The Abdus Salam
International Centre
for Theoretical Physics

IAEA

Joint ICTP-IAEA School on
Systems-on-Chip based on
FPGA for Scientific Instrumentation
and Reconfigurable Computing

Introduction to Laboratory 1

Hello World and GPIO In/Out

Prepared by
Romina Molina and Maynor Ballina

Multidisciplinary Laboratory
ICTP

Trieste - Italy
2023

Lab 1: Objectives

- Acquire the knowledge for a **SoC-FPGA design flow** developed with the Vitis Unified Software Platform.

Lab 1: Objectives

- Acquire the knowledge for a **SoC-FPGA design flow** developed with the Vitis Unified Software Platform.

- **Create the hardware** to configure the FPGA part of the SoC, configure the PS (GPIO MIO, UART, DDR3 memory controller...), instantiate the GPIO blocks, and understand the communication between the different components of the design (Vivado).

Lab 1: Objectives

- Acquire the knowledge for a **SoC-FPGA design flow** developed with the Vitis Unified Software Platform.

- **Create the hardware** to configure the FPGA part of the SoC, configure the PS (GPIO MIO, UART, DDR3 memory controller...), instantiate the GPIO blocks, and understand the communication between the different components of the design (Vivado).

- **Create the 'C' application** (in Vitis) that will run on the PS to control the reading and the writing of the generated hardware (in Vivado).

Lab 1: Objectives

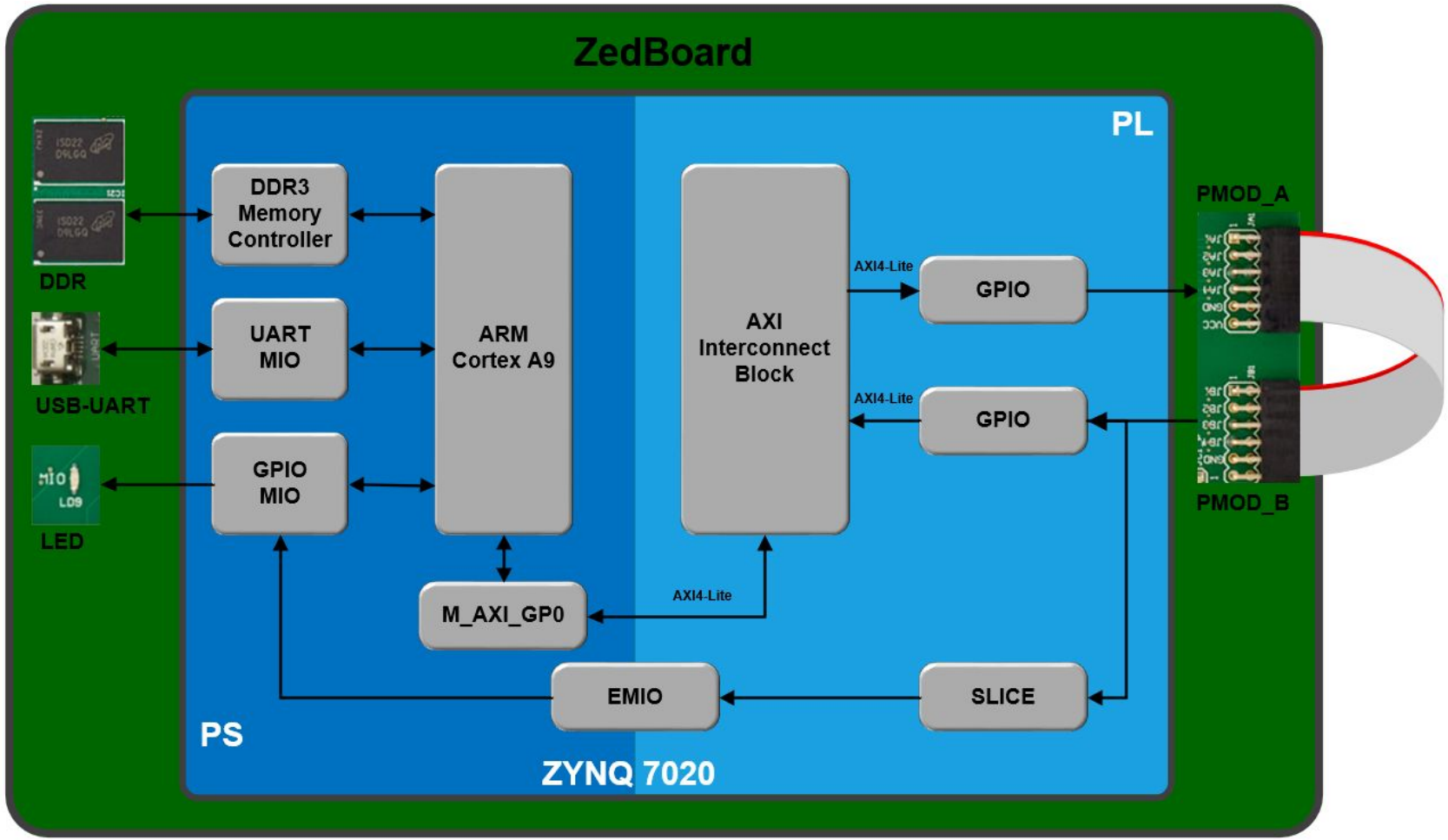
- Acquire the knowledge for a **SoC-FPGA design flow** developed with the Vitis Unified Software Platform.

- **Create the hardware** to configure the FPGA part of the SoC, configure the PS (GPIO MIO, UART, DDR3 memory controller...), instantiate the GPIO blocks, and understand the communication between the different components of the design (Vivado).

- **Create the 'C' application** (in Vitis) that will run on the PS to control the reading and the writing of the generated hardware (in Vivado).

- **Test the complete design on the ZedBoard platform** to verify the implementation (Vitis and serial port).

Design description



Acquire the knowledge for a **SoC-FPGA design flow**

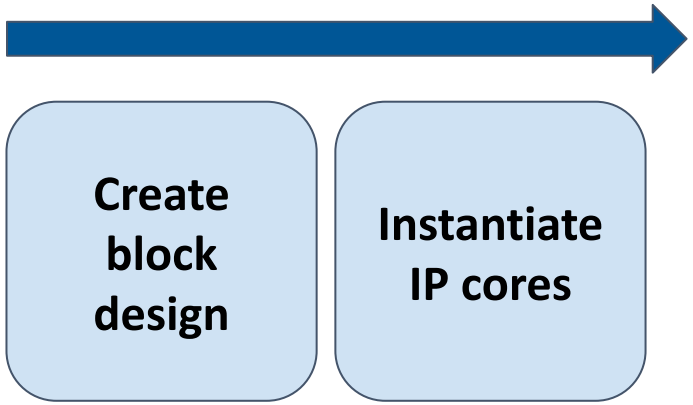
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation

Create block design

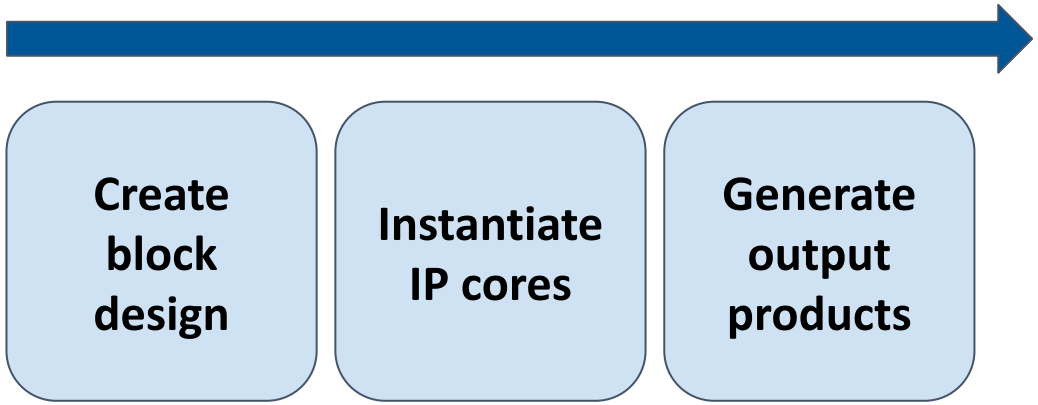
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



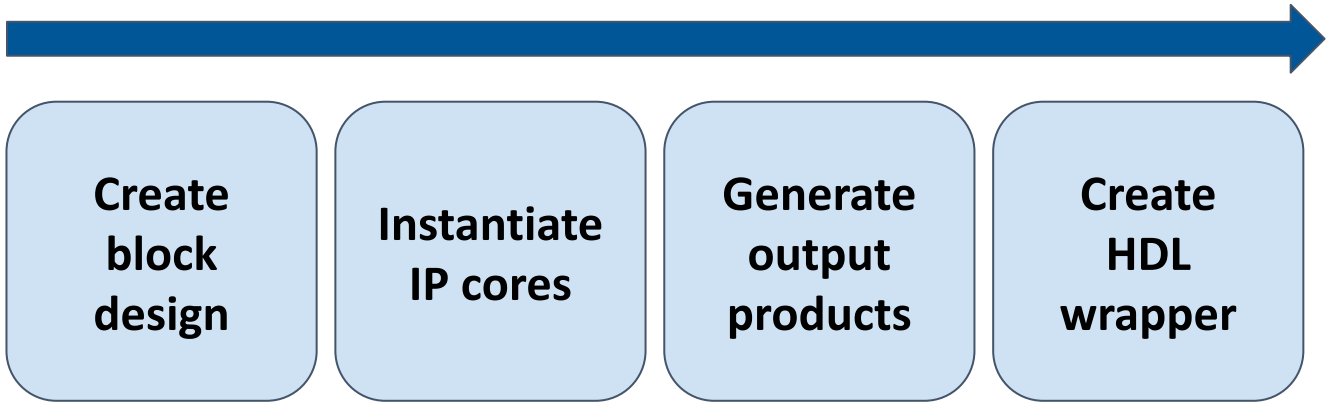
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



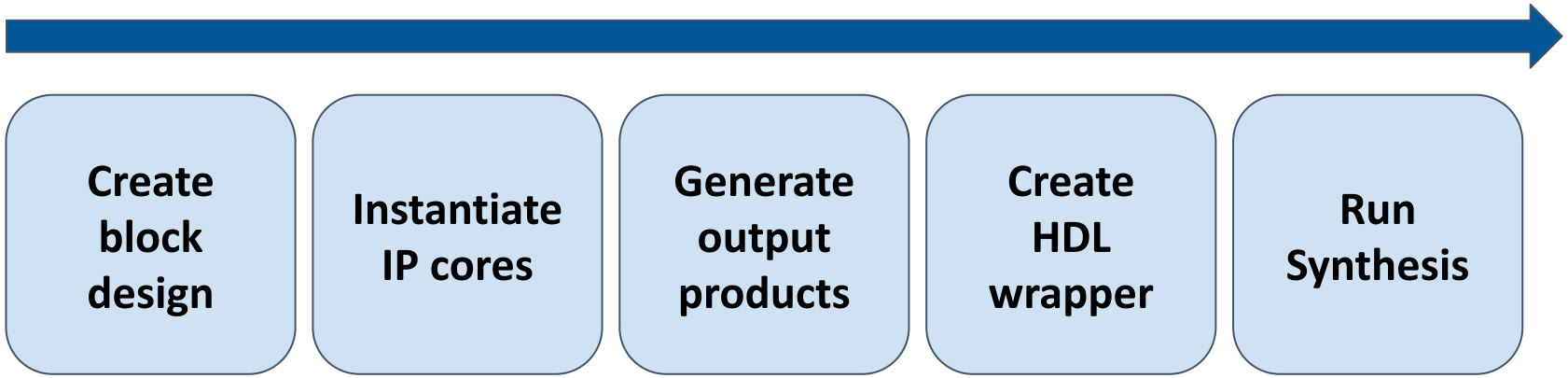
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



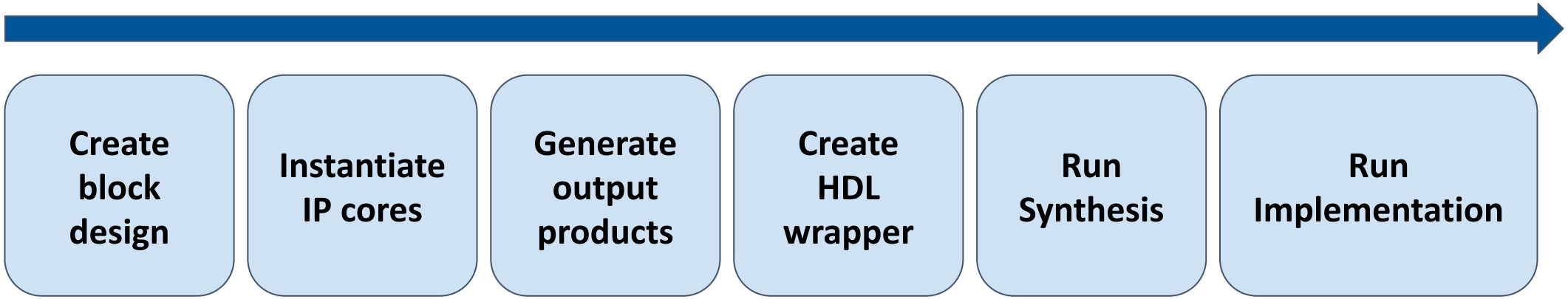
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



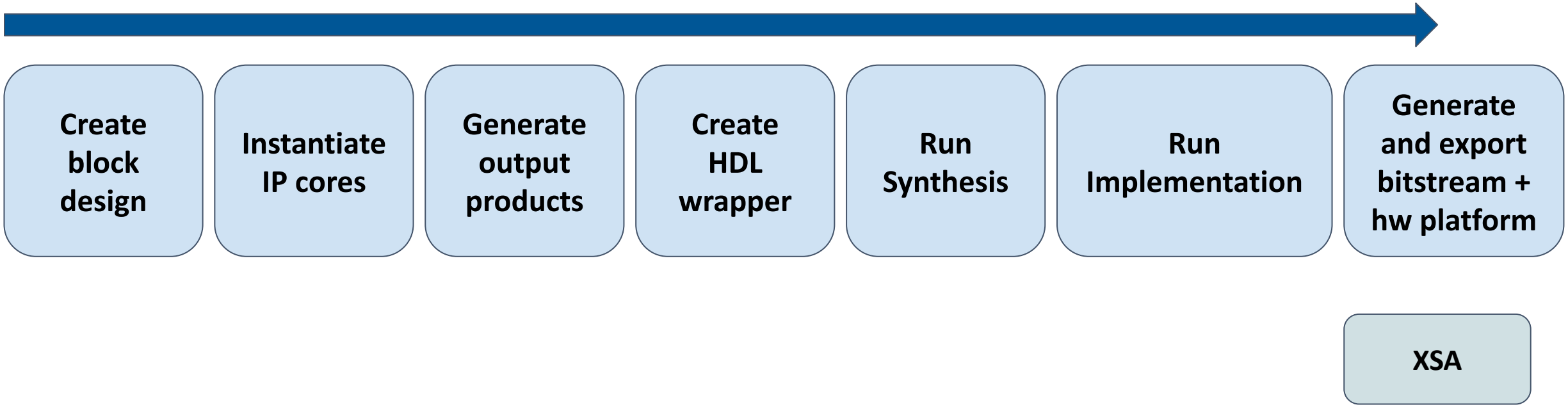
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow

Exported hardware from Vivado

XSA

SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow

Export hardware from Vivado

Import hardware spec from Vivado

XSA

SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow

Export hardware from Vivado

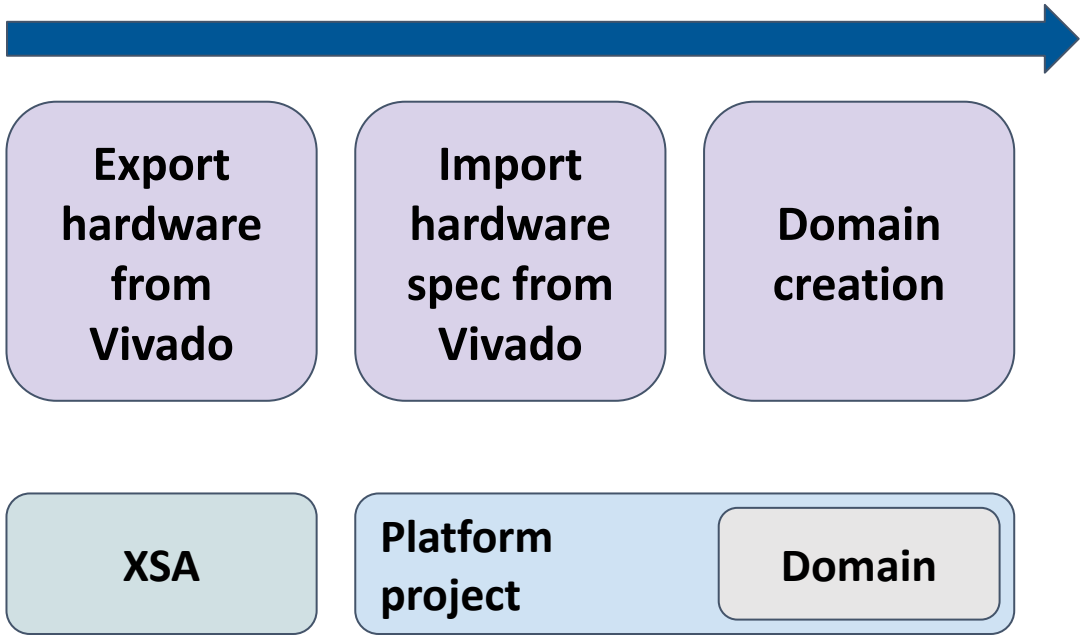
Import hardware spec from Vivado

Domain creation

XSA

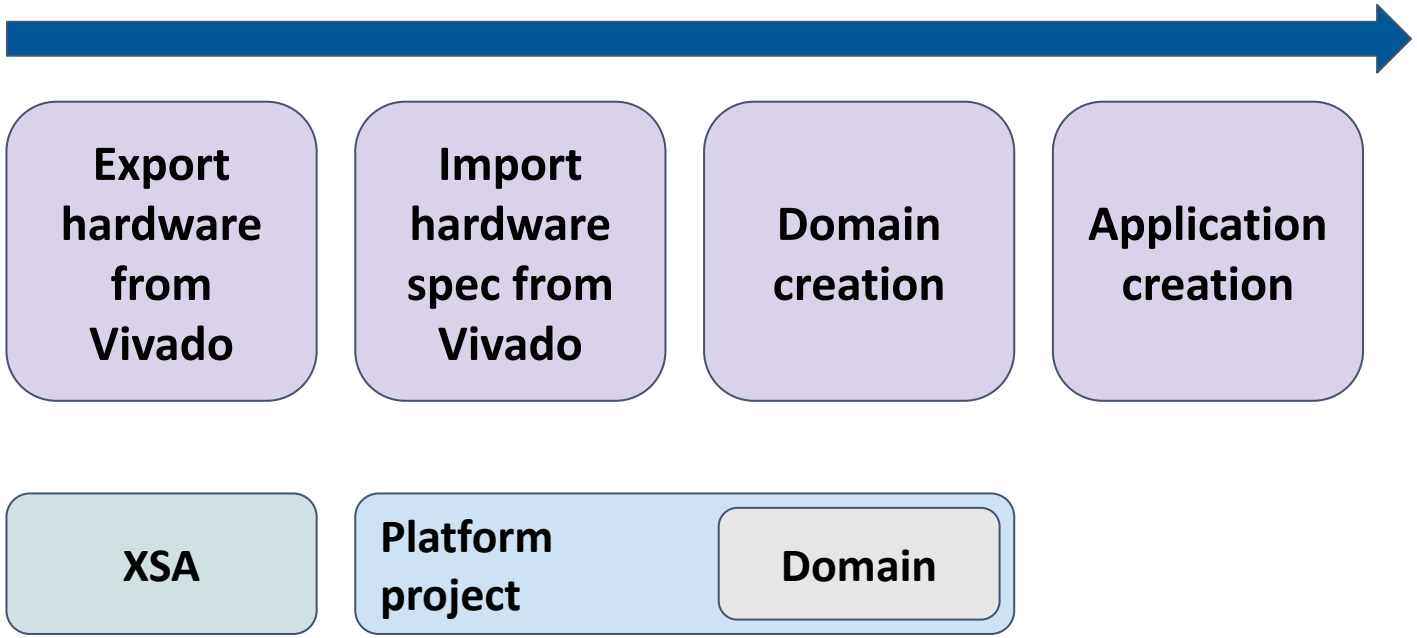
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow

Export hardware from Vivado

Import hardware spec from Vivado

Domain creation

Application creation

Application debugging

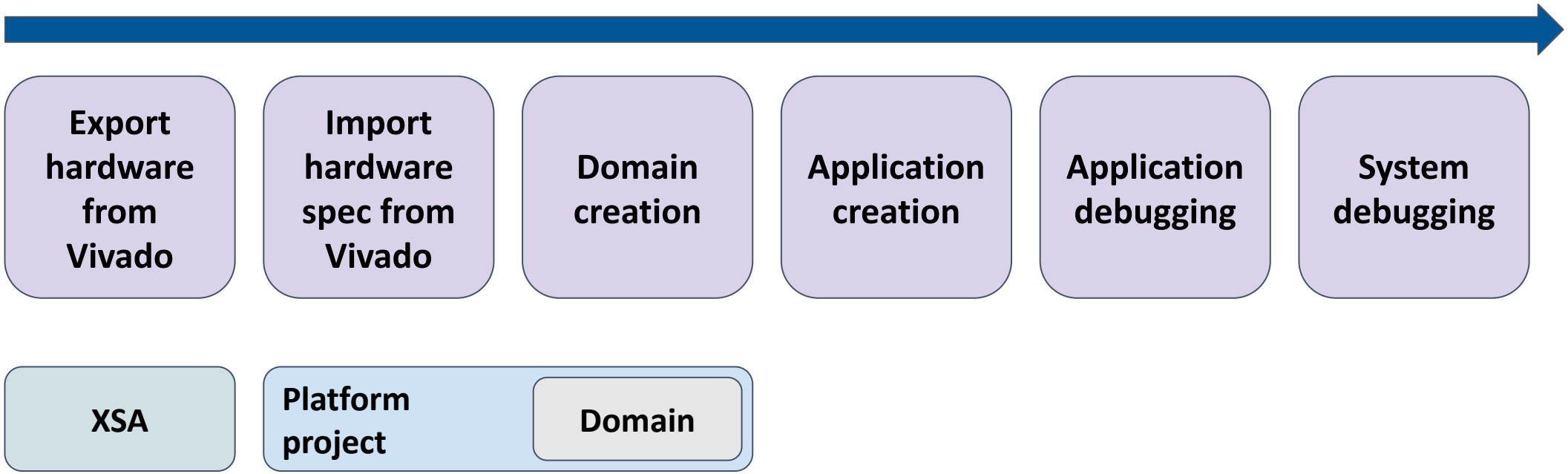
XSA

Platform project

Domain

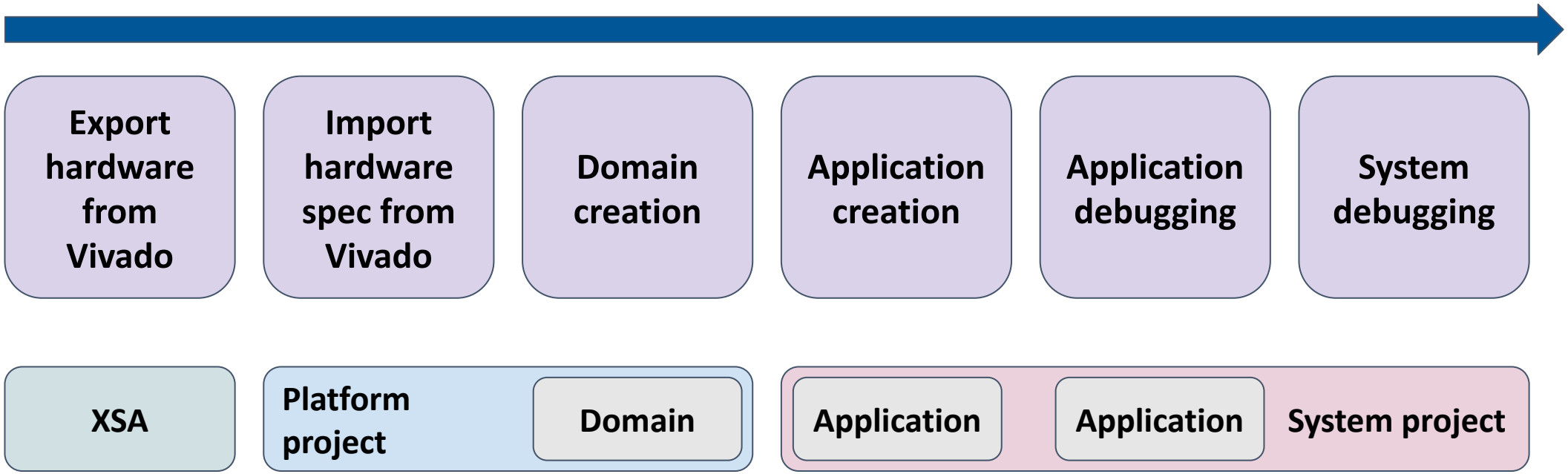
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



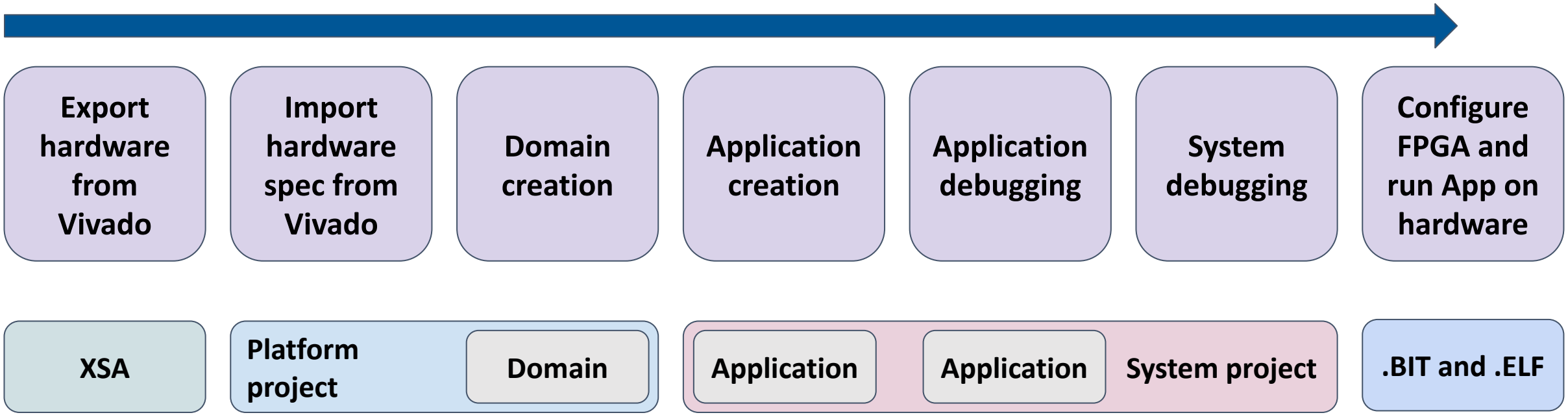
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



SoC-FPGA design flow for Lab 1

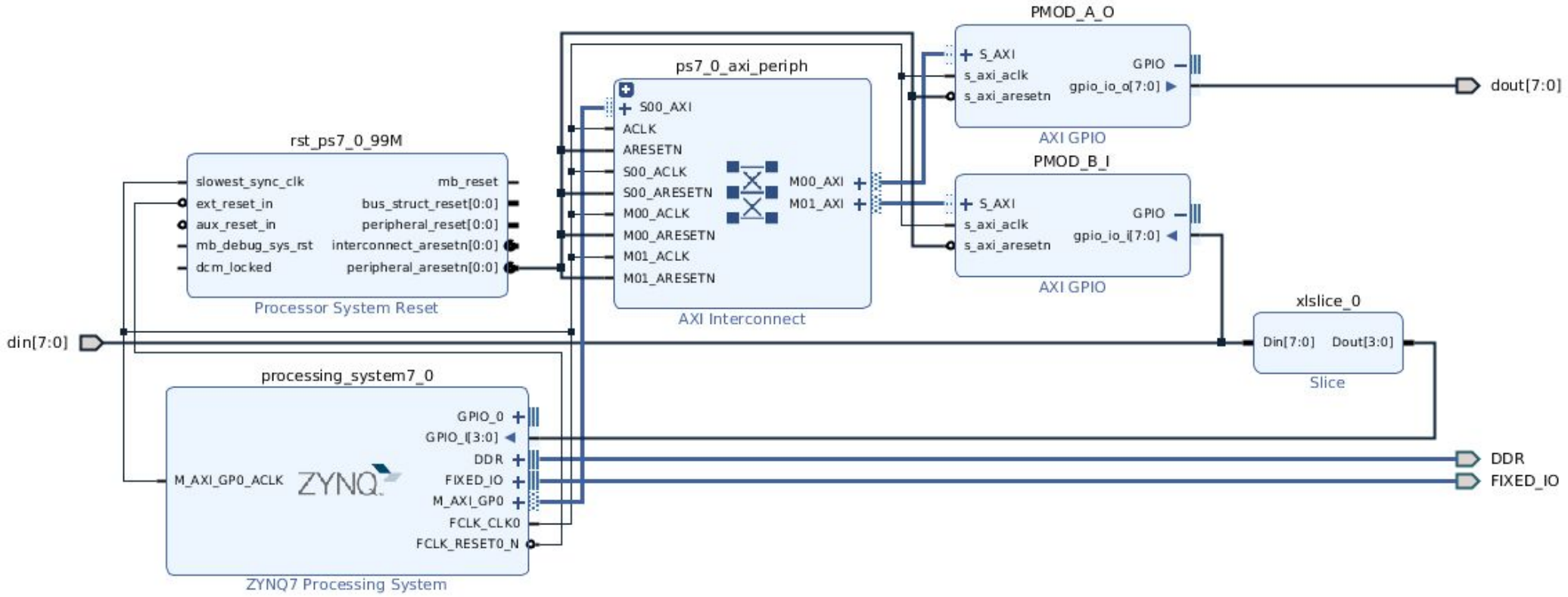
Vitis Software Development Workflow



Create the hardware to configure the FPGA part of the SoC,
configure the PS (GPIO MIO, UART, DDR3 memory controller...), instantiate the GPIO blocks, and understand the communication between the different components of the design (Vivado).

Hardware creation

Vivado



Create the 'C' application (in Vitis) that will run on the PS to control the reading and the writing of the generated hardware (in Vivado).

Software

C Application

- The code tests a loopback using PMOD connectors. Also, support is added to interface with MIO LED (MIO7).

```
34 #include "xparameters.h"
35 #include "xgpio.h"
36 #include "xgpiops.h"
37 #include "xil_printf.h"
38 #include "sleep.h"
39
40 static XGpioPs psGpioInstancePtr;
41
42 #define EMIO_GPIO_BANK 2 // EMIO Bank
43 #define MIO_GPIO_BANK 0  // GPIOs connected directly to PS
44 #define MIO_LED_PIN 7 // LED pin on MIO is directly connected to pin 7 (MIO7)
45
46
47 int main(void)
48 {
49         //PL GPIOs
50         XGpio pmoda_o, pmodb_i;
51
52         //PS GPIOs
53         XGpioPs_Config *PSConfigPtr;
54
55         int readVal, psVal;
56         int Status;
57
58         xil_printf("\r\n--- Start the Program ---\r\n");
59
60         /*
61          *  ======================
62          *  PL GPIO Initialization
63          *  ======================
64          *
65     */
66         Status = <???>(&pmoda_o, XPAR_PMOD_A_O_DEVICE_ID);
67
68
69         if (Status != XST_SUCCESS) {
70                 xil_printf("Gpio PMODA Initialization Failed\r\n");
71                 return XST_FAILURE;
72         }
73
74
75         Status = XGpio_Initialize(&pmodb_i, <???>);
76
77         if (Status != XST_SUCCESS) {
78                 xil_printf("Gpio PMODB Initialization Failed\r\n");
79                 return XST_FAILURE;
80         }
81
```

Software

C Application

- The code tests a loopback using PMOD connectors. Also, support is added to interface with MIO LED (MIO7).

- You have to complete the code by replacing **<???>** with the right instruction or value.

```
#include "xparameters.h"
#include "xgpio.h"
#include "xgpiops.h"
#include "xil_printf.h"
#include "sleep.h"

static XGpioPs psGpioInstancePtr;

#define EMIO_GPIO_BANK 2 // EMIO Bank
#define MIO_GPIO_BANK 0  // GPIOs connected directly to PS
#define MIO_LED_PIN 7 // LED pin on MIO is directly connected to pin 7 (MIO7)


int main(void)
{
	//PL GPIOs
	XGpio pmoda_o, pmodb_i;

	//PS GPIOs
	XGpioPs_Config *PSConfigPtr;

	int readVal, psVal;
	int Status;

	xil_printf("\r\n--- Start the Program ---\r\n");

	/*
	 *  ======================
	 *  PL GPIO Initialization
	 *  ======================
	 *
    */
	Status = <???>(&pmoda_o, XPAR_PMOD_A_O_DEVICE_ID);


	if (Status != XST_SUCCESS) {
		xil_printf("Gpio PMODA Initialization Failed\r\n");
		return XST_FAILURE;
	}


	Status = XGpio_Initialize(&pmodb_i, <???>);

	if (Status != XST_SUCCESS) {
		xil_printf("Gpio PMODB Initialization Failed\r\n");
		return XST_FAILURE;
	}
```

Software

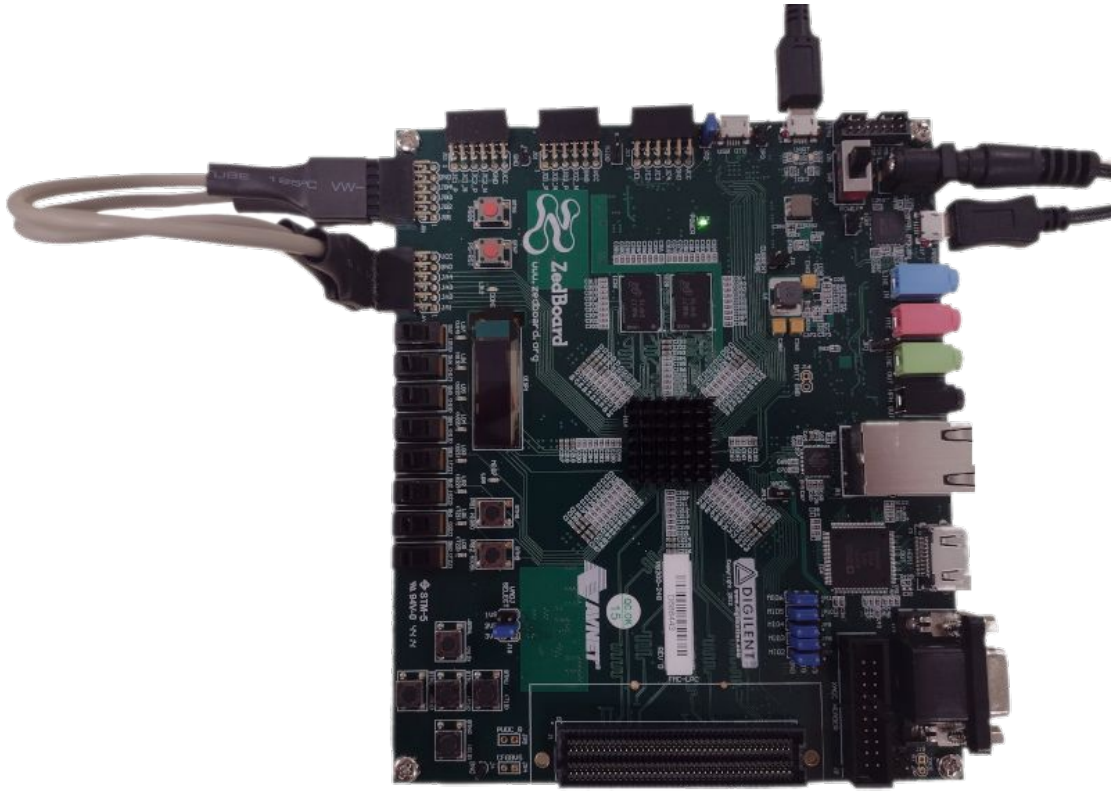
C Application

- The code tests a loopback using PMOD connectors. Also, support is added to interface with MIO LED (MIO7).
- You have to complete the code by replacing **<???>** with the right instruction or value.
- The output values will be displayed on the Host PC's monitor through serial communication between the PC-Host and the ZedBoard.

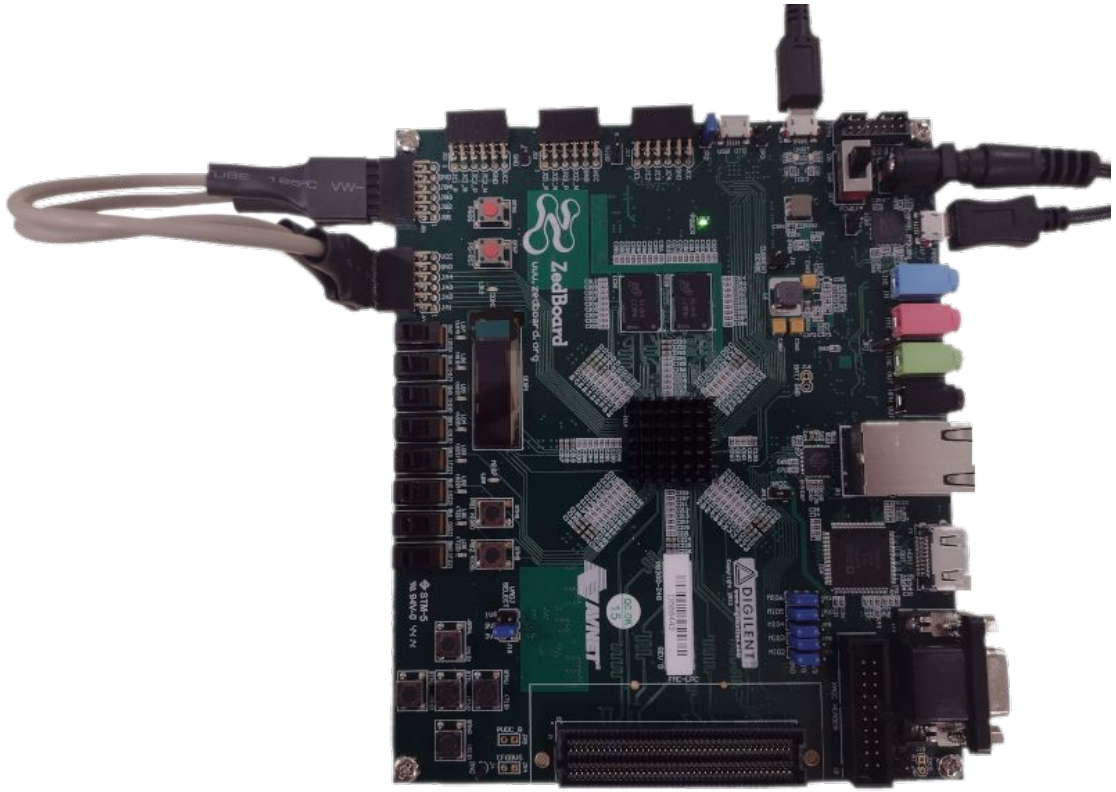
```
34 #include "xparameters.h"
35 #include "xgpio.h"
36 #include "xgpiops.h"
37 #include "xil_printf.h"
38 #include "sleep.h"
39
40 static XGpioPs psGpioInstancePtr;
41
42 #define EMIO_GPIO_BANK 2 // EMIO Bank
43 #define MIO_GPIO_BANK 0  // GPIOs connected directly to PS
44 #define MIO_LED_PIN 7 // LED pin on MIO is directly connected to pin 7 (MIO7)
45
46
47 int main(void)
48 {
49         //PL GPIOs
50         XGpio pmoda_o, pmodb_i;
51
52         //PS GPIOs
53         XGpioPs_Config *PSConfigPtr;
54
55         int readVal, psVal;
56         int Status;
57
58         xil_printf("\r\n--- Start the Program ---\r\n");
59
60         /*
61          *  ======================
62          *  PL GPIO Initialization
63          *  ======================
64          *
65      */
66         Status = <???>(&pmoda_o, XPAR_PMOD_A_O_DEVICE_ID);
67
68
69         if (Status != XST_SUCCESS) {
70                 xil_printf("Gpio PMODA Initialization Failed\r\n");
71                 return XST_FAILURE;
72         }
73
74
75         Status = XGpio_Initialize(&pmodb_i, <???>);
76
77         if (Status != XST_SUCCESS) {
78                 xil_printf("Gpio PMODB Initialization Failed\r\n");
79                 return XST_FAILURE;
80         }
81
```

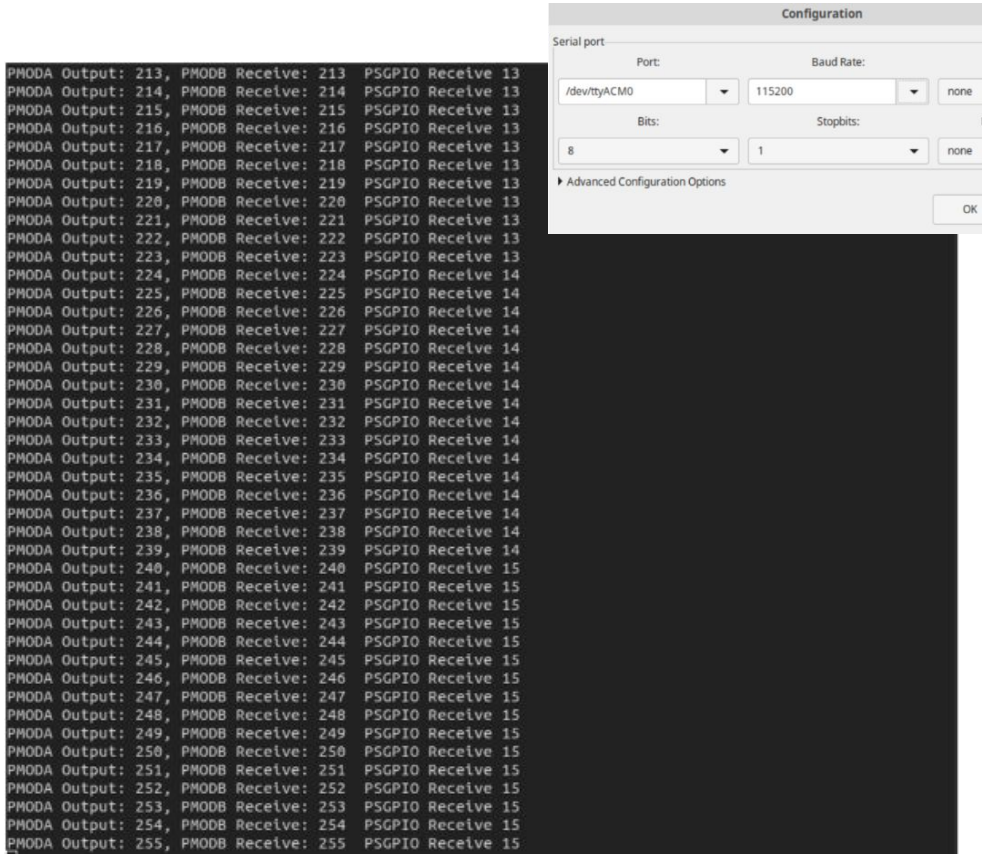
Test the complete design on the ZedBoard platform
to verify the implementation (Vitis and serial port).

Test on hardware



Test on hardware





The Abdus Salam
International Centre
for Theoretical Physics

IAEA

Joint ICTP-IAEA School on
Systems-on-Chip based on
FPGA for Scientific Instrumentation
and Reconfigurable Computing

Thank you!

Trieste - Italy
2023