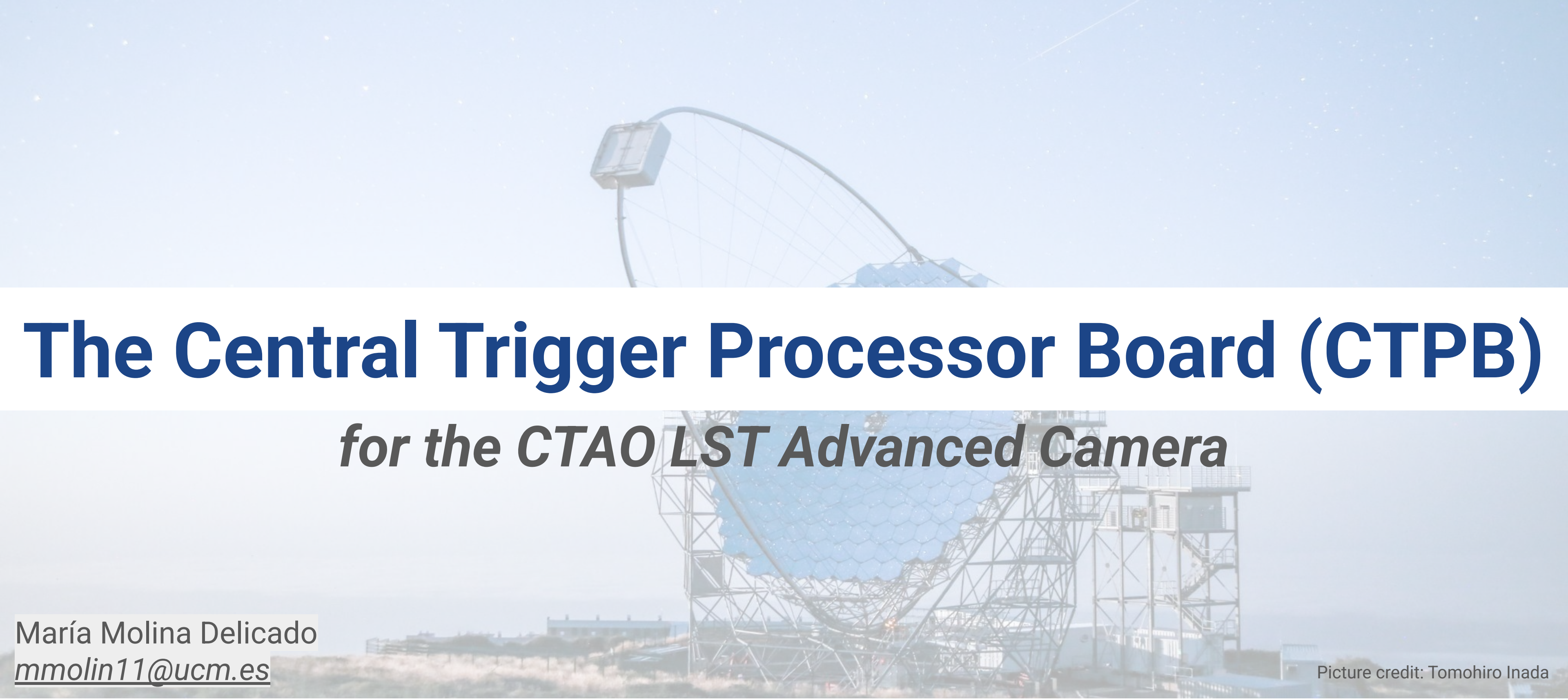
The Central Trigger Processor Board (CTPB)

for the CTAO LST Advanced Camera

María Molina Delicado
mmolin11@ucm.es

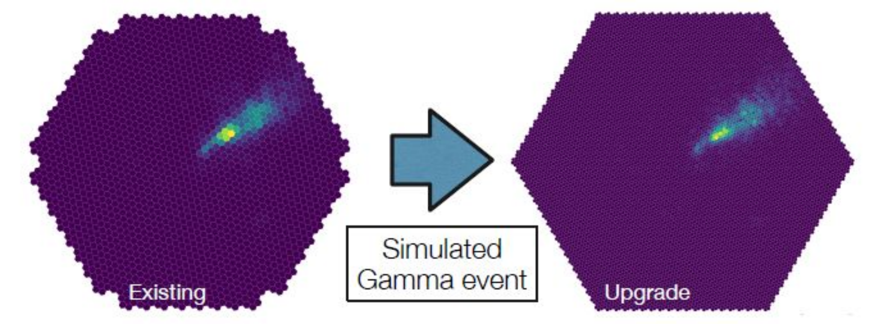


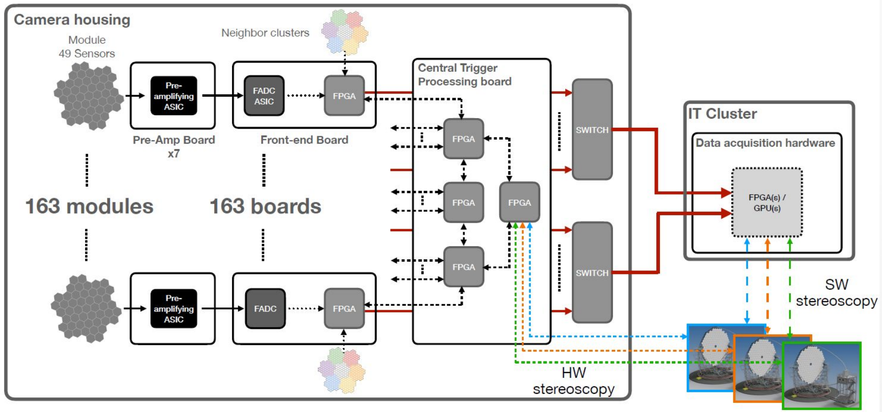
Picture credit: Tomohiro Inada

Contents

The LST Advanced Camera

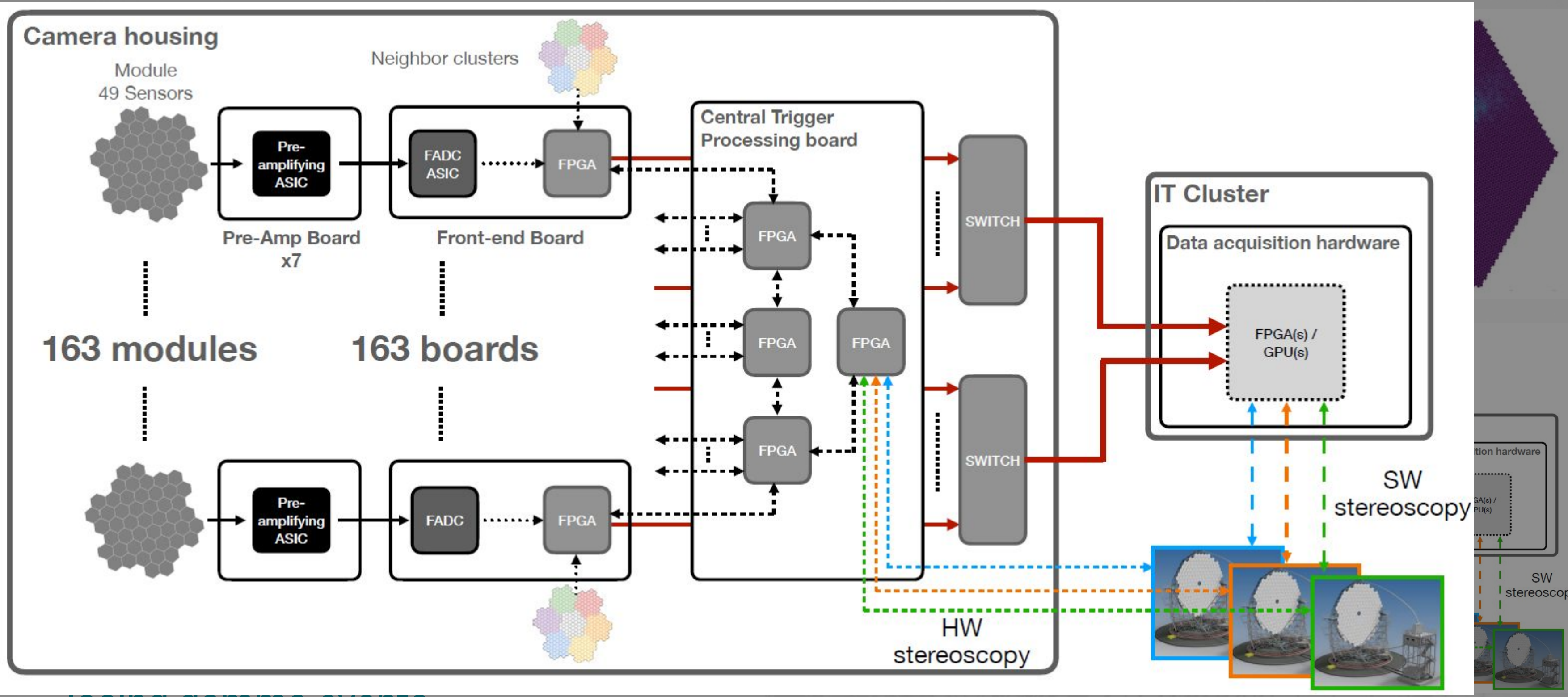
- ➔ Transition from photomultiplier tubes (PMTs) to silicon photomultipliers (SiPMs)
- ➔ Higher granularity
  - ◆ Smaller pixels increase 4x resolution
- ➔ Fully digital architecture
  - ◆ Replacing analog memories with a continuous digital readout
- ➔ The data challenge
  - ◆ Input rate: 1 GHz (1 frame/ns)
  - ◆ DAQ limit: ~40 kHz
  - ◆ **Goal:** Maximize NSB rejection without losing gamma events





Images credit: M. Heller (UniGe)

The LST Advanced Camera



losing gamma events

Images credit: M. Heller (UniGe)

The Central Trigger Processor Board (CTPB)

→ Input stage

- Receives L1 data @ 1GHz
- Handles data deserialization

→ Processing stage

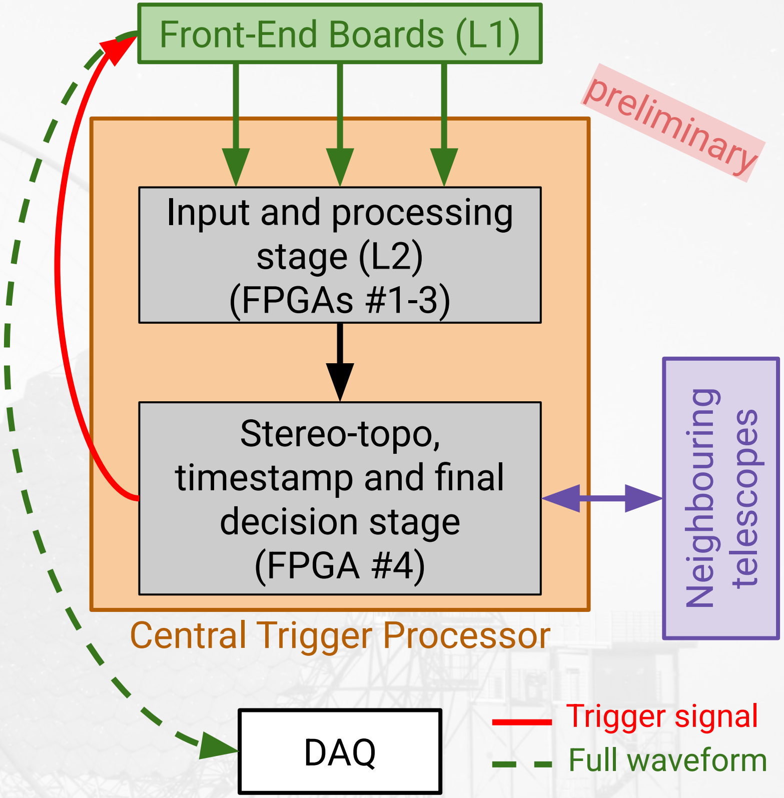
- Executes L2 algorithm in real-time
- First rate reduction step to ~2 MHz

→ Stereo-topological trigger stage

- Searches for spatio-temporal coincidences with neighboring telescopes
- Reduces mono rate to ~40 kHz

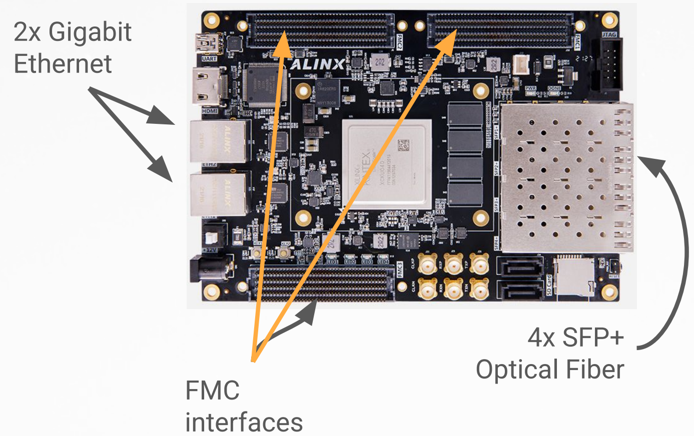
→ Final decision and readout

- Event timestamping
- Trigger signal is sent to the front-end to initiate DAQ readout



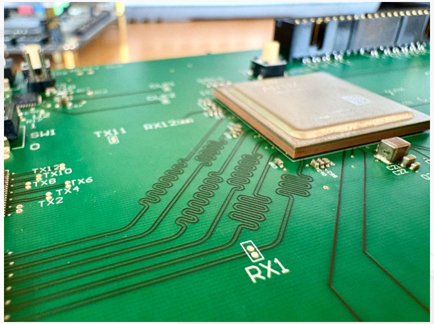
CTP test benches

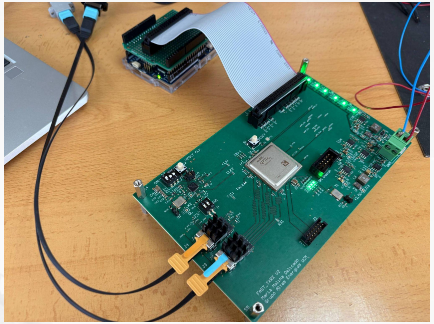
#1 Machine Learning @ FPGAs



- 2x ALINX AMD Xilinx Kintex UltraScale XCKU040
  - 20 gigabit transceivers @ 16.3 Gbps
  - 4GB high-speed DDR4 RAM
- Data transfer between PC and FPGA using IPBus protocol

#2 High-speed lines testing





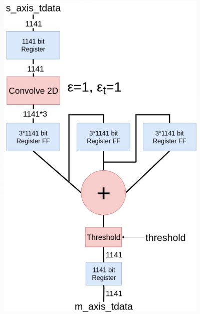
- Main components
  - Xilinx UltraScale+ with 12 gigabit transceivers
  - Two 12-channel Samtec FireFly optical connectors (TX and RX)
- Design and manufacturing
  - 12 layer PCB design and high-speed differential pair routing
  - Validation of the substrate and the PCB manufacturer

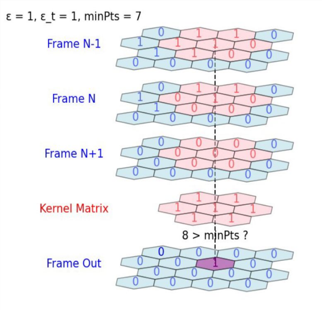
L2 algorithms: Real-time pattern recognition

Approach A: Clustering (TDSCAN)

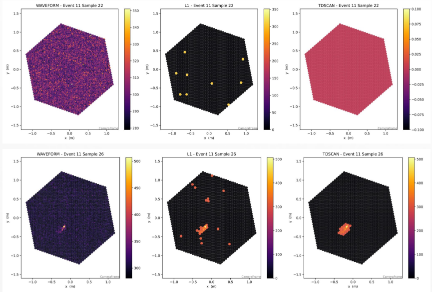
Parallel 2D+1 convolution over the whole camera, as a DBSCAN simplification

credit: T. Dietrich (HESGE-HEPIA)





credit: J. Buces (UCM-GAE)

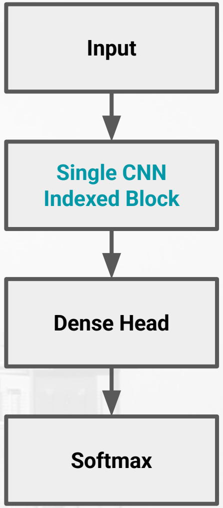


- ➔ Achieves 1 GHz processing rate being hardware-friendly
- ➔ High noise rejection but lower sensitivity than other algorithms

Approach B: Machine Learning (CNNs)

High Level Synthesis (HLS) implementation of CNN models based on L1 data

- ➔ HLS enables the hardware implementation of complex algorithms using high-level languages (C/C++)
- ➔ Small architectures (3.6k parameters) that include *custom layers* (cannot use HLS4ML)
- ➔ Two firmware implementations
  - ◆ XCKU040 (our test bench)
  - ◆ XCKU115 (CTPB candidate)



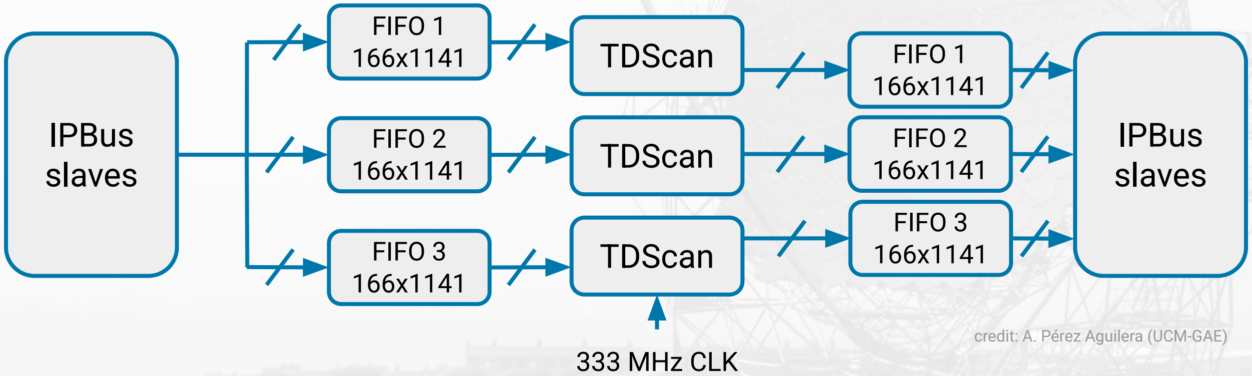
TDSCAN @ 1 GHz: firmware architecture

1 GHz effective rate

- Achieved implementing three parallel pipelines clocked at 333 MHz
- Events are split using a cyclic Round-Robin scheme per channel
  - e.g., events 1, 4, 7, … to FIFO1, events 2, 5, 8, … to FIFO2, etc
- Latency tests were performed using `s_axis_tvalid` and `m_axis_tvalid` signals
- Proper operation was verified using simulated events generated on Corsika-Simtel (J. Buces)

Data management

- Host handles pre-ordering and re-assembling to simplify hardware logic and minimize latency



credit: A. Pérez Aguilera (UCM-GAE)

From Python to Vitis (C++): CNN hardware implementation

➔ Reconstruction of the original CNN architecture

- Extraction of layer parameters and development of C++ core functions and wrappers

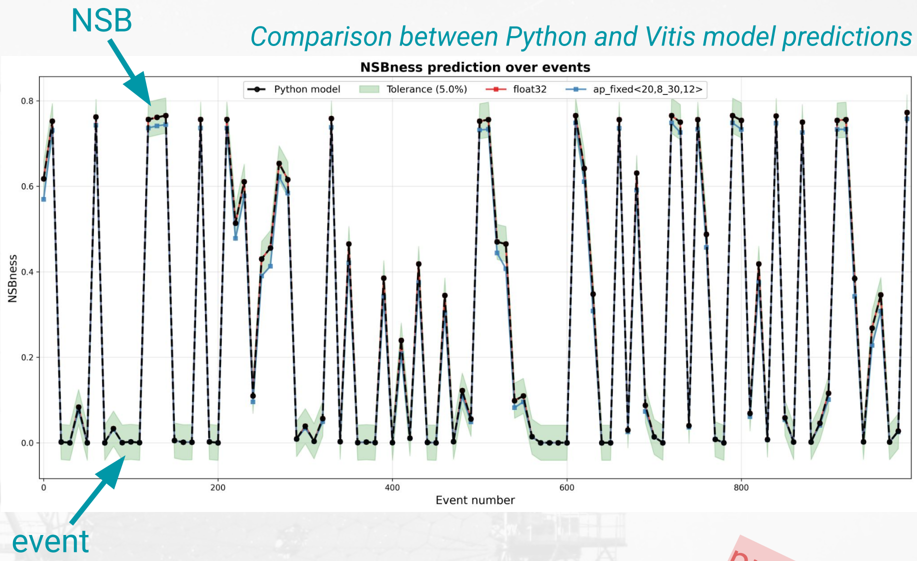
➔ Optimization for hardware

- Batch-normalization folding (hw-friendly)
- Data quantization from `float32` to `fixed point` reduces DSP usage, fitting the model into the FPGA while maintaining the performance

➔ HLS optimization #pragmas

- In order to maximize parallelism and reduce latency/interval
- Trade-off between parallelism and resources

Comparison between Python and Vitis model predictions



Synthesis results (XCKU115)

preliminary

- ➔ Binary model (1141,10,1 shape)
  - New inference each ~1.7 µs
- ➔ Quantised model (163,10,2 shape)
  - New inference each ~525 ns

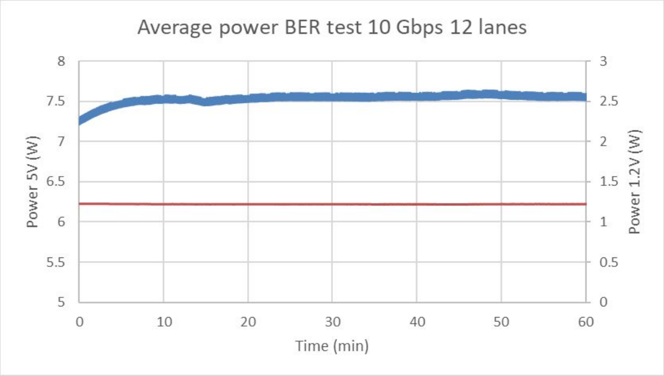
High-speed links validation

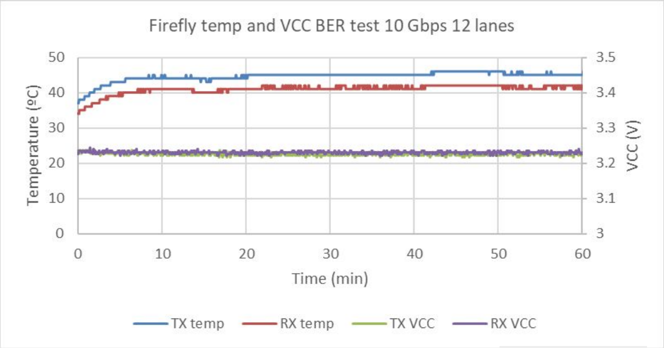
➔ Hardware description

- FireFly Optical modules (TX and RX)
- High density: 12 optical lanes in a single compact module with bandwidth up to 14 Gbps
- Slow control firmware: Full real-time monitoring of temperature and voltage via I2C to ensure stability

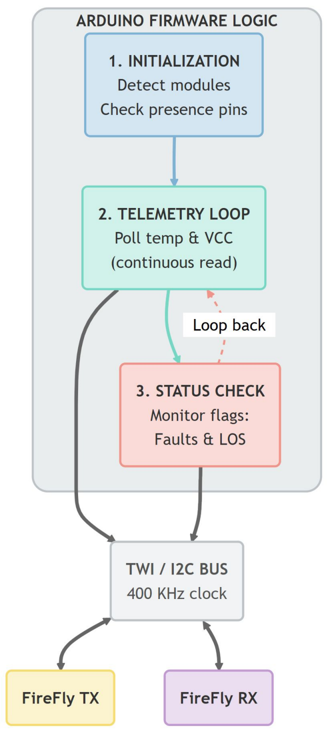
➔ Performance verification

- BER analysis: Tests performed at 2, 6, and 10 Gbps
- Results: Error-free transmission detected during 1-hour runs with stable temperature and flat power consumption (BER < $2.63 \cdot 10^{-14}$)





BER test monitorization



FireFly Slow Control firmware

Summary and outlook

- Preliminary architecture defined for the Central Trigger Processor of the LST Advanced Camera (performing L2 & stereo-topo)
- Ongoing evaluation of FPGA-based trigger algorithms, ranging from clustering (TDSCAN) to Machine Learning approaches (CNNs)
- Successful tests of high-speed optical links and FireFly modules, achieving stable transmission with BER $< 10^{-13}$ at 10 Gbps

Next steps

- Protocol benchmarking: comparative analysis of protocol candidates for the communication between Front-End Boards ↔ CTPB FPGAs to minimize latency
- Algorithm selection: final decision on the L2 trigger logic based on resource usage and discrimination performance
- Prototype definition: consolidating test bench results to define the specifications and architecture for the first CTPB prototype

Acknowledgements

The research here presented has been partially supported by the MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, under grant PDC2023-145839-I00