



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).

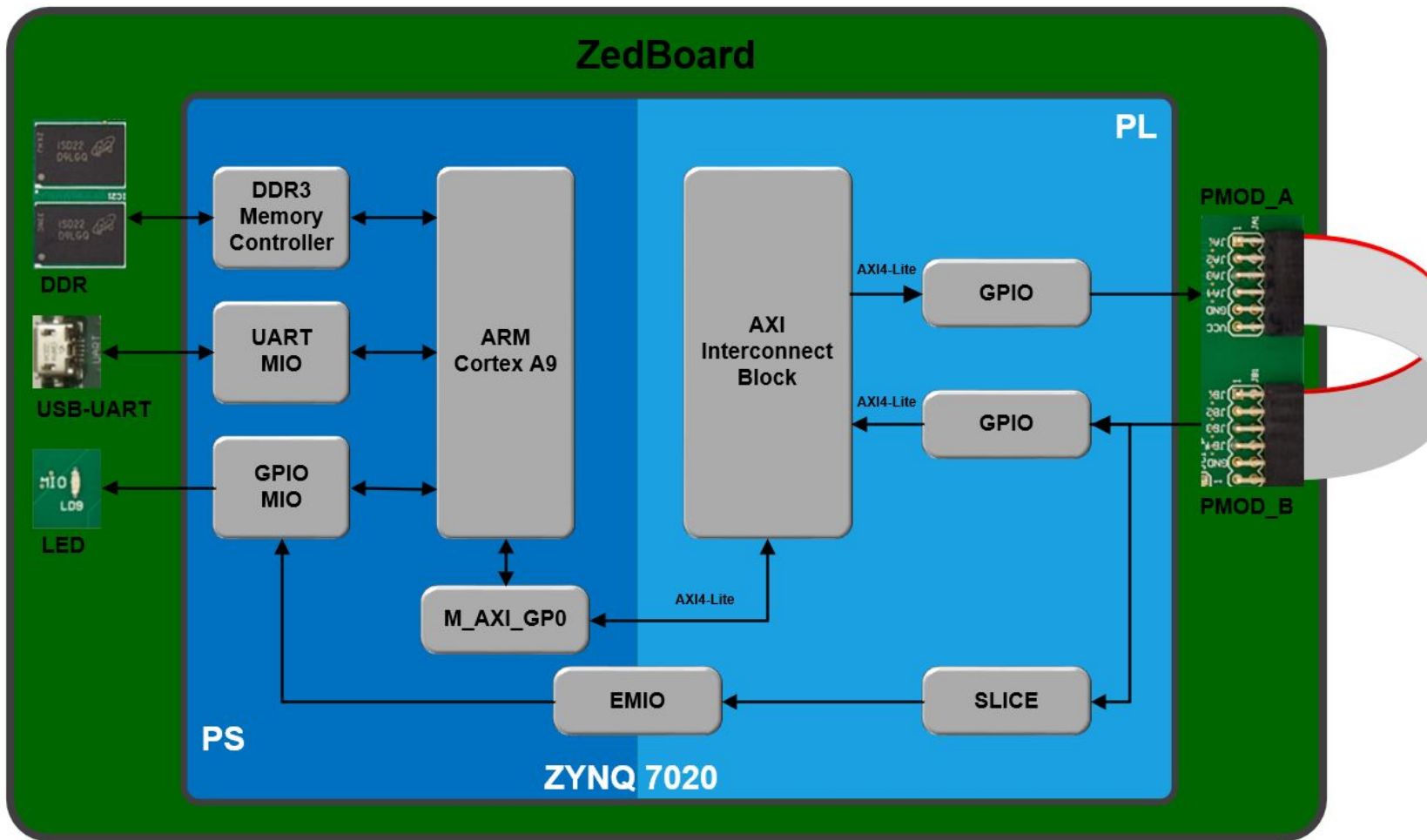
Lab 1: Objectives

- 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

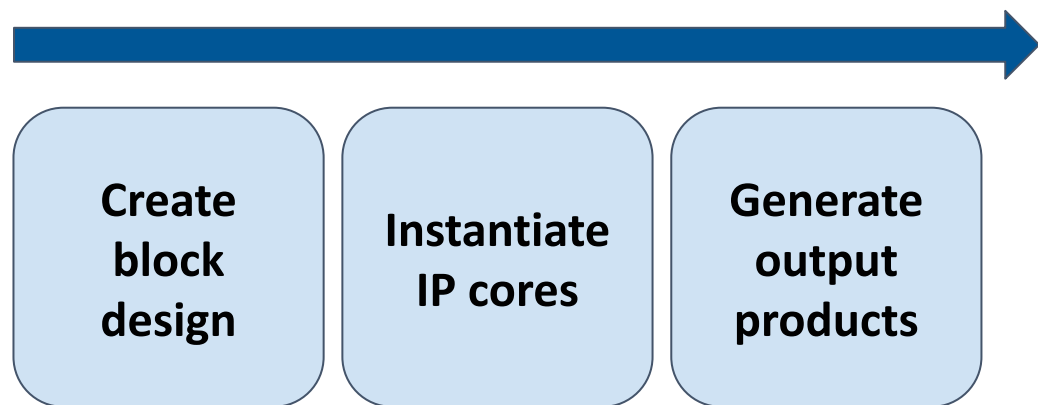


**Create
block
design**

**Instantiate
IP cores**

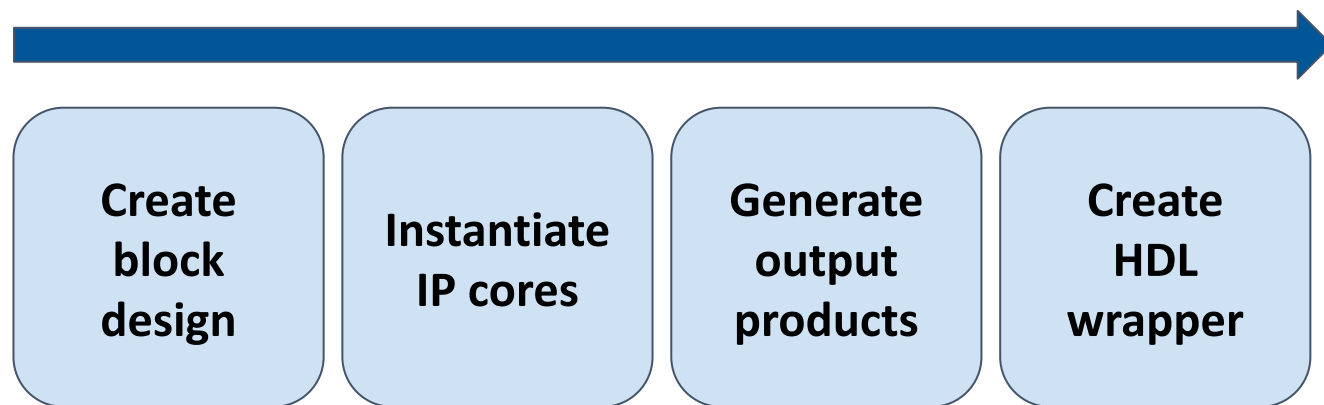
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



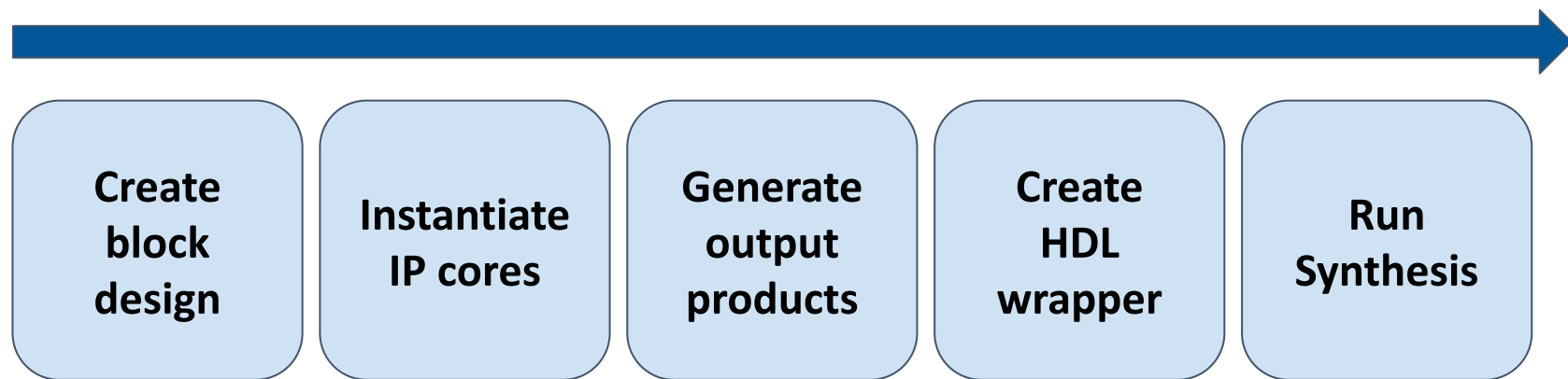
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



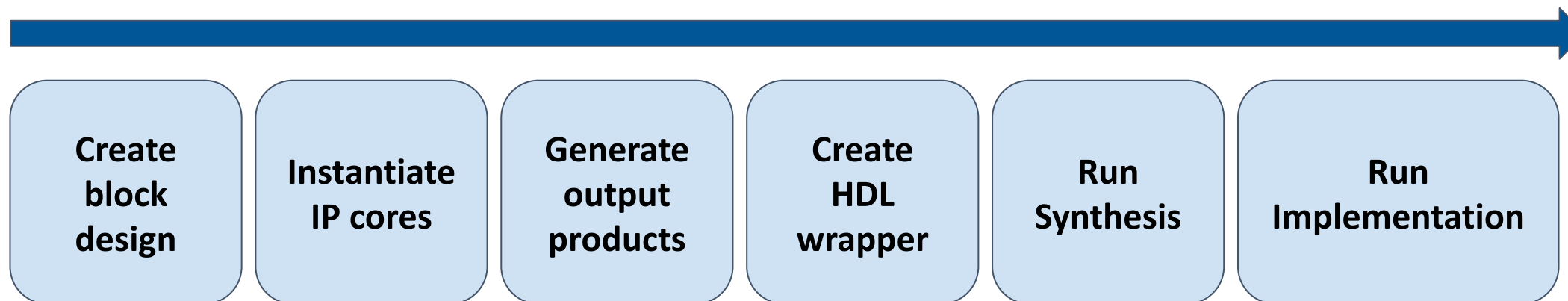
SoC-FPGA design flow for Lab 1

Vivado - Hardware creation



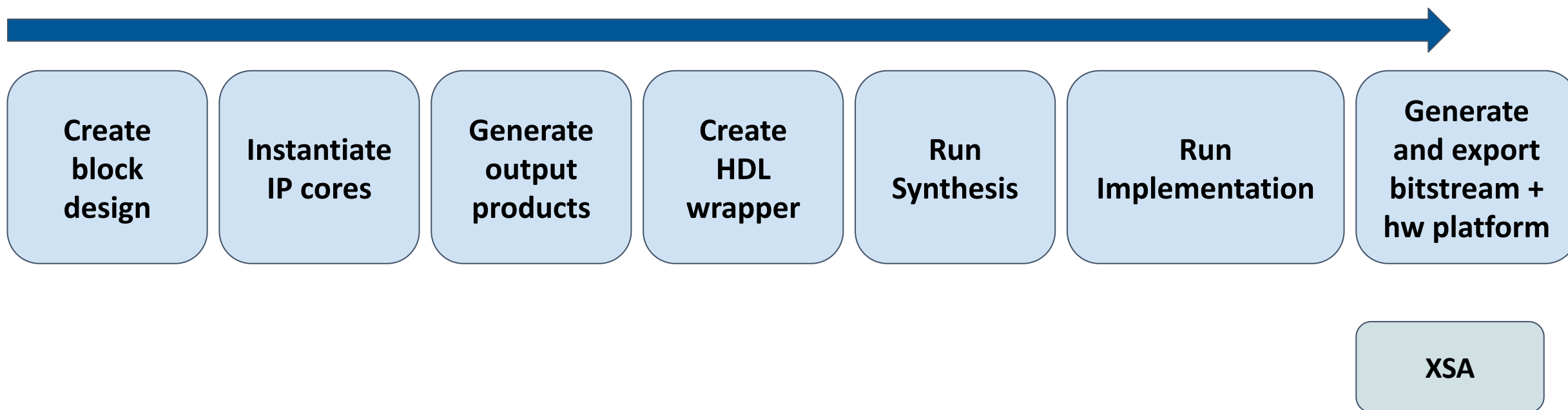
SoC-FPGA design flow for Lab 1

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



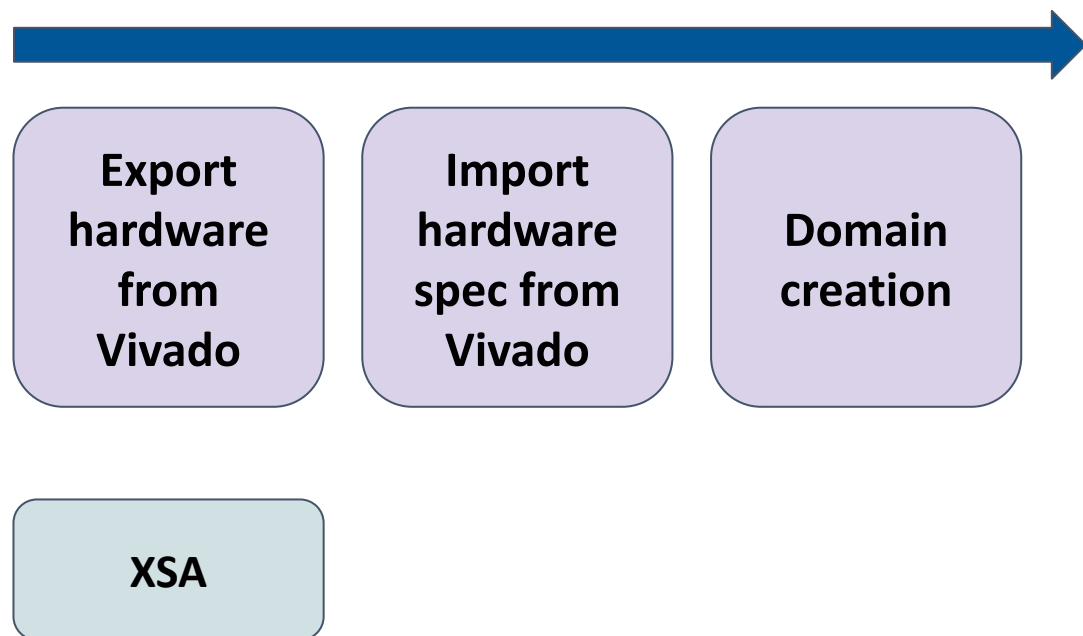
**Export
hardware
from
Vivado**

**Import
hardware
spec from
Vivado**

XSA

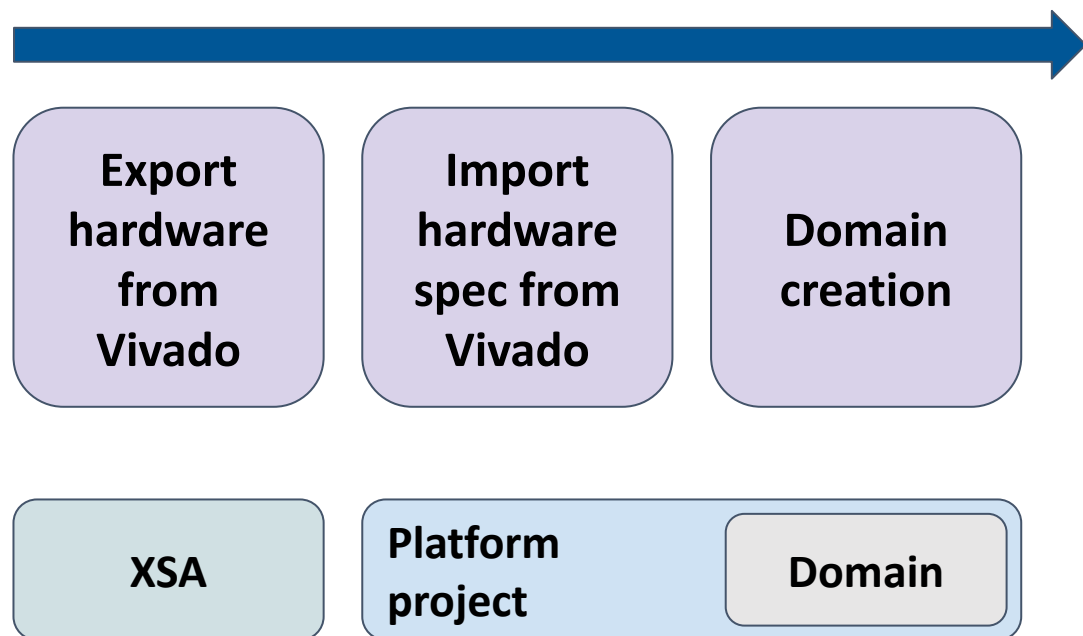
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



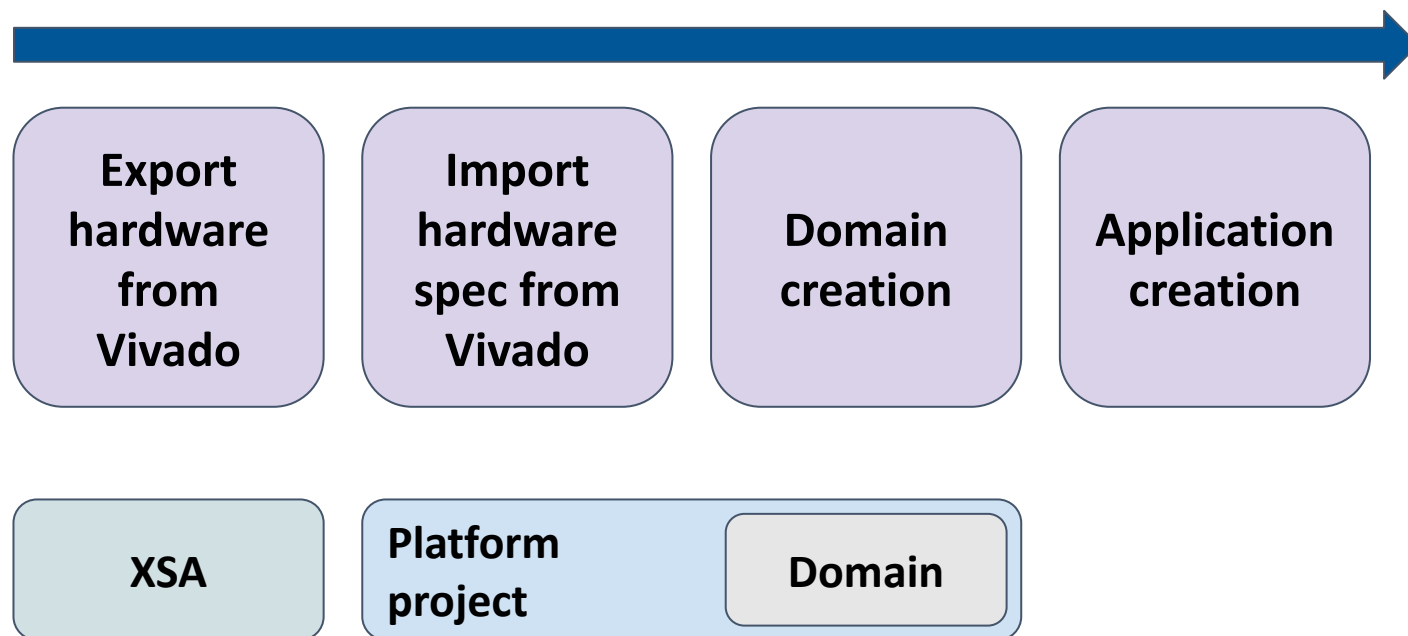
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



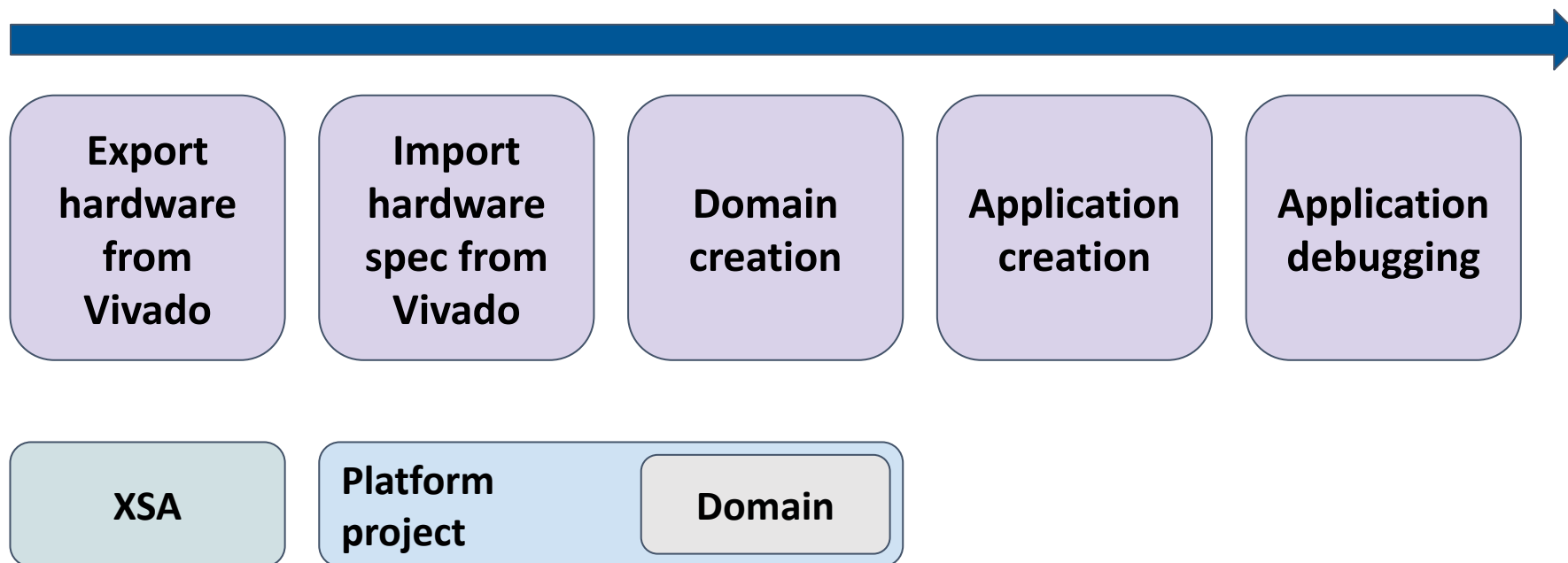
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



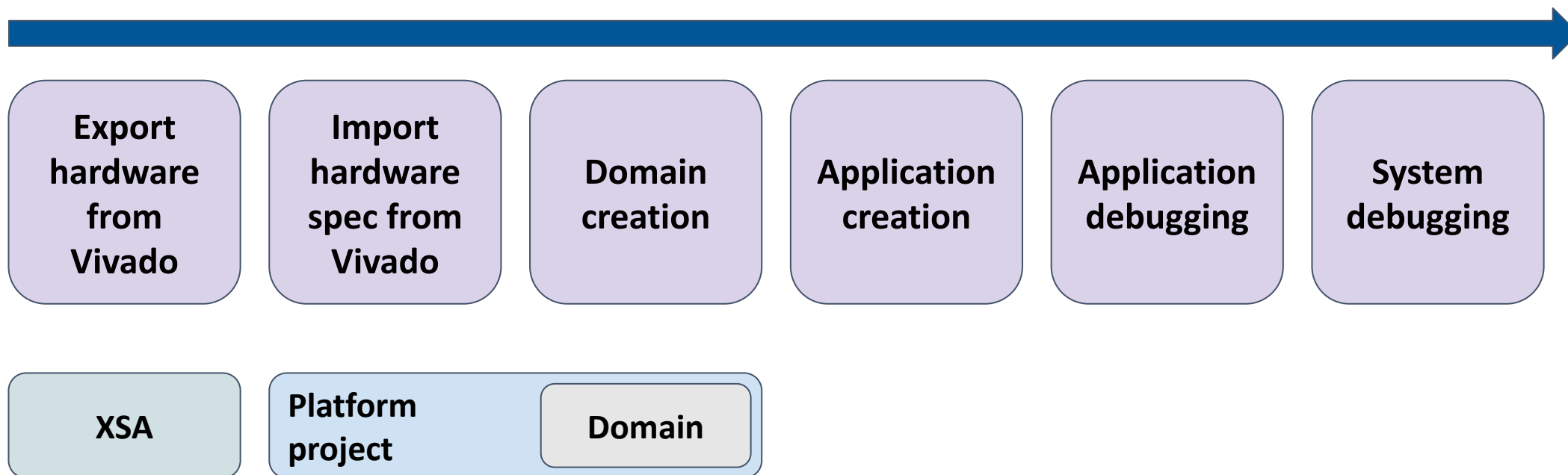
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



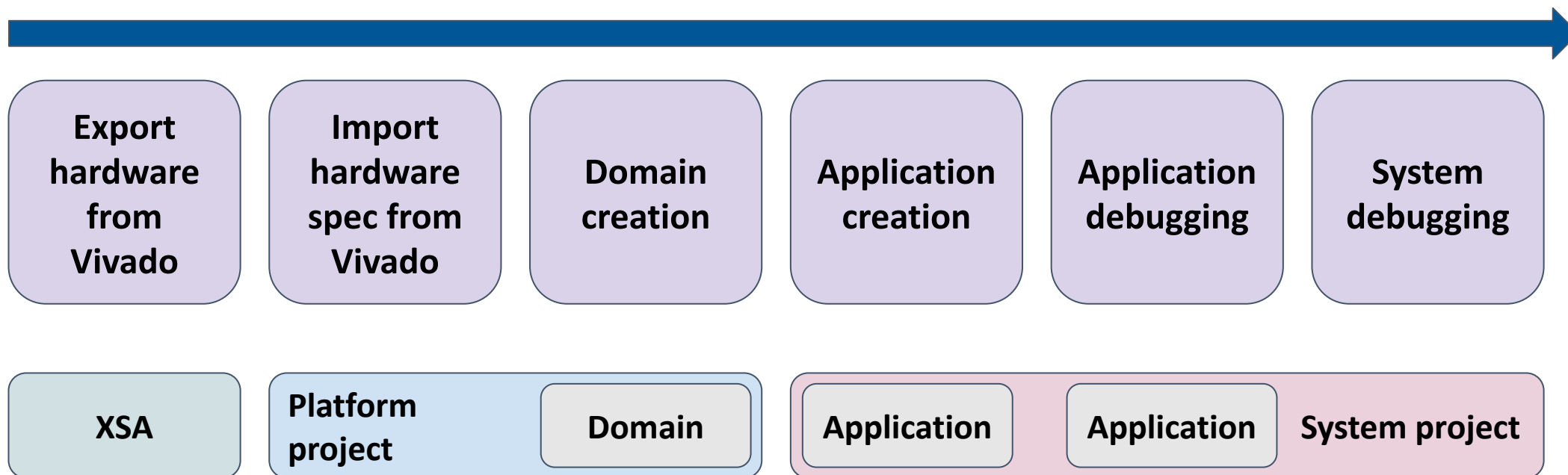
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



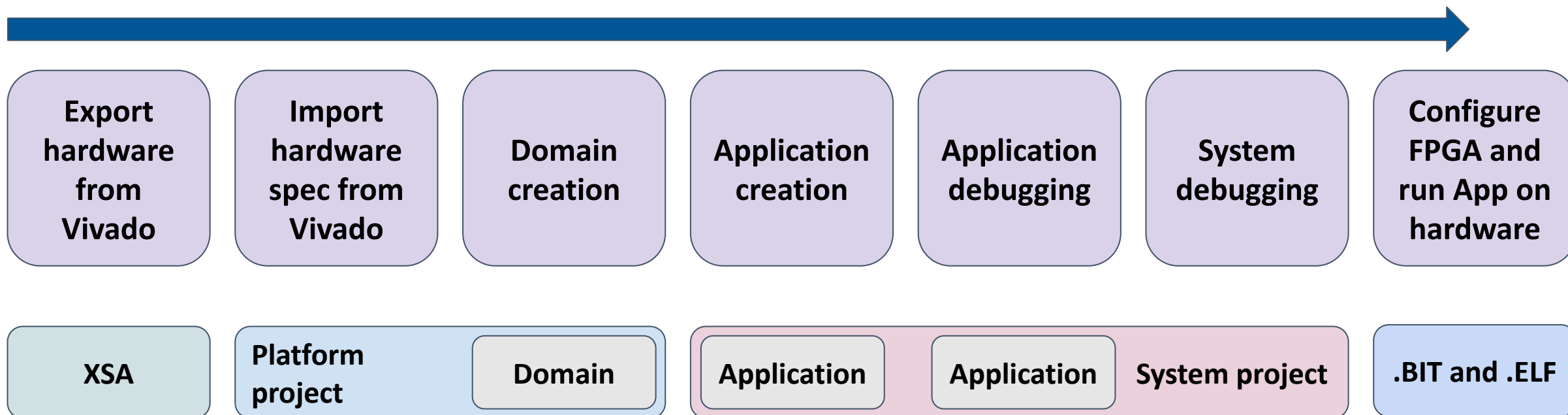
SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow



SoC-FPGA design flow for Lab 1

Vitis Software Development Workflow





The Abdus Salam

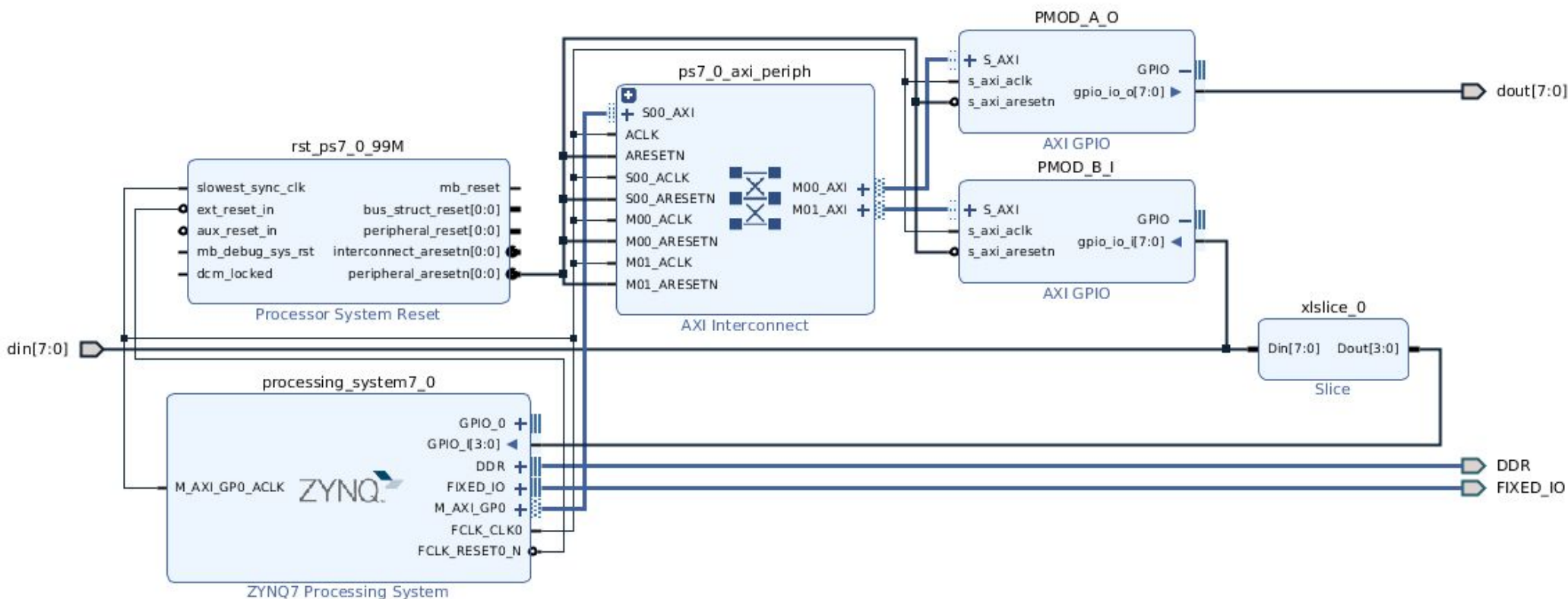
International Centre
for Theoretical Physics



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





The Abdus Salam

International Centre
for Theoretical Physics



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
82

```

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.

```

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
82

```

Software

C Application

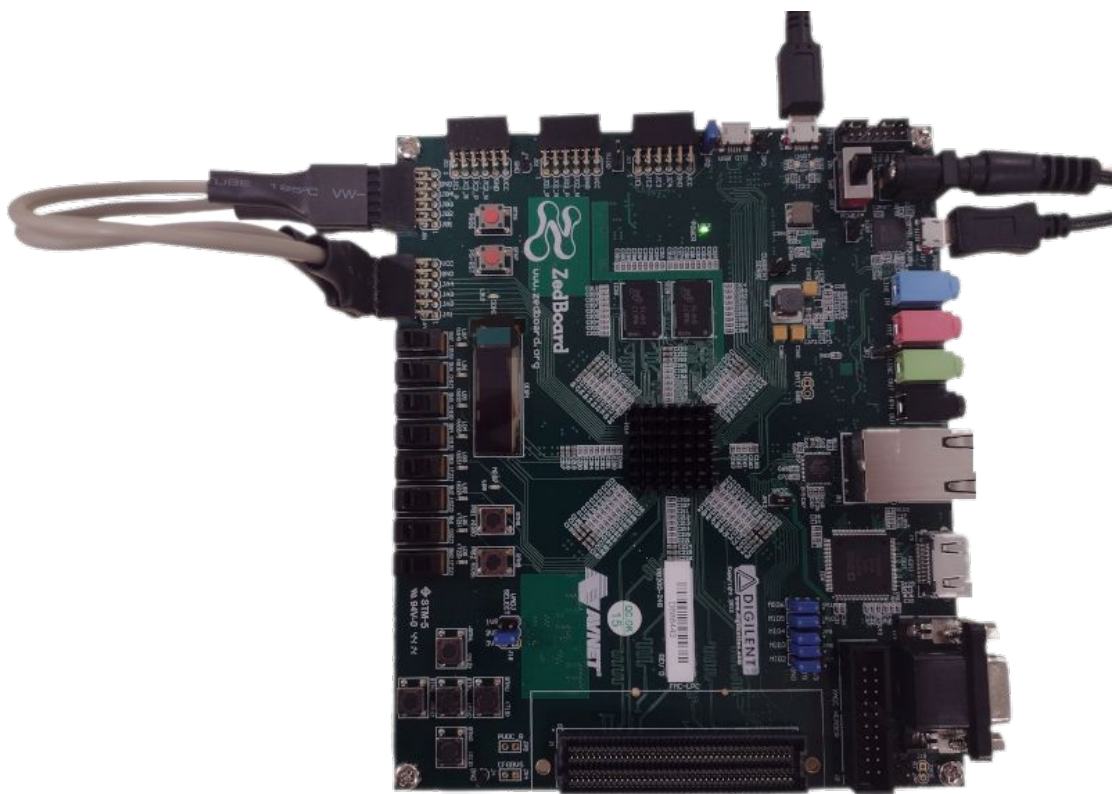
- The code tests a loopback using PMOD connectors. Also, support is added to interface with MIO LED (MIO7).
- You have to complete the code by replacing `<???` with the right instruction or value.
- The output values will be displayed on the Host PC's monitor through serial communication between the 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
82
```

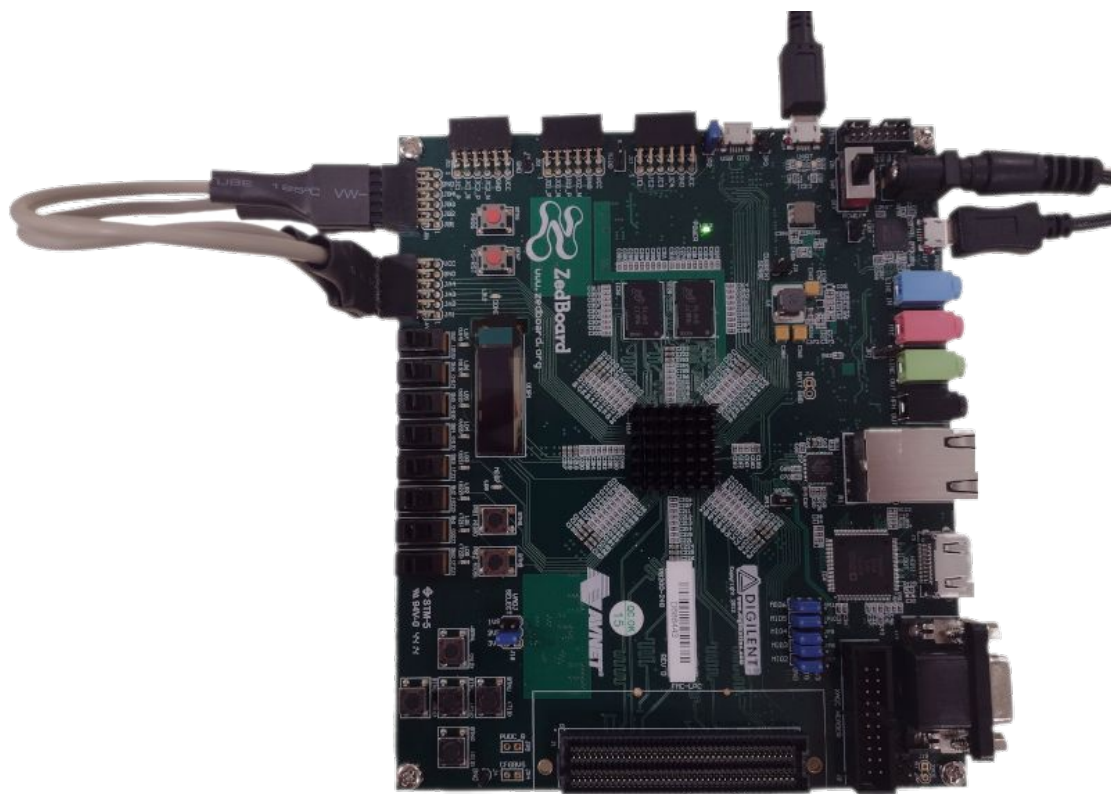


**Test the complete design on the ZedBoard platform
to verify the implementation (Vitis and serial port).**

Test on hardware

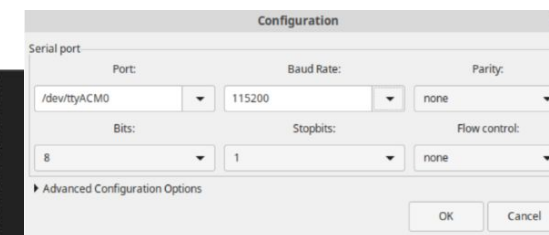


Test on hardware



```

PHODA Output: 213, PMODB Receive: 213 PSGPIO Receive 13
PHODA Output: 214, PMODB Receive: 214 PSGPIO Receive 13
PHODA Output: 215, PMODB Receive: 215 PSGPIO Receive 13
PHODA Output: 216, PMODB Receive: 216 PSGPIO Receive 13
PHODA Output: 217, PMODB Receive: 217 PSGPIO Receive 13
PHODA Output: 218, PMODB Receive: 218 PSGPIO Receive 13
PHODA Output: 219, PMODB Receive: 219 PSGPIO Receive 13
PHODA Output: 220, PMODB Receive: 220 PSGPIO Receive 13
PHODA Output: 221, PMODB Receive: 221 PSGPIO Receive 13
PHODA Output: 222, PMODB Receive: 222 PSGPIO Receive 13
PHODA Output: 223, PMODB Receive: 223 PSGPIO Receive 13
PHODA Output: 224, PMODB Receive: 224 PSGPIO Receive 14
PHODA Output: 225, PMODB Receive: 225 PSGPIO Receive 14
PHODA Output: 226, PMODB Receive: 226 PSGPIO Receive 14
PHODA Output: 227, PMODB Receive: 227 PSGPIO Receive 14
PHODA Output: 228, PMODB Receive: 228 PSGPIO Receive 14
PHODA Output: 229, PMODB Receive: 229 PSGPIO Receive 14
PHODA Output: 230, PMODB Receive: 230 PSGPIO Receive 14
PHODA Output: 231, PMODB Receive: 231 PSGPIO Receive 14
PHODA Output: 232, PMODB Receive: 232 PSGPIO Receive 14
PHODA Output: 233, PMODB Receive: 233 PSGPIO Receive 14
PHODA Output: 234, PMODB Receive: 234 PSGPIO Receive 14
PHODA Output: 235, PMODB Receive: 235 PSGPIO Receive 14
PHODA Output: 236, PMODB Receive: 236 PSGPIO Receive 14
PHODA Output: 237, PMODB Receive: 237 PSGPIO Receive 14
PHODA Output: 238, PMODB Receive: 238 PSGPIO Receive 14
PHODA Output: 239, PMODB Receive: 239 PSGPIO Receive 14
PHODA Output: 240, PMODB Receive: 240 PSGPIO Receive 15
PHODA Output: 241, PMODB Receive: 241 PSGPIO Receive 15
PHODA Output: 242, PMODB Receive: 242 PSGPIO Receive 15
PHODA Output: 243, PMODB Receive: 243 PSGPIO Receive 15
PHODA Output: 244, PMODB Receive: 244 PSGPIO Receive 15
PHODA Output: 245, PMODB Receive: 245 PSGPIO Receive 15
PHODA Output: 246, PMODB Receive: 246 PSGPIO Receive 15
PHODA Output: 247, PMODB Receive: 247 PSGPIO Receive 15
PHODA Output: 248, PMODB Receive: 248 PSGPIO Receive 15
PHODA Output: 249, PMODB Receive: 249 PSGPIO Receive 15
PHODA Output: 250, PMODB Receive: 250 PSGPIO Receive 15
PHODA Output: 251, PMODB Receive: 251 PSGPIO Receive 15
PHODA Output: 252, PMODB Receive: 252 PSGPIO Receive 15
PHODA Output: 253, PMODB Receive: 253 PSGPIO Receive 15
PHODA Output: 254, PMODB Receive: 254 PSGPIO Receive 15
PHODA Output: 255, PMODB Receive: 255 PSGPIO Receive 15
    
```





The Abdus Salam
**International Centre
for Theoretical Physics**



IAEA

Thank you!

Trieste - Italy
2023

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

