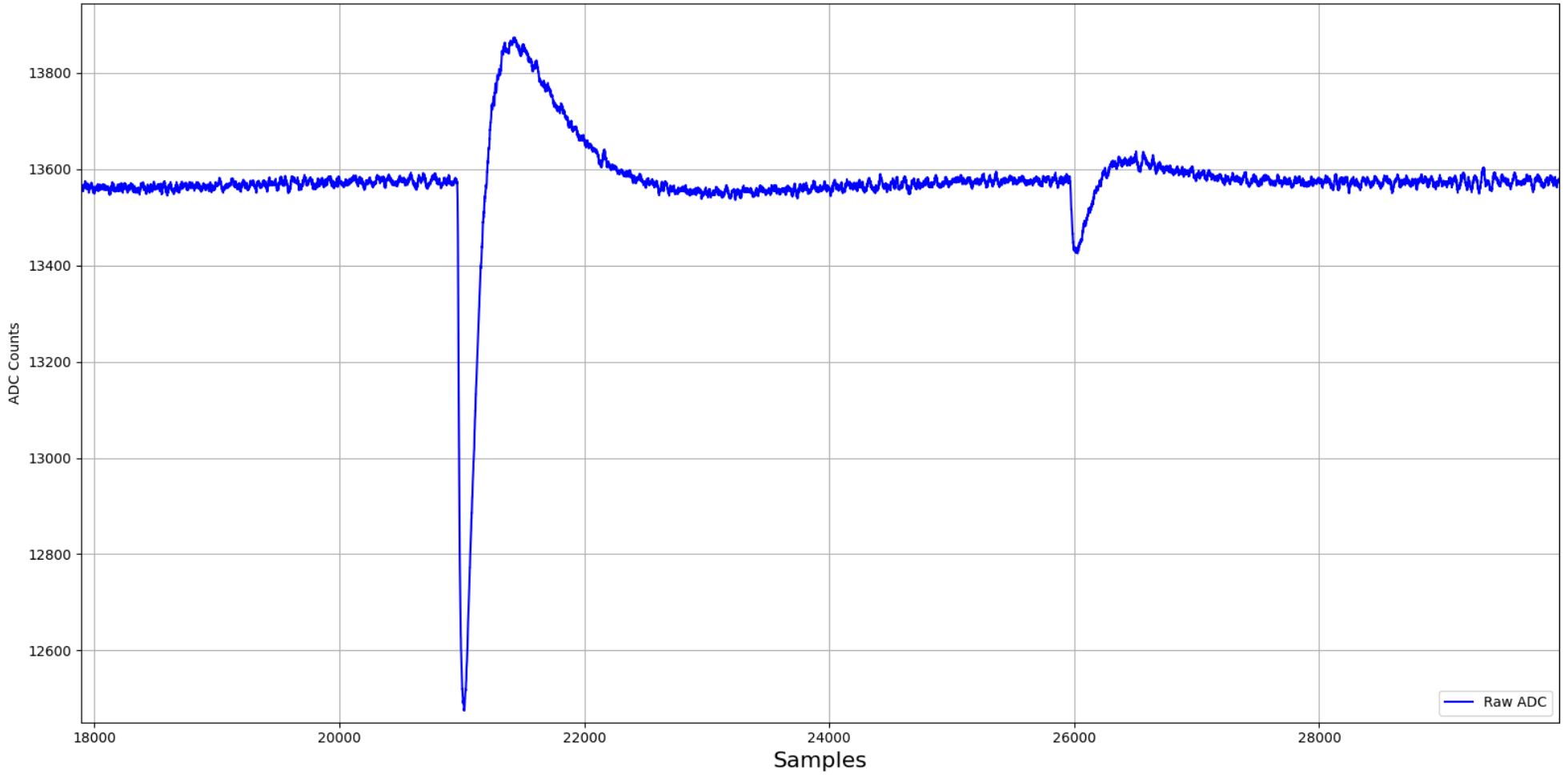
# Daphne v2A at CIEMAT

Antonio Verdugo, Jose Ignacio Crespo, **Ignacio López de Rego,**
CIEMAT

**Warm Electronics Meeting** 9th November 2023

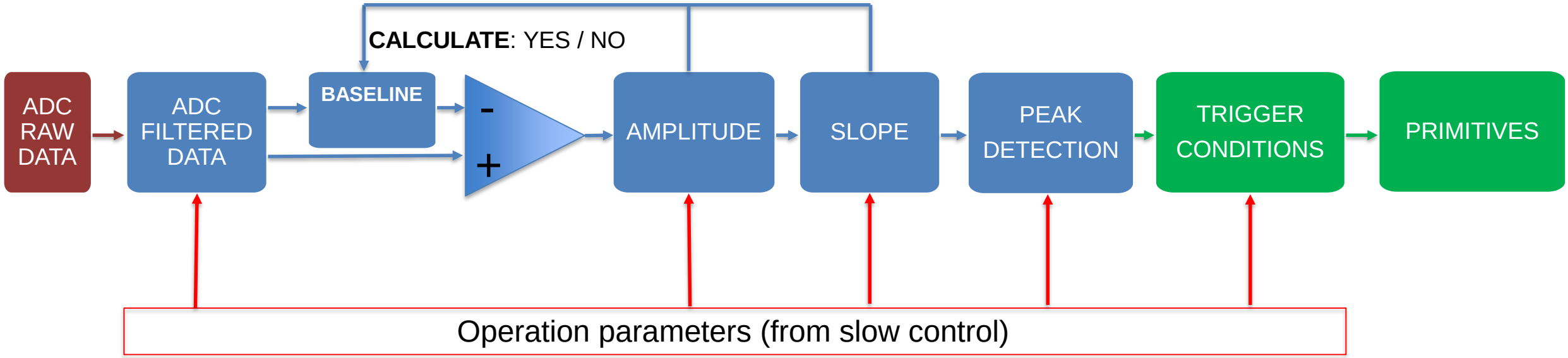
# Self-trigger & Primitive Calculation ALGORITHM (1)

## Light detection signal



# Self-trigger & Primitive Calculation ALGORITHM (2)

## Peak Detection

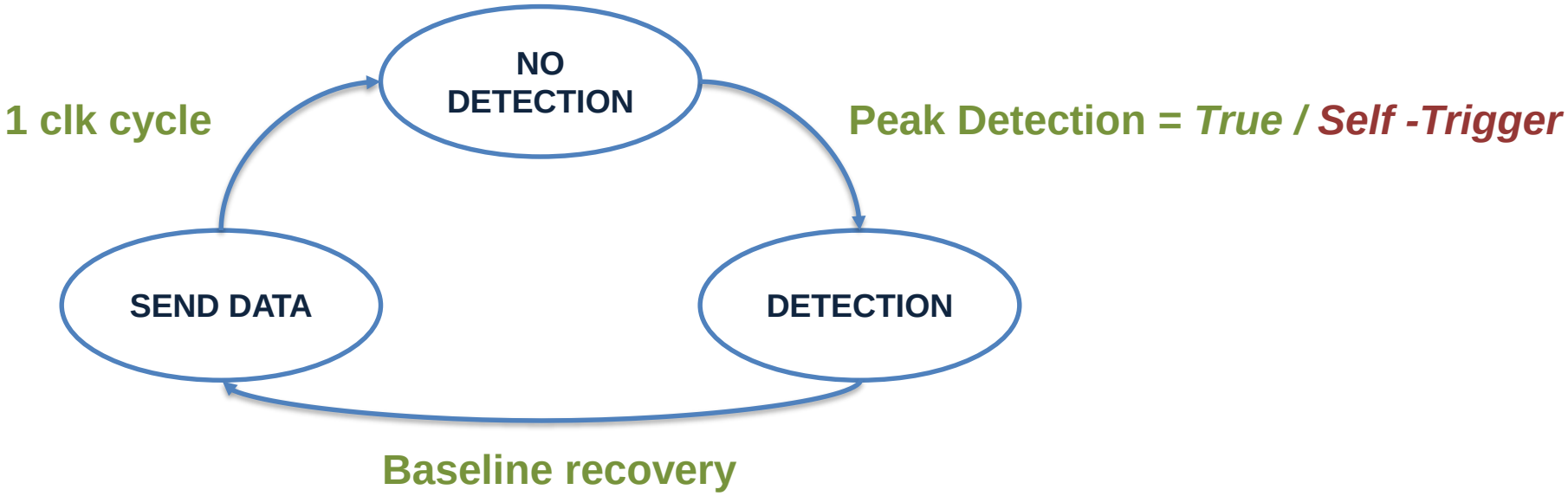



- **ADC FILTERED DATA**: Moving average of 2 ADC RAW DATA SAMPLES
  → Reduces High Frequency noise.
- **BASELINE:**
  - Based on cumulative average over previous N (4) samples.
  - Stop baseline calculation if peak is detected.
- **AMPLITUDE =** Filtered Data – Baseline.
- **SLOPE =** Amplitude last simple – Amplitude previous sample.
- **PEAK DETECTION**: Threshold over the slope.

**Actualized value each CLK cycle**

# Self-trigger & Primitive Calculation ALGORITHM (3)

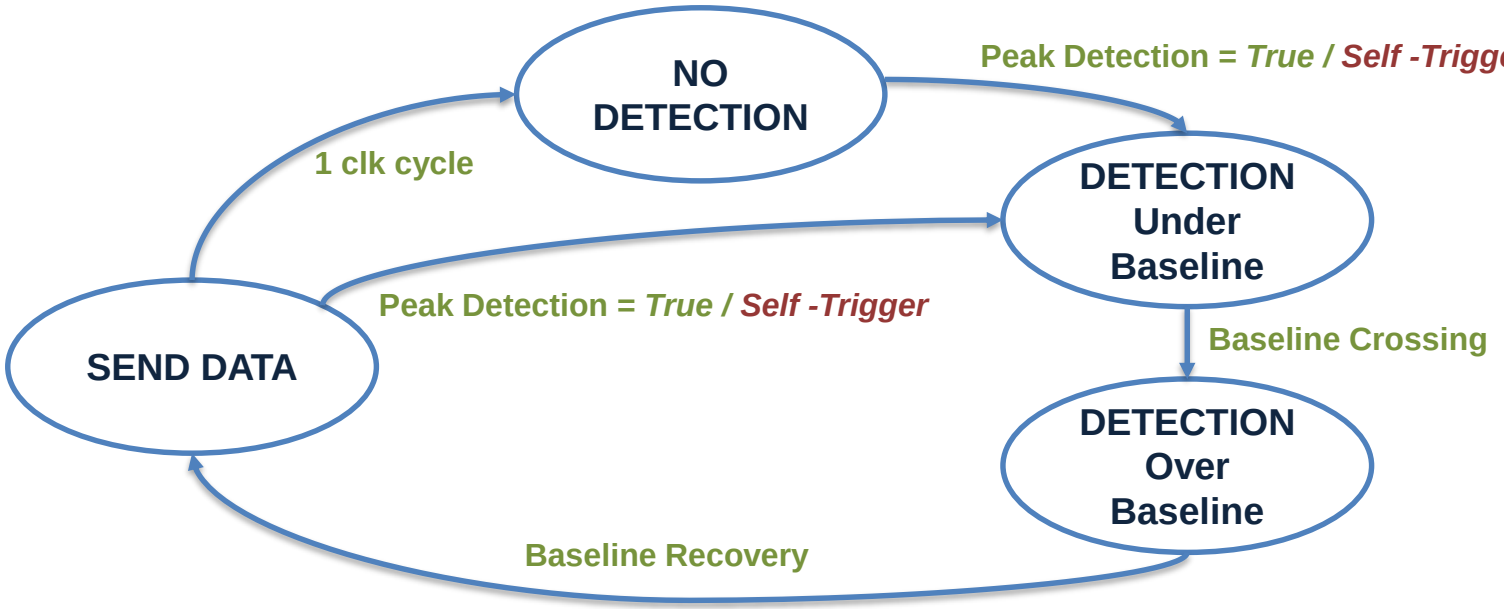
## Trigger Condition & Primitive Calculation: OLD APPROACH




- **TRIGGER CONDITION**: When a Peak is detected in NO DETECTION *State.*
- **NO DETECTION *State*:**
  - Peak detection variables calculation (Baseline Calculation).
- **DETECTION *State* :**
  - Peak detection variables calculation (Baseline remains constant)
  - Waveform`s Primitive Calculation.
  - Peak detection does not generate a self-trigger signal.
- **SEND DATA *State*:** Waveform's Primitve Data available.

# Self-trigger & Primitive Calculation ALGORITHM (4)

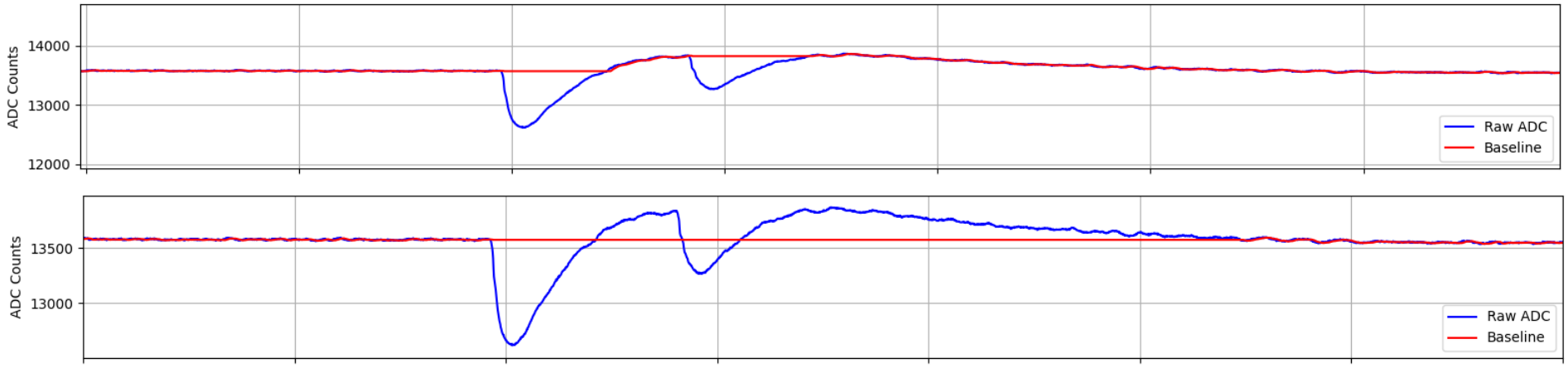
## Trigger Condition & Primitive Calculation: NEW APPROACH




- **TRIGGER CONDITION**: When a Peak is detected in NO DETECTION *State.*
- **NO DETECTION *State*:**
  - Peak detection variables calculation (Baseline Calculation).
- **DETECTION *States* :**
  - Peak detection variables calculation (Baseline remains constant)
  - Waveform`s Primitive Calculation.
  - Peak detection does not generate a self-trigger signal.
- **SEND DATA *State*:** Waveform's Primitve Data available.

# Self-trigger & Primitive Calculation ALGORITHM (5)

## Comparison: OLD APPROACH vs NEW APPROACH




**MAIN DIFFERENCES**

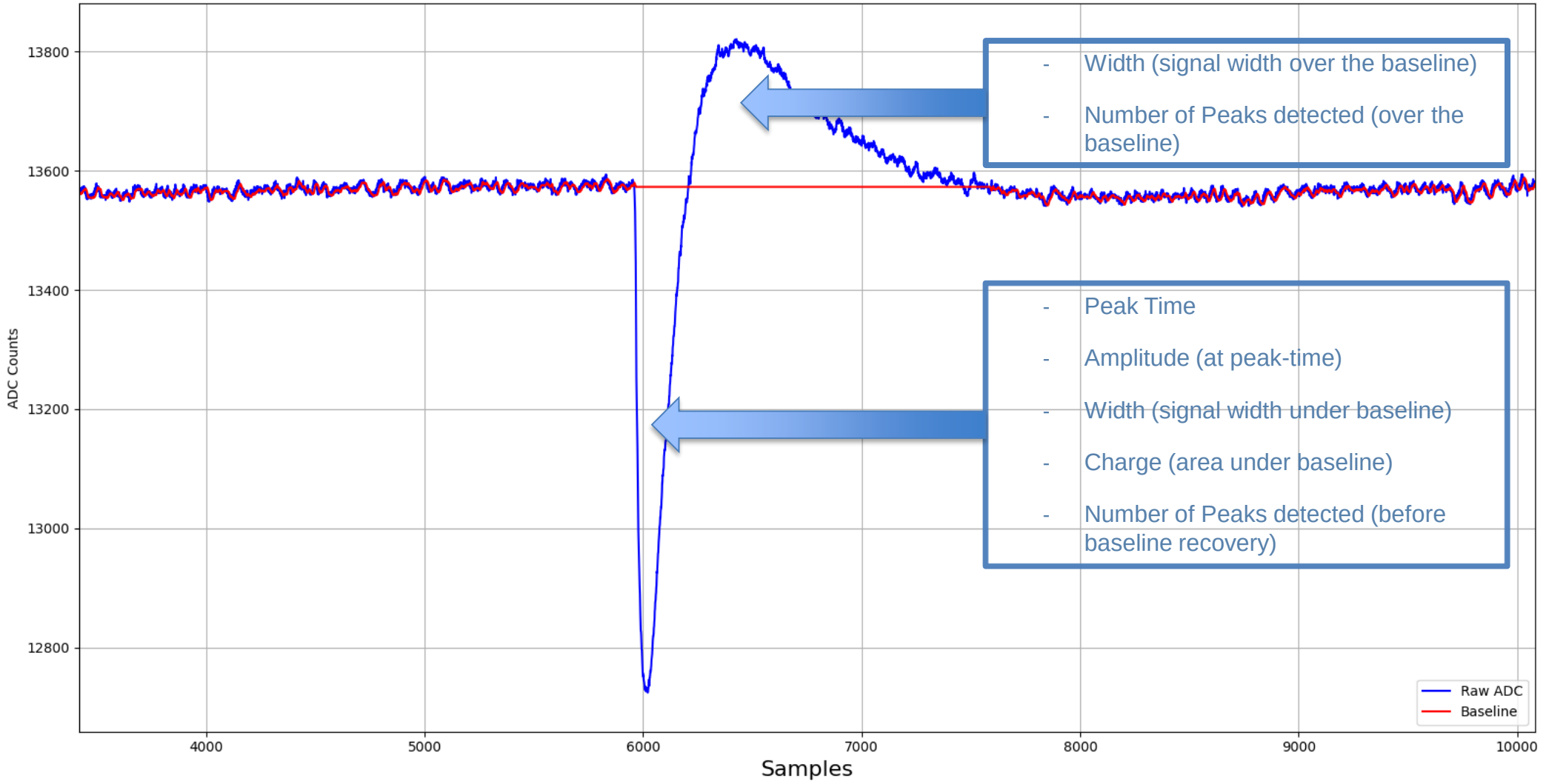
- **Baseline:**
  - *OLD APPROACH: Follows undershoot.*
  - *NEW APPROACH: Remains constant.*
- **Self-Trigger:**
  - *OLD APPROACH: It is allowed during undershoot.*
  - *NEW APPROACH: It is  not allowed during undershoot.*

**MOTIVATION**

- Self-Trigger Event → Waveform's Primitive Calculation.
- Waveform`s Primitive Calculation is not accurate in the undershoot.
- Baseline calculation stability.

# Self-trigger & Primitive Calculation ALGORITHM (6)

## Waveform's Primitives




- Width (signal width over the baseline)
- Number of Peaks detected (over the baseline)

- Peak Time
- Amplitude (at peak-time)
- Width (signal width under baseline)
- Charge (area under baseline)
- Number of Peaks detected (before baseline recovery)

# Self-trigger & Primitive Calculation QUESTIONS (1)

## Self-Trigger Frame Format

```
Data_Available: out std_logic;                    -- Primitives calculation available. Active HIGH
Time_Peak: out  std_logic_vector(7 downto 0);     -- Time in Samples to achieve de Max peak
Time_Pulse_UB: out  std_logic_vector(9 downto 0);  -- Time in Samples of the light pulse signal is UNDER BASELINE (without undershoot)
Time_Pulse_OB: out  std_logic_vector(10 downto 0);  -- Time in Samples of the light pulse signal is OVER BASELINE (undershoot)
Max_Peak: out  std_logic_vector(15 downto 0);      -- Amplitude in ADC counts od the peak
Charge: out  std_logic_vector(19 downto 0);        -- Charge of the light pulse (without undershoot) in ADC*samples
Number_Peaks_UB: out  std_logic_vector(3 downto 0);-- Number of peaks detected when signal is UNDER BASELINE (without undershoot).
Number_Peaks_OB: out  std_logic_vector(3 downto 0);-- Number of peaks detected when signal is OVER BASELINE (undershoot).
```

**73 bits required**

```
d <= X"0000003C" when (state=sof) else -- sof of frame word = D0.0 & D0.0 & D0.0 & K28.1
     link_id & slot_id & crate_id & detector_id & version_id when (state=hdr0) else
     ts_reg(31 downto 0)  when (state=hdr1) else
     ts_reg(63 downto 32) when (state=hdr2) else
     ("00" & trigsample & x"00" & "00" & ch_id(5 downto 0)) when (state=hdr3) else  -- trigger sample and channel ID
     ("00" & baseline & "00" & threshold) when (state=hdr4) else -- average baseline and user-threshold
     (afe_dly0( 3 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto  0))                          when (state=dat0)
     (afe_dly0( 7 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto  4))                          when (state=dat2)
     (afe_dly0(11 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto  8))                          when (state=dat4)
     ( afe_dly( 1 downto 0) & afe_dly0(13 downto 0) & afe_dly1(13 downto  0) & afe_dly2(13 downto 12)) when (state=dat6)
     (afe_dly0( 5 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto  2))                          when (state=dat9)
     (afe_dly0( 9 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto  6))                          when (state=dat11
     (afe_dly0(13 downto 0) & afe_dly1(13 downto 0) & afe_dly2(13 downto 10))                          when (state=dat13
     X"FFFFFFFF" when (state=trailer) else -- OR indicate some overflow condition here!?
     "0000" & crc20 & X"DC" when (state=eof) else -- "0000" & CRC[19..0] & K28.6
     X"00000000";
```

# Self-trigger & Primitive Calculation QUESTIONS (2)

Once a waveform passes the **self-trigger condition (“detection state”)**, a **fixed-length readout window** is sent to DAQ RU server (1024 samples × 16 ns ≈ 16 us).

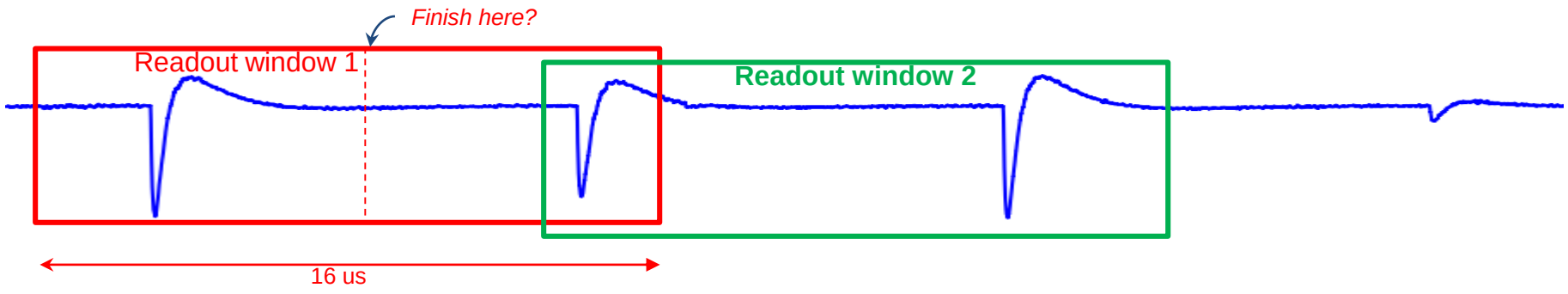
Questions:

What drives the **1024-sample length** (16 us)?

If waveform satisfies the **self-trigger condition again** before the end of the fixed-length window, **open a new-fixed length window**?

How to treat the **overlap** for fixed windows? fill with “zeroes/EOF” the remaining old window? or duplicate the ADC samples?

Or since scintillation light emission is stochastic, shall we use a **variable-length readout window**? (i.e. finish the readout window when we recover baseline and we exit the detection state)



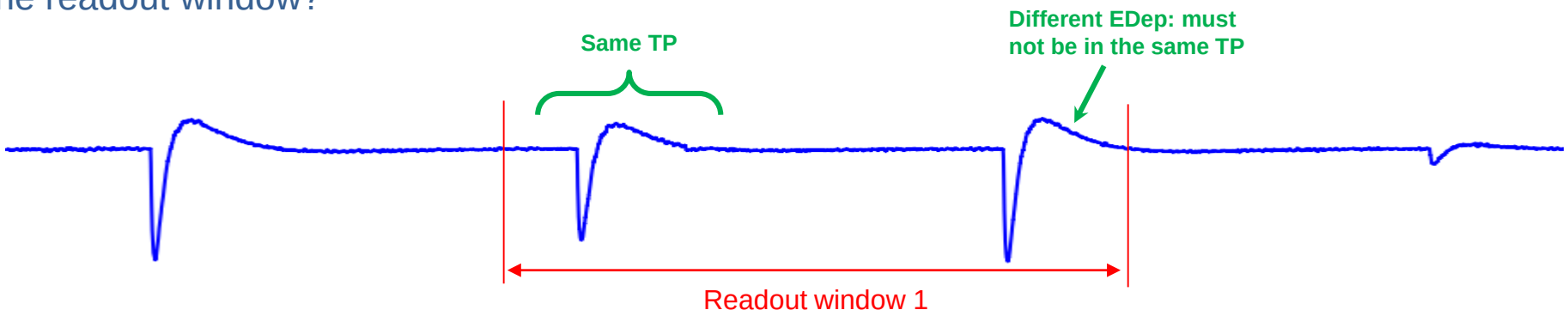
# Self-trigger & Primitive Calculation QUESTIONS (3)

Current DAPHNE Frame structure imposes **1 self-trigger = 1 trigger-primitive (TP)**

Merging fast and slow light components (“**peak detections**”) from the **same “energy deposition”** into the **same TP should be OK** (some info can be retained e.g. number of peaks).

Merging two **different energy depositions** into the same TP is **NOT OK**. For some pulses it will be unavoidable unless:

- Allow > 1 TP per self-trigger?
- Shorten the readout window?



# Thanks for your attention!

# Peak Detection - BACKUP

**Baseline calculation is based in calculating cumulative average.**

$$\bar{x} = \frac{\sum x_i}{n} \rightarrow \overline{x_{i+1}} = \bar{x}_i + \frac{x_{i+1} - \bar{x}_i}{n+1}$$

$$\overline{x_{i+1}} = \bar{x}_i + \frac{x_{i+1} - \bar{x}_i}{2^N}$$

| FILTERED DATA | BASELINE | AMPLITUDE | SOLPE | PEAK DETECTION |
|---|---|---|---|---|
| **Initial condition:** $F_0 = x_0$ **Algorithm:** $F_{i+1} = \frac{x_i + x_{i+1}}{2}$ | **Initial condition:** $B_0 = x_0$ **Algorithm:** **If** $Detection$ $B_{i+1} = B_i$ **Else** $B_{i+1} = B_i + \frac{F_{i+1} - B_i}{8}$ | **Initial condition:** $A_0 = 0$ **Algorithm:** $A_{i+1} = F_{i+1} - B_{i+1}$ | **Initial condition:** $S_0 = 0$ **Algorithm:** $S_{i+1} = A_{i+1} - A_i$ | **Initial condition:** $P_0 = false$ **Algorithm:** **If** $S_{i+1} < -10$ $P_0 = false$ **Else** $P_0 = true$ |

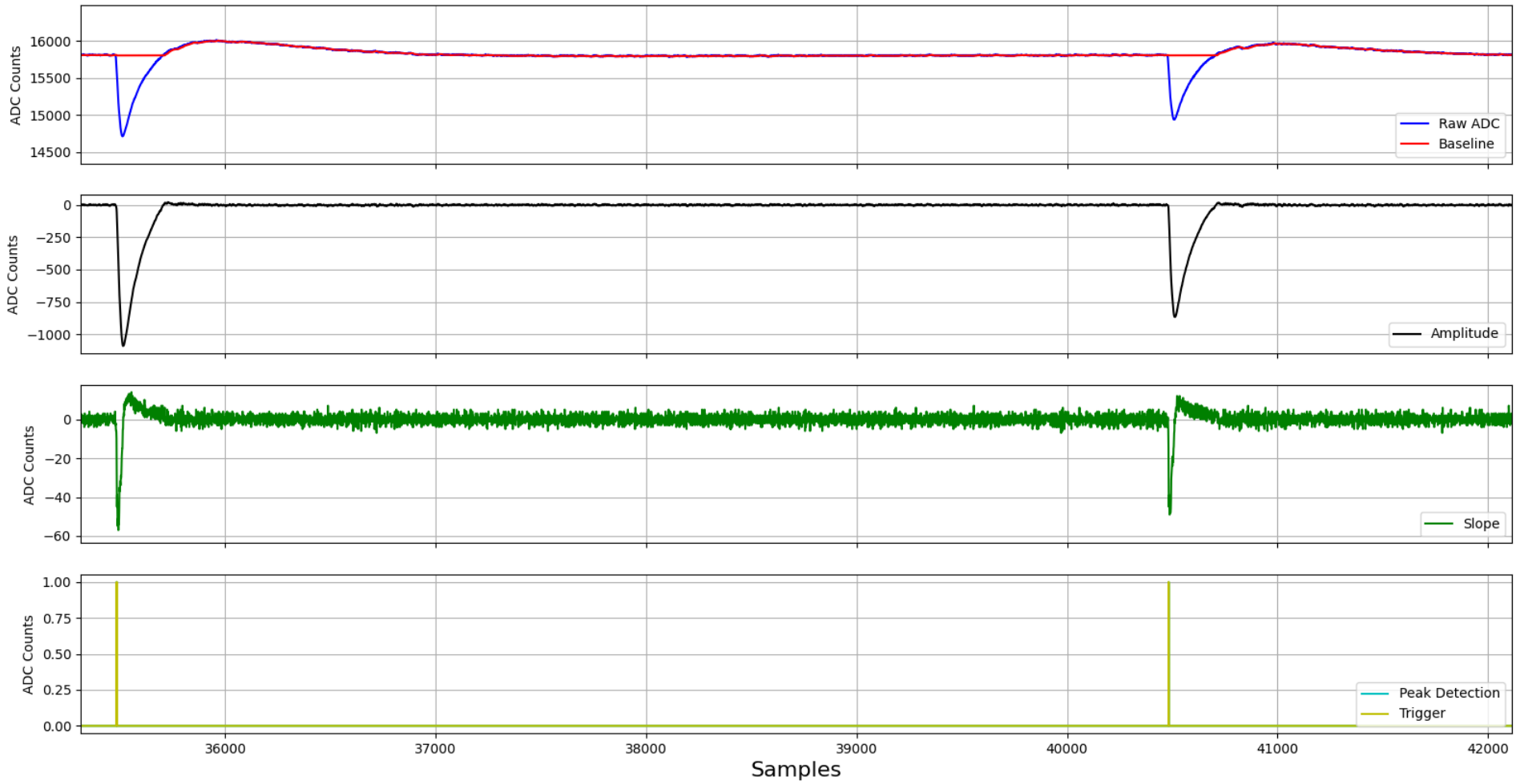
# Waveform's Primitive Calculation- BACKUP

While **DETECTION *State***

| PULSE WITH | TIME TO PEAK | MAX AMPLITUDE | CHARGE | NUMBER OF PEAKS |
|---|---|---|---|---|
| **Initial condition:** $W_0 = 0$ **Algorithm:** $W_{i+1} = W_i + 1$ | **Initial condition:** $TP_0 = 0$ **Algorithm:** **If** $Amplitude_{i+1} < MA_i$ $T_{i+1} = W_{i+1}$ **Else** $T_{i+1} = T_i$ | **Initial condition:** $MA_0 = 0$ **Algorithm:** **If** $Amplitude_{i+1} < MA_i$ $MA_{i+1} = Amplitude_{i+1}$ **Else** $MA_{i+1} = MA_i$ | **Initial condition:** $C_0 = 0$ **Algorithm:** $C_{i+1} = C_i + Amplitude_{i+1}$ | **Initial condition:** $NP_0 = 0$ **Algorithm:** **If** $Peak\_Detection$ $NP_{i+1} = NP_i + 1$ **Else** $NP_{i+1} = NP_i$ |

# Self-trigger & Primitive Calculation ALGORITHM (7)

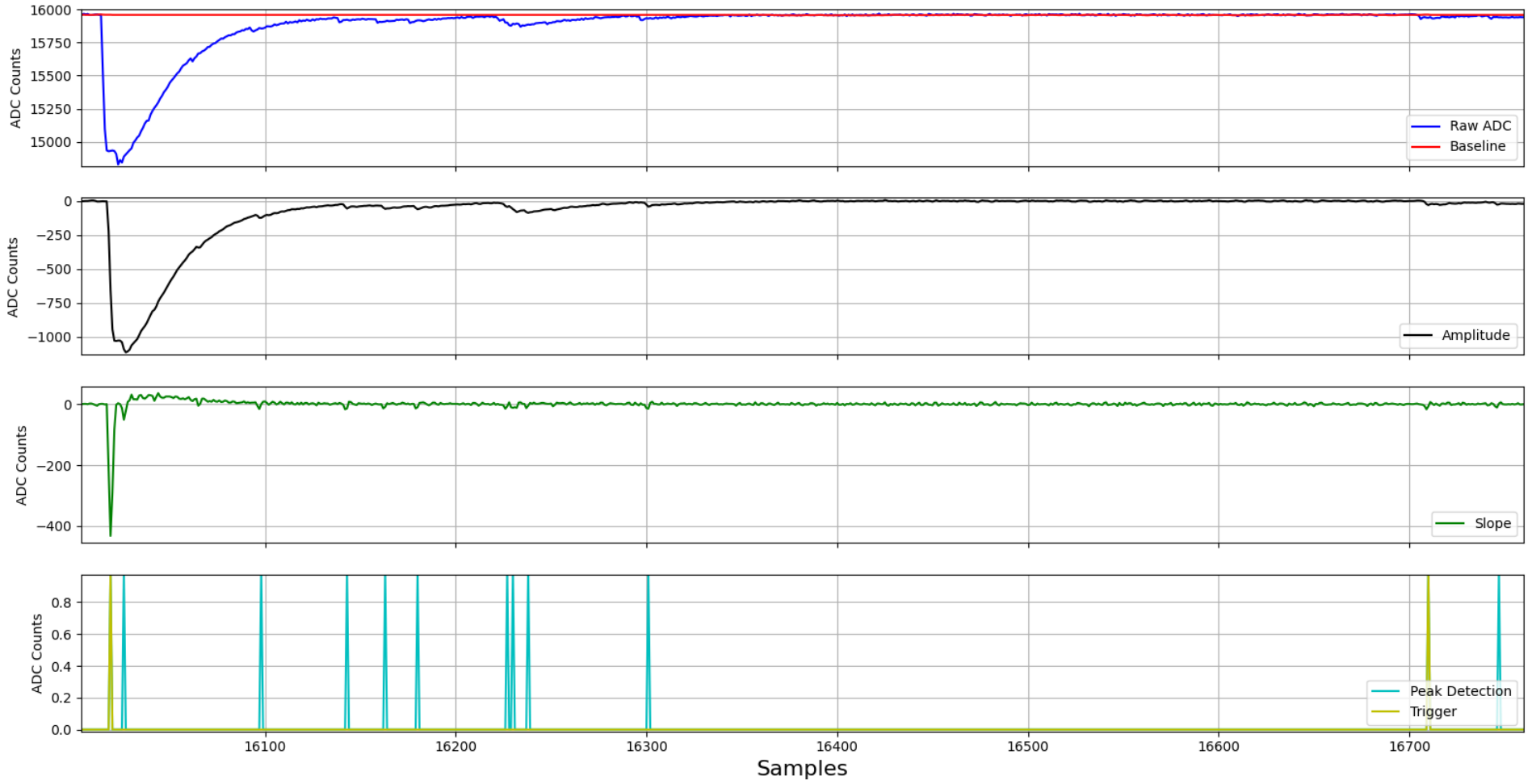
## Post-synthesis timing simulation → X-Arapuca (SuperCell)




```
---------- DETECTED PULSE  8 ----------
Time to Peak      =  30
Pulse Width       =  222
Max Amplitude     =  -1093
Charge            =  -102784
Number of Peaks   =  1
---------- DETECTED PULSE  9 ----------
Time to Peak      =  30
Pulse Width       =  226
Max Amplitude     =  -868
Charge            =  -80940
Number of Peaks   =  1
```

# Self-trigger & Primitive Calculation ALGORITHM (6)

## Post-synthesis timing simulation → SiPM




```
---------- DETECTED PULSE  4 ----------
Time to Peak      =  8
Pulse Width       =  343
Max Amplitude     =  -1114
Charge            =  -50108
Number of Peaks   =  10
---------- DETECTED PULSE  5 ----------
Time to Peak      =  6
Pulse Width       =  97
Max Amplitude     =  -27
Charge            =  -1164
Number of Peaks   =  2
```