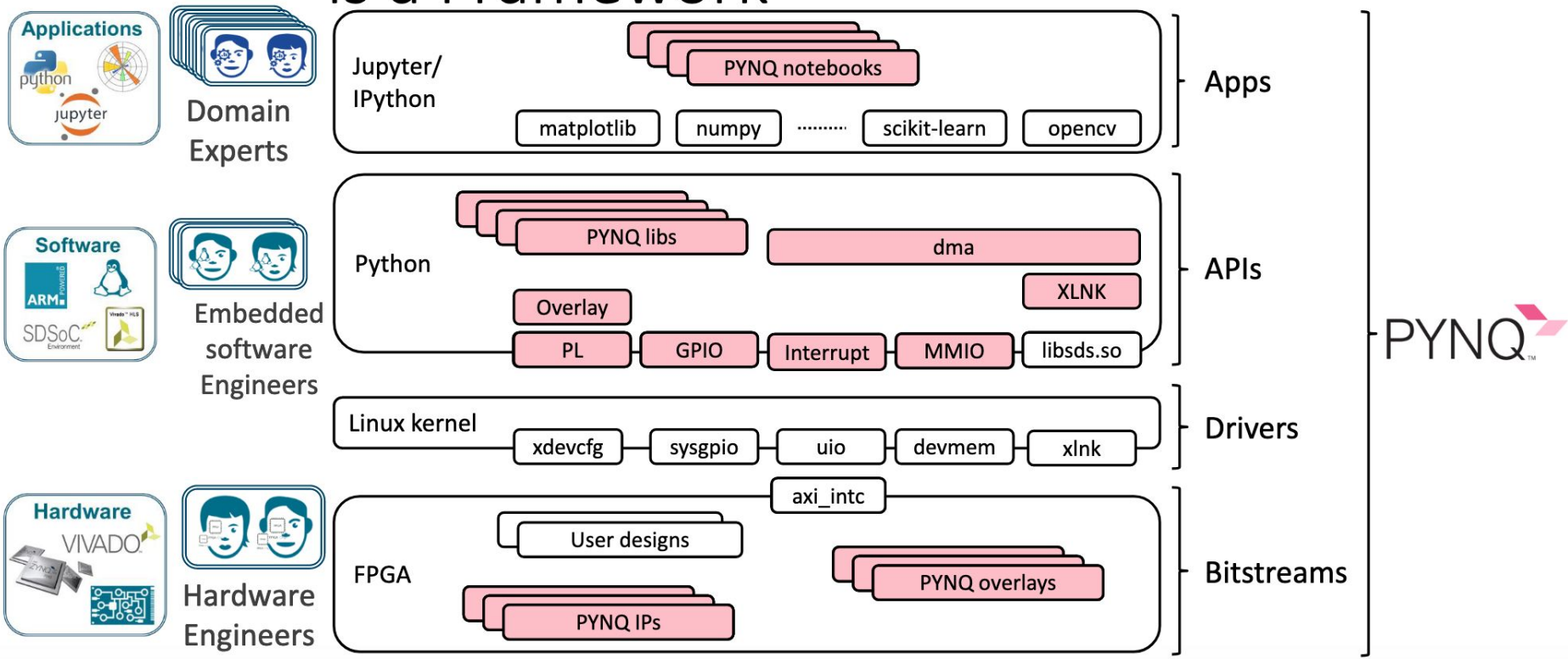
PYNQ Introduction

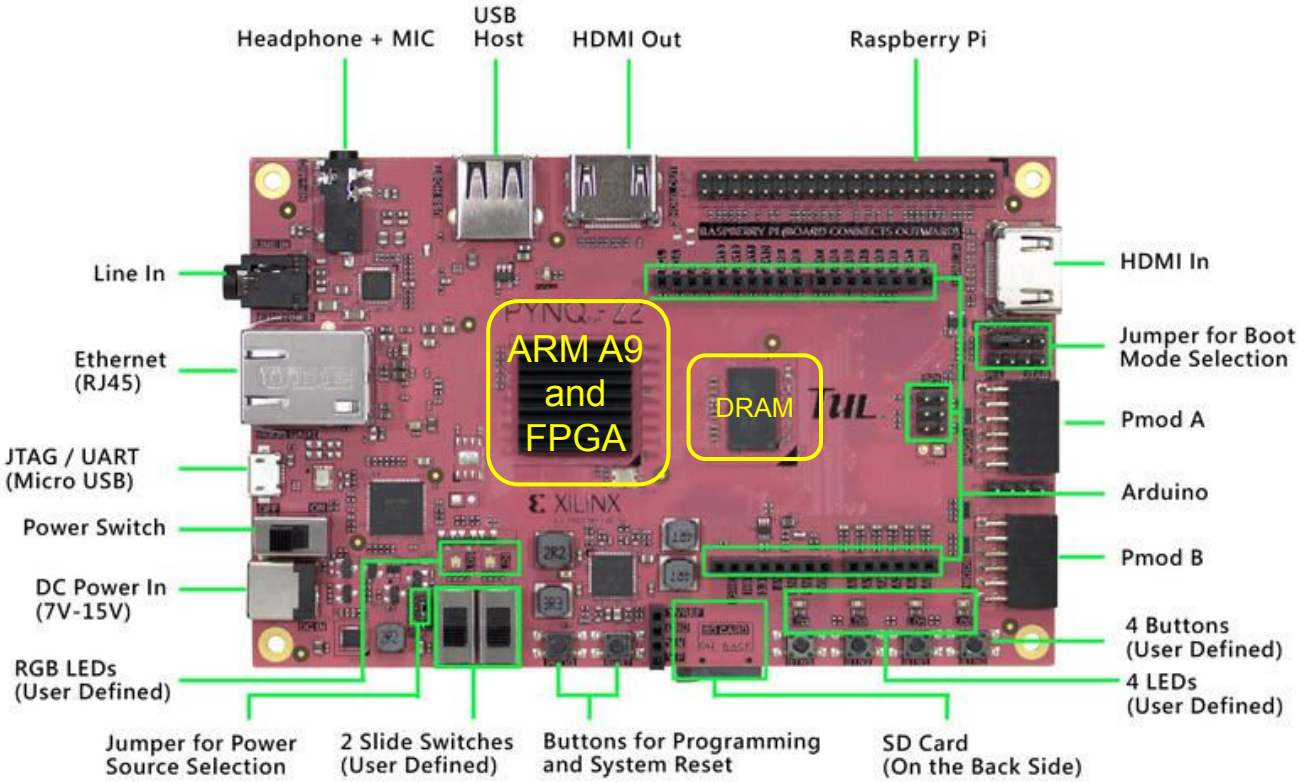
Speaker: JiaMing Lin

PYNQ

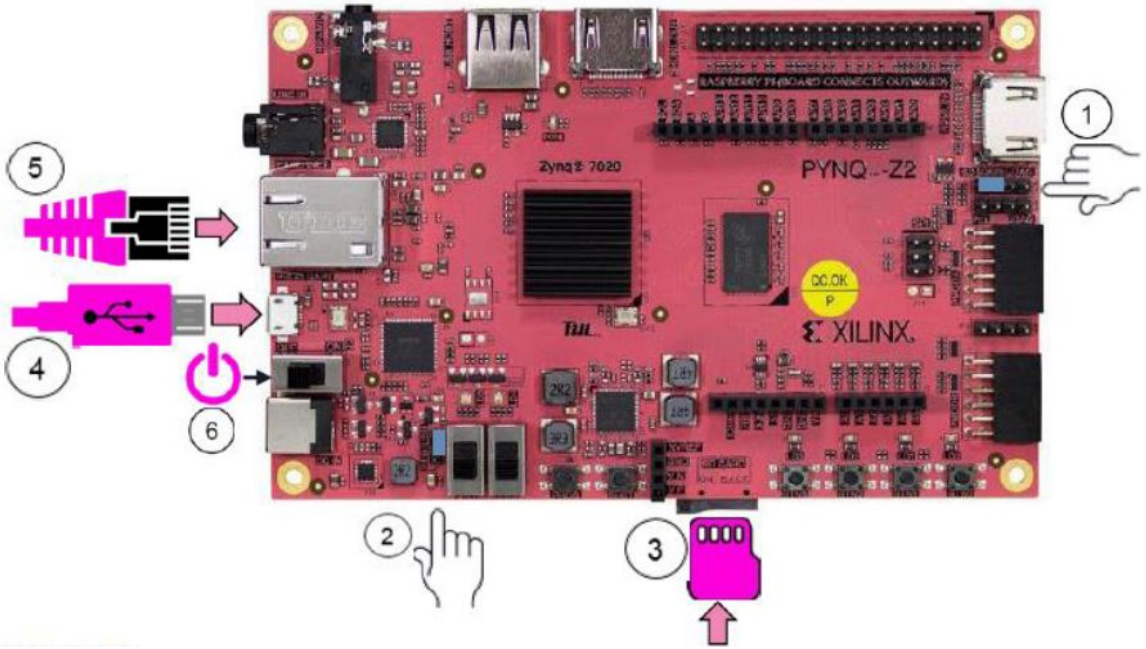
is a Framework



PYNQ-Z2


Headphone + MIC
USB Host
HDMI Out
Raspberry Pi
Line In
HDMI In
Ethernet (RJ45)
Jumper for Boot Mode Selection
ARM A9 and FPGA
DRAM
Pmod A
JTAG / UART (Micro USB)
Arduino
Power Switch
Pmod B
DC Power In (7V-15V)
4 Buttons (User Defined)
RGB LEDs (User Defined)
4 LEDs (User Defined)
Jumper for Power Source Selection
2 Slide Switches (User Defined)
Buttons for Programming and System Reset
SD Card (On the Back Side)

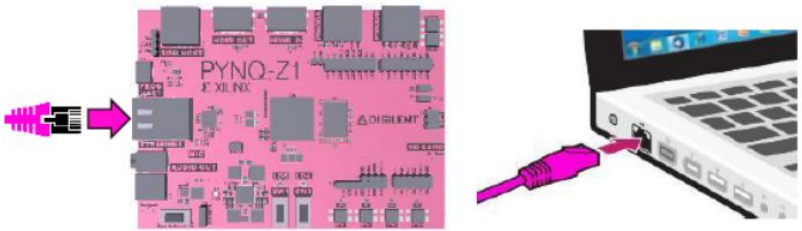
Connecting to the Board



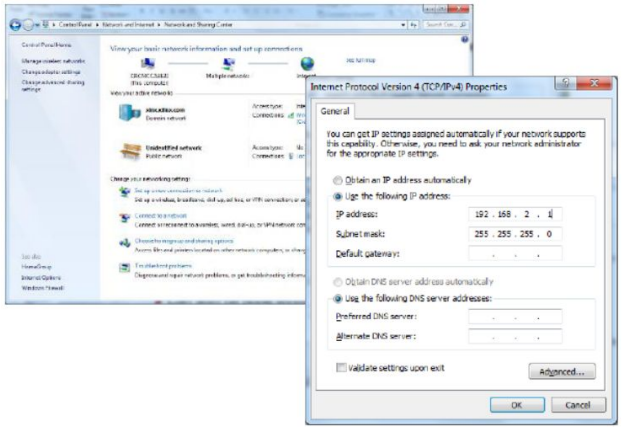
1. **Configure the board to boot from SD card**
2. **Set jumper to power from USB**
3. **Insert SD card**
4. **Connect USB cable**
5. **Connect Ethernet cable to PC**
6. **Power on**

Connecting to the Board: Direct Connection to PC

1. Connect board to Ethernet port on PC
2. Board IP address is 192.168.2.99 by default
3. Manually configure static IP address for PC
    - In the same network segment, e.g. 192.168.2.1



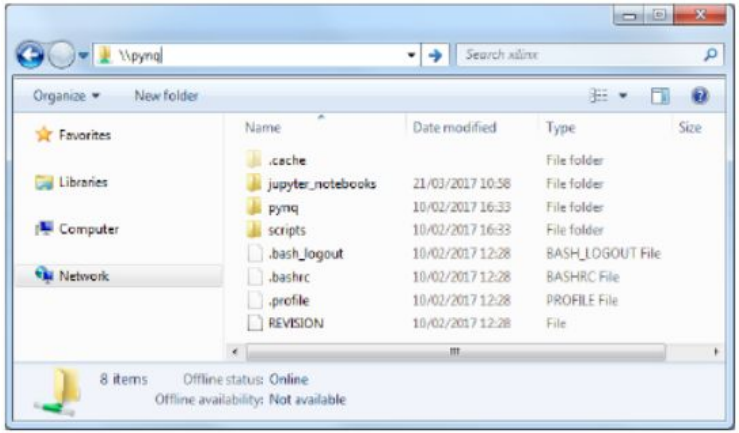
Connect board directly to PC



Samba Share

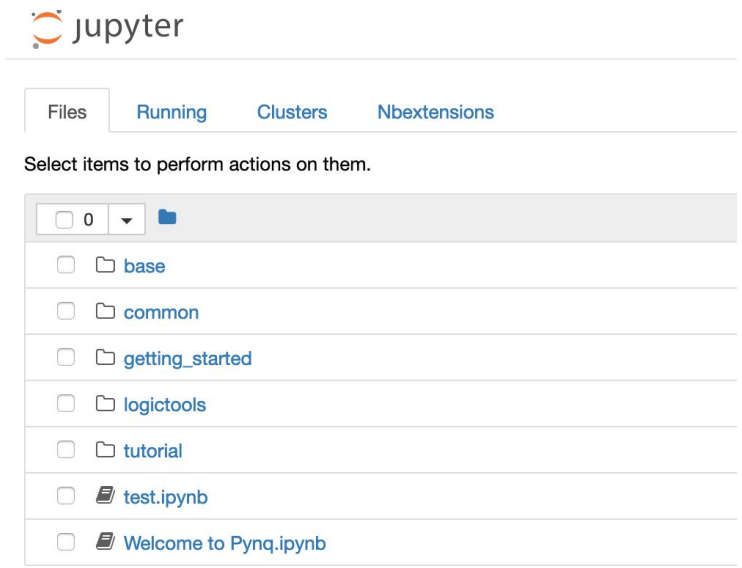
- Board can also be accessed as shared drive
  - Windows: \\192.168.2.99\xilinx
  - MAC OS: smb://192.168.2.99/xilinx
  - Linux: smb://192.168.2.99/xilinx
- Log-in
  - User name: xilinx
  - Password: xilinx
- Copy files easily between PC and Board

Window File Browser



Log-in to Jupyter Notebook

1. Open a browser(e.g. Chrome)
2. Browse to http://192.168.2.99:9090
3. Password: xilinx



- Documents:
  - https://pynq.readthedocs.io
- Support:
  - https://discuss.pynq.io/

Labs 0-1: Matrix Multiplication by using Python library, Numpy

- Create new IPython-Notebook

Labs 0-1: Matrix Multiplication by using Python library, Numpy

- Following Python script shows matrix multiplication by using Numpy

Initialize matrices

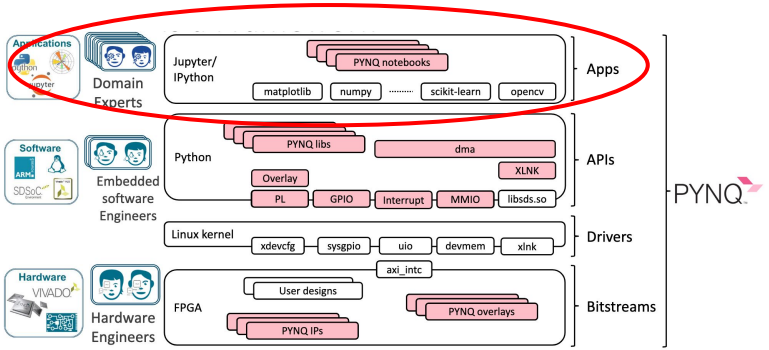
```
In [1]: import numpy as np

        A = np.random.randint(0,256, size = (32,64))
        B = np.random.randint(0,256, size = (64,96))
```

Matrix Multiplication using Python library, Numpy

```
In [2]: C = np.dot(A, B)
```

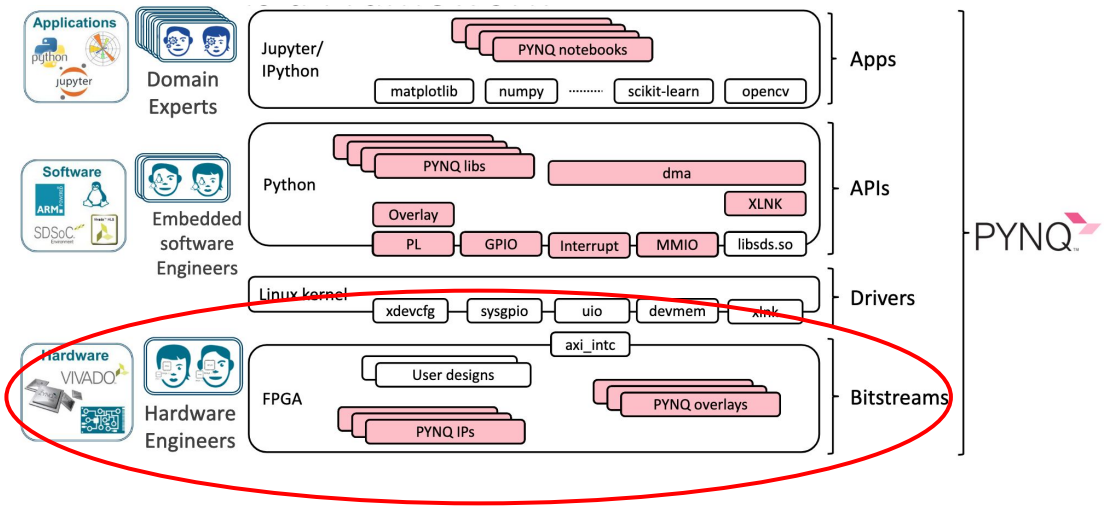
Performance Evaluation

```
In [3]: %timeit np.dot(A, B)

        1000 loops, best of 3: 1.87 ms per loop
```



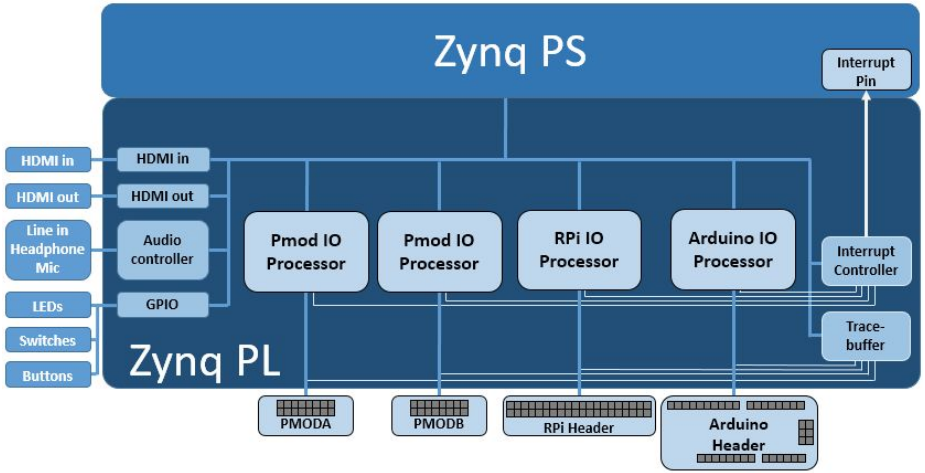
- **No FPGA, Only CPU involved.**

Introduction to Overlay



What is an Overlay?

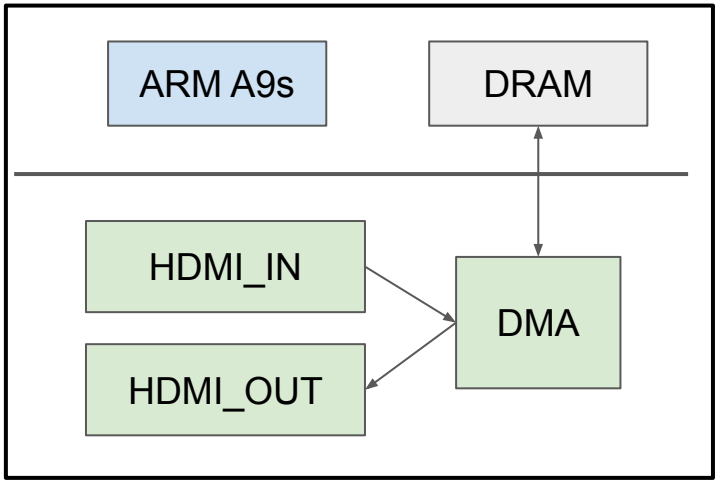
- Hardware library, consisted by one or several IPs
- Extends user application from Processing System(CPU) to Processing Logic(FPGA)
  - Speed-up or customize the hardware platform for a particular application.
- Example: PYNQ base overlay



Case Study: Video input/output

- Two Methods
  a. HDMI_IN and HDMI_OUT on Processing Logic
  b. OpenCV on ARM A9s
- Which one is better? In terms of throughput(FPS), energy consumption(Watt)

Method (a)



Method (b)

- USB Webcam on USB port of Zynq PS(CPU)
- Capture image and stored to DRAM
- OpenCV package is used

Source code reference:
https://gist.github.com/cathalmccabe/b0ab8917f748840f0d3959f7eabf0f82

Lab 0-2: LED Blinking and Button Access(1/2)

1. Create a new IPython-Notebook
2. Following script shows LED and button control via GPIO on Processing Logic

```
# import Python API
from pynq.overlays.base import BaseOverlay

# Download Bitstream of PYNQ-Z2 base overlay
base_overlay = BaseOverlay("base.bit")
```

```
help(base_overlay)
```

```
help(base_overlay.leds)
```

Informations about components

Blink LED

```
import time

# turn on the first LED
base_overlay.leds[0].on()

# wait for five second
time.sleep(5)

# turn off the first LED
base_overlay.leds[0].off()
```

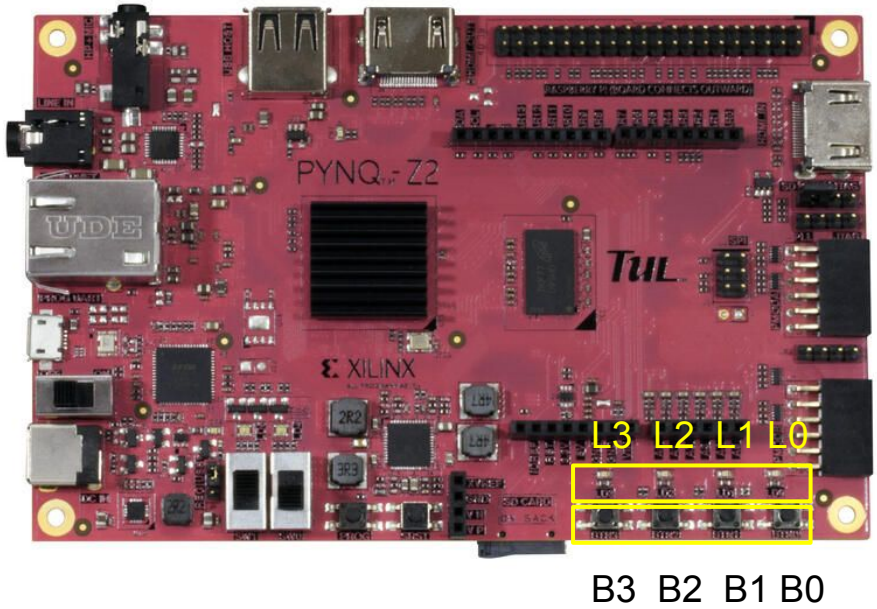
```
# button is released, value = 0
print(base_overlay.buttons[0].read())

# press botton for 5 seconds
time.sleep(3)

# button is pressed, value = 1
print(base_overlay.buttons[0].read())
```

```
0
1
```

Button Access

Lab 0-3: LED Blinking and Button Access(2/2)

- Follow the instructions below to complete this Lab
  - Lights on the LED while the corresponding button is pressed
  - And lights off the LED while the corresponding button is released
  - EX, B1 is pressed then L1 is on; B1,B3 are pressed then L1,L3 are on



Summarize

- This week we
  - Connect to the PYNQ-Z2 board
  - Login to the Jupyter-Notebook
  - Try the pre-built overlay, provided by Xilinx.
- Next week Goal
  - Build our own overlay by using Vivado tool chain.
- One bounce
  - Compare the two methods for video input/output.