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

School on FPGA-based SoC
Lab 3: UDMA

Agustin Silva

Laboratory Objectives

- Import HLS IP block
- Understand the usage of the UDMA
- Transfer data between the PC and the FPGA
- Interact within JupyterLab with custom hardware
- Every step is described in Laboratory 3 UDMA

Laboratory Objectives

- Import HLS IP block
- Understand the usage of the UDMA
- Transfer data between the PC and the FPGA
- Interact within JupyterLab with custom hardware
- Every step is described in Laboratory 3 UDMA

Laboratory Objectives

- Import HLS IP block
- Understand the usage of the UDMA
- Transfer data between the PC and the FPGA
- Interact within JupyterLab with custom hardware
- Every step is described in Laboratory 3 UDMA

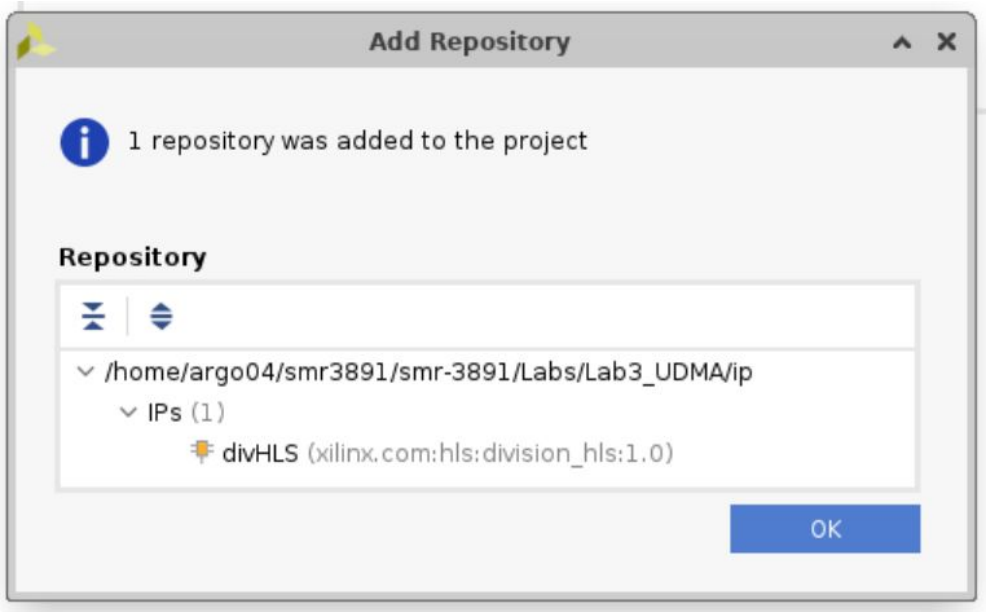
Laboratory Objectives

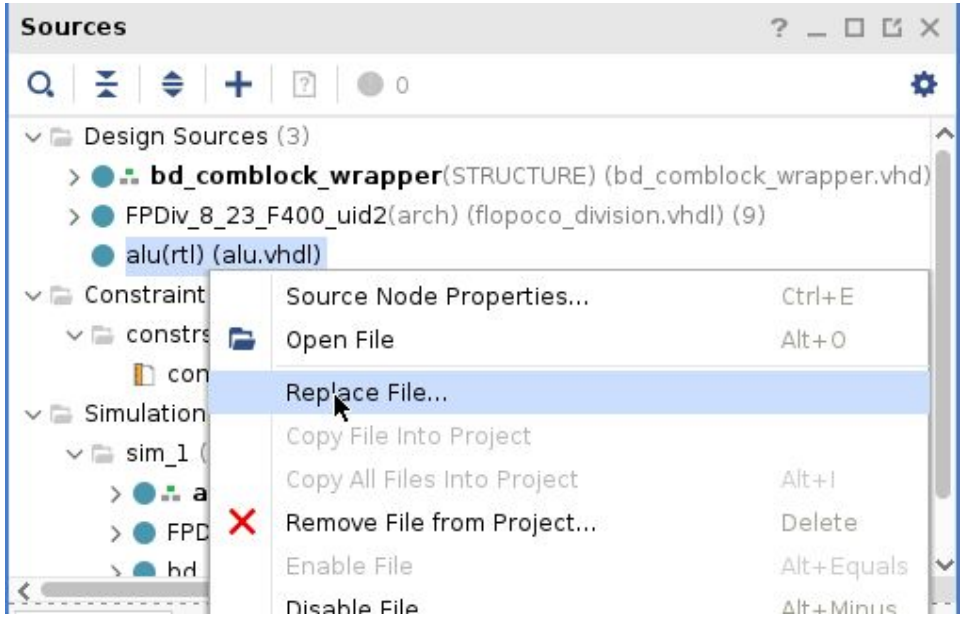
- Import HLS IP block
- Understand the usage of the UDMA
- Transfer data between the PC and the FPGA
- Interact within JupyterLab with custom hardware
- Every step is described in Laboratory 3 UDMA

Laboratory Objectives

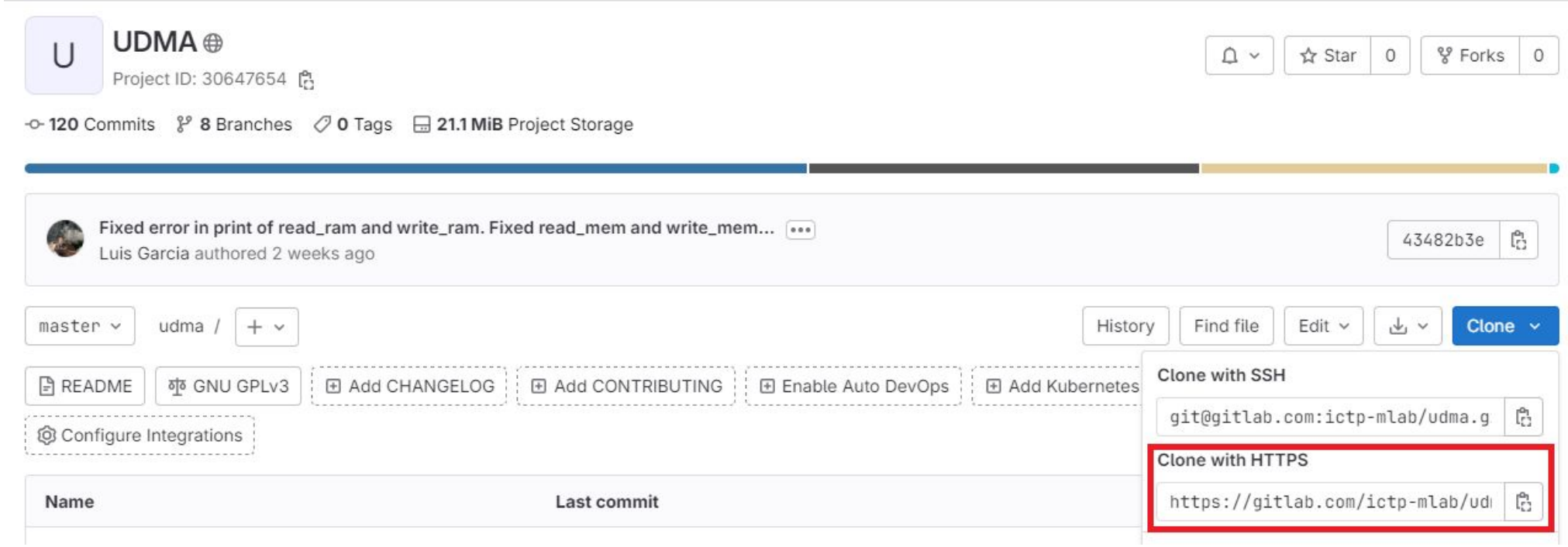
- Import HLS IP block
- Understand the usage of the UDMA
- Transfer data between the PC and the FPGA
- Interact within JupyterLab with custom hardware
- Every step is described in Laboratory 3 UDMA

Import HLS IP and update modules



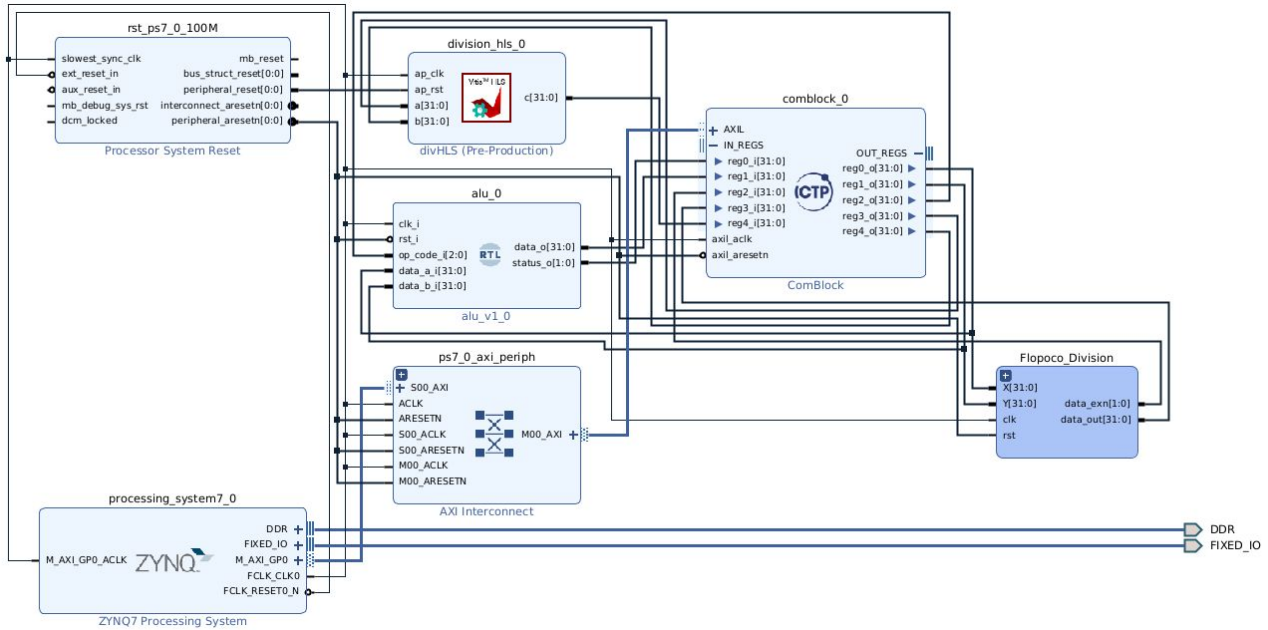


Install UDMA



Software/hardware co-design

- Vhdl same as always
- Python same as always

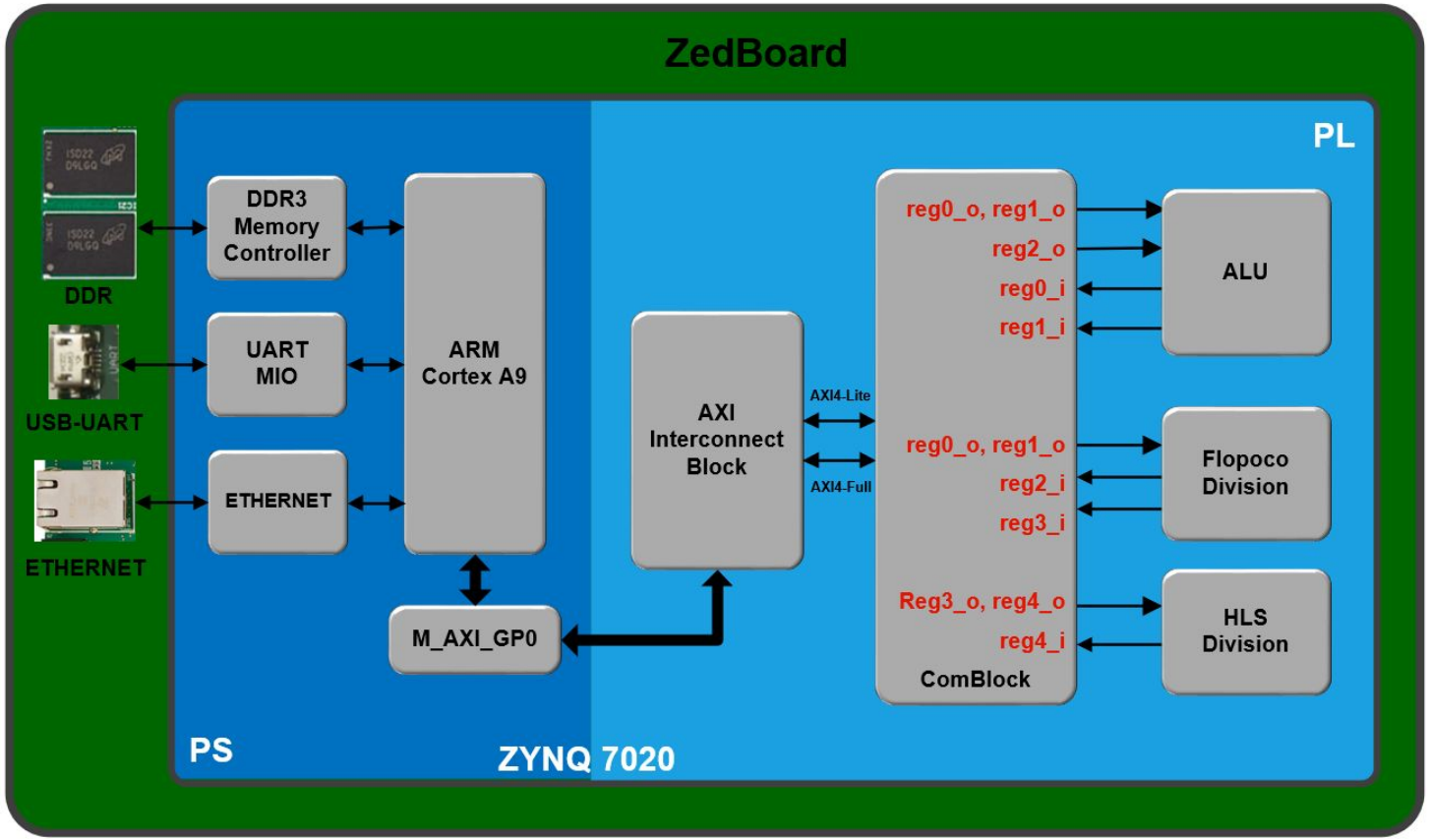


Software/hardware co-design

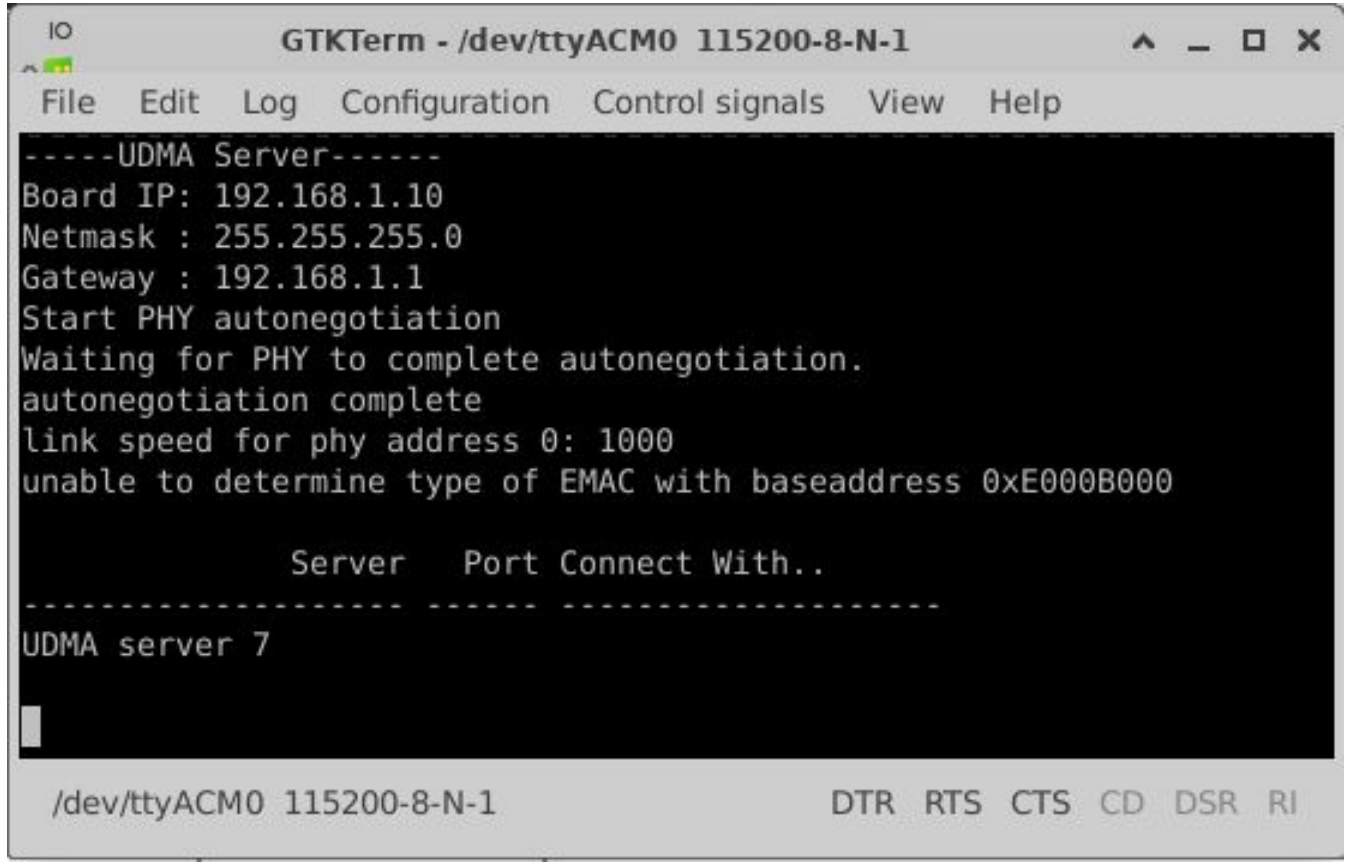
- Vhdl same as always
- Python same as always

```
import udma
import numpy as np
import matplotlib.pyplot as plt
from tqdm import trange
```

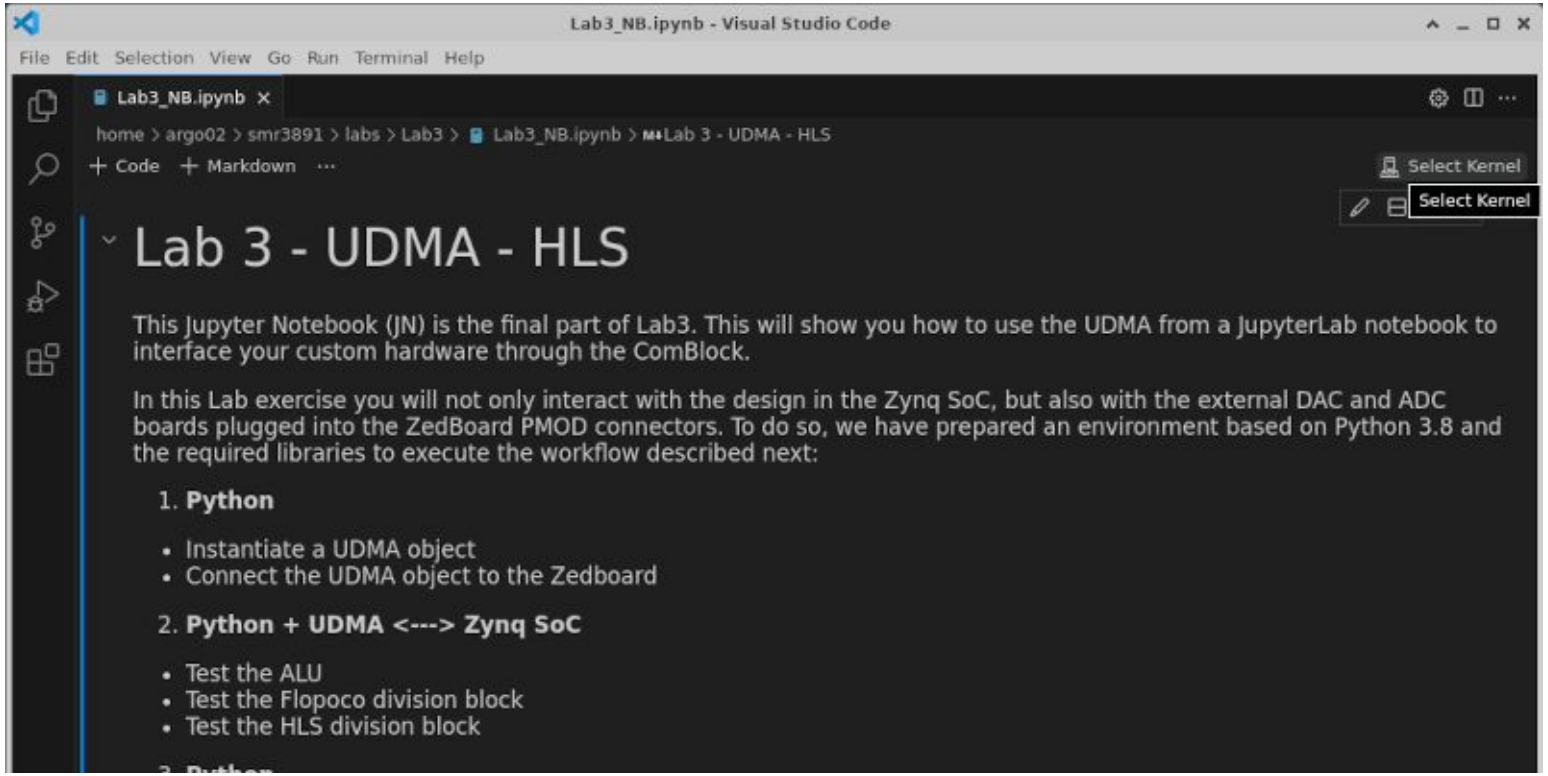
FreeRTOS + UDMA server in the PS



Make sure connection is established



Jupyter Notebook and UDMA



How to use it

- Import udma class to write and read the comblock
- Interact with registers, fifos and RAM with python

```
cb = udma.UDMA_CLASS('192.168.1.10', 7)
cb.connect()
cb.log(0)
```

How to use it

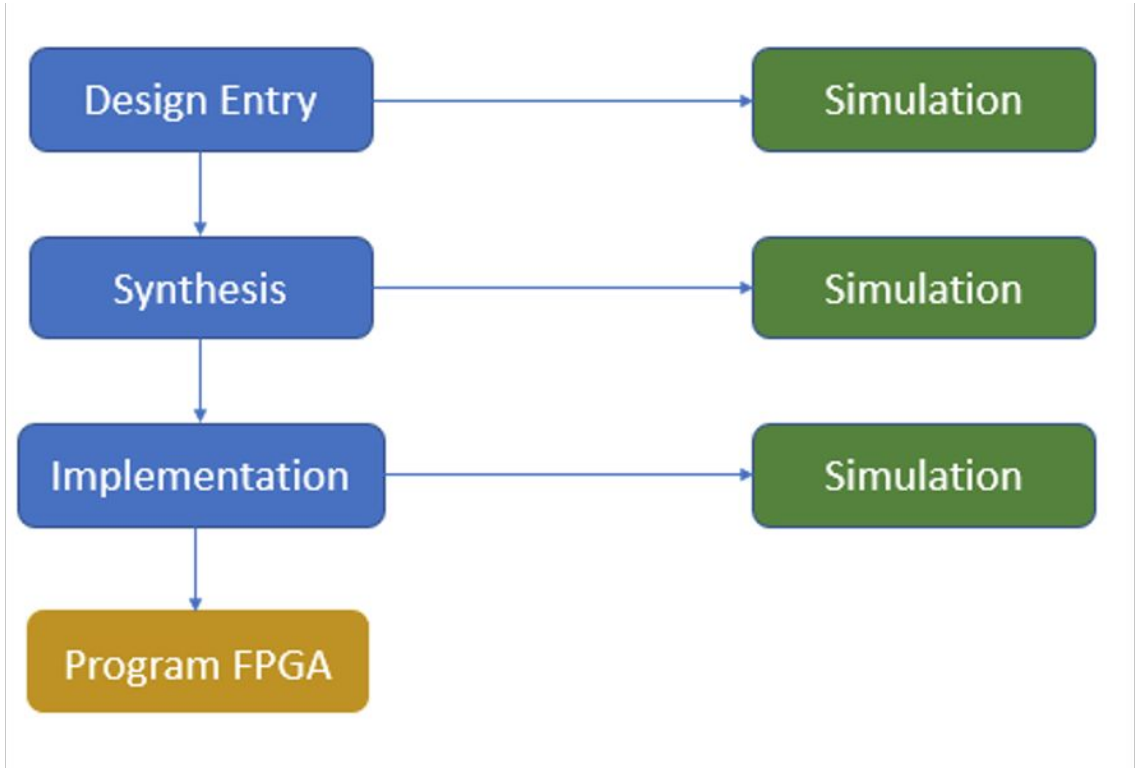
- Import udma class to write and read the comblock
- Interact with registers, fifos and RAM with python

```
for i in trange(100):
    x = np.random.randint(0,32767)
    cb.write_fifo(1, [x])
    #print(x)
    while (cb.read_reg(1)[1][0] & 1 == 0):
        pass
```

UDMA Methods

```
13 UDMA class
14     Attributes:
15         ip: str
16         port: int
17         s: socket
18     Methods
19         set_ip(ip)                          / get_ip()
20         set_port(port)                      / get_port()
21         connect()                           / disconnect()
22         read_reg(reg)                       / write_reg(reg, data)
23         read_fifo(N)                        / write_fifo(N, data)
24         read_ram(addr, length, inc)         / write_ram(addr, length, inc, data)
25         read_mem(addr, length, inc)         / write_mem(addr, length, inc, data)
26         select_comblock()
27         log()
```

Debugging, testbench and verification



Examples 1.

https://github.com/agustinsilva447/QMARL_FPGA/blob/main/QMARL_6/qmarl_udma.ipynb

Examples 2.

https://github.com/agustinsilva447/QMARL_FPGA/blob/main/final_2qagents/qdma.ipynb

Thank you!
Have fun =D