DPP for X-ray Photon Detection

1st Mesoamerican Workshop on Reconfigurable X-ray Scientific Instrumentation for Cultural Heritage

Luis Guillermo García Ordóñez

IAEA International Atomic Energy Agency
Atoms for Peace and Development

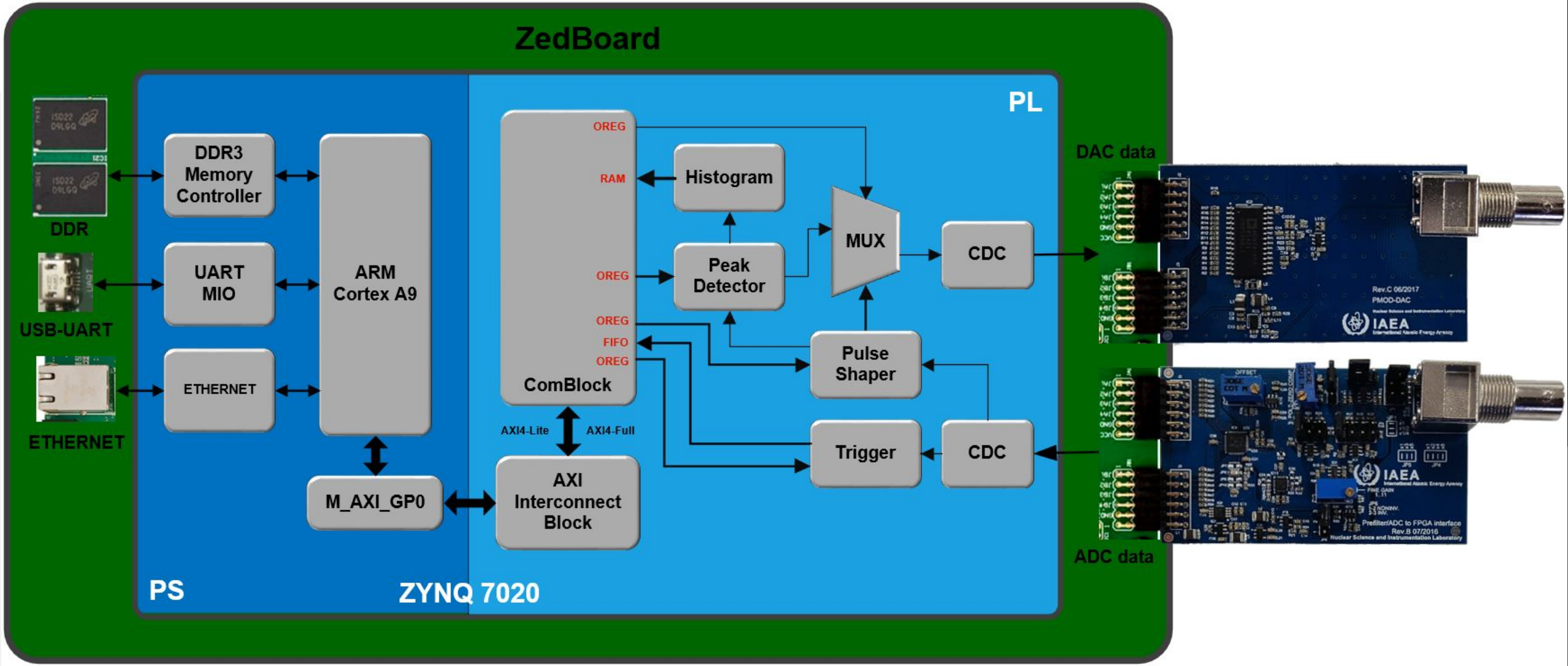
IUPAP Centenary YEARS 1922 - 2022

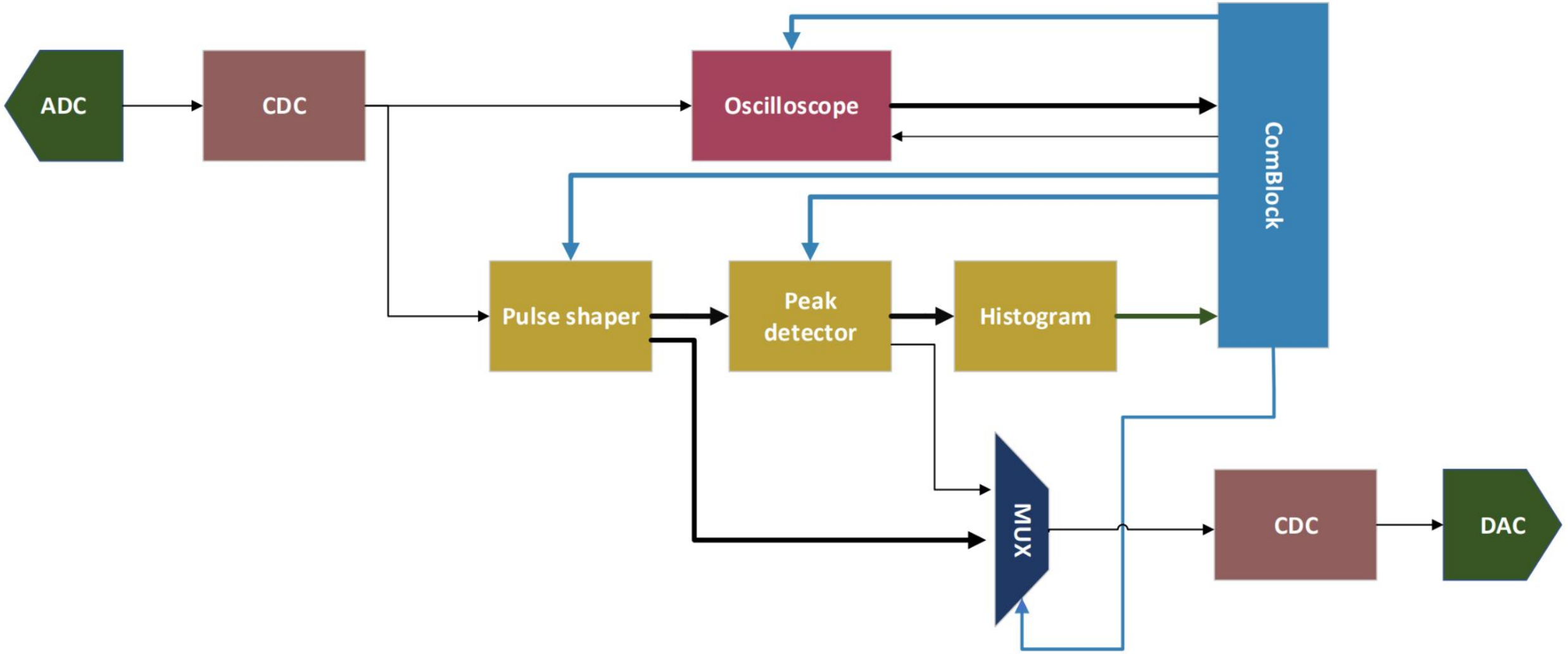
Secretaría Nacional de Ciencia y Tecnología

The Abdus Salam International Centre for Theoretical Physics

Outline

- Create the hardware logical blocks for a DPP system.
- Set the parameters for the oscilloscope using the ComBlock output registers
- Acquire the raw pulse shapes from the oscilloscope using the ComBlock input FIFO
- Set the parameters for baseline remover using the Comblock output registers
- Set the parameters for the pulse shaper using the ComBlock output registers
- Set the parameters for the peak detector using the ComBlock output registers
- Acquire the computed histogram (spectrum) using the ComBlock true dual-port RAM


ZedBoard
PL
PS
ZYNQ 7020
DDR
USB-UART
ETHERNET
DDR3 Memory Controller
UART MIO
ETHERNET
ARM Cortex A9
M_AXI_GP0
AXI Interconnect Block
AXI4-Lite
AXI4-Full
OREG
RAM
OREG
OREG
FIFO
OREG
ComBlock
Histogram
Peak Detector
MUX
CDC
Pulse Shaper
Trigger
CDC
DAC data
ADC data
Rev.C 06/2017
PMOD-DAC
IAEA
Prefilter/ADC to FPGA interface
Rev.B 07/2016


ADC
CDC
Oscilloscope
ComBlock
Pulse shaper
Peak detector
Histogram
MUX
CDC
DAC

Trapezoidal Filter

The discrete-time output of a trapezoidal filter can be expressed as:

$$y[n] = \sum_{i=0}^{L-1} x[n-i] - \sum_{i=K+L}^{K+2L-1} x[n-i]$$

where:

- $x[n]$ is the input signal,
- $y[n]$ is the filtered output,
- L is the length of the integration window (rise and fall time),
- K is the gap between the two integration windows (flat top width).

Trapezoidal Filter

$$a^{K,L}[n] = x[n] - x[n-K] - x[n-L] + x[n-K-L]$$

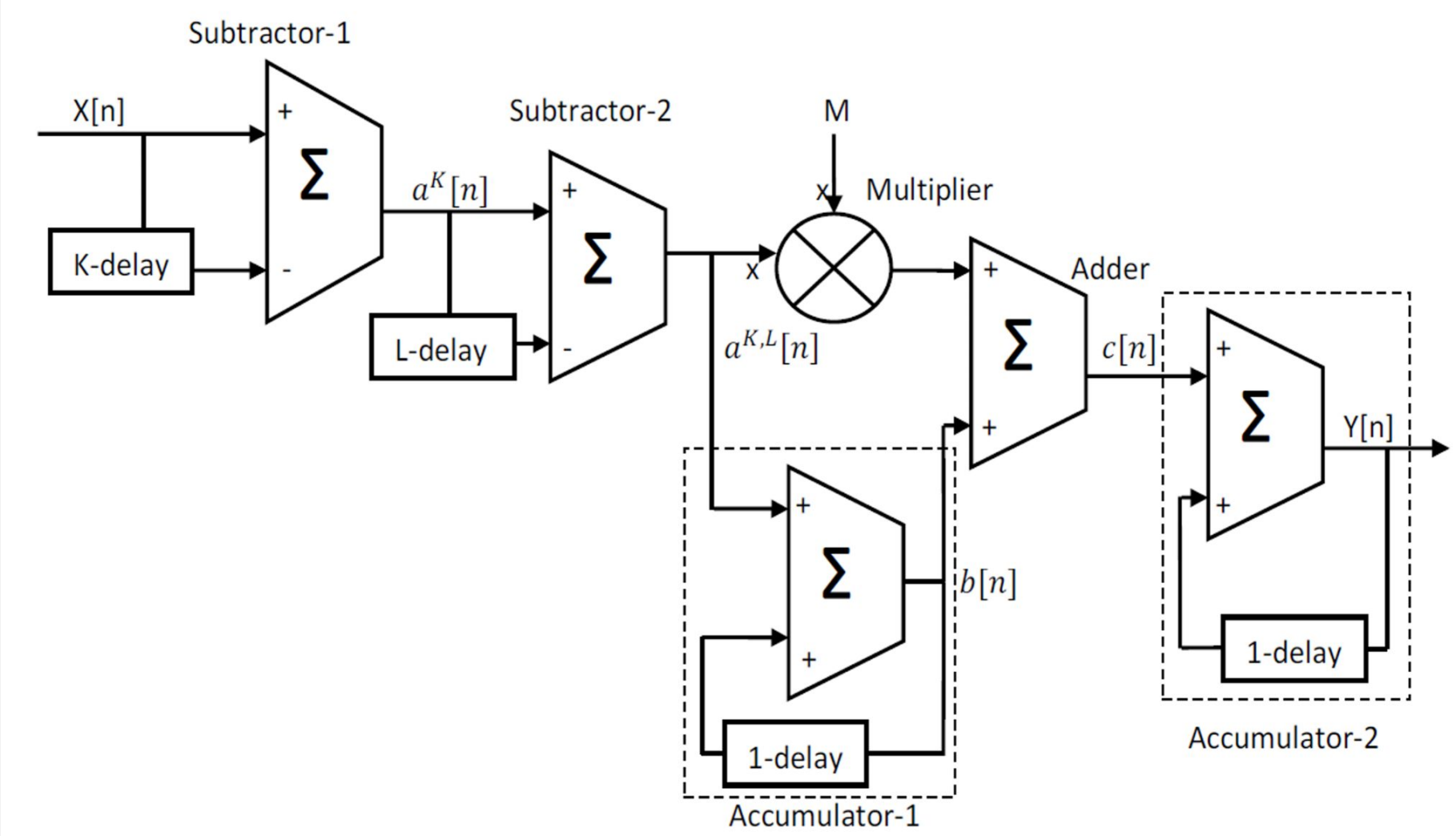
$$b[n] = b[n-1] + a^{K,L}[n], n \geq 0$$

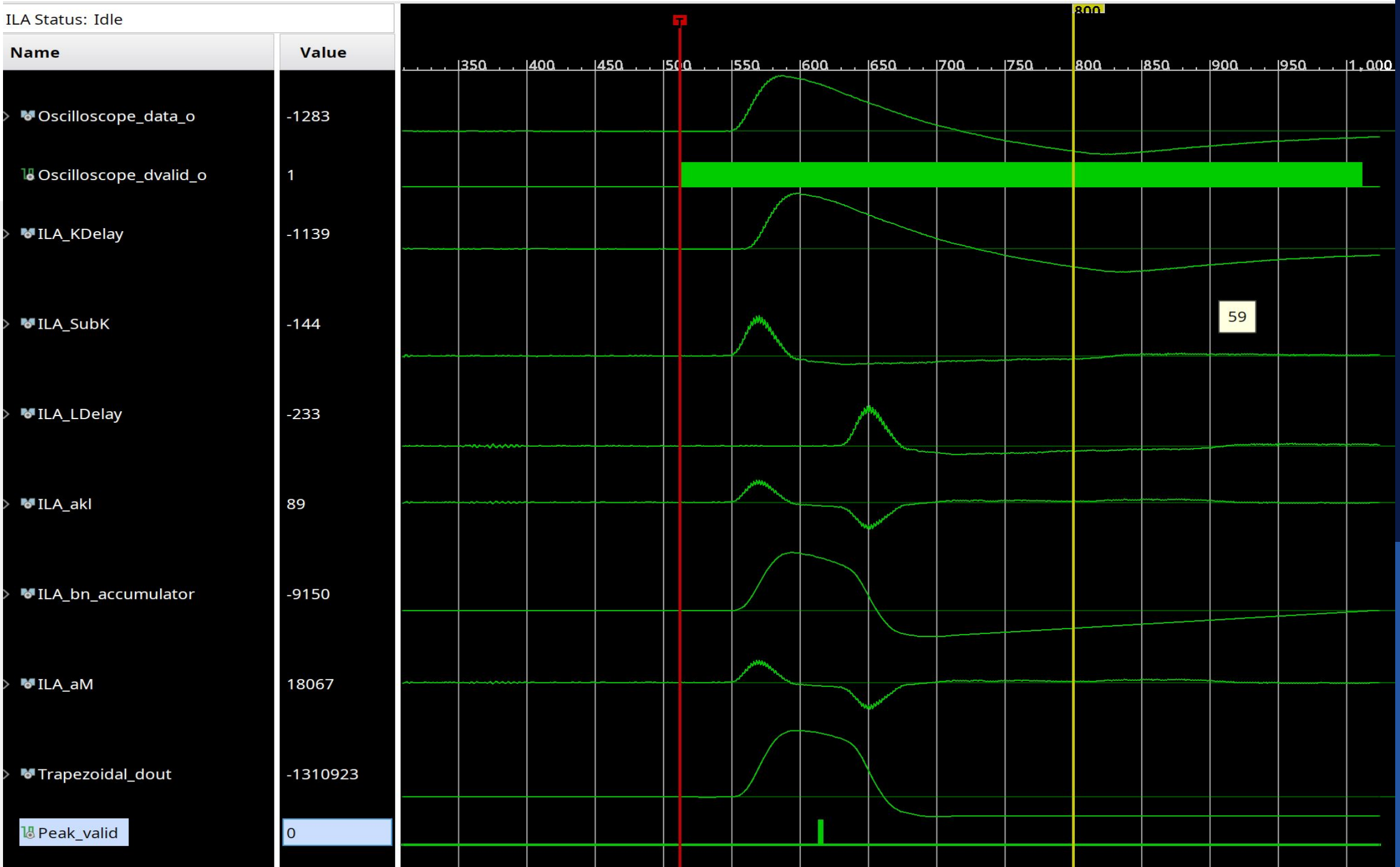
$$c[n] = b[n] + Ma^{K,L}[n]$$

$$y[n] = y[n-1] + c[n], n \geq 0$$

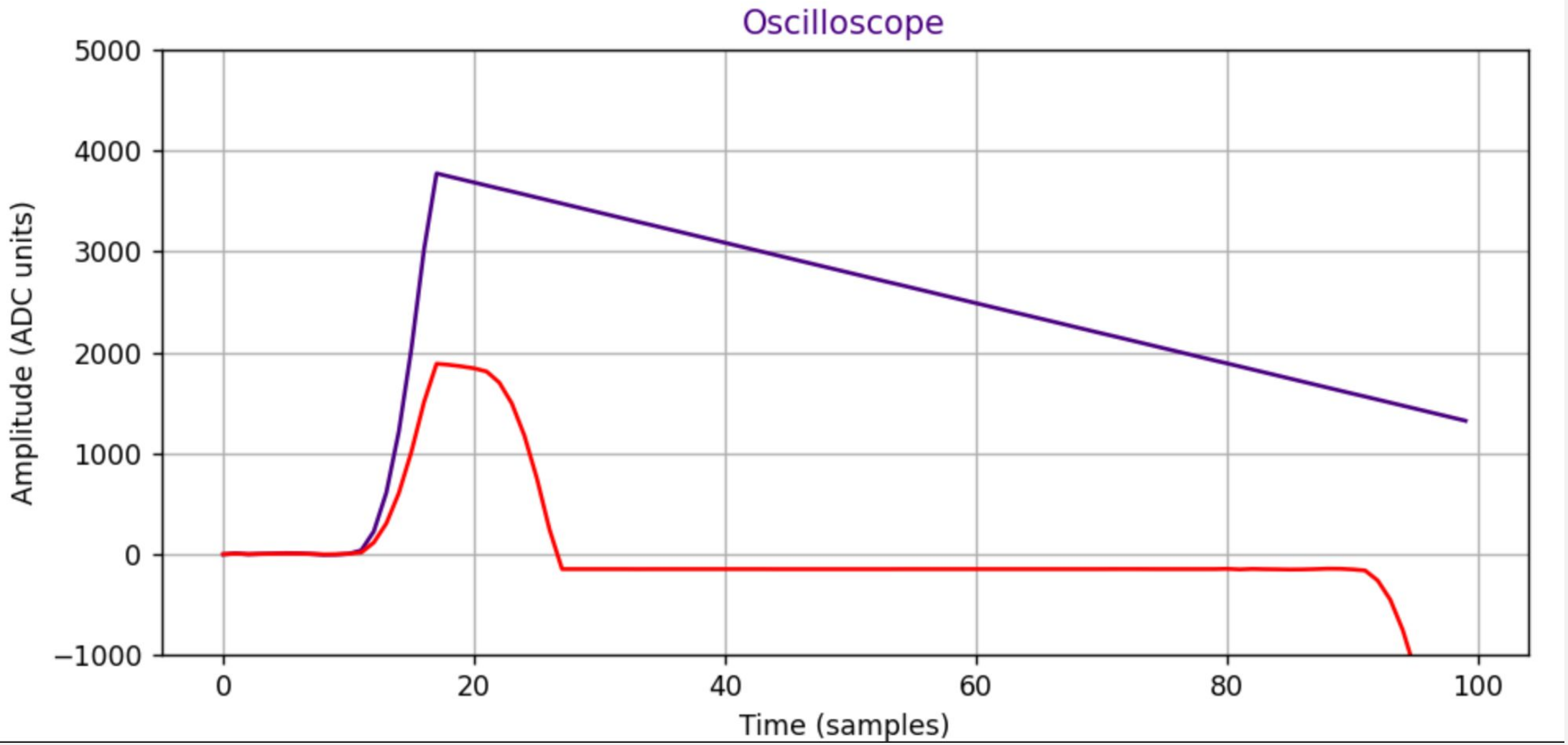
$$M = \frac{1}{e^{Tp/\tau} - 1}$$

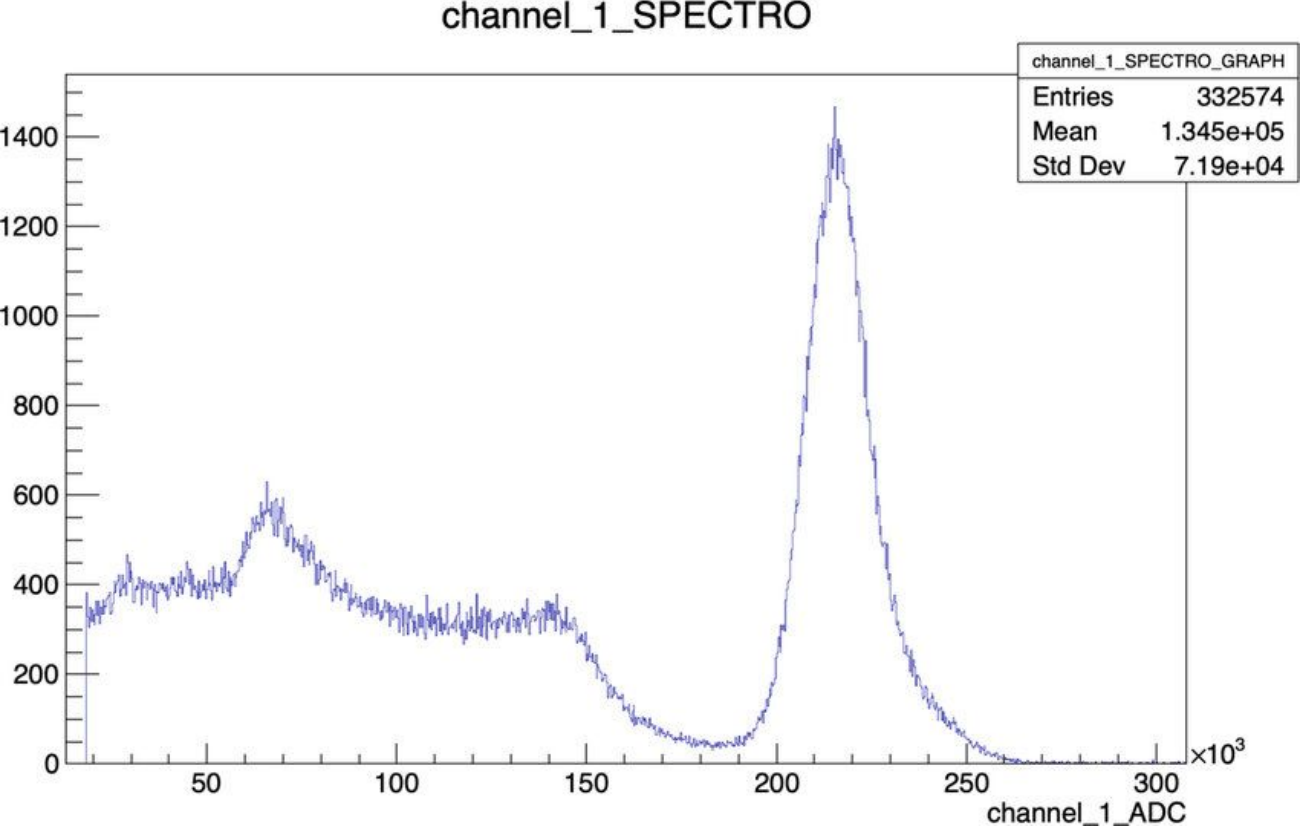
Trapezoidal Filter




ILA Status: Idle
Name
Value
Oscilloscope_data_o
-1283
Oscilloscope_dvalid_o
1
ILA_KDelay
-1139
ILA_SubK
-144
ILA_LDelay
-233
ILA_akl
89
ILA_bn_accumulator
-9150
ILA_aM
18067
Trapezoidal_dout
-1310923
Peak_valid
0
350
400
450
500
550
600
650
700
750
800
850
900
950
1,000
800
59

Trapezoidal Filter



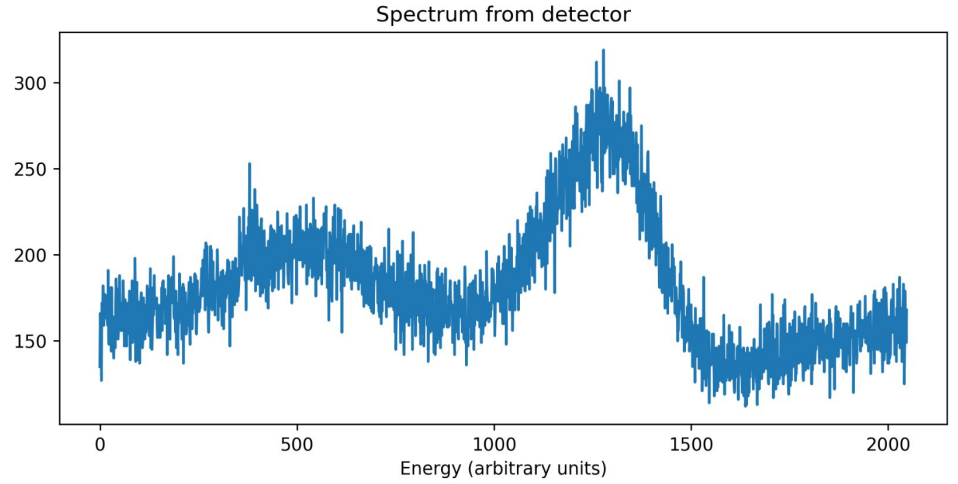


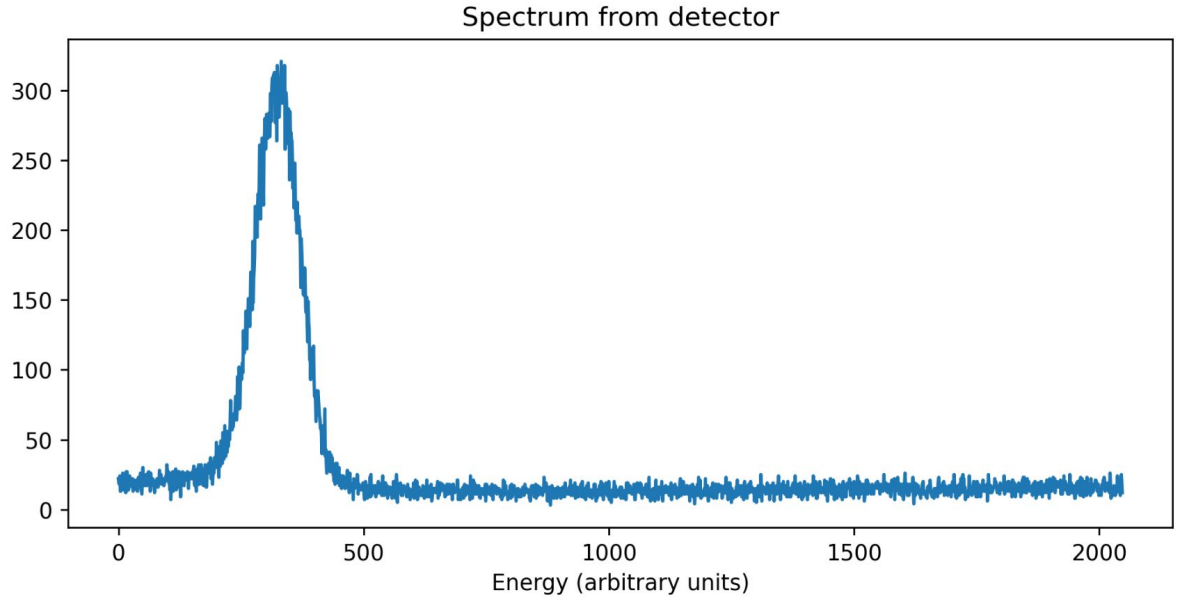
Plot the contents of the RAM using a `plt.plot`. Do not use a `plt.hist`, since the histogram has already been computed in the FPGA

```
spectrum = np.array(spectrum).astype(np.uint32)

xAxis = np.linspace(0, HISTO_LEN - 1, HISTO_LEN)

plt.figure()
plt.title("Spectrum from detector")
plt.plot(xAxis, spectrum, '-')
plt.xlabel("Energy (arbitrary units)")
# plt.yscale('log')
plt.show()
```

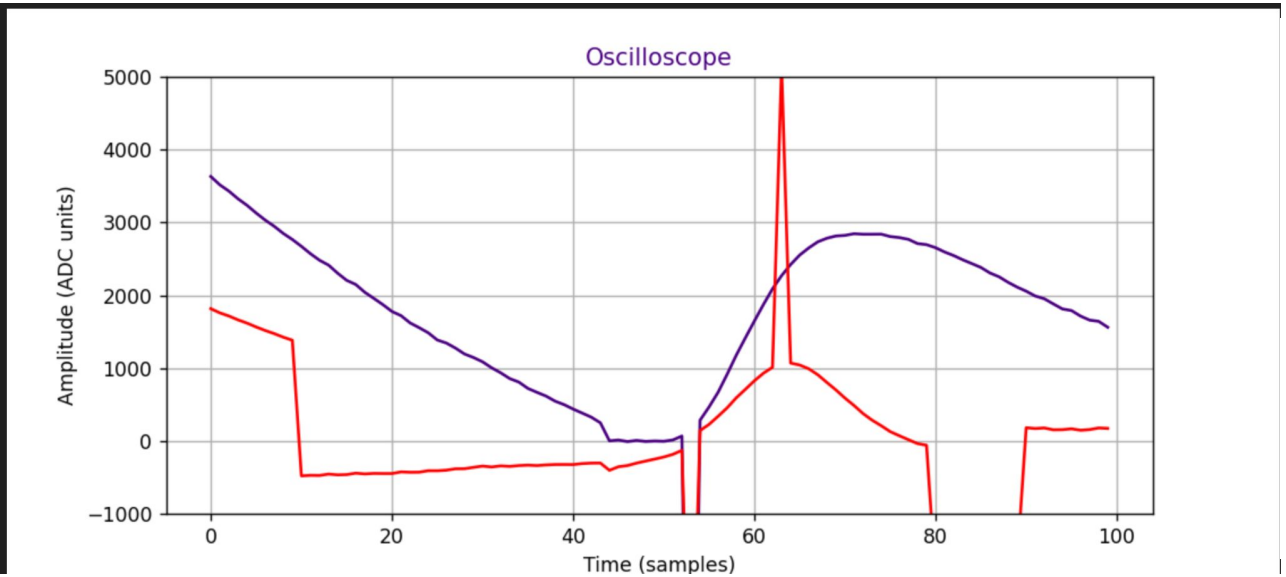
✓ 0.0s

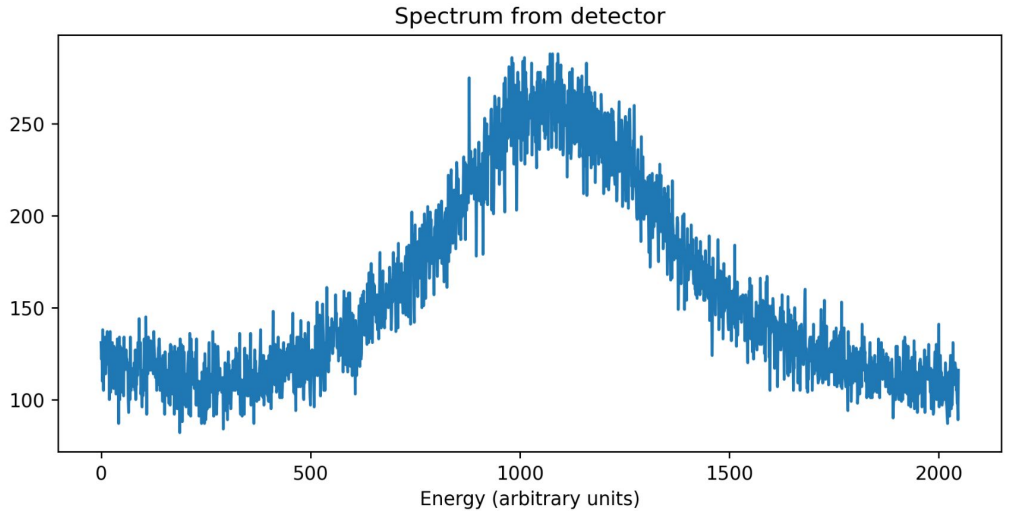
Figure 6

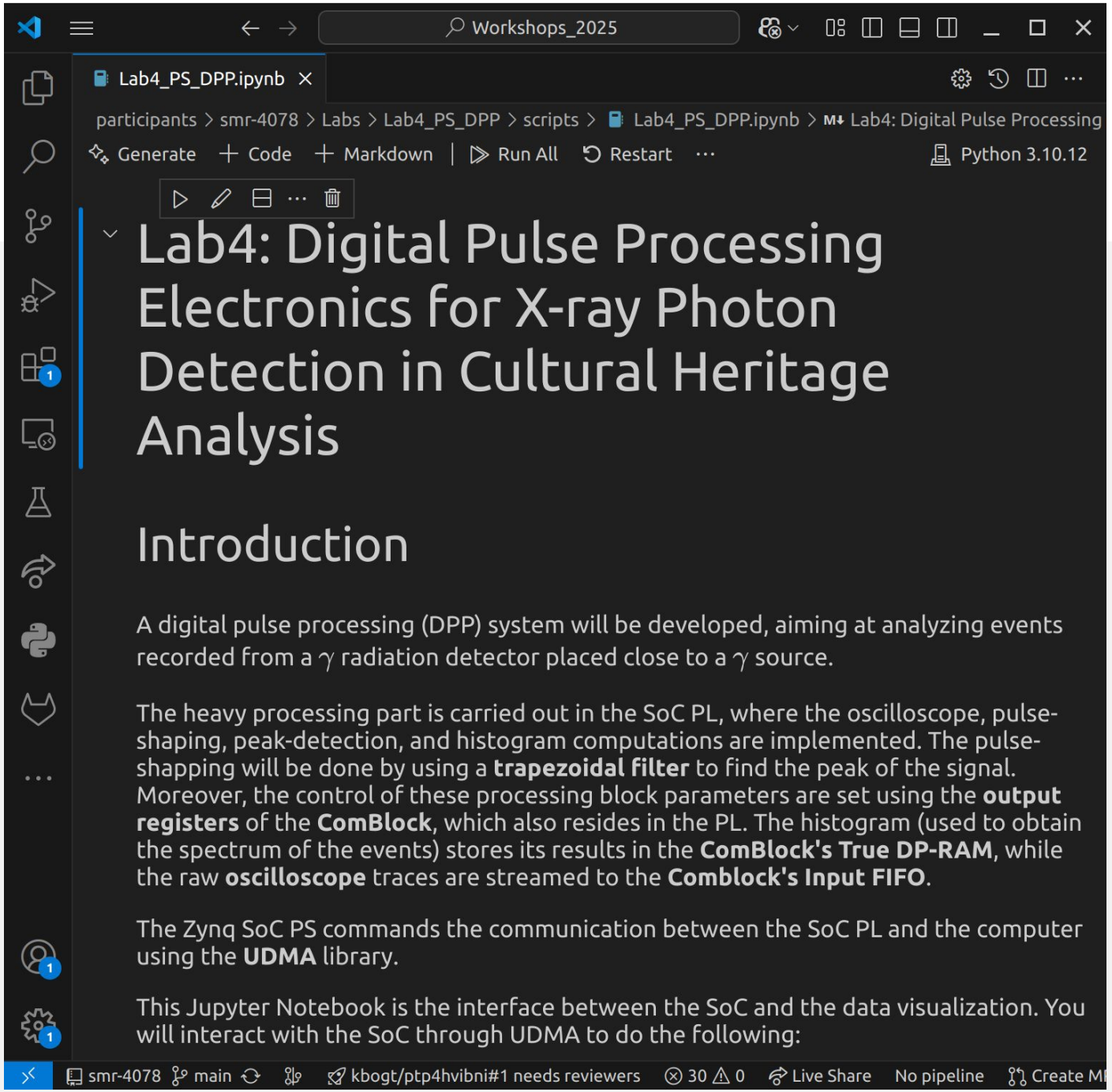




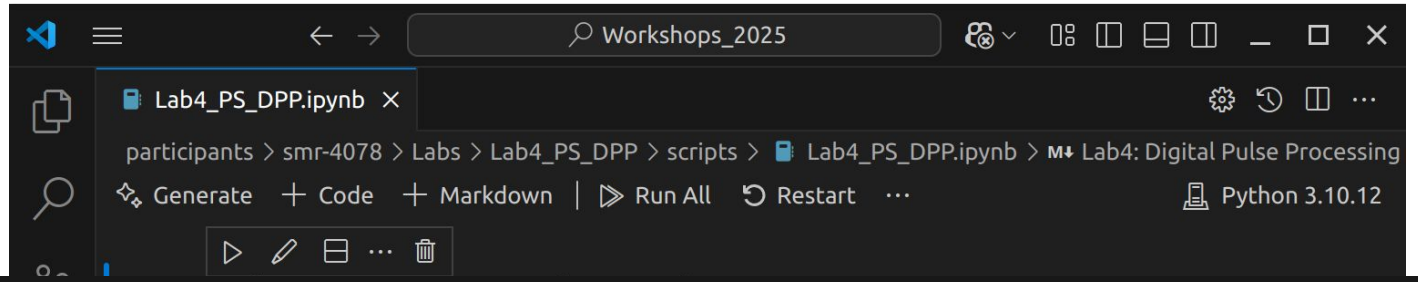
Every system is different








```
Oscillo_Threshold_reg= ???
Oscillo_SBT_reg= ???
Oscillo_Pulse_len_reg= ???
Oscillo_Edge_reg= ???
Oscillo_Ext_ret= ??? ## Delete on the Lab

# Baseline Remover (BR)

BR_Threshold_reg= ???
BR_Set_reg= ???

# Trapezoidal Filter Registers (TF)
TF_K_reg= ???
TF_L_reg= ???
TF_M_reg= ???
TF_NSamp_reg= ???

# DAC Mux (DMUX)
DMUX_sel_reg= ???
```




```
Oscillo_Threshold_reg= ???
Oscillo_SBT_reg= ???
Oscillo_Pulse_len_reg= ???
Oscillo_Edge_reg= ???
Oscillo_Ext_ret= ??? ## Delete on the Lab
```

```
### Change this values to fit your pulse
K=30
L=10
M=250

zedBoard.write_reg(TF_K_reg, K)
zedBoard.write_reg(TF_L_reg, L)
zedBoard.write_reg(TF_M_reg, M)
zedBoard.write_reg(TF_NSamp_reg, 2*K+L)

pulseRecorder = PulseFit(pulseLenSamples=TRACE_LEN,
                         samplingRate=SAMPLING_RATE,
                         resolutionBits=ADC_RESOLUTION+2) ##Notice that by removing the baseline our resolution increase, why?

dataset = pulseRecorder.recordDataset(PULSES_TO_RECORD, zedBoard)

Trapezoidal_plot(dataset, l=L, k=K, M=M, scale=1, PLOT_Y_AXIS_LIMITS=(-1000,5000))
```

the raw **oscilloscope** traces are streamed to the **Comblock's input FIFO**.

The Zynq SoC PS commands the communication between the SoC PL and the computer using the **UDMA** library.

This Jupyter Notebook is the interface between the SoC and the data visualization. You will interact with the SoC through UDMA to do the following:

We will continue in the lab.

1st Mesoamerican Workshop on Reconfigurable X-ray Scientific Instrumentation for Cultural Heritage

Luis Guillermo García Ordóñez

IAEA International Atomic Energy Agency
Atoms for Peace and Development

IUPAP Centenary YEARS 1922 - 2022

Gobierno de la República Guatemala
Secretaría Nacional de Ciencia y Tecnología

Academia Caroliana Orbis Conspicua Interfaces Guatemalensis

ICTP The Abdus Salam International Centre for Theoretical Physics