



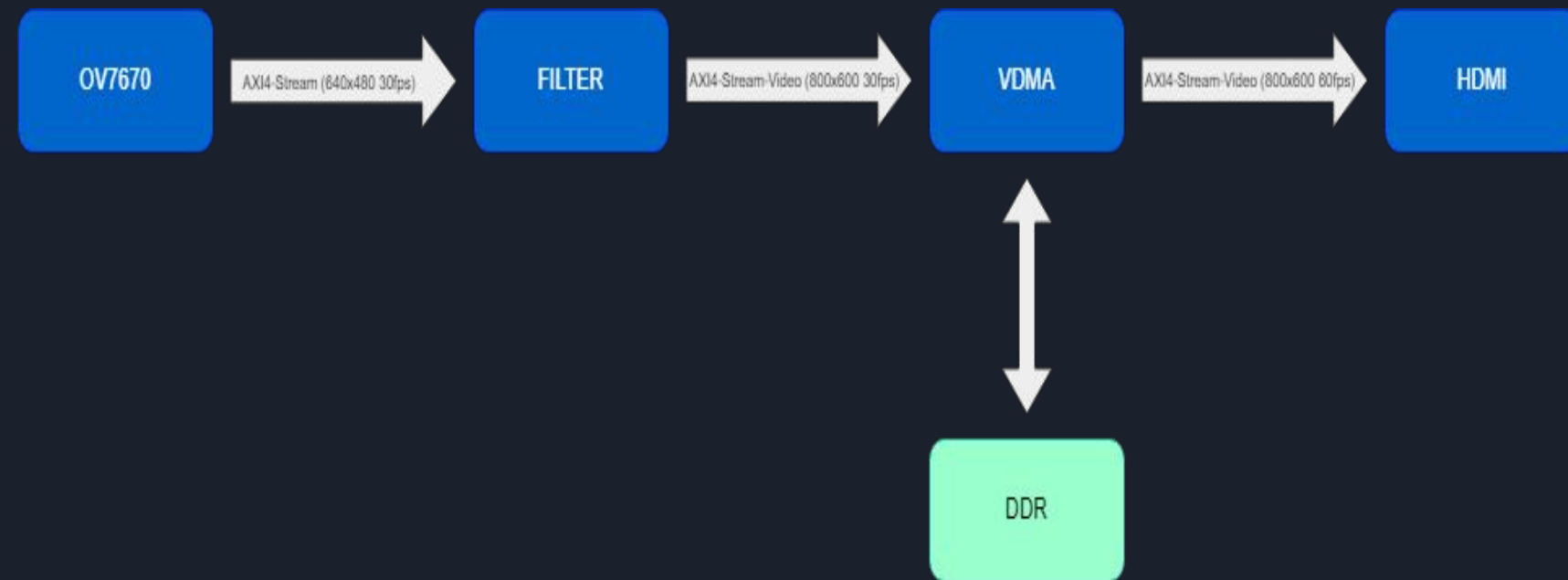
OV7670 e filtro convoluzionale su Pynq-Z2



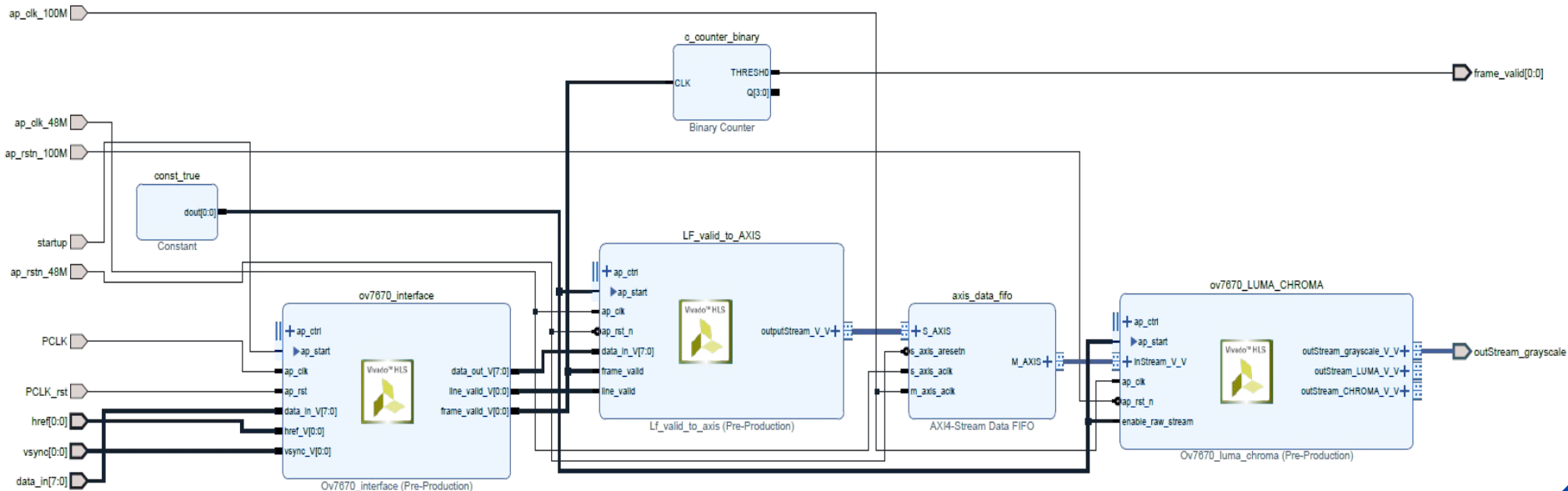
Mazzoli Mattia
mattia.mazzoli@studio.unibo.it


Di Lorenzi Matteo
matteo.dilorenzi@studio.unibo.it

Struttura base di funzionamento

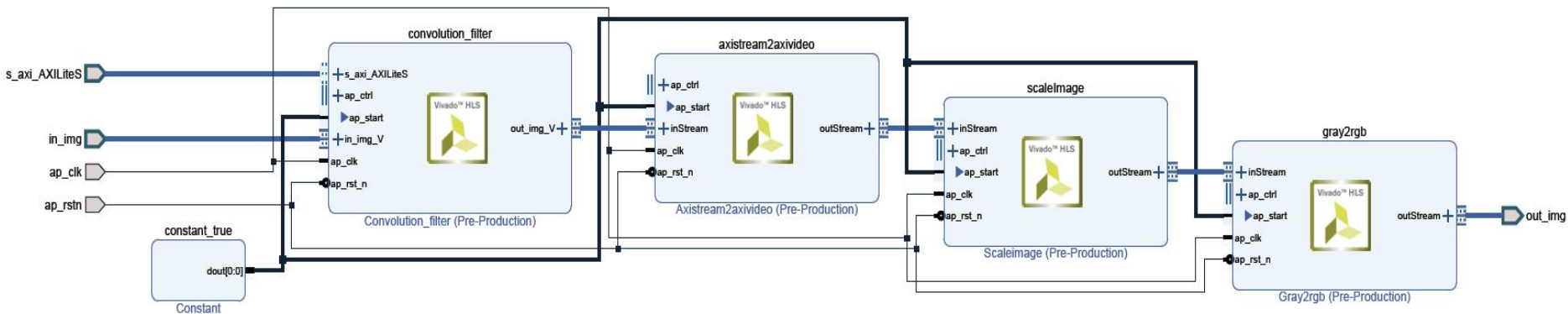



OV7670



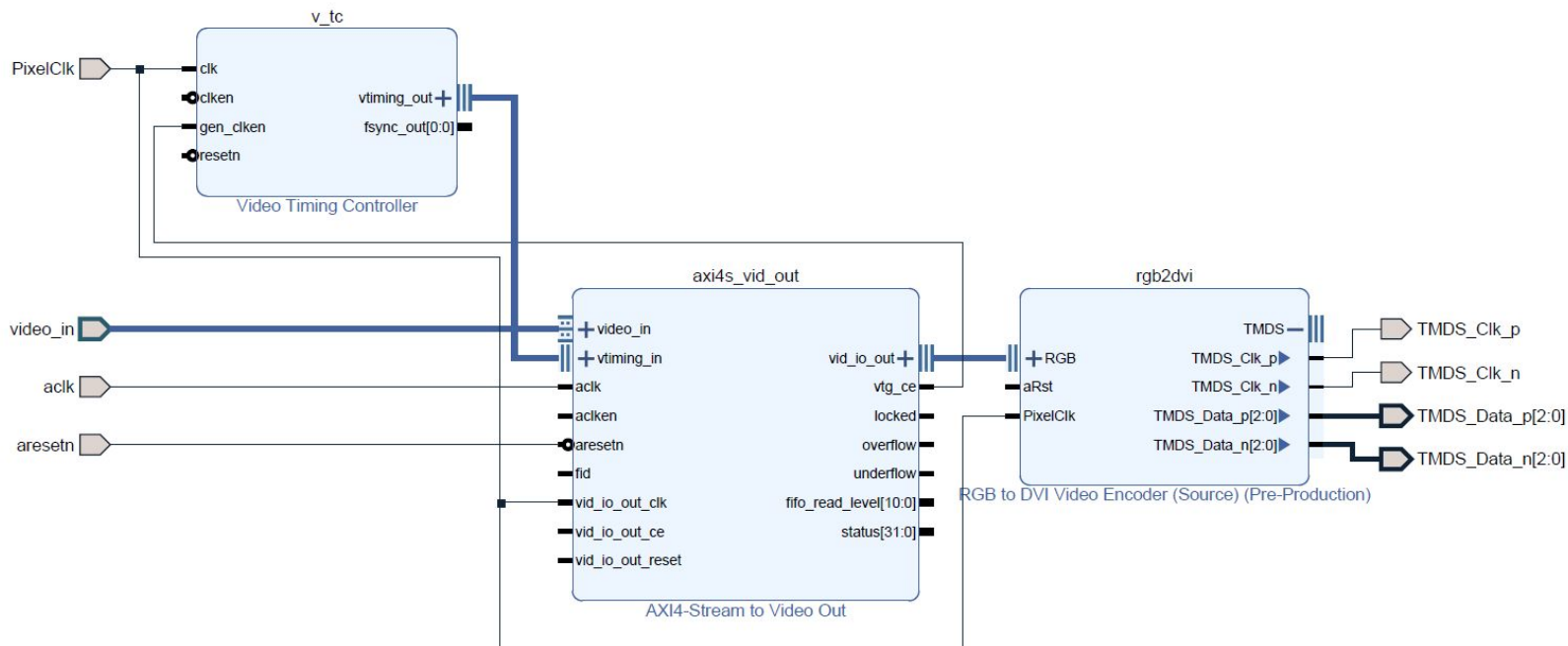
- 
- ap_start di OV7670_interface connesso a SW0 della board per sincronizzare i blocchi all'avvio.
 - Clock a 24 MHz per IP OV7670_interface, 48 MHz per LF_valid_to_AXIS, 100MHz per i restanti blocchi.
 - Trasforma i segnali provenienti dal sensore in un AXI4-Stream.
 - Frame_valid è connesso ad un led che lampeggia all'arrivo di nuovi frame.


FILTER



- 
- Clock a 100MHz per tutti gli IP.
 - Filtro convoluzionale configurabile tramite AXILite.
 - Gli IP successivi al filtro sono necessari a rendere lo stream compatibile con gli IP presenti in HDMI:
 - `axistream2axivideo` → inserisce i segnali *tuser* e *tlast*
 - `scaleImage` → converte lo stream a una risoluzione di 800x600
 - `gray2rgb` → converte lo stream da scala di grigi al formato RGB

HDMI



- 
- `axi4s_vid_out` → converte in un Video Protocol un AXI Stream Video inserendo i segnali per la sincronizzazione tramite il Video Timing Controller.
 - `v_tc` → genera i segnali di timing necessari all'AXI4-Stream to Video Out.
 - `rgb2dvi` → IP importato dalle librerie Digilent, permette di controllare l'uscita HDMI della board.
 - PixelCLK impostato a 40MHz, il minimo supportato dal blocco rgb2DVI per ottenere la risoluzione di 800x600 in uscita.



Python

1. Configurazione dei componenti tramite Overlay

```
from pynq import Overlay

overlay = Overlay("design_1.bit")
```

2. Interazione diretta con gli IP Core configurabili che compongono il progetto

```
IP Blocks
-----
axi_gpio           : pynq.lib.axigpio.AxiGPIO
axi_iic            : pynq.lib.iic.AxiIIC
Interrupt/axi_intc : pynq.overlay.DefaultIP
VDMA/axi_vdma      : pynq.lib.video.dma.AxiVDMA
filter/convolution_filter : pynq.overlay.DefaultIP
```

Configurazione del filtro convoluzionale

```
def update_filter(self, fil):
    if(len(fil) != 51):
        print("La lunghezza del filtro deve essere di 51 elementi")

    address = self.address_offset
    data = 0x00000000
    bits_shift = 0
    counter = 0

    for el in fil:
        if(bits_shift >= 32):
            self.mmio.write(address, data)
            data = 0x00000000
            bits_shift = 0
            address = address + self.offset

        counter += 1
        data = data | (el << bits_shift)
        bits_shift += 8
        if(counter >= 51):
            self.mmio.write(address, data)
```

```
def __init__(self, overlay, base_address=0x43C10000, address_range=0x10000, address_offset=0x40):
    self.base_address = base_address
    self.address_range = address_range
    self.address_offset = address_offset
    self.offset = 0x04
    self.mmio = MMIO(base_address, address_range)
    self.conv = overlay.filter.convolution_filter
```

- Definizione di una classe dedicata ***Convolution_Filter***
- Aggiornamento del filtro in real-time da Jupyter Notebook

Configurazione OV7670

```
def __init__(self, iic):
    self.OV7670_SLAVE_ADDRESS = 0x21

    _ffi = cffi.FFI()
    self.tx_buf = _ffi.new("unsigned char [32]")
    self.rx_buf = _ffi.new("unsigned char [32]")

    self.iic = iic
```

```
def write_register(self, reg, data):
    self.tx_buf[0] = reg
    self.tx_buf[1] = data
```

```
self.iic.send(self.OV7670_SLAVE_ADDRESS, self.tx_buf, 2, 0)
```

- Classe dedicata **OV7670**
- Scritture e letture dei registri del sensore tramite protocollo I2C

```
def read_register(self, reg):
    self.tx_buf[0] = reg

    self.iic.send(self.OV7670_SLAVE_ADDRESS, self.tx_buf, 1, 0)
    self.iic.receive(self.OV7670_SLAVE_ADDRESS, self.rx_buf, 1, 0)

    return self.rx_buf[0]
```

Risultati ottenuti



Immagine neutra catturata dal sensore



1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

Sharpen filter

-1	-2	-4	0	4	2	1
-1	-2	-4	0	4	2	1
-2	-4	-6	0	6	4	2
-4	-6	-8	0	8	6	4
-2	-4	-6	0	6	4	2
-1	-2	-4	0	4	2	1
-1	-2	-4	0	4	2	1

Vertical filter





Grazie per l'attenzione