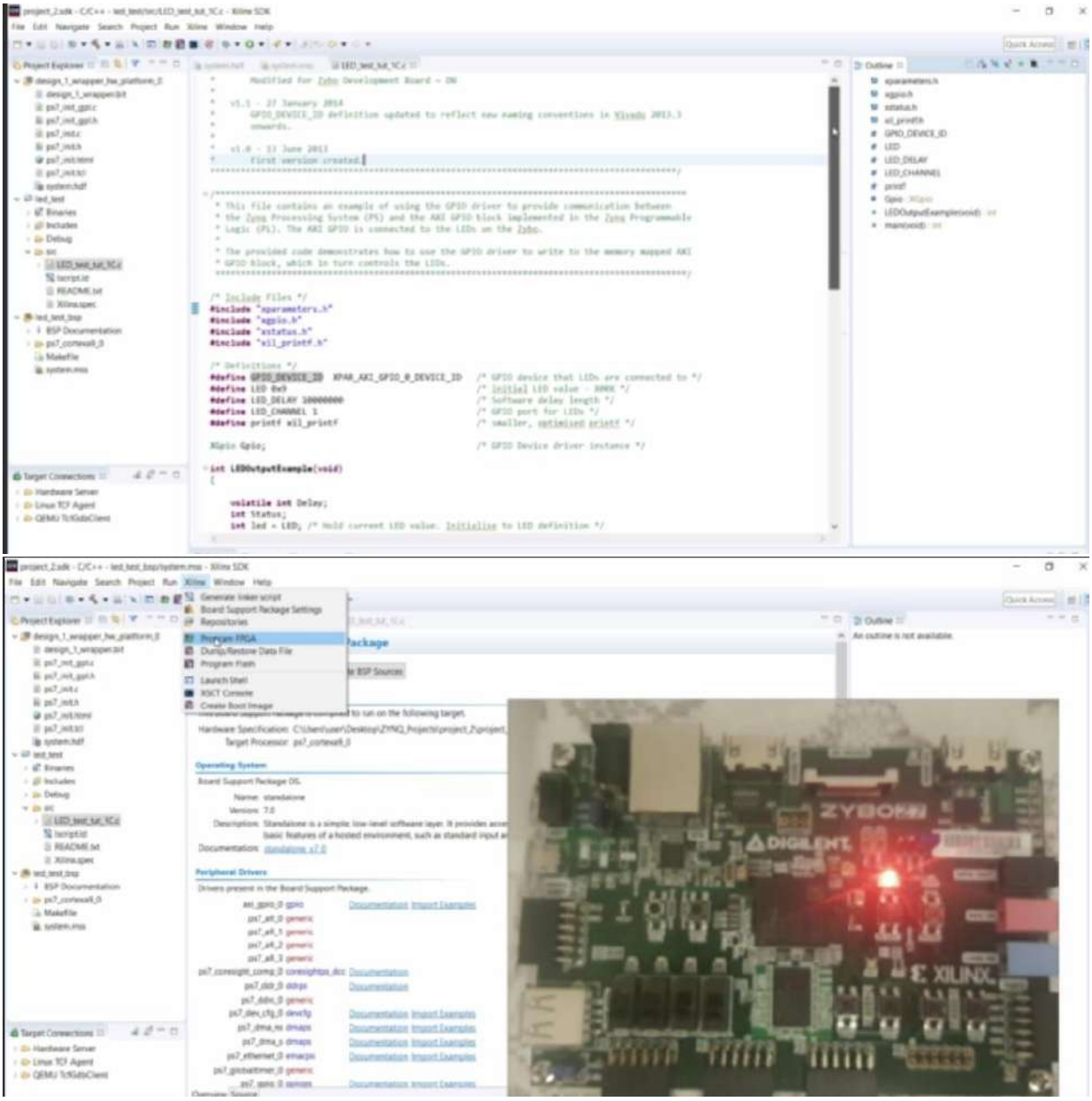

led_test_bsp Board Support Package
File system
Source must not be empty.
From directory:
Browse...
Import from directory
Select a directory to import from.
Into folder:
Options
Advanced >>
< Back
Next >
Finish
Cancel

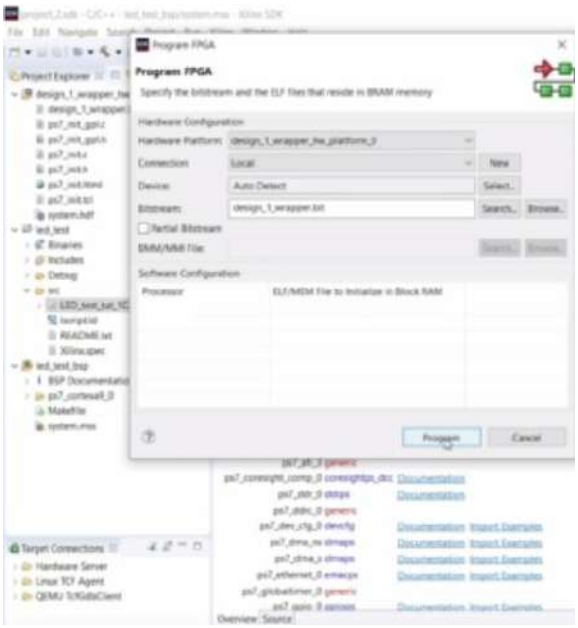

led_test_bsp Board Support Package
File system
Source must not be empty.
Import from directory
Select a directory to import from.
vivadoProjects
digilent-xdc-master
project_1
The_Zynq_Sources
zynq
From directory:
Into folder:
Options
Overwrite existing
Create top-level
Advanced ++
Browse...
< Back
Next >
Finish
Cancel
led_test_bsp Board Support Package
File system
Source must not be empty.
Import from directory
Select a directory to import from.
hls
hls_nco
input
zedboard
zynq
The_Zynq_Sources
From directory:
Into folder:
Options
Overwrite existing
Create top-level
Advanced ++
Browse...
< Back
Next >
Finish
Cancel

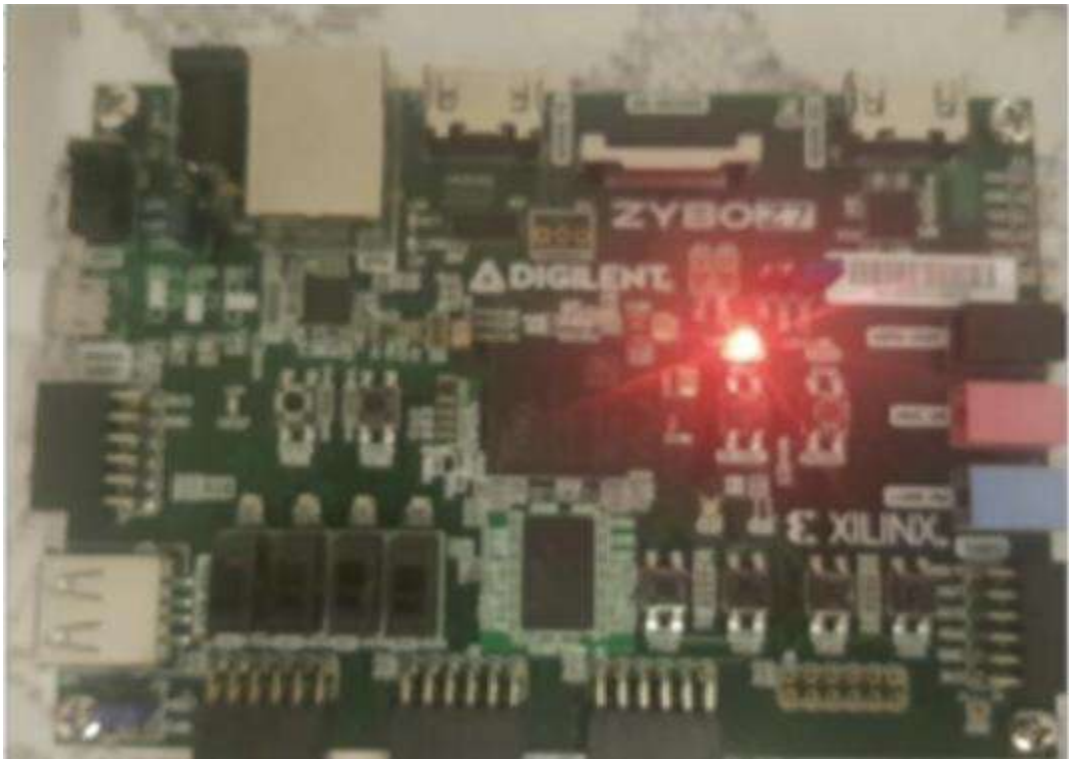

led_test_bsp Board Support Package
Import
File system
Source must not be empty.
From directory:
Browse...
Import from directory
Select a directory to import from:
zybo
adventures_with_ip_integrator
first_zynq_design
test_controller
setup
Into folder:
Options
Overwrite existing
Create top-level
Advanced ++
Klasör:
first_zynq_design
Yeni Klasör Oluştur
Tamam
İptal
< Back
Next >
Finish
Cancel
Outline
An outline is not available.
Target Connections
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
Problems
Description
Resource
Path
Location
20:26:34 INFO : XSCT server has started successfully.
20:26:34 INFO : Successfully done setting XSCT server connection channel


led_test_bsp Board Support Package
Import
File system
Import resources from the local file system.
From directory:
Browse...
first_zynq_design
LED_test_full_TC.c
Filter Types...
Select All
Deselect All
Into folder:
led_test/src
Options
Overwrite existing resources without warning
Create top-level folder
Advanced ++
< Back
Next >
Finish
Cancel
Outline
An outline is not available.
Target Connections
Hardware Server
Linux TCF Agent
QEMU TcfGdbClient
Problems
Description
Resource
Path
Location
20:26:34 INFO : XSCT server has started successfully.
20:26:34 INFO : Successfully done setting XSCT server connection channel
20:26:34 INFO : Successfully done setting SDK workspace


led_test_bsp Board Support Package
Modify this BSP's Settings
Re-generate BSP Sources
Target Information
Operating System
Peripheral Drivers
Window
Show View
Console
Outline
Problems
Project Explorer
Properties
Search
Tasks
Other...
#include "xparameters.h"
#include "xgpio.h"
#include "xstatus.h"
#include "xil_printf.h"
#define GPIO_DEVICE_ID XPAR_AXI_GPIO_0_DEVICE_ID
#define LED 0x9
#define LED_DELAY 10000000
#define LED_CHANNEL 1
#define printf xil_printf
XGpio Gpio;
int LEDOutputExample(void)
volatile int Delay;
int Status;
int led = LED;

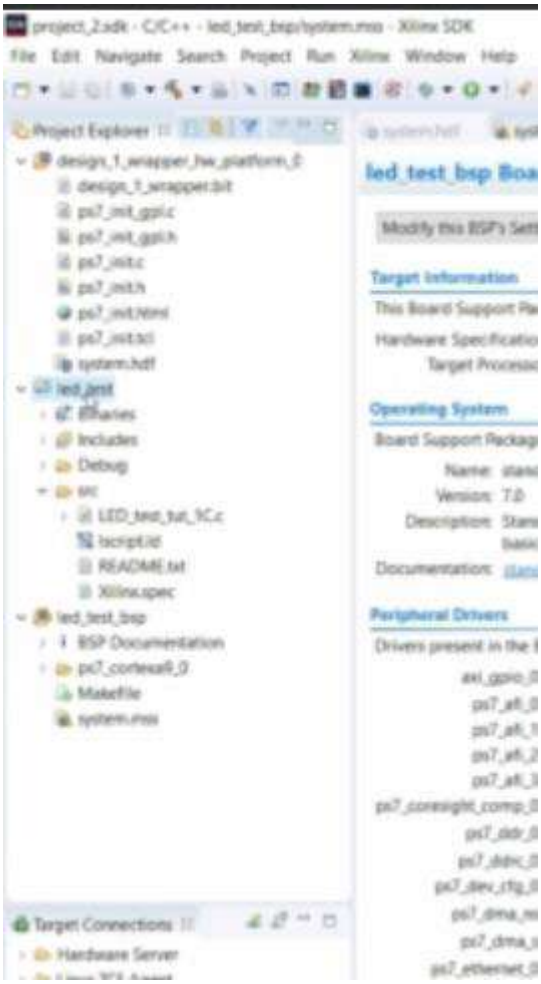

Program FPGA
Dump/Restore Data File
Program Flash
Launch Shell
XSCT Console
Create Boot Image
ZYBO
DIGILENT
XILINX

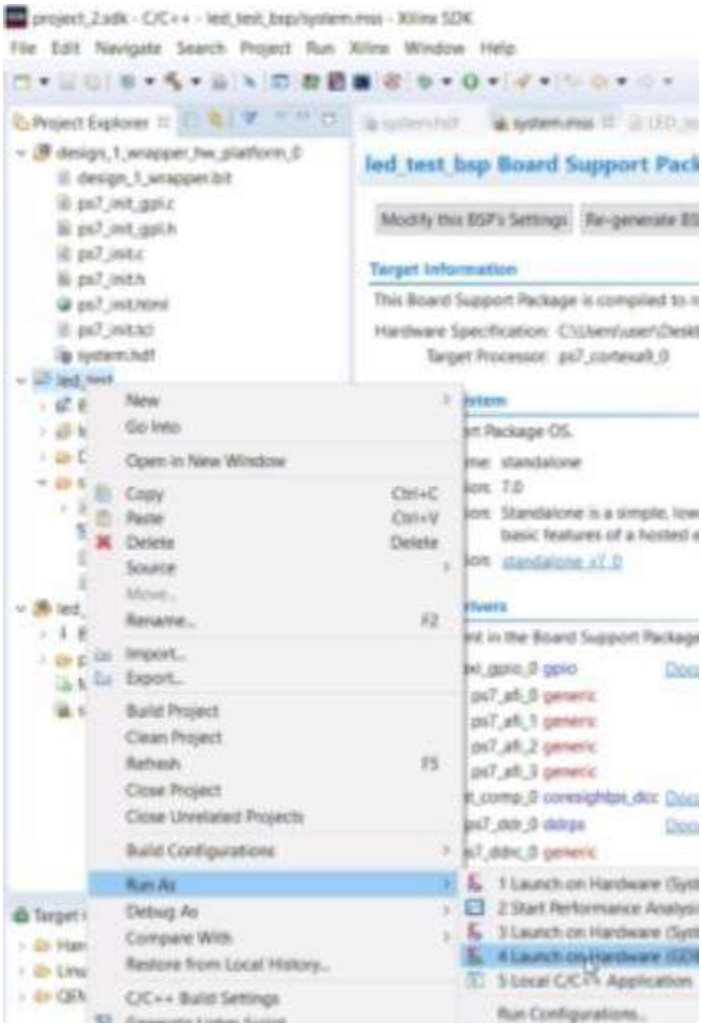


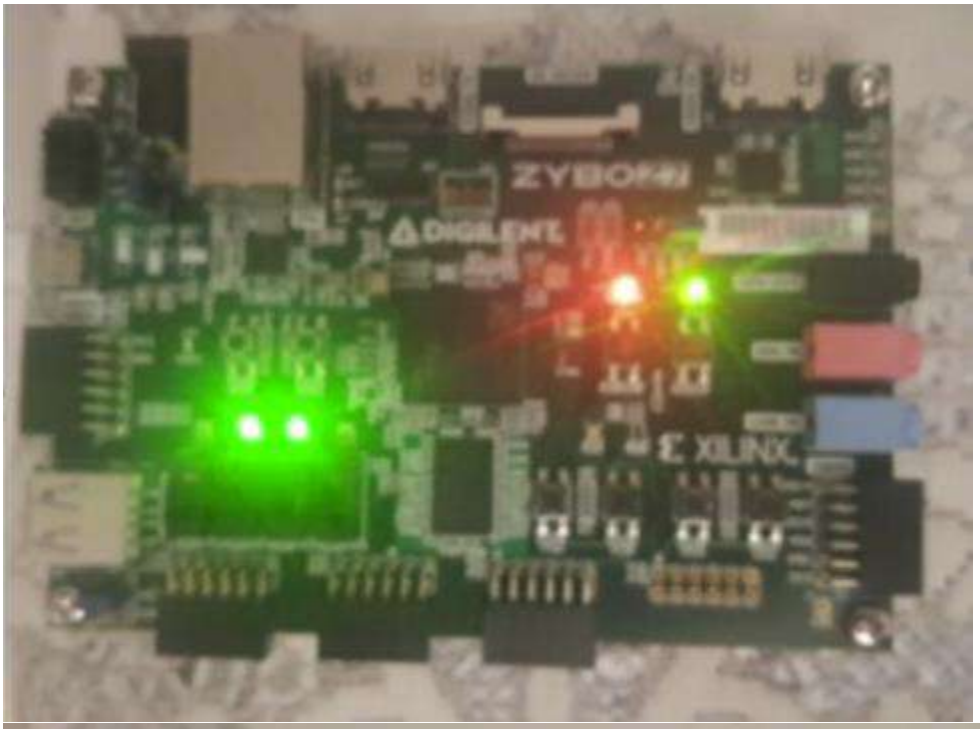


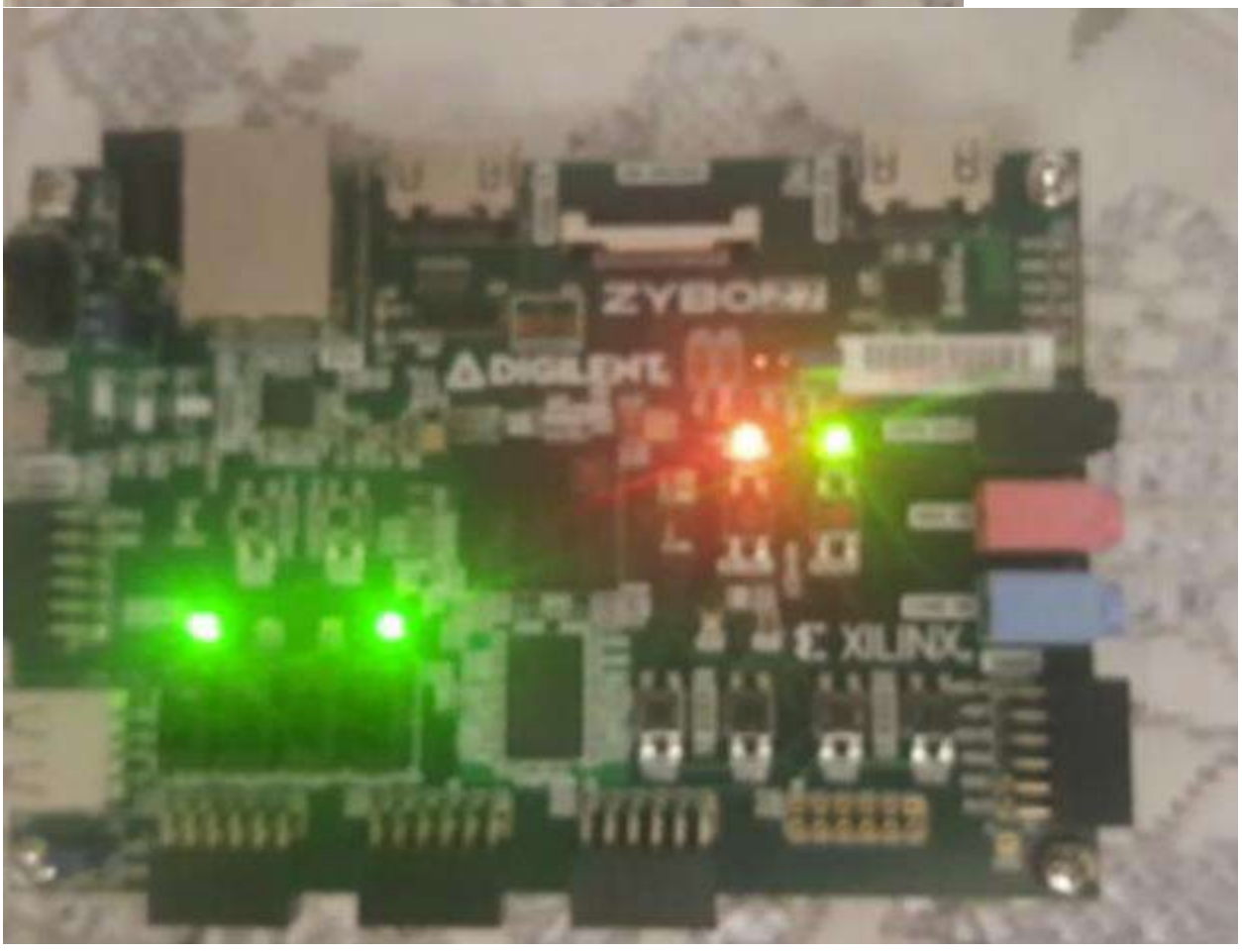
→when the program is complete we'll get green light which shows that our bitsream file is being uploaded:

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


project_2.sdk - C/C++ - led_test_bsp/system.mss - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Project Explorer
design_1_wrapper_hw_platform_0
design_1_wrapper.bit
ps7_init_gpl.c
ps7_init_gpl.h
ps7_init.c
ps7_init.h
ps7_init.html
ps7_init.tcl
system.hdf
led_test
Binaries
Includes
Debug
src
LED_test_bit_1C.c
lscript.ld
README.txt
Xilinx.spec
led_test_bsp
BSP Documentation
ps7_cortexa9_0
Makefile
system.mss
Target Connections
Hardware Server
led_test_bsp Bo
Modify this BSP's Sett
Target Information
This Board Support Pa
Hardware Specificatio
Target Processo
Operating System
Board Support Packag
Name: stan
Version: 7.0
Description: Stan
basic
Documentation:
Peripheral Drivers
Drivers present in the
axi_gpio_0
ps7_afi_0
ps7_afi_1
ps7_afi_2
ps7_afi_3
ps7_coresight_comp_0
ps7_ddr_0
ps7_ddrc_0
ps7_dev_cfg_0
ps7_dma_ns
ps7_dma_s
ps7_ethernet_0


project_2.sdk - C/C++ - led_test_bsp/system.mss - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Project Explorer
design_1_wrapper_hw_platform_0
design_1_wrapper.bit
ps7_init_gpl.c
ps7_init_gpl.h
ps7_init.c
ps7_init.h
ps7_init.html
ps7_init.tcl
system.hdf
led_test
led_test_bsp Board Support Pack
Modify this BSP's Settings
Target Information
This Board Support Package is compiled to
Hardware Specification:
Target Processor: ps7_cortexa9_0
New
Go Into
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Build Configurations
Run As
Debug As
Compare With
Restore from Local History...
C/C++ Build Settings
Package OS.
standalone
7.0
Standalone is a simple, low
basic features of a hosted
standalone v7.0
in the Board Support Package
gpio
generic
generic
generic
generic
coresightps_dcc
dcrga
generic
1 Launch on Hardware (Syst
2 Start Performance Analysi
3 Launch on Hardware (Syst
4 Launch on Hardware (GDB
5 Local C/C++ Application
Run Configurations...


ZYBO Rev.B
DIGILENT
XILINX


ZYBO Rev.B
DIGILENT
XILINX