MOCAST 2023

Data Preparation and Optimization for Real Time Track Reconstruction

Konstantinos Axiotis
Université de Genève

MOCAST
29/06/2023

Outline

- High Luminosity Large Hadron Collider (LHC) Challenges
- Phase II Trigger and Data AcQuisition (TDAQ) Upgrade
- Hardware Track for the Trigger
    - Pattern Recognition Mezzanine
    - Firmware blocks that I developed
- Hardware Tests

High Luminosity LHC Challenges

High Luminosity consequences

- High pile-up up to 200 events per bunch crossing (previously ~ <40>)
- High granularity detectors that need to be read out
  - New, all-silicon Inner Tracker (ITk)
  - front-end/back-end electronics updates
- Larger event size ~5.2 MB (~2 MB previously)

Operating points for ATLAS TDAQ

- L1 latency increase to ~ 10 μs (~2.5 μs previously)
- Readout rate increase to 1-4 MHz (100 kHz previously)
- Rate to permanent storage ~ 10 kHz (~ 1 kHz previously)

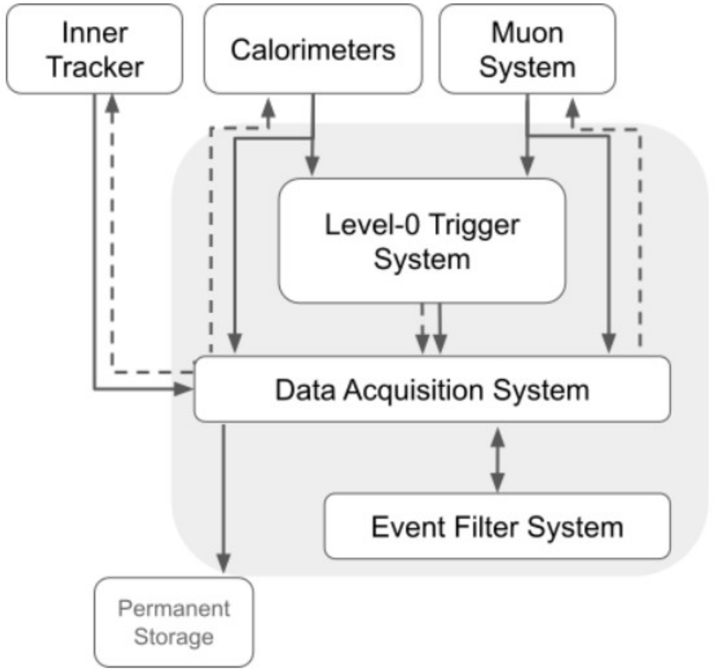
ATLAS Phase-II TDAQ upgrade

TDAQ Phase II architecture

Single level hardware trigger: Level 0

- Hardware Track Trigger (HTT) → Reduce backgrounds after L0 trigger selections and mitigate pile-up
  - Custom designed boards
  - **Regional tracking (rHTT)** at 1 MHz
    - on up to 10% of the ITk data, Pt > 2 GeV
  - **Global tracking (gHTT)** at 100 kHz
    - Full coverage Pt > 1 GeV
- Data are provided by the Event Filter processing farm and tracks are returned to it

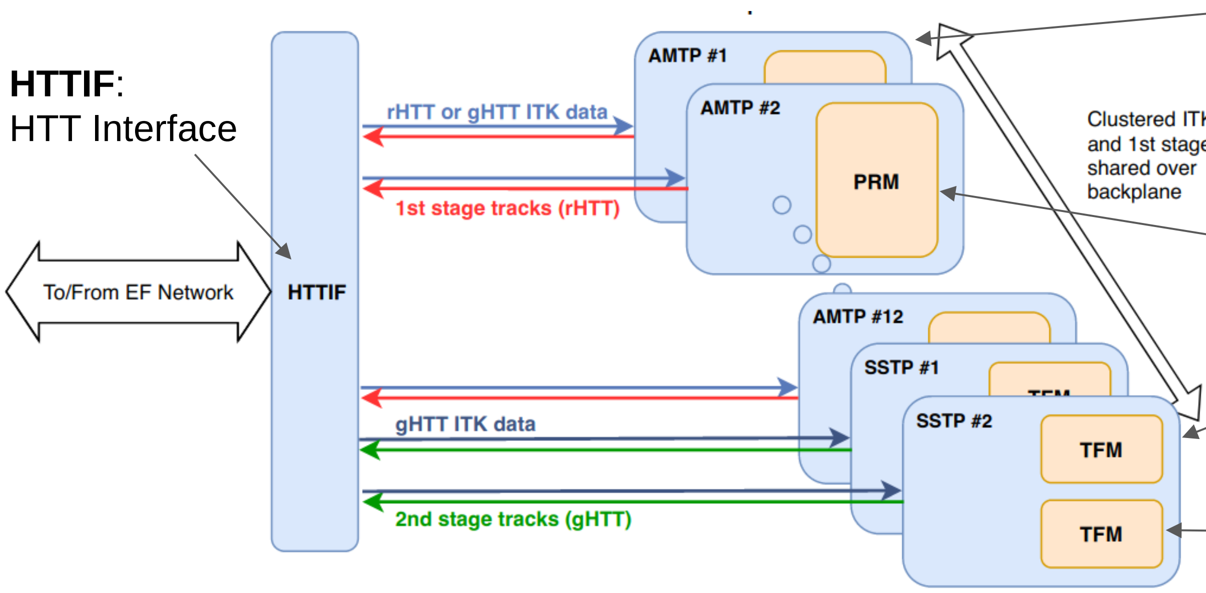
HTT evolved into commodity based system, so no longer happening, but the experience with the demonstrator quite unique!



Intel Stratix 10 FPGA design for track reconstruction for the ATLAS experiment at the HL-LHC

Hardware Tracking for the Trigger (HTT)

HTTIF:
HTT Interface



AMTP: Associative memory Tracking Processor

PRM: Pattern Recognition Mezzanine (Intel Stratix 10 FPGA)

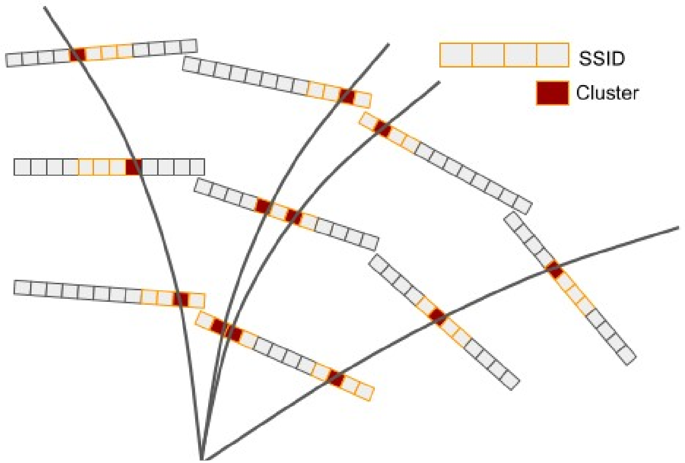
SSTP: Second-Stage Tracking Processor

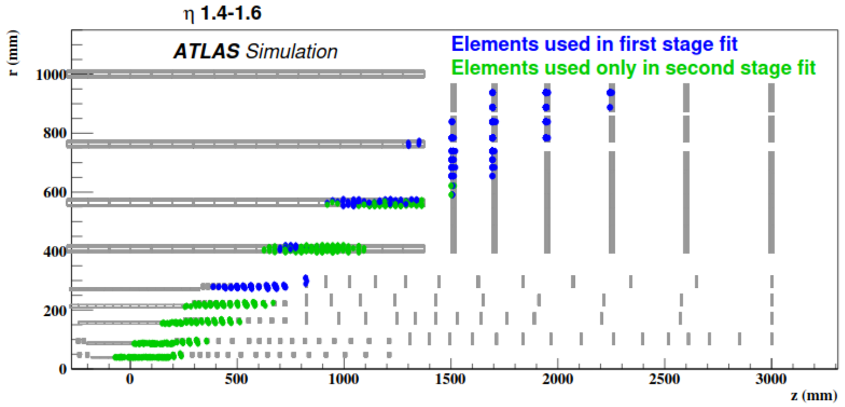
TFM: Track-Fitting Mezzanine

HTT Track reconstruction

Stage 1

- The **clustered** data are grouped into consecutive strip or pixel channels (so-called “**superstrips**”)
- For each track candidate that matches a pattern, the track χ^2 and **helix parameters** are computed from the corresponding clustered hits in an FPGA
- Transfer back to the processing unit





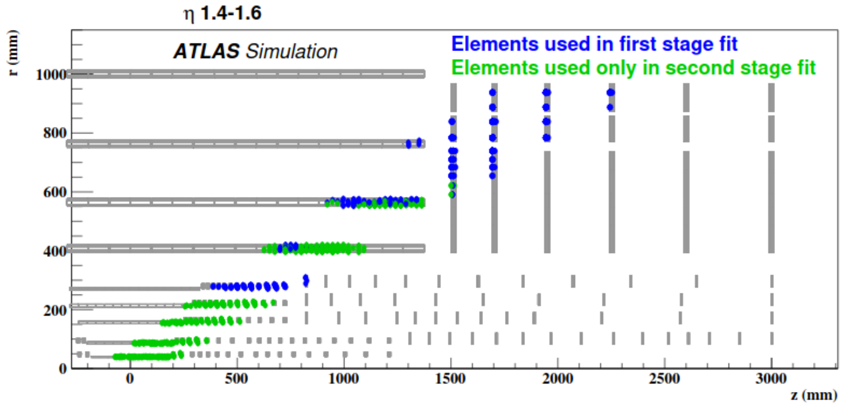
HTT Track reconstruction

Stage 1

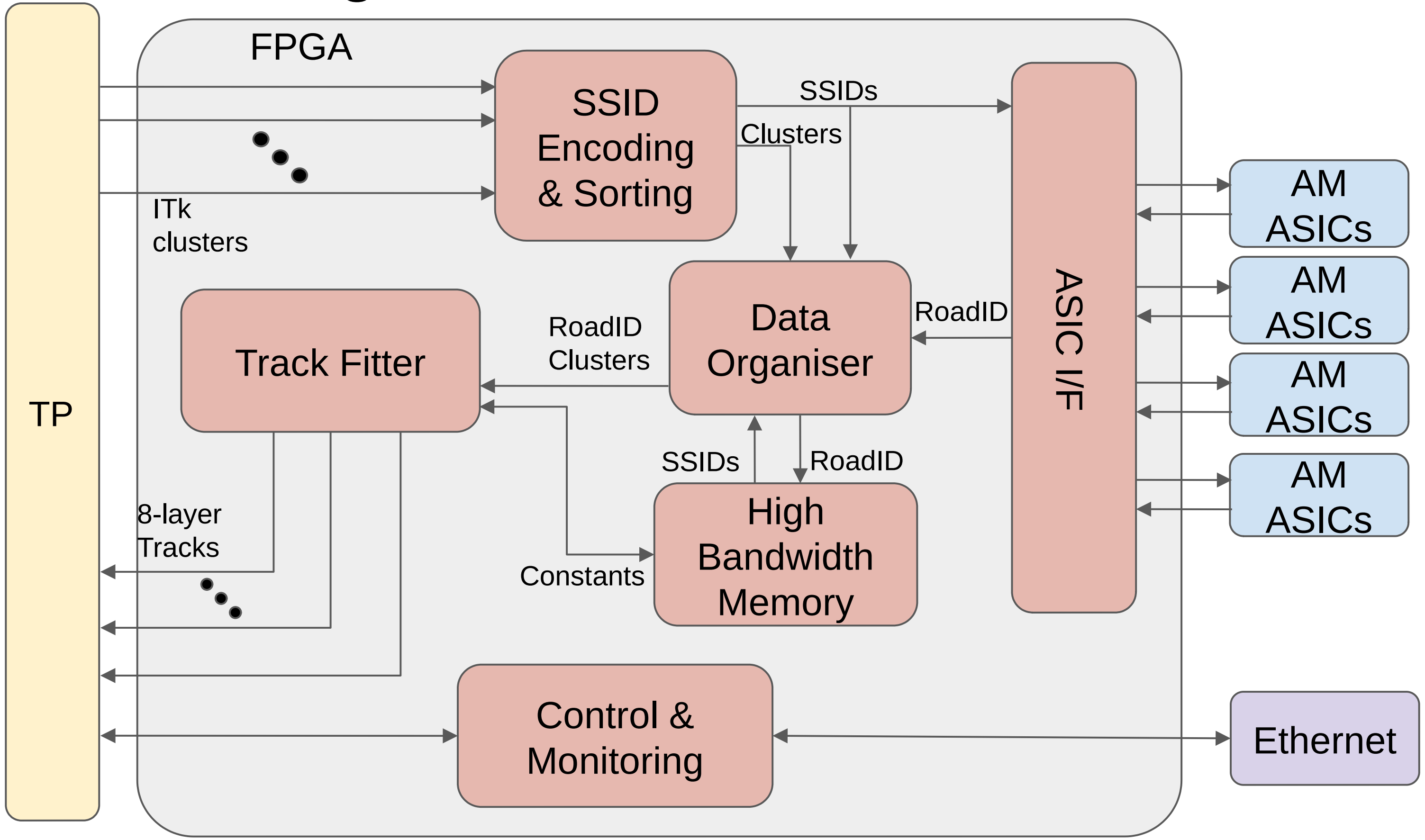
- The **clustered** data are grouped into consecutive strip or pixel channels (so-called "**superstrips**")
- For each track candidate that matches a pattern, the track χ^2 and **helix parameters** are computed from the corresponding clustered hits in an FPGA
- Transfer back to the processing unit

Stage 2

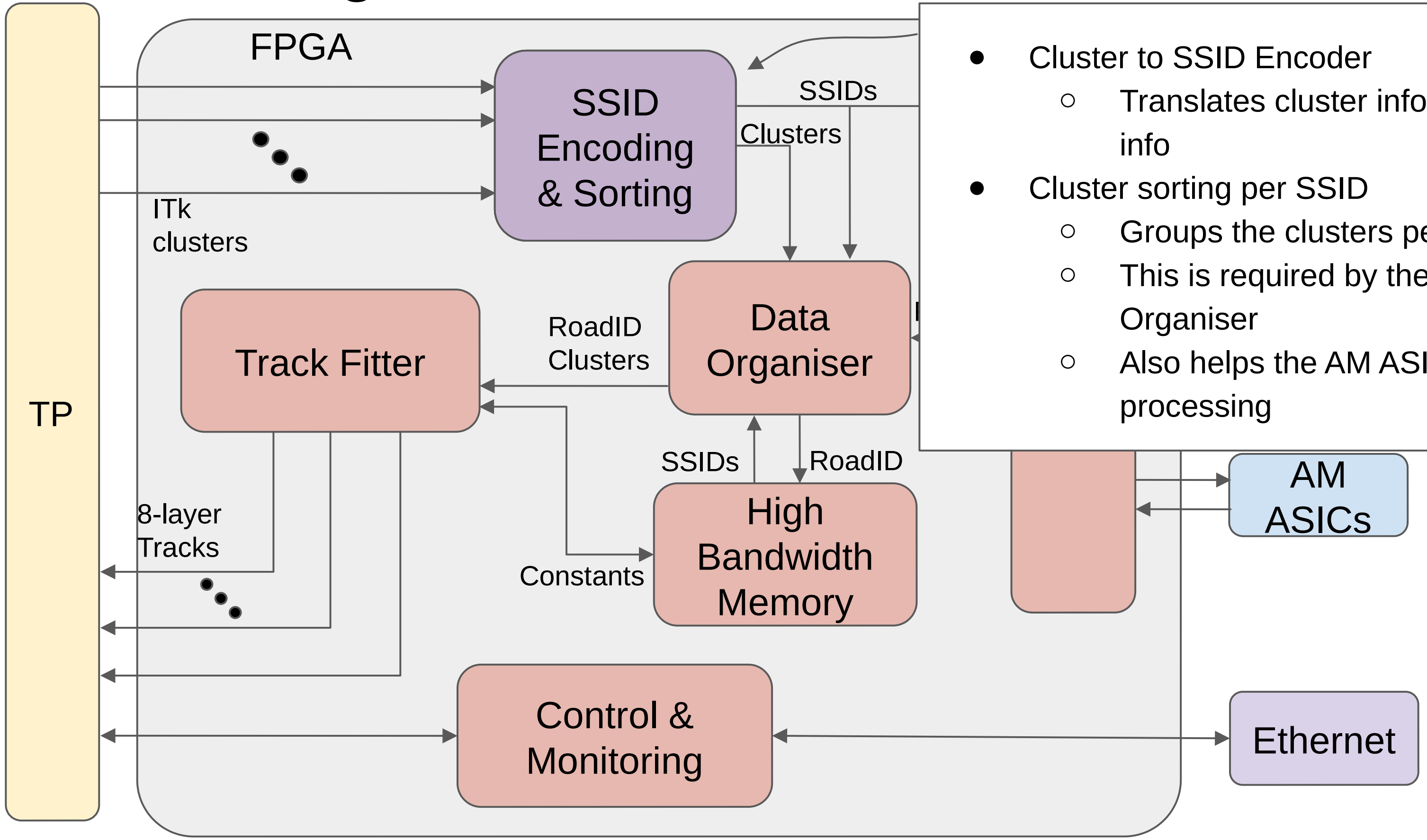
- Each 8-layer track candidate goes through a second stage of processing
- χ^2 and **helix parameters** $\rightarrow$ 5 remaining ITk layers



Pattern Recognition Mezzanine Firmware

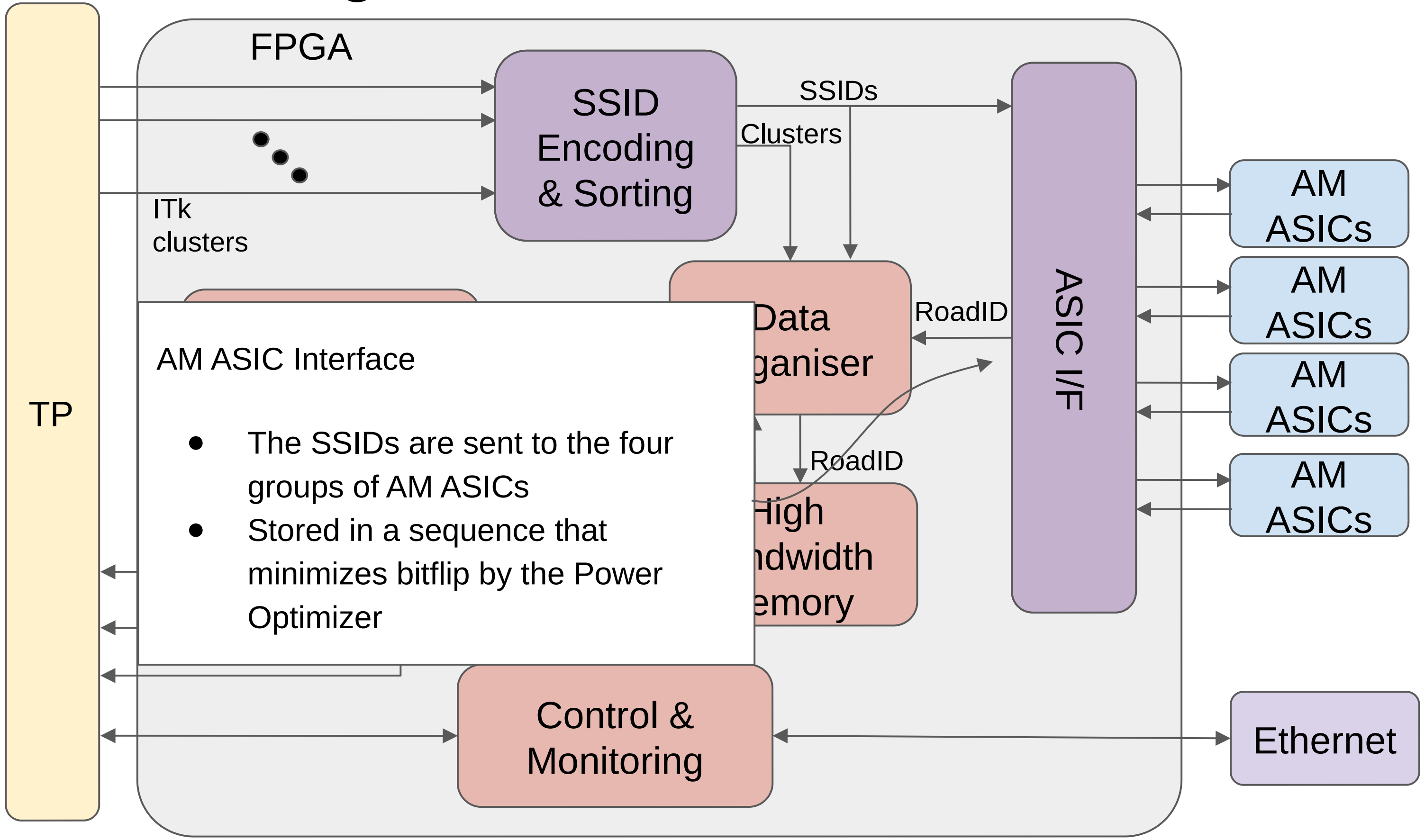


Pattern Recognition Mezzanine Firmware



- Cluster to SSID Encoder
  - Translates cluster info to SSID info
- Cluster sorting per SSID
  - Groups the clusters per SSID
  - This is required by the Data Organiser
  - Also helps the AM ASIC processing

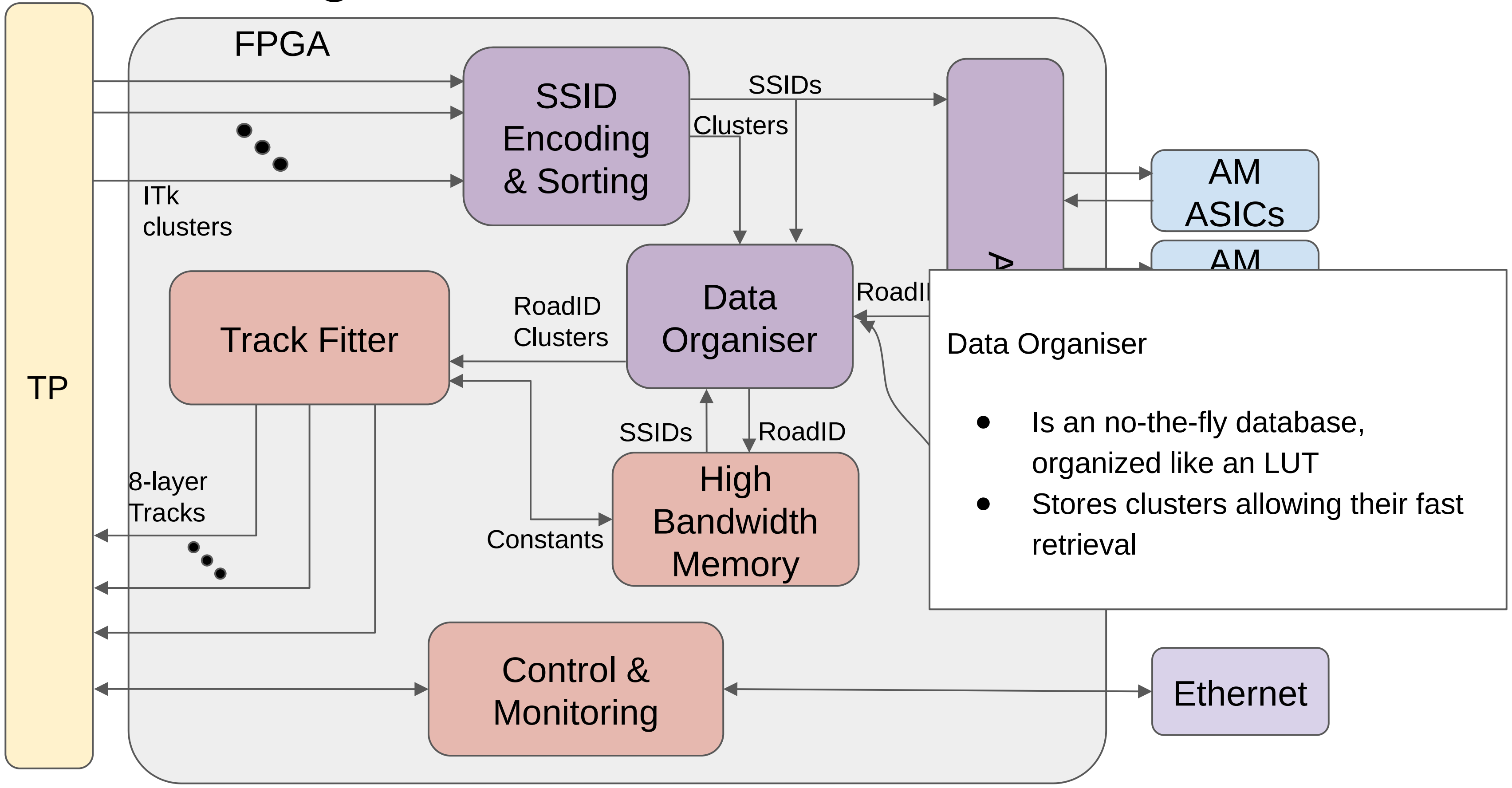
Pattern Recognition Mezzanine Firmware



AM ASIC Interface

- The SSIDs are sent to the four groups of AM ASICs
- Stored in a sequence that minimizes bitflip by the Power Optimizer

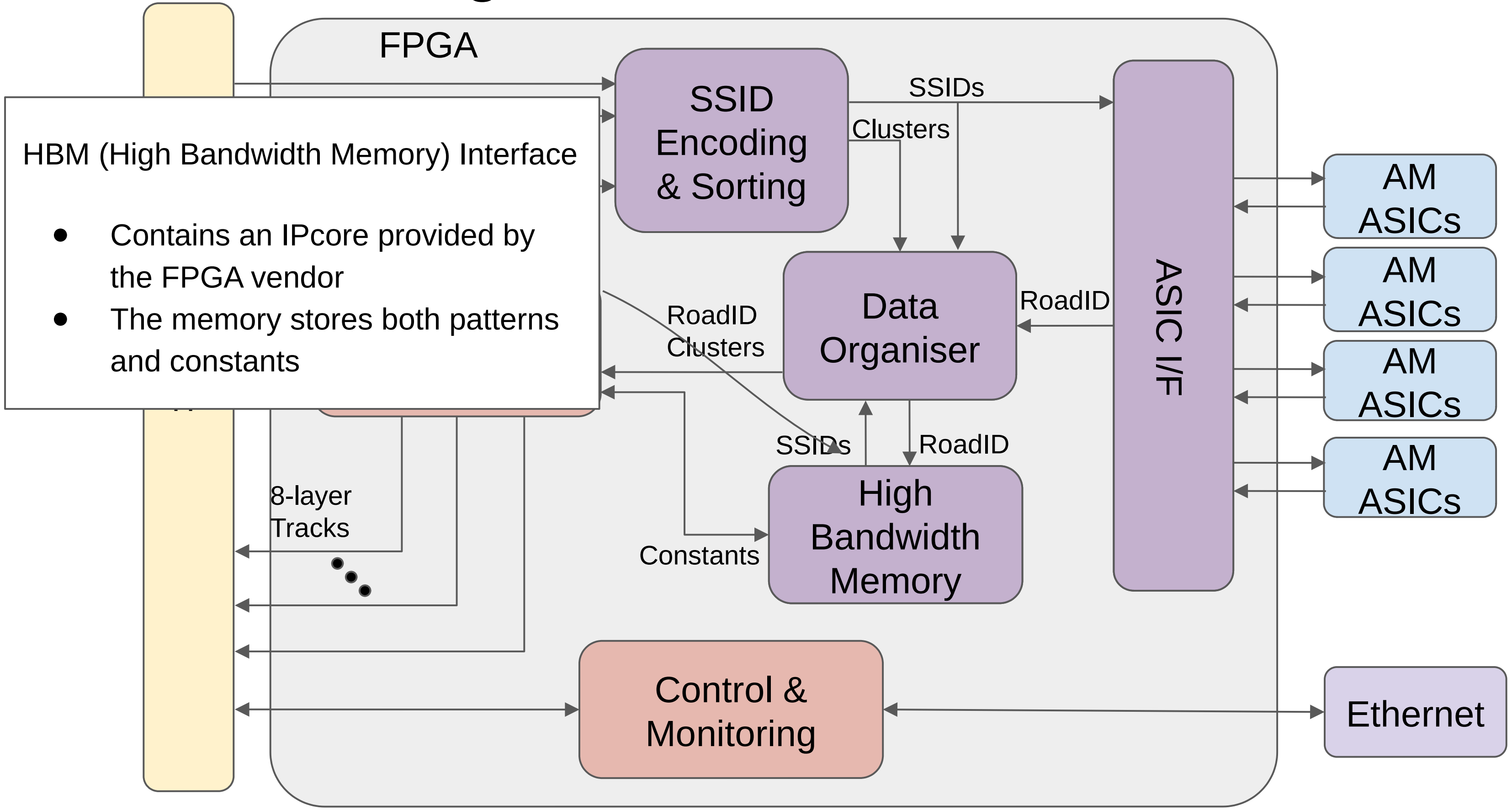
Pattern Recognition Mezzanine Firmware



Data Organiser

- Is an no-the-fly database, organized like an LUT
- Stores clusters allowing their fast retrieval

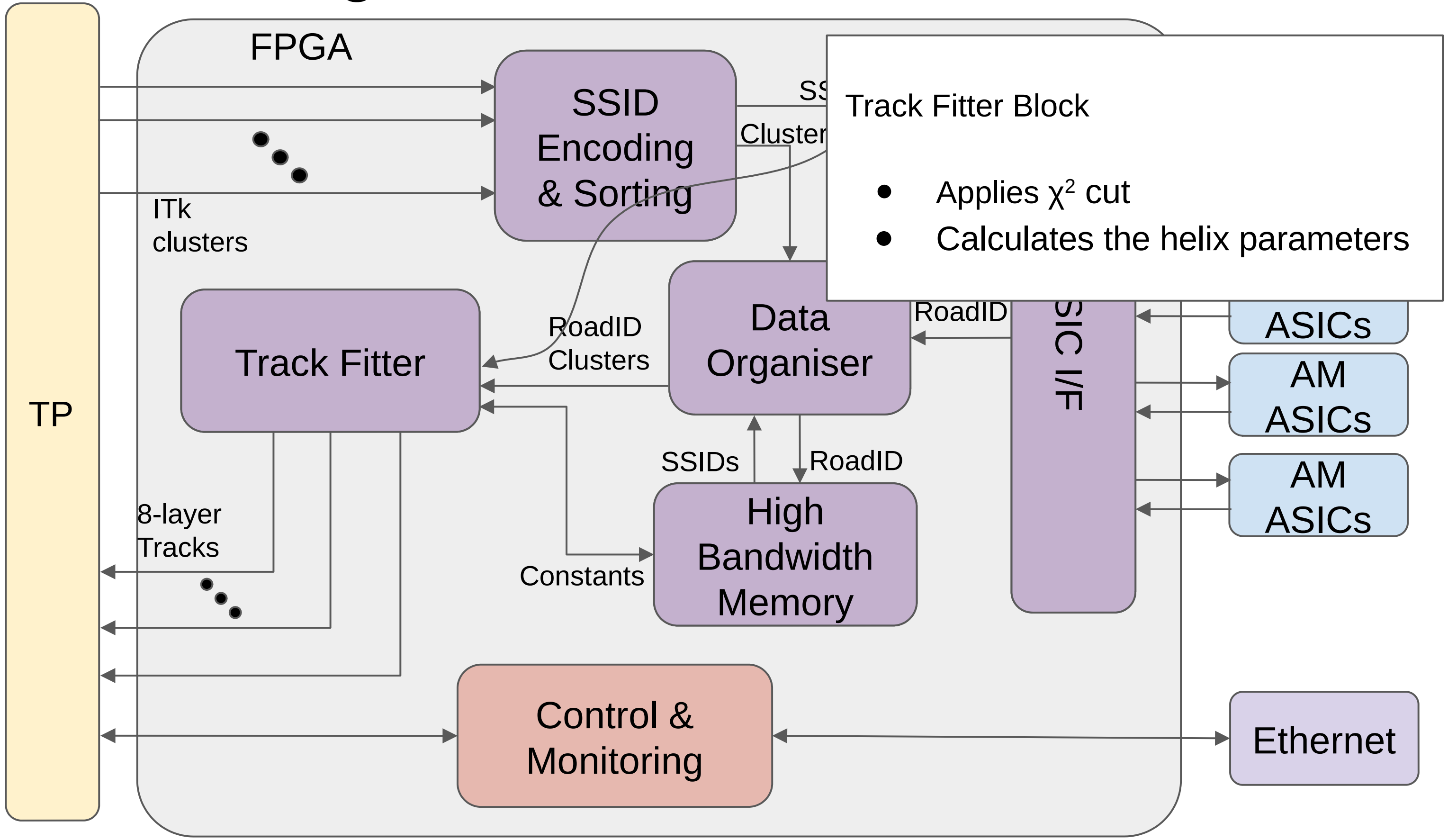
Pattern Recognition Mezzanine Firmware



HBM (High Bandwidth Memory) Interface

- Contains an IPcore provided by the FPGA vendor
- The memory stores both patterns and constants

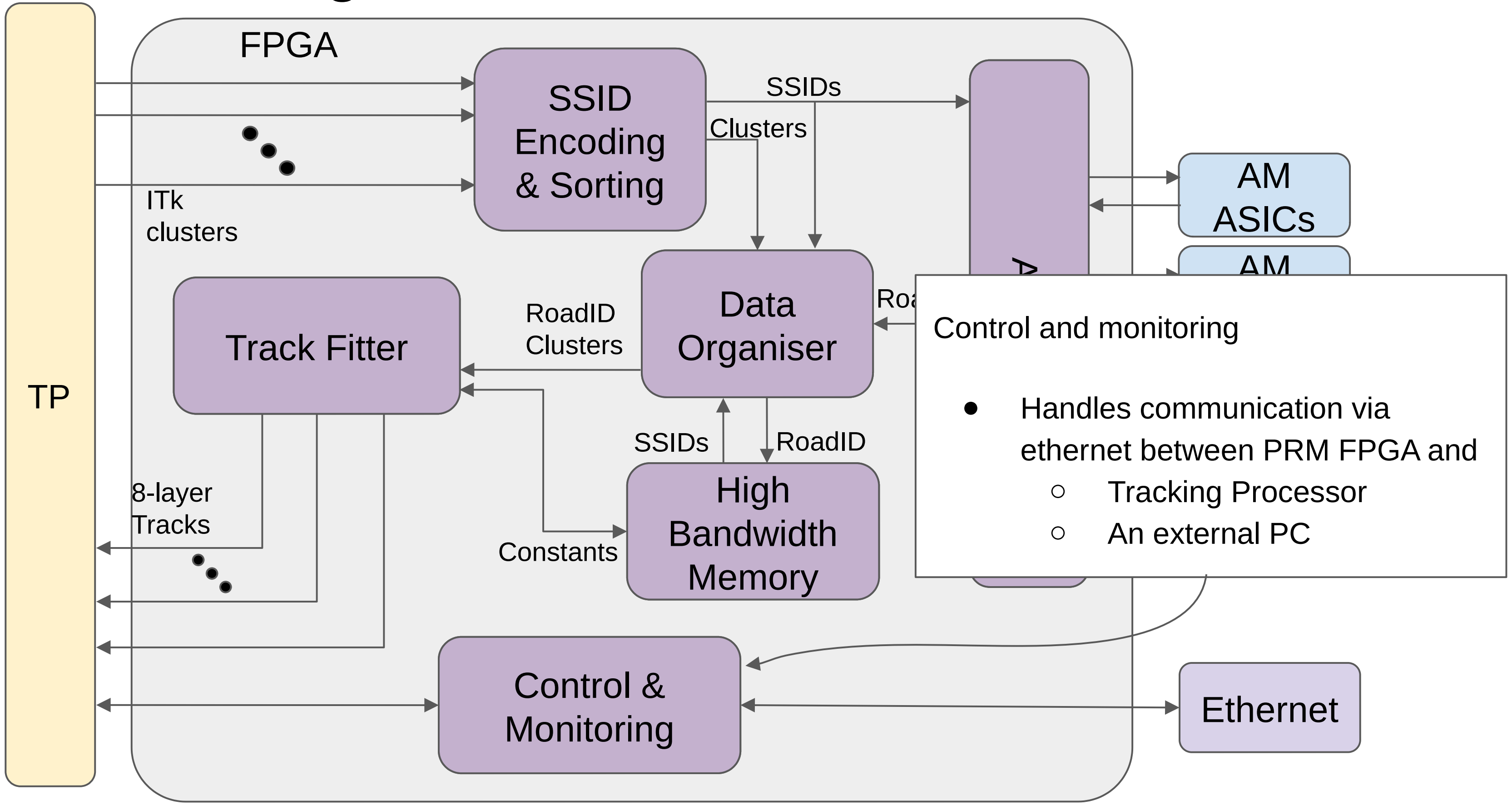
Pattern Recognition Mezzanine Firmware



Track Fitter Block

- Applies χ^2 cut
- Calculates the helix parameters

Pattern Recognition Mezzanine Firmware



Control and monitoring

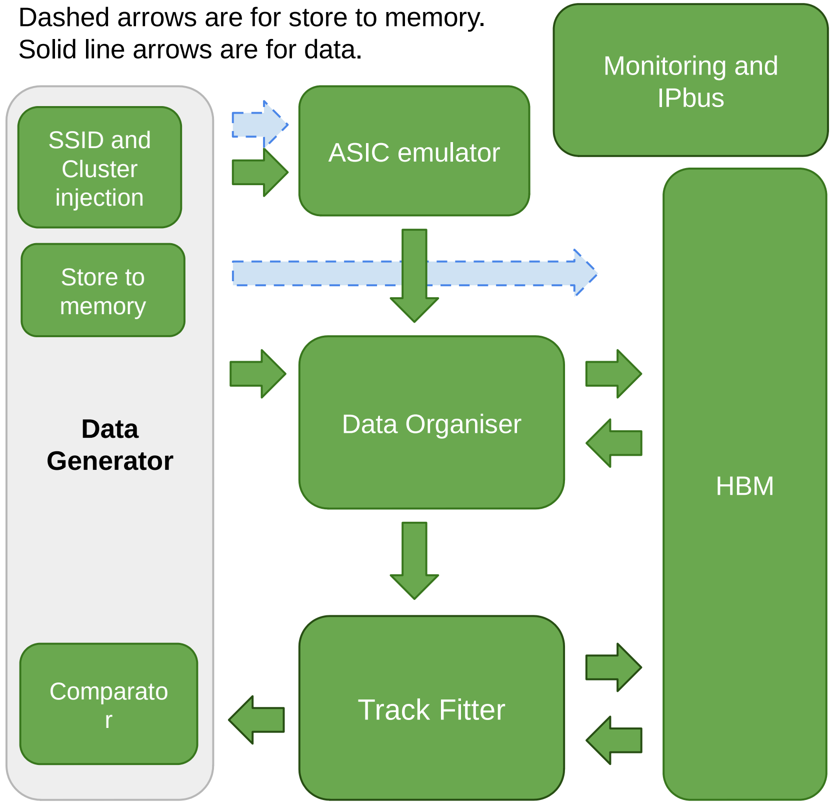
- Handles communication via ethernet between PRM FPGA and
  - Tracking Processor
  - An external PC

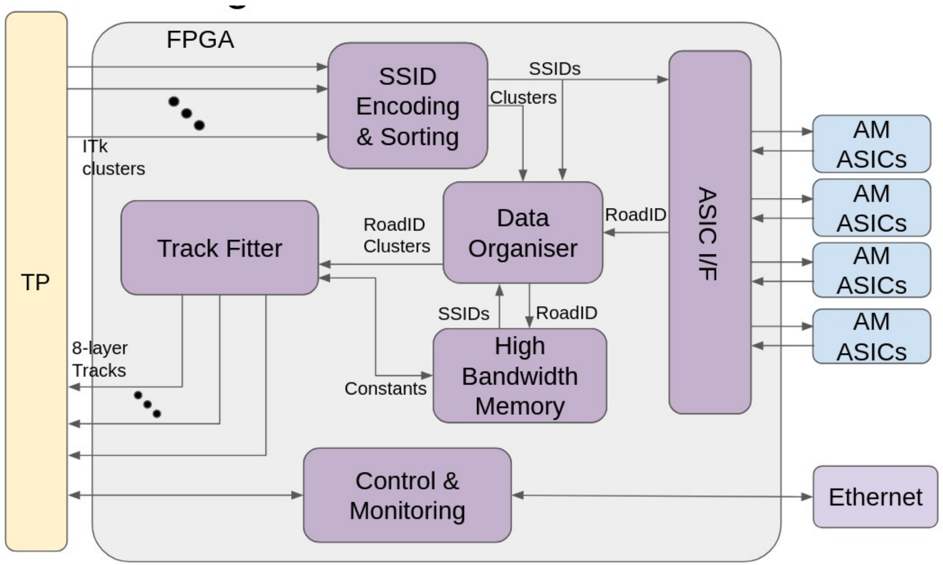
PRM Standalone Project

- Test the functionality for the PRM FW without the need of having the TP card
- Validate the PRM firmware before the integration stage
- Upload and store in the FW the Test Vectors (monte carlo data by the offline software team)
    - Input data →**Clusters, SSIDs** and **Constants**
    - Output data → **Chi2 and Track Parameters**

PRM Standalone Project

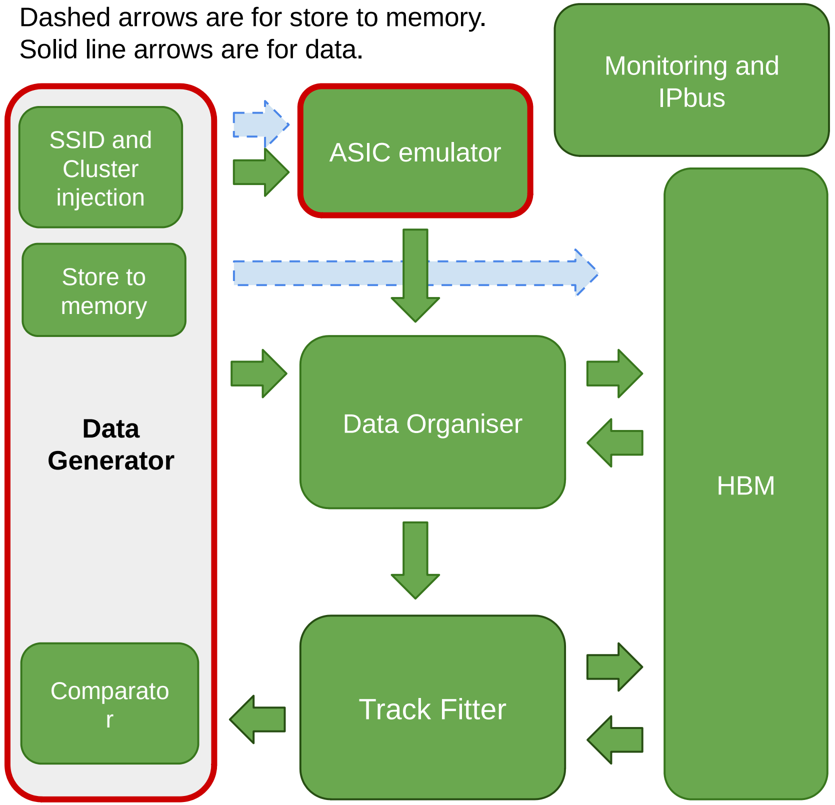
Dashed arrows are for store to memory.
Solid line arrows are for data.

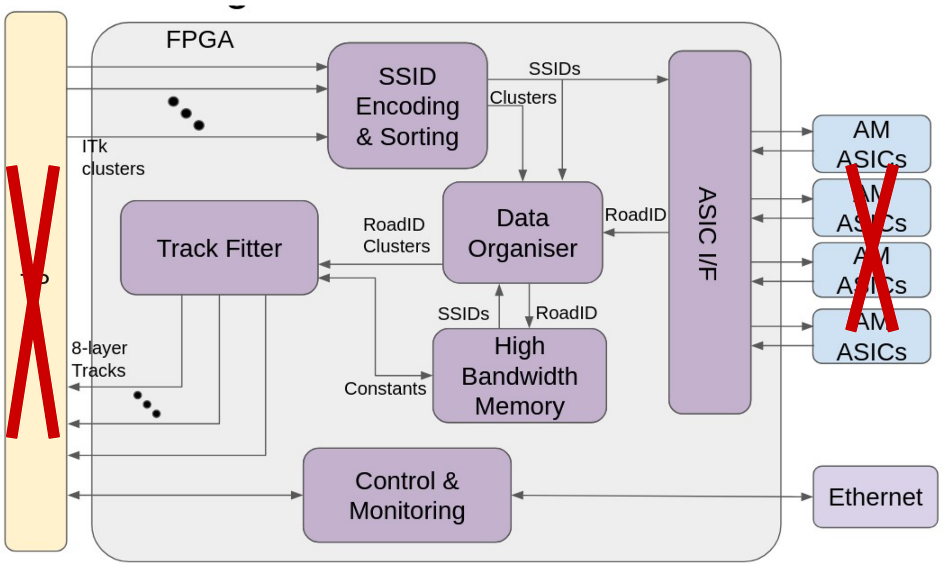




PRM Standalone Project

Dashed arrows are for store to memory.
Solid line arrows are for data.



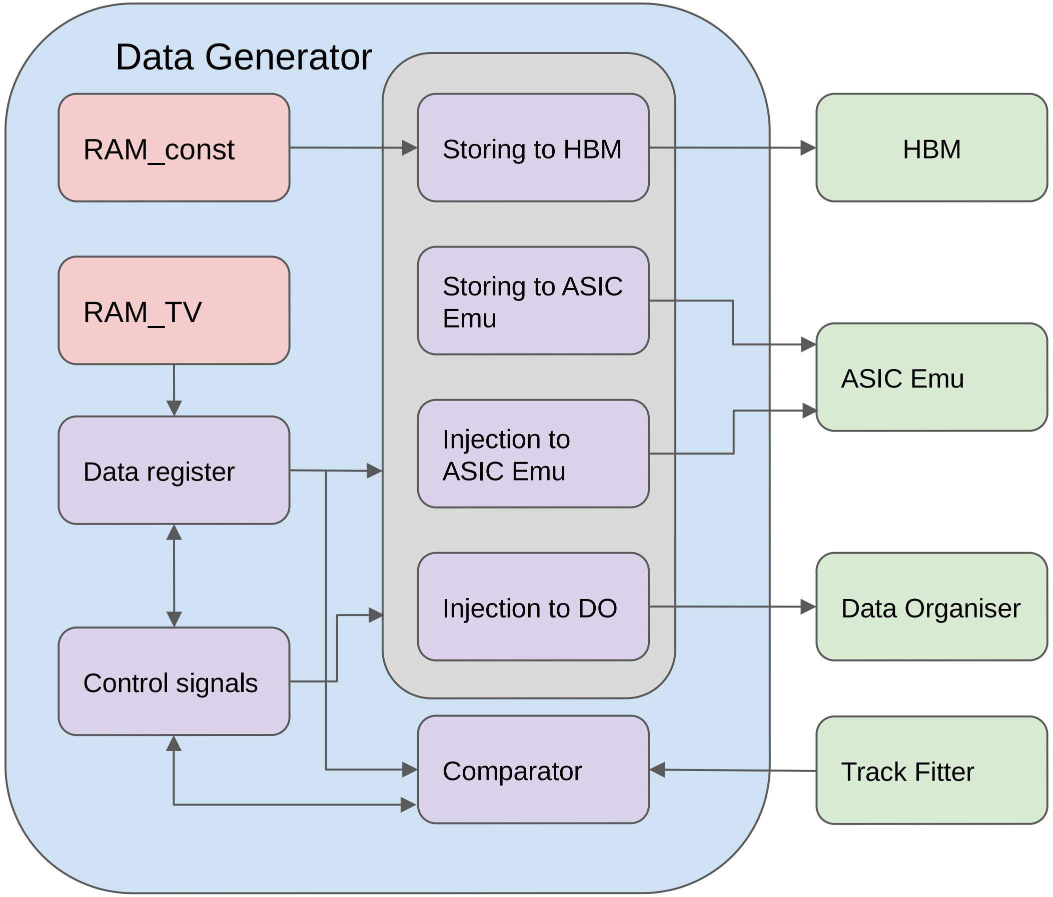


Data Generator

The Task

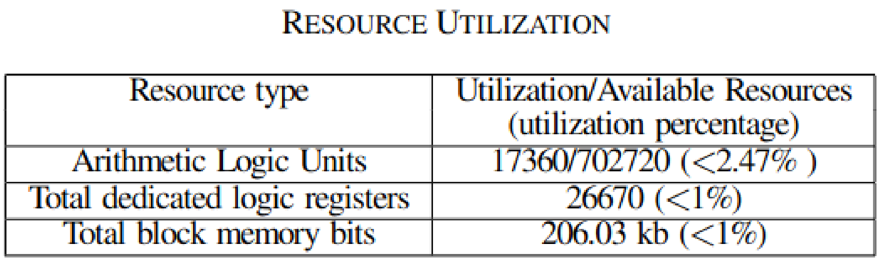
- Initializes the memories
- Injects data to the rest of the pipeline
- Compares the output with the input values

- A set of data are propagated to the rest of the components
- A set of input registers
  - **Control** the injection
  - **Define** the number of events to be injected
  - **Define** frequency of event injection
  - **Enable** the injection of **fake** clusters



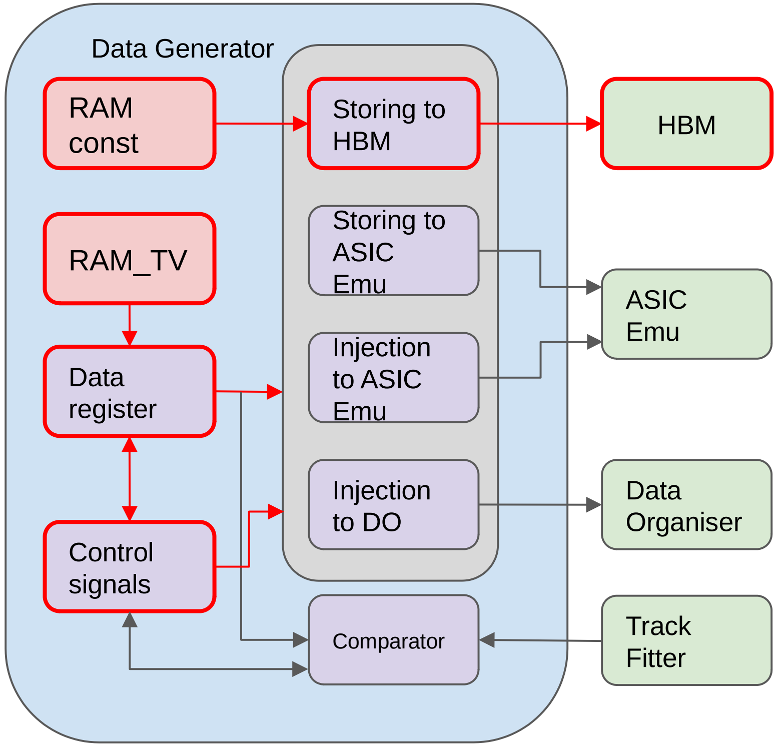
Data Generator #2

- **HBM** *initialization*
- **ASIC Emu** *initialization* & data *injection*
- **DO** data *injection*
- Output **comparator**



RESOURCE UTILIZATION

Resource type	Utilization/Available Resources (utilization percentage)
Arithmetic Logic Units	17360/702720 (<2.47%)
Total dedicated logic registers	26670 (<1%)
Total block memory bits	206.03 kb (<1%)

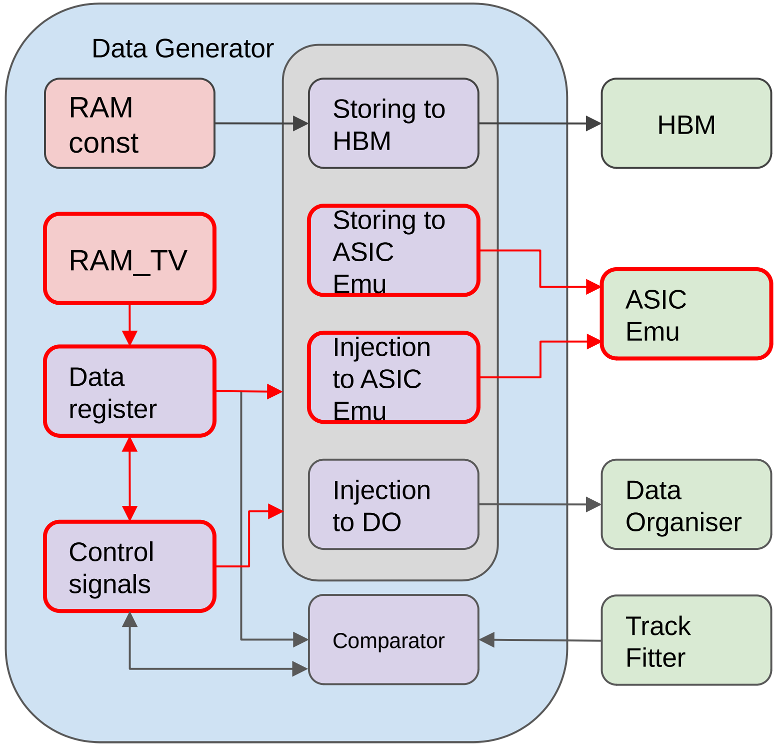


Data Generator #2

- **HBM** *initialization*
- **ASIC Emu** *initialization* & data *injection*
- **DO** data *injection*
- Output **comparator**

RESOURCE UTILIZATION

Resource type	Utilization/Available Resources (utilization percentage)
Arithmetic Logic Units	17360/702720 (<2.47%)
Total dedicated logic registers	26670 (<1%)
Total block memory bits	206.03 kb (<1%)

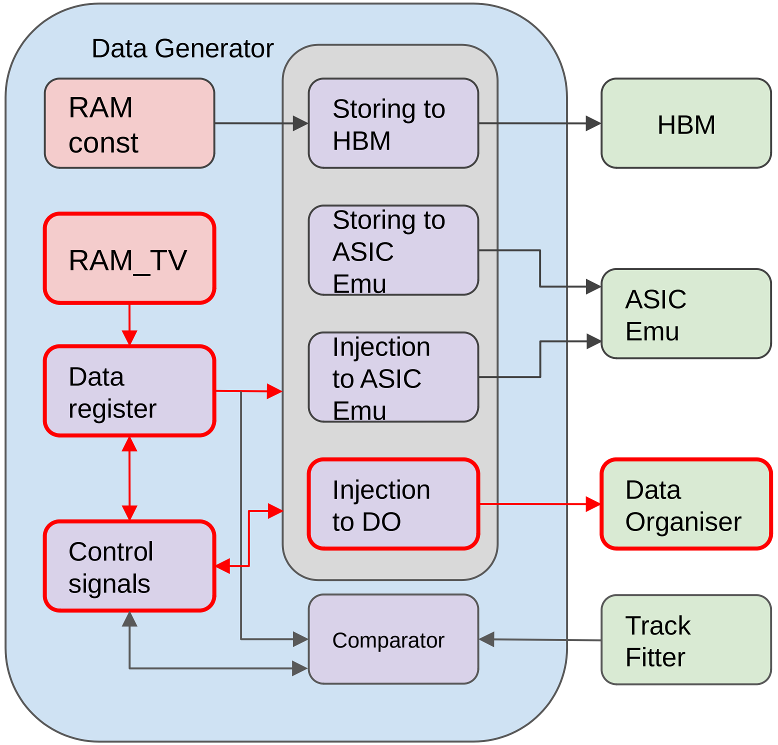


Data Generator #2

- **HBM** *initialization*
- **ASIC Emu** *initialization* & data *injection*
- **DO** data *injection*
- Output **comparator**

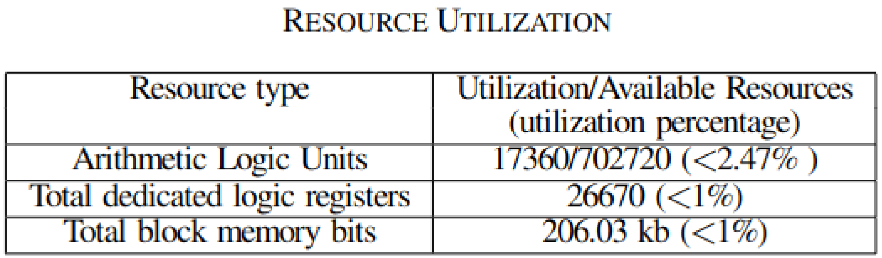
RESOURCE UTILIZATION

Resource type	Utilization/Available Resources (utilization percentage)
Arithmetic Logic Units	17360/702720 (<2.47%)
Total dedicated logic registers	26670 (<1%)
Total block memory bits	206.03 kb (<1%)



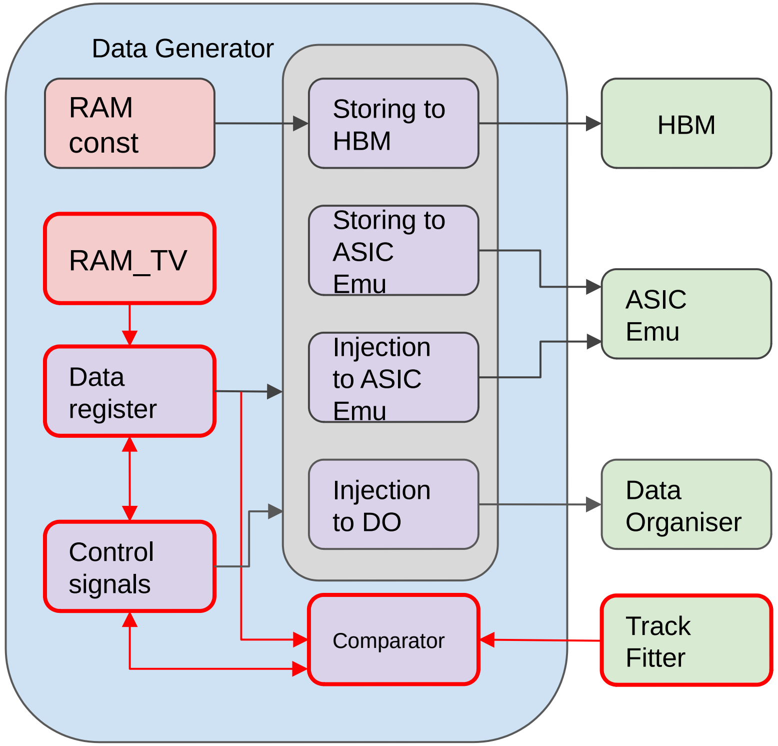
Data Generator #2

- **HBM** *initialization*
- **ASIC Emu** *initialization* & data *injection*
- DO data *injection*
- Output **comparator**



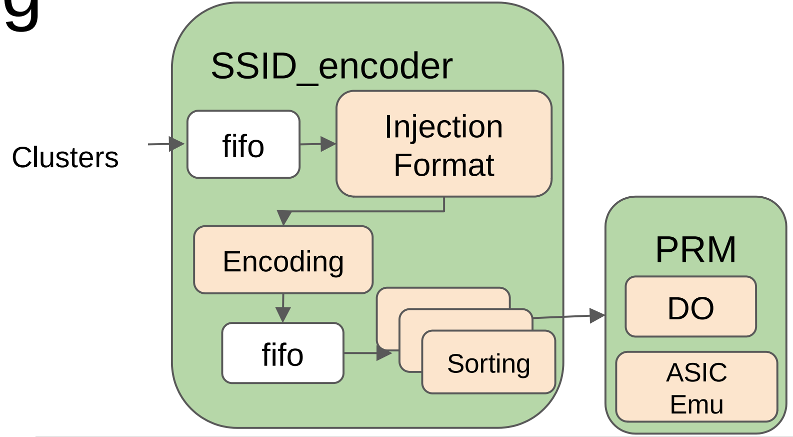
RESOURCE UTILIZATION

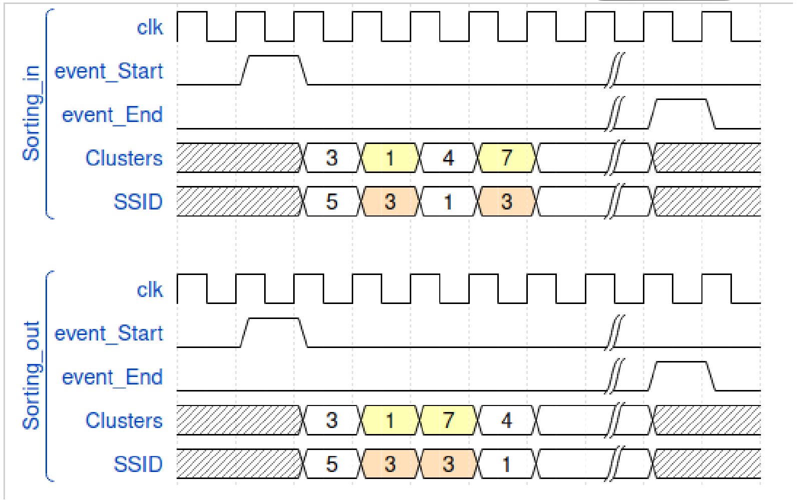
Resource type	Utilization/Available Resources (utilization percentage)
Arithmetic Logic Units	17360/702720 (<2.47%)
Total dedicated logic registers	26670 (<1%)
Total block memory bits	206.03 kb (<1%)



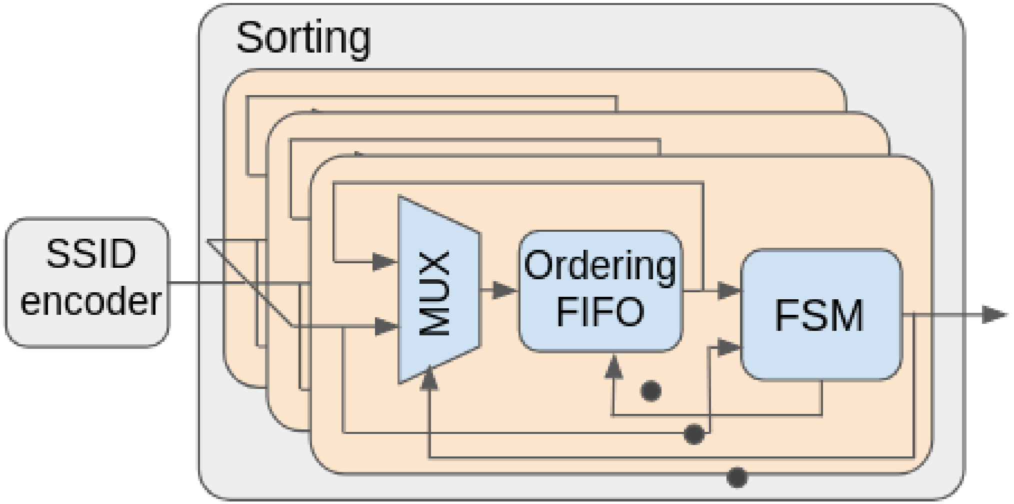
SSID Encoding and Sorting

- Receives the cluster data from the TP
- Encodes clusters into SSIDs (Coarser information)
- Groups the clusters per SSID (Sorting)
- Provide the data to the AM ASICs and Data Organiser
- High throughput
- Reasonable resource usage





SSID Encoding and Sorting #2



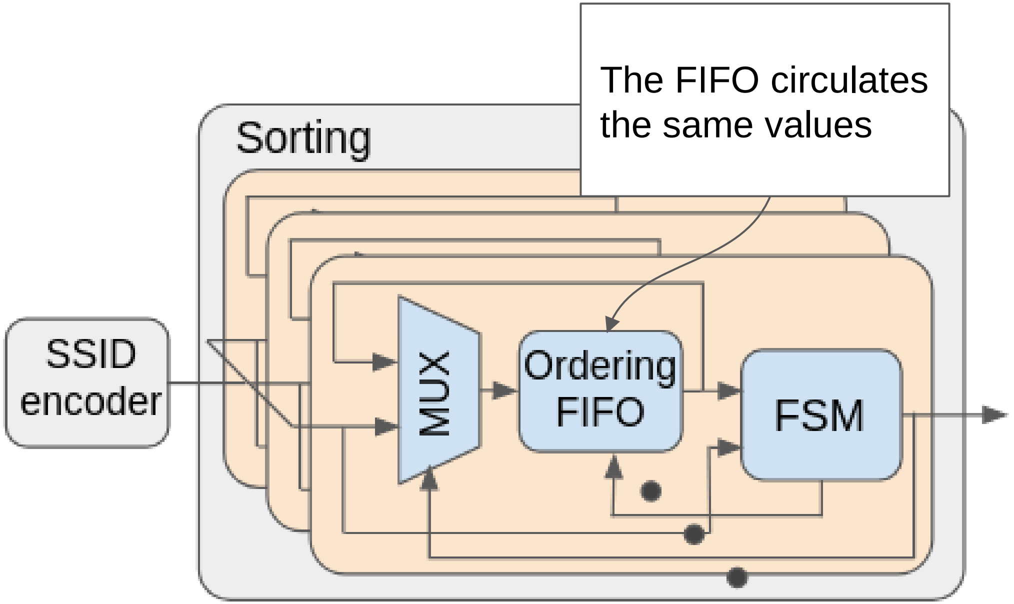
A design that fits:

- The FPGA architecture
- The needs of the application

Proposed solution:

- Grouping the SSIDs inside the "Sorting FIFO"
- Generate many instances of the ordering design → Optimize performance
- After the ordering is finished, propagate the output of the fifo that has the lowest value

SSID Encoding and Sorting #2



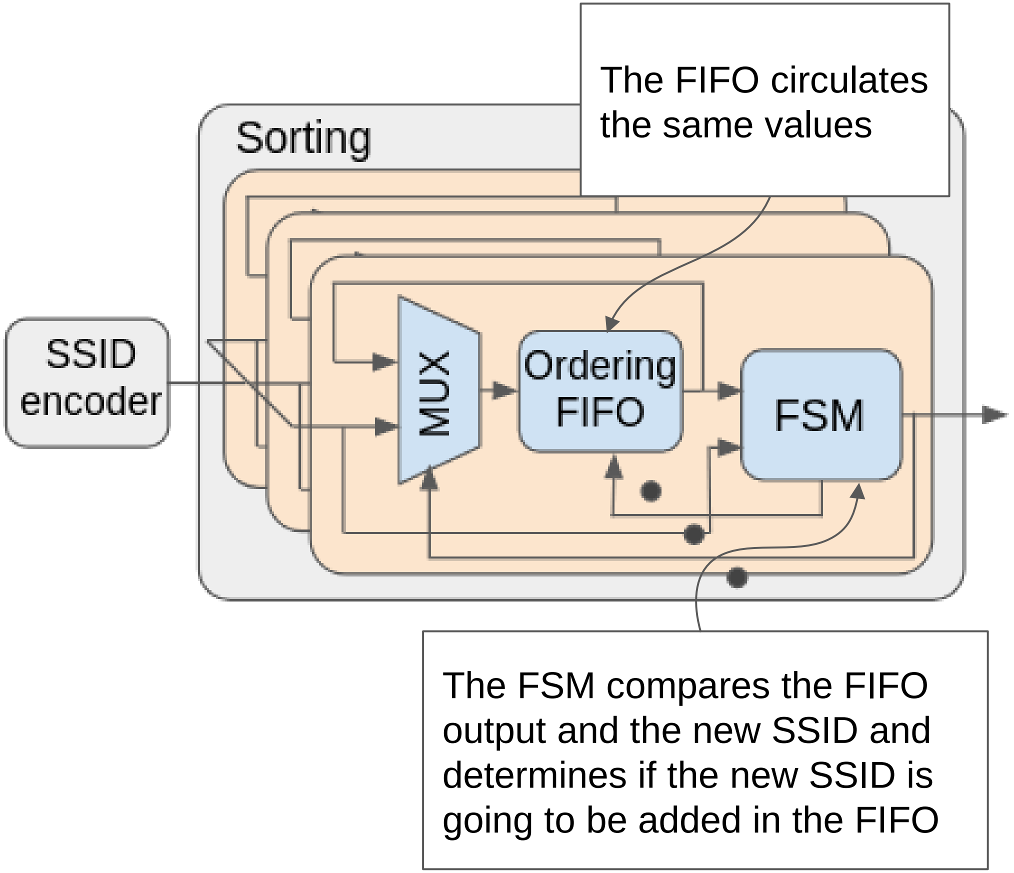
A design that fits:

- The FPGA architecture
- The needs of the application

Proposed solution:

- Grouping the SSIDs inside the "Sorting FIFO"
- Generate many instances of the ordering design → Optimize performance
- After the ordering is finished, propagate the output of the fifo that has the lowest value

SSID Encoding and Sorting #2


The FIFO circulates the same values
Sorting
SSID encoder
MUX
Ordering FIFO
FSM
The FSM compares the FIFO output and the new SSID and determines if the new SSID is going to be added in the FIFO

A design that fits:

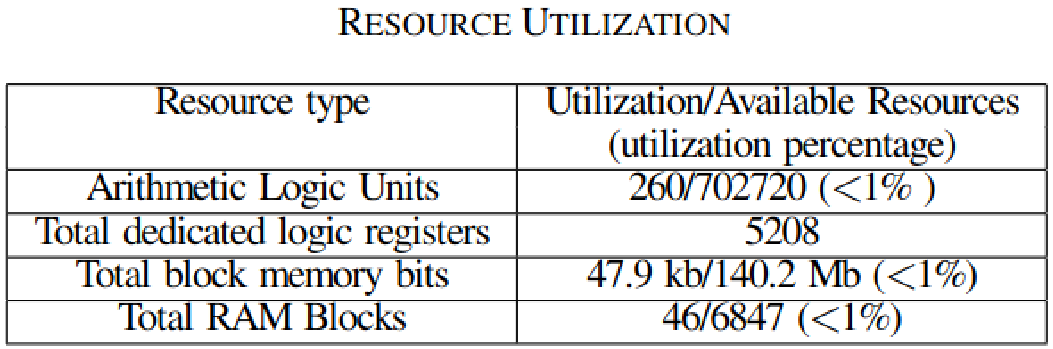
- The FPGA architecture
- The needs of the application

Proposed solution:

- Grouping the SSIDs inside the “Sorting FIFO”
- Generate many instances of the ordering design → Optimize performance
- After the ordering is finished, propagate the output of the fifo that has the lowest value

SSID Encoding and Sorting #3

- **165 ALU** per Sorting Block
- **13312 Block Memory Bits** per sorting block
- **Latency:**
  - 1 sorting block: 528 ns
  - 2 sorting blocks: 424 ns
  - ~ 20% improvement



RESOURCE UTILIZATION

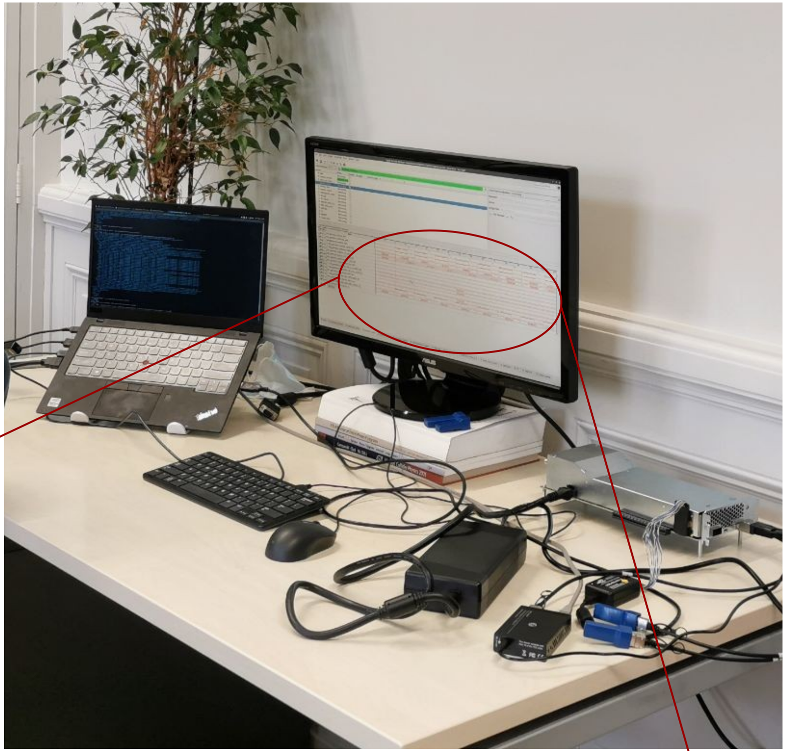
Resource type	Utilization/Available Resources (utilization percentage)
Arithmetic Logic Units	260/702720 (<1%)
Total dedicated logic registers	5208
Total block memory bits	47.9 kb/140.2 Mb (<1%)
Total RAM Blocks	46/6847 (<1%)

Tests on Hardware

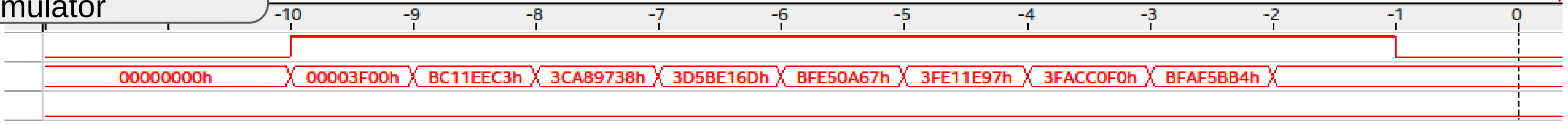
- Tests on the development kit (ID : 1SM21BHU2F53E1VG)
- Programing over **JTAG**
- Communication with linux machine over **Ethernet**
  - **Converter** from RJ45 to QSFP
  - **IPBus** protocol
    - Packet-based control protocol
- Quartus Signaltap
- Verify that the simulation matches the HW tests

Tests on Hardware

- Tests on the development kit (ID : 1SM21BHU2F53E1VG)
- Programing over **JTAG**
- Communication with linux machine over **Ethernet**
  - **Converter** from RJ45 to QSFP
  - **IPBus** protocol
    - Packet-based control protocol
- Quartus Signaltap
- Verify that the simulation matches the HW tests

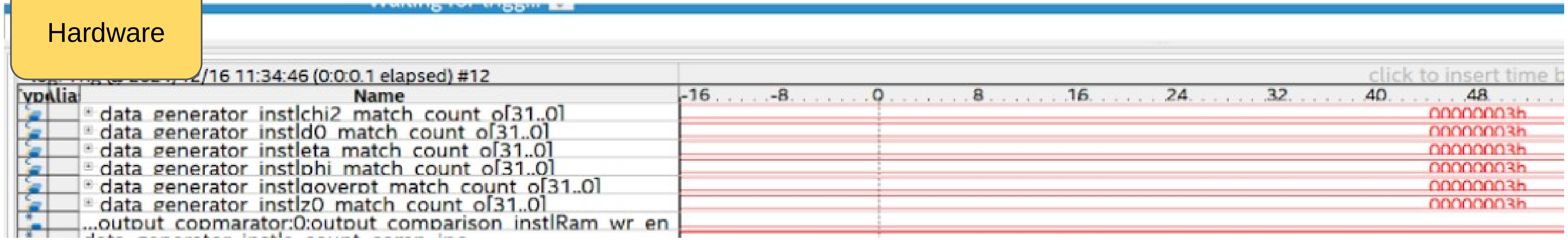


Initialization of the AM ASIC Emulator



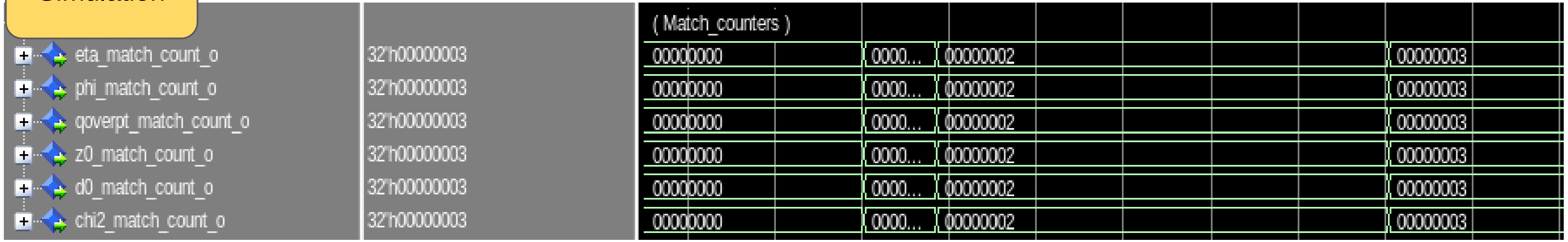
Tests on Hardware #2

Hardware



```
/16 11:34:46 (0:0:0.1 elapsed) #12                                         click to insert time b
ype Alia   Name                                                 -16 . . -8 . . 0 . . 8 . . 16 . . 24 . . 32 . . 40 . . 48
           data generator inst|chi2 match count o[31..0]                                          0000003h
           data generator inst|d0 match count o[31..0]                                            0000003h
           data generator inst|eta match count o[31..0]                                           0000003h
           data generator inst|phi match count o[31..0]                                           0000003h
           data generator inst|qoverpt match count o[31..0]                                       0000003h
           data generator inst|z0 match count o[31..0]                                            0000003h
           ...output copmarator:0:output comparison inst|Ram wr en
```

Simulation



```
                                         ( Match_counters )
eta_match_count_o       32'h00000003     00000000   0000...  00000002     00000003
phi_match_count_o       32'h00000003     00000000   0000...  00000002     00000003
qoverpt_match_count_o   32'h00000003     00000000   0000...  00000002     00000003
z0_match_count_o        32'h00000003     00000000   0000...  00000002     00000003
d0_match_count_o        32'h00000003     00000000   0000...  00000002     00000003
chi2_match_count_o      32'h00000003     00000000   0000...  00000002     00000003
```

Design challenges and further optimizations

Challenges

- Data Generator
  - Dealing with all the possible types of input
    - Number of pixel and strip clusters
    - Missing hits
    - Generation of random clusters
  - Keeping the injections and the comparison in sync
- SSID Encoder
  - Designing and implementing the logic for the SSID grouping
    Simple concept, challenging to describe in hardware

Further optimizations

- Adding more layers of parallelization during the sorting
  - Dealing with each layer separately
  - Dealing with each group of clusters separately
  - Implementing different algorithms and comparing the performance
    - More conventional approach → Using SSIDs as pointers

Summary

- Design and Verification of the **Data Generator**
- Design and Verification of the **SSID Encoder**
- Verification on **Hardware**

A **paper** on the full PRM Stratix 10 firmware has been published!

“Intel Stratix 10 FPGA design for track reconstruction for the ATLAS experiment at the HL_LHC”

Thank You!

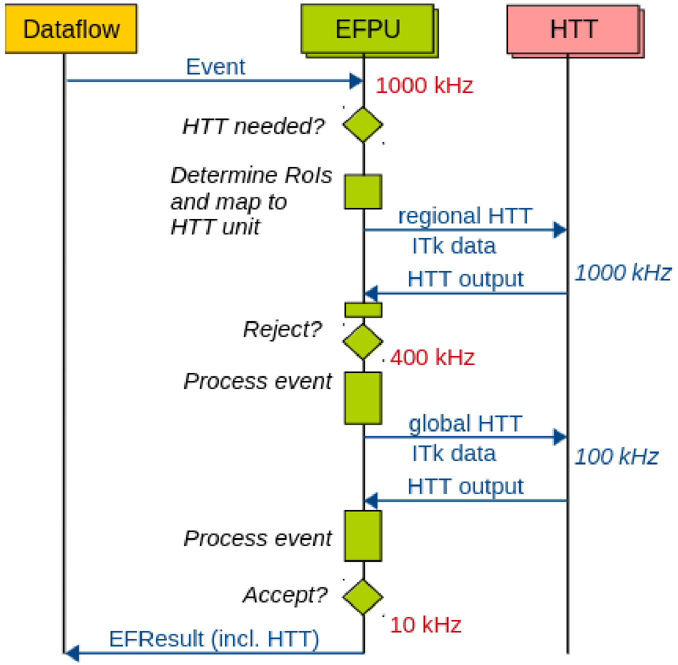
ATLAS Phase-II TDAQ upgrade

TDAQ Phase II architecture

Single level hardware trigger: Level 0

- Hardware Track Trigger (HTT) → Reduce backgrounds after L0 trigger selections and mitigate pile-up
  - Custom designed boards
  - **Regional tracking (rHTT)** at 1 MHz
    - on up to 10% of the ITk data, Pt > 2 GeV
  - **Global tracking (gHTT)** at 100 kHz
    - Full coverage Pt > 1 GeV
- Data are provided by the Event Filter processing farm and tracks are returned to it

HTT evolved into commodity based system, so no longer happening, but the experience with the demonstrator quite unique!



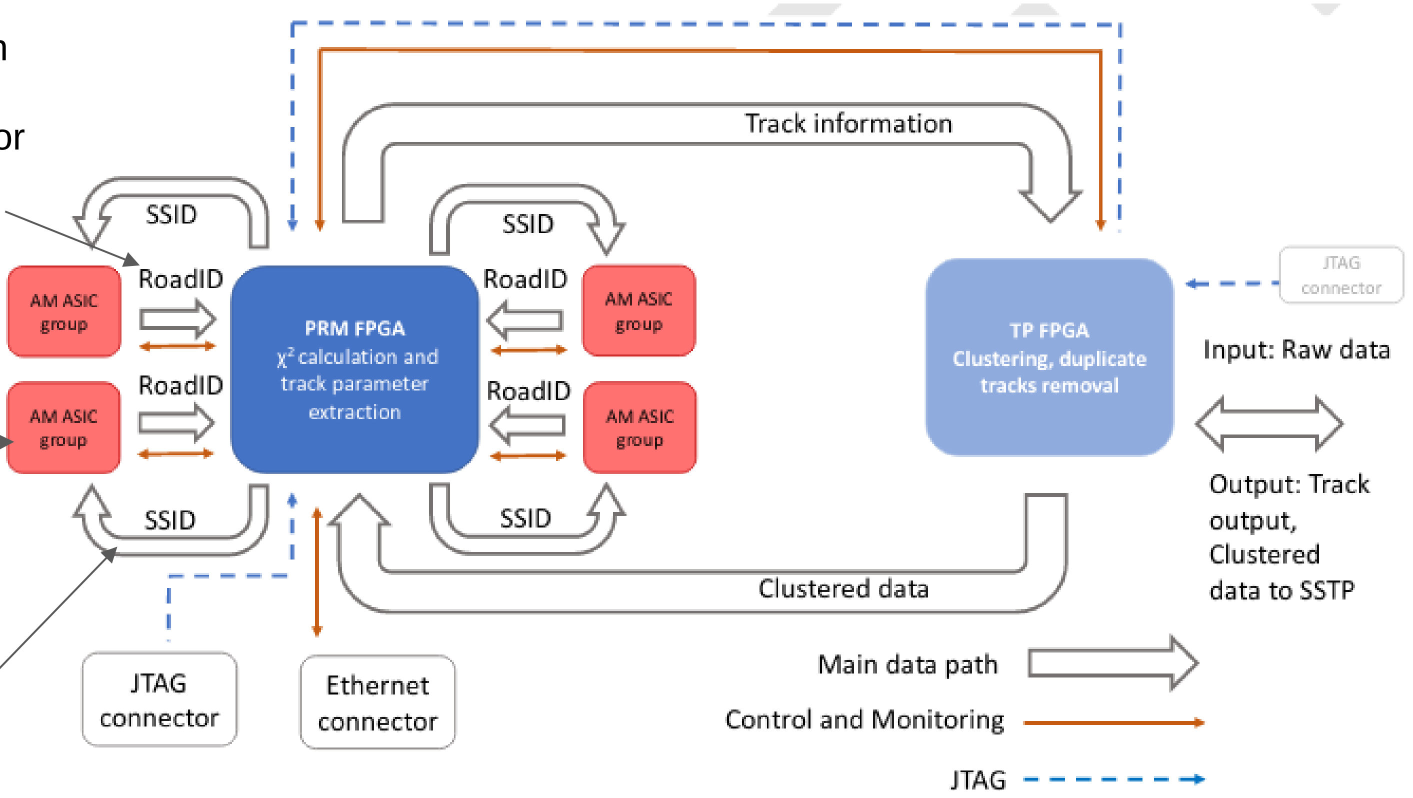
https://cds.cern.ch/record/2285584/files/ATLAS-TDR-029.pdf

Pattern Recognition Mezzanine

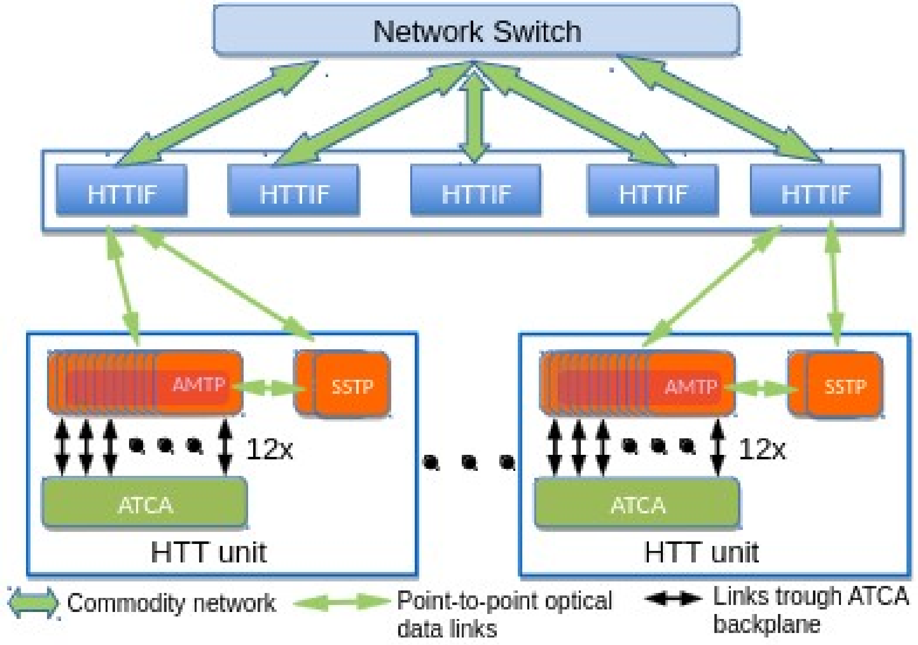
Identifier of the match between a list of clusters in the detector layers and the patterns in the AM ASICs

Associative Memory Application Specific Integrated Circuit

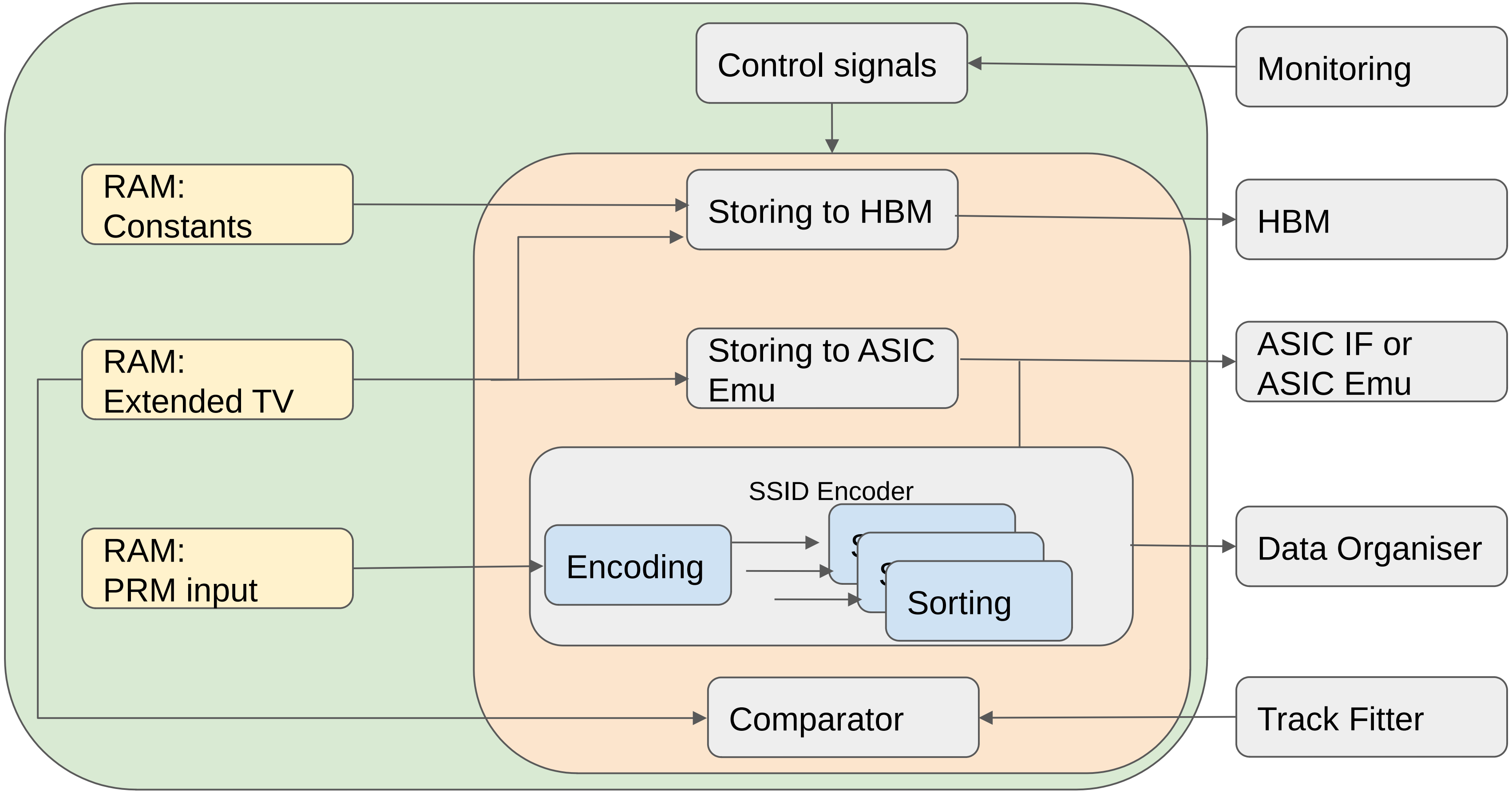
Superstrip Identifier



Hardware Track Trigger (HTT)



Data Generator with SSID encoder



Tests on Hardware

Configuration, control and monitoring

- FPGA and PC **communication** by **monitoring block**
- Communication over **Ethernet** using **IPBus protocol**