KØBENHAVNS UNIVERSITET

Niels Bohr Institute

ATLAS EXPERIMENT

CERN

Hardware-based track reconstruction for the ATLAS Trigger and Data Acquisition Phase-II system

Alessandra Camplani – University of Copenhagen

December 5th, 2019

LHC and ATLAS

The Large Hadron Collider

Large Hadron Collider (LHC) at CERN

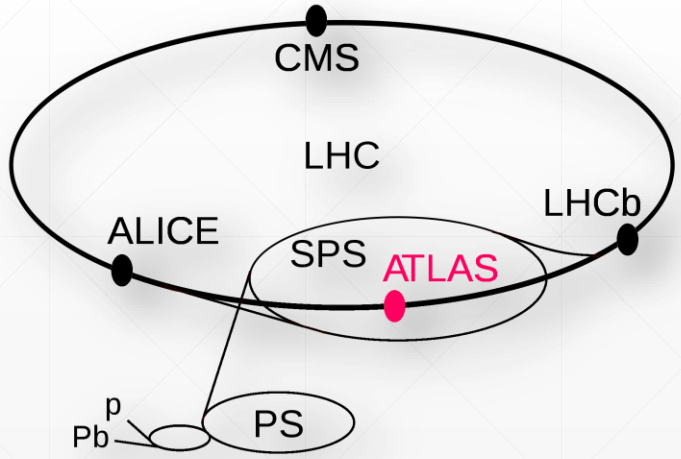
- 27 km ring placed underground at an average depth of 100 m
- two-rings collider able to accelerate protons up to 6.5 TeV and lead ions up to 2.5 TeV per nucleon

Four main experiments where collisions happen

- ATLAS, ALICE, CMS, LHCb

Physics goals for the next years

- Study the Higgs boson properties to confirm the Standard Model predictions and search for new physics phenomena



The ATLAS experiment

A Toroidal LHC ApparatuS (ATLAS)

- Cylindrical shape, 46 m long, 25 m in diameter
- Weight of 7,000 tonnes (similar to the Eiffel Tower)

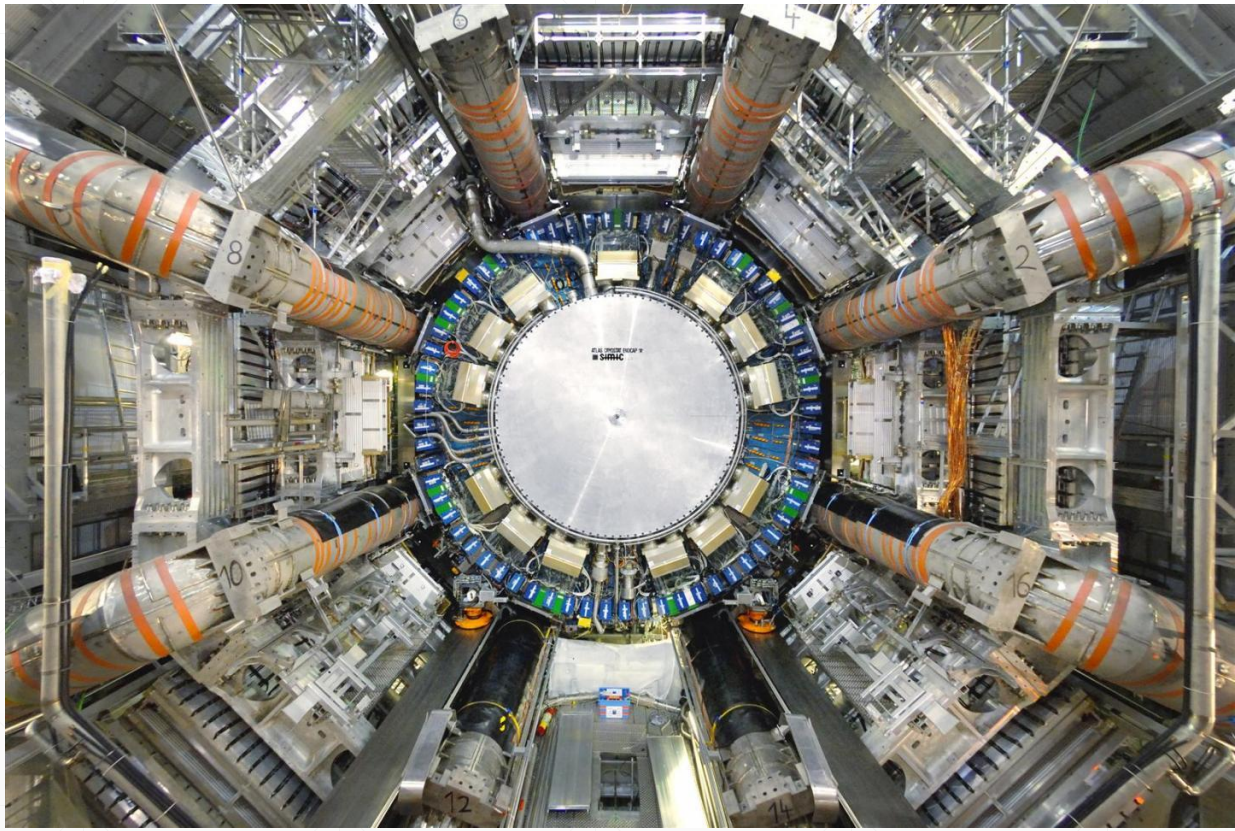
ATLAS like a giant camera that takes pictures **every 25 ns**

➢ bunches of protons cross 40 million times a second

(When any of the particles collide, the process is called “event”)

Every second more than 60 terabytes is produced (link)

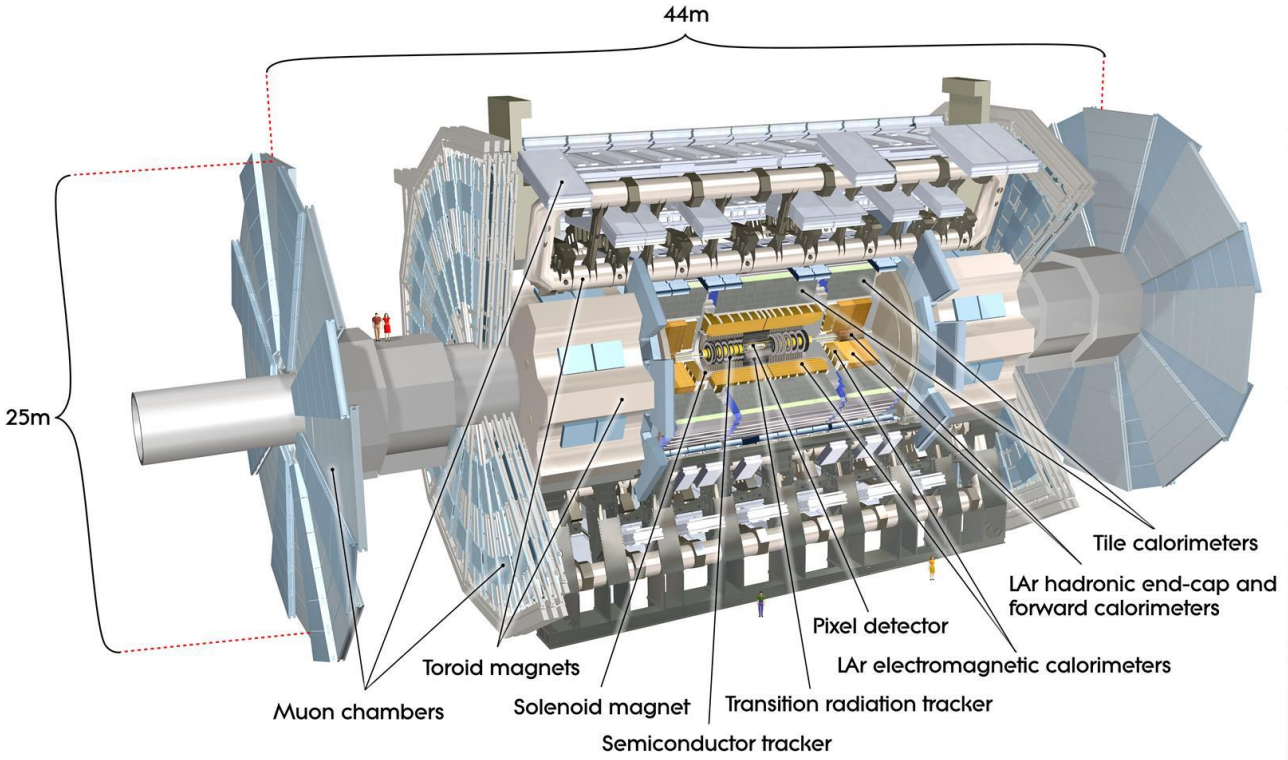
- specialized multi-level computing system selects only few events depending on their characteristics
- **Event selection is done in real time during the data taking**



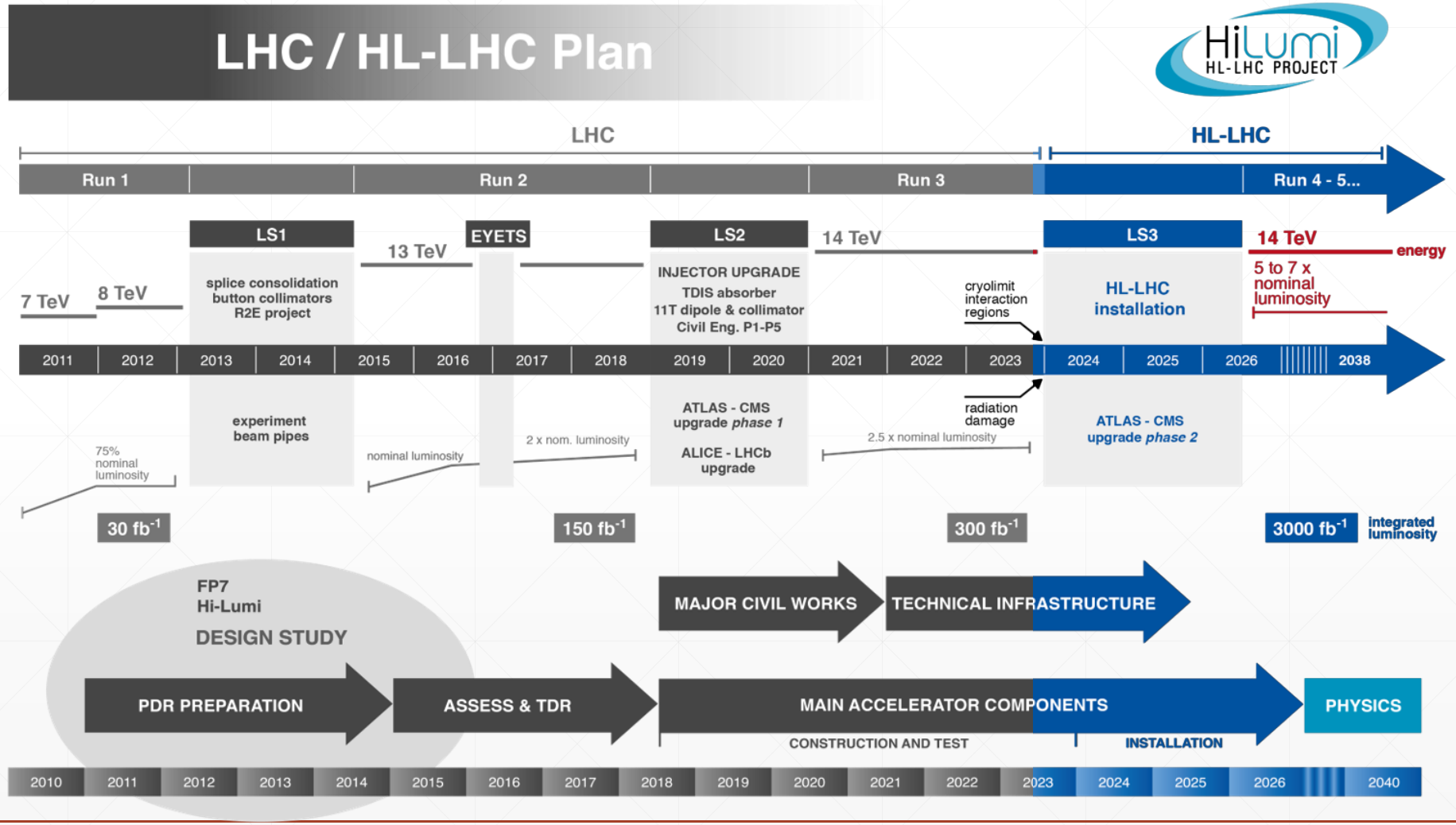
The ATLAS detectors

Four major components:

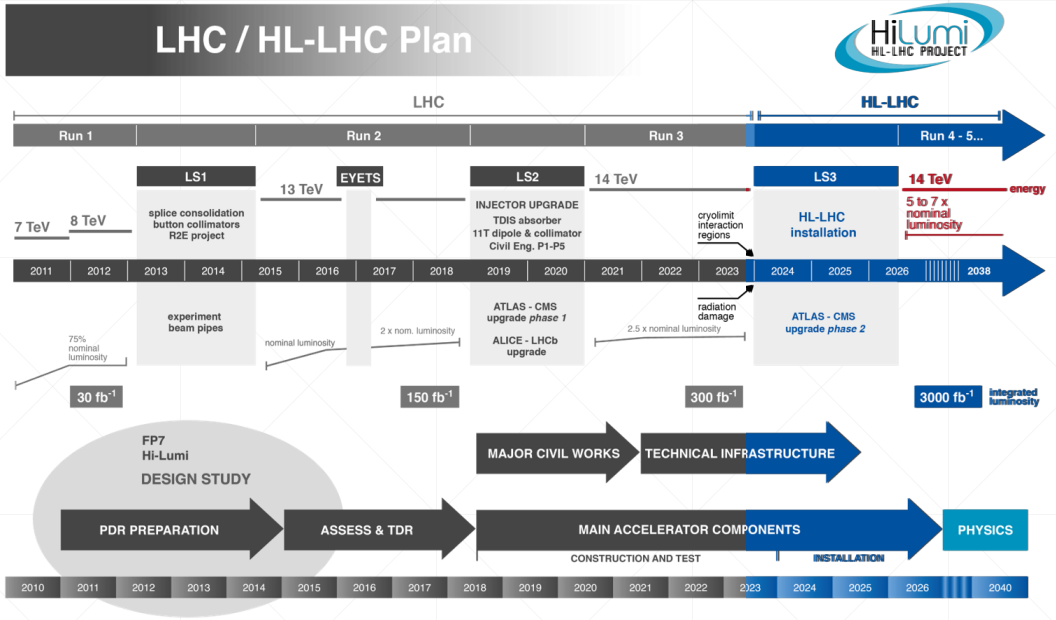
- **Inner detectors:** pixel, semiconductor tracker and transition radiation tracker
  - Measure the momentum and direction of charged particles
- **Calorimeters:** electromagnetic and hadronic
  - Measure the energies and direction of electrons, photons and hadron jets
- **Muon chambers**
  - Identify and measure the momentum of muons
- **Magnet systems:** solenoid and toroid
  - Bend charged particles for momentum measurements



LHC schedule



Schedule and upgrade plans



Currently, LHC is in **technical pause**

- Started on the 10th of December 2018 and will last for two years
- Time for **machine maintenance and upgrades**
  - Part of the trigger and readout electronic has to be substituted and tested

ATLAS is currently installing the new **Phase I** electronics, necessary for Run 3 starting in 2021.

Preparing ATLAS also for the **High-Luminosity LHC** (HL-LHC) (link)

- LHC upgrade will allow to achieve a luminosity value (how many particles are brought to the collision point) **a factor of ten larger than the LHC nominal value**

For this purpose, scientist, engineers and technicians are working together on the development of new electronics and detector components.

Phase-II Trigger and Data Acquisition system

What is TDAQ?

Trigger and Data Acquisition (TDAQ) system:

- selects events with distinguishing characteristics that make them interesting for physics analyses
  - reducing the flow of data to manageable levels
- passes interesting events on to a data storage system for offline analysis

Currently the ATLAS trigger system carries out the selection process in **two stages**:

- **The Level-1 hardware trigger**, constructed with custom-made electronics, works on a subset of information from the calorimeter and muon detectors.
- **The High Level Trigger (HLT)** is a large farm of CPUs (i.e. a software based trigger) which refines the analysis of the hardware-based Level-1 trigger.

Challenges and physics motivation

High luminosity consequences

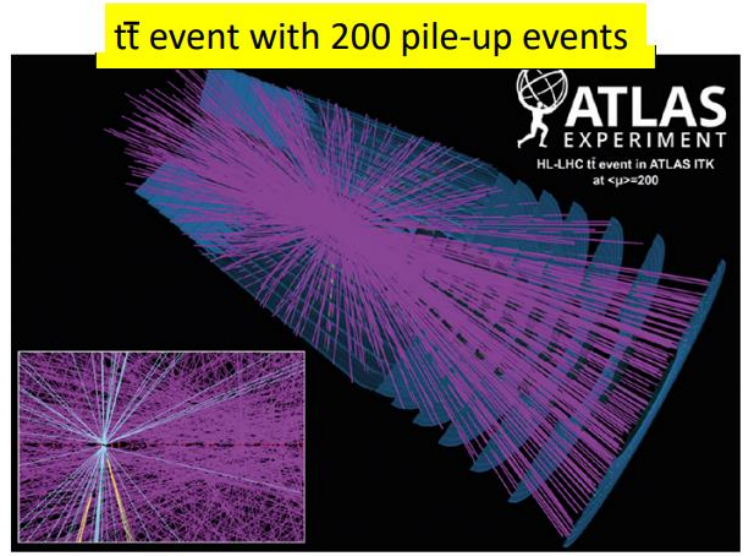
- High pile-up up to 200 events per bunch crossing (<μ>~40 today)
- High granularity detectors that need to be read out
  - new subdetector: Inner Tracker (ITk)
  - front-end/back-end electronics updates
- Larger event size ~5.2 MB (~2 MB today)

The **challenging and broad HL-LHC program** requires the pT of the various trigger objects as low as possible, e.g.:

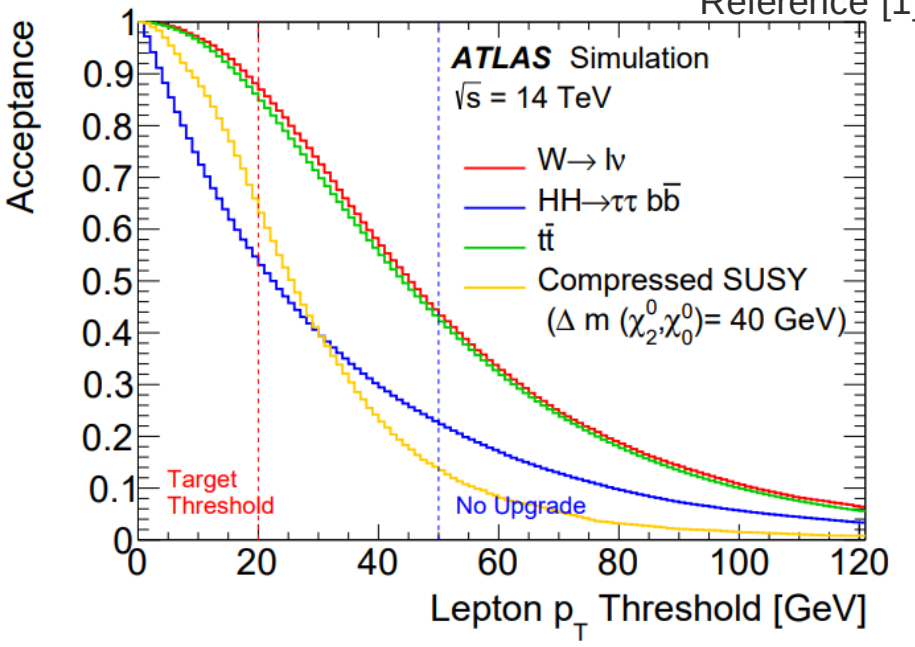
- Electroweak scale requires low pT leptons
- Searches for new physics with low Δm
- HH measurements requires low pT jets /b-jets

Operating points for ATLAS TDAQ:

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



Reference [1]



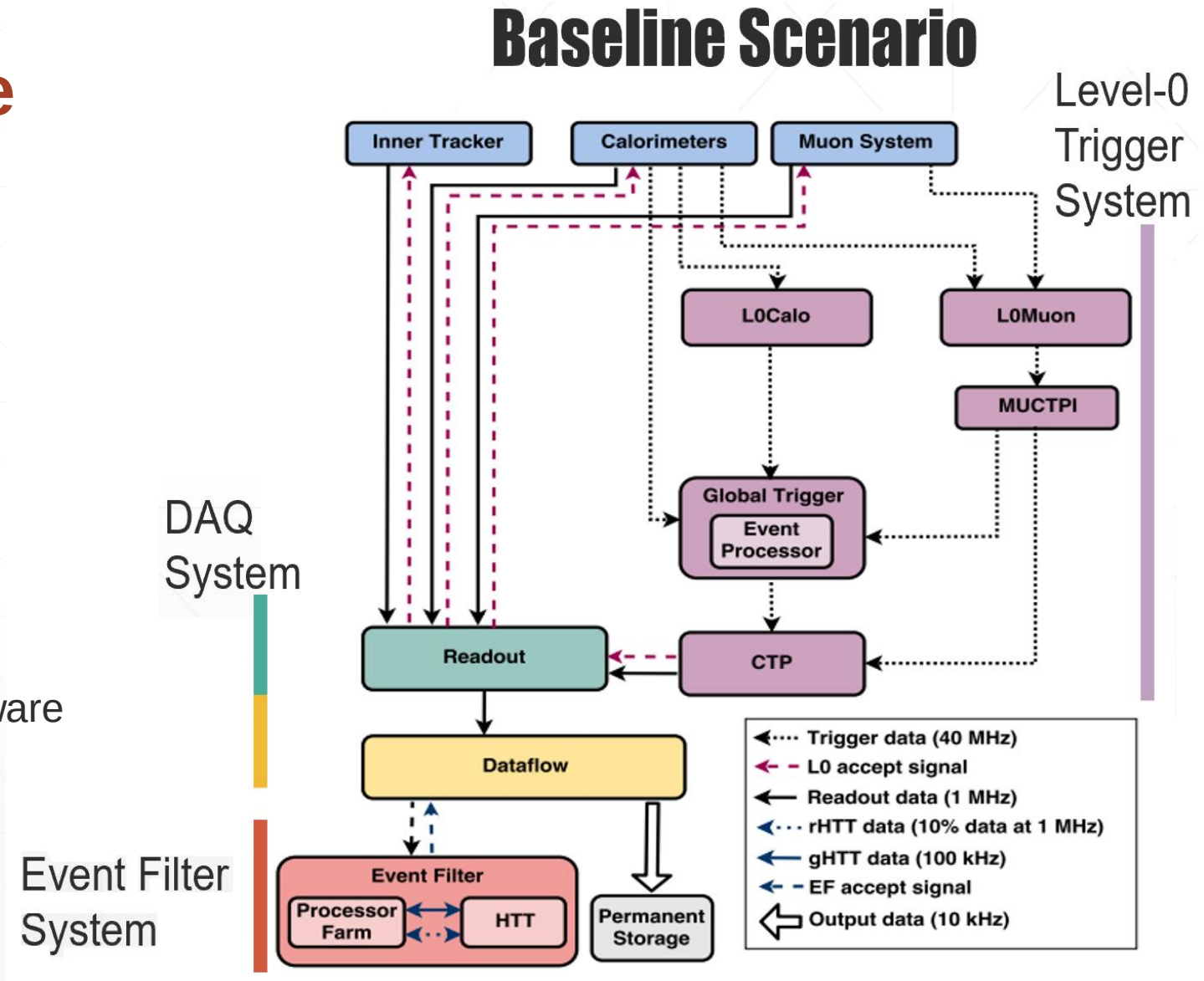
Phase-II TDAQ upgrade

Three main systems of the TDAQ Phase-II upgrade architecture:

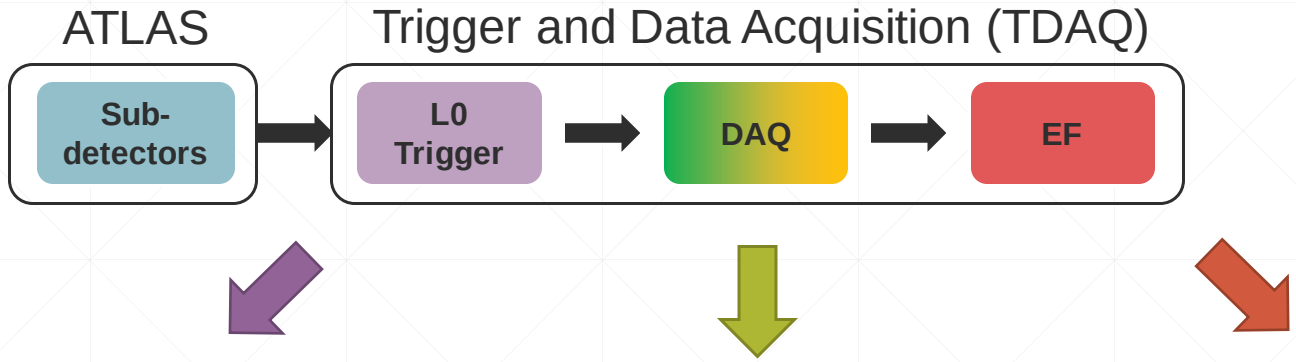
- Level-0 Trigger
- DAQ (Readout and Dataflow subsystems)
- Event Filter

Single-hardware-level trigger architecture (baseline scenario):

➢ capable of evolving into a two-level hardware trigger system (evolved scenario)



Overview of the Phase II TDAQ systems



- Data from calorimeters and muons are used for a first selection.
- Identification algorithm, are applied.
- Digital trigger inputs are aligned and combined to make the final L0A decision.

- Data from the ATLAS detector front-end electronics are received and formatted.
- Dara are buffered before, during and after the Event Filter decision.
- Finally selected events are transferred to permanent storage.

- Algorithms close to the offline reconstruction methods will run on a processor farm.
- Track reconstruction is performed by a co-processor
- Info are sent back to DAQ.

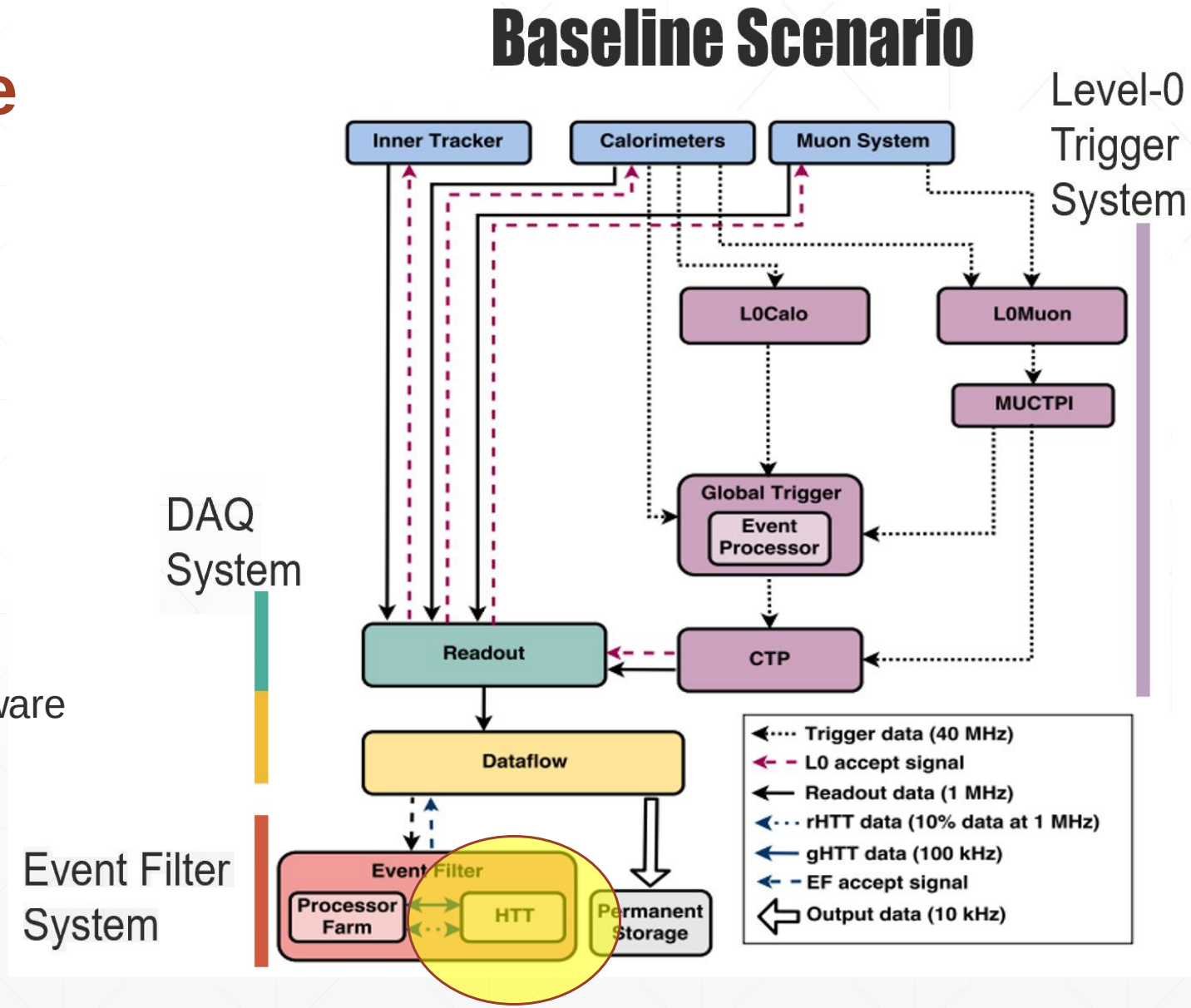
Phase-II TDAQ upgrade

Three main systems of the TDAQ Phase-II upgrade architecture:

- Level-0 Trigger
- DAQ (Readout and Dataflow subsystems)
- Event Filter

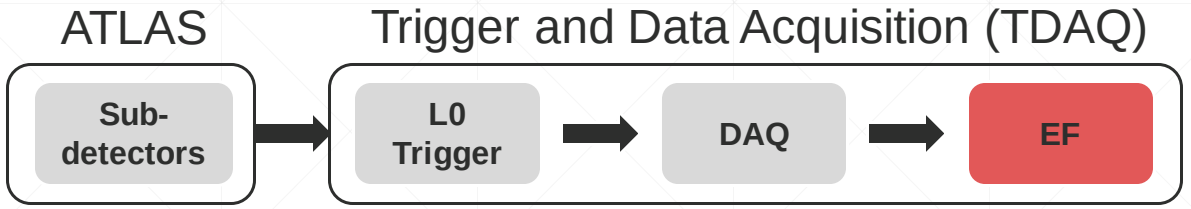
Single-hardware-level trigger architecture (baseline scenario):

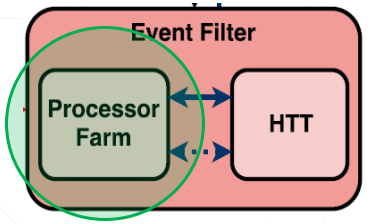
➢ capable of evolving into a two-level hardware trigger system (evolved scenario)



Hardware Track Trigger system

Event Filter





High pile-up conditions: higher occupancy of tracking detectors and reduced energy resolution in calorimeters.

To maintain threshold similar to Run1:

- algorithms close to the offline reconstruction methods
- tracking to identify a primary vertex and associate reconstructed objects

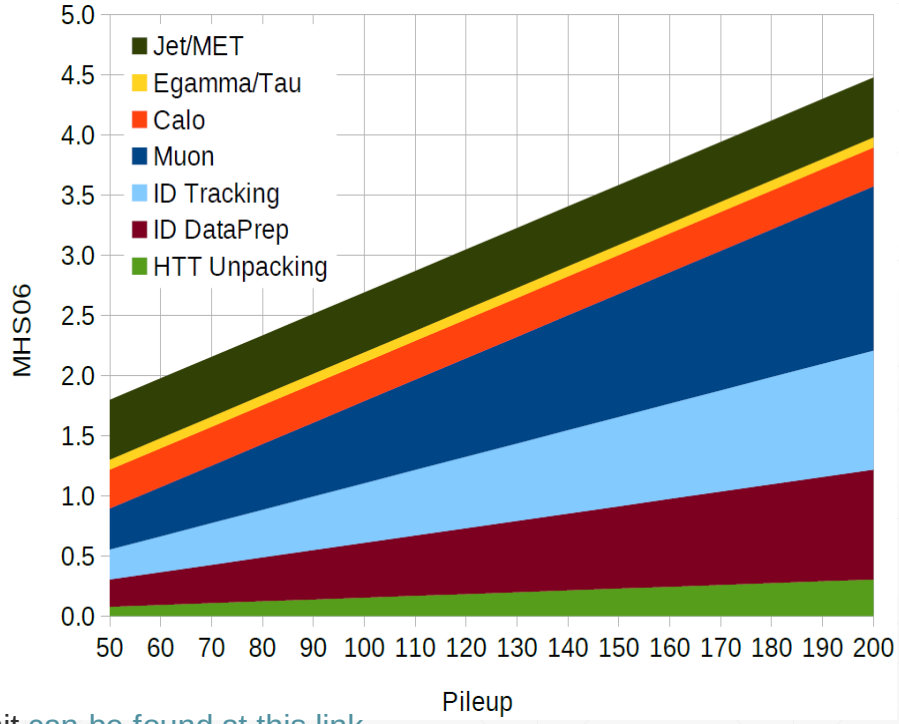
Processor Farm:

The current baseline assumption is that **CPUs will provide the required compute density** on the time-scale of Phase-II:

Current estimate of farm size:

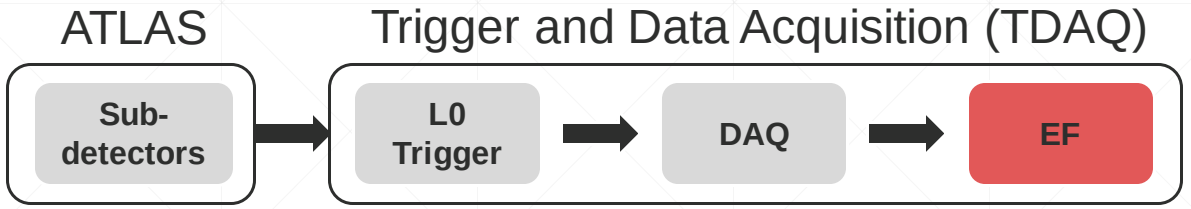
- 4.5 MHS06(1) (+Hardware Track Trigger) to handle a L0 rate of 1 MHz

CPU extrapolation versus event pileup

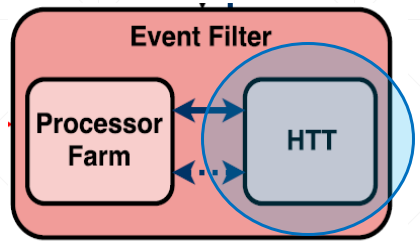


(1) More about HEP-SPEC06 (HS06) unit can be found at this link

HTT system



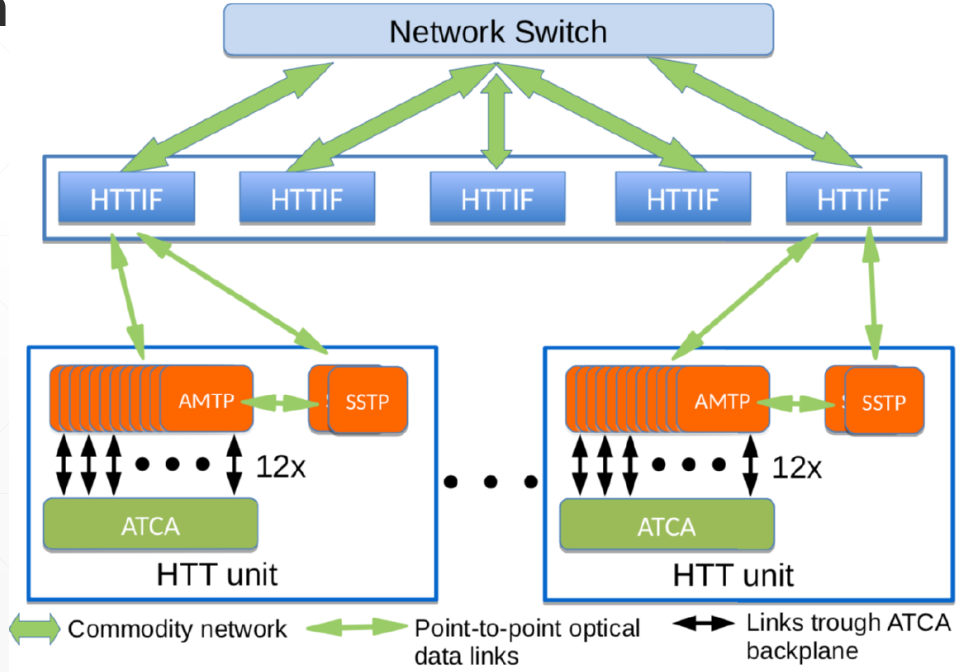
HTT (Hardware Track Trigger) hardware co-processor is a new **massively parallel system**, based on Field Programmable Gate Arrays (**FPGAs)** and custom Associative Memories (**AM ASICs**).



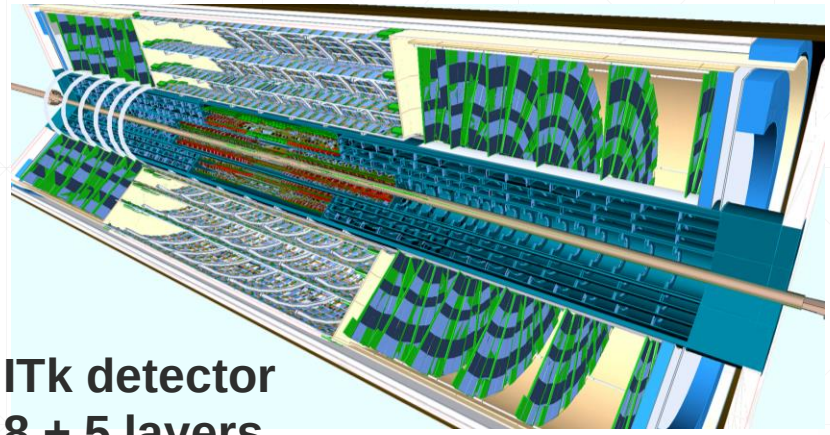
Goal of the HTT project: perform **very fast reconstruction** (faster than what can be done in software) of particle tracks.

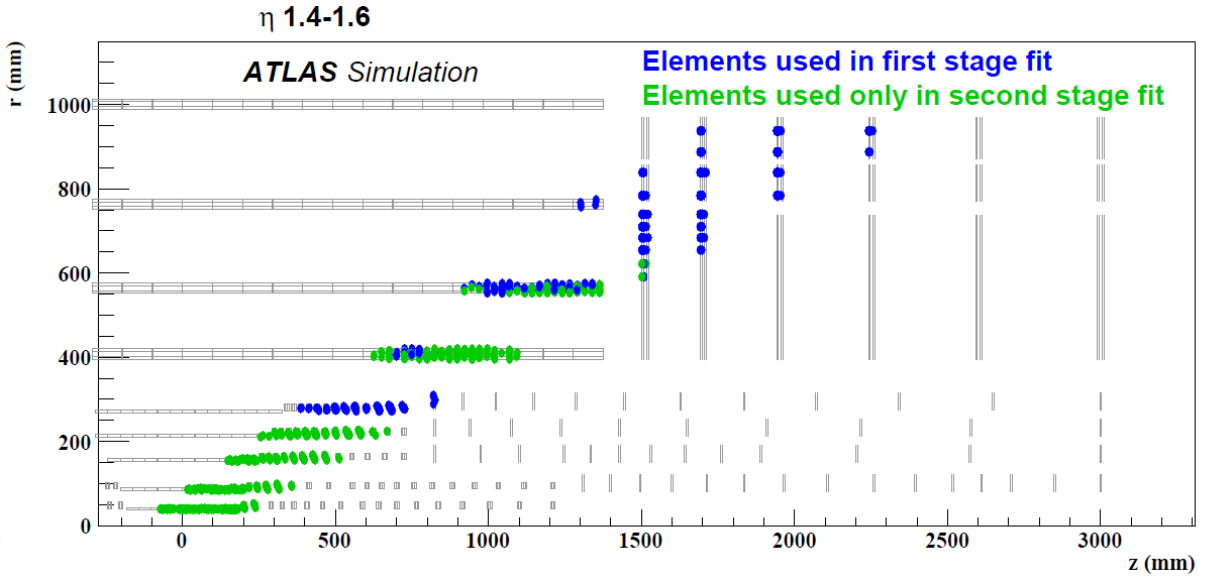
The HTT is organized as an array of independent tracking units called HTT units, each containing:

- **12 Associative Memory Tracking Processors (AMTPs)**:
  - Tracking Processor (TP) carrier board
  - One Pattern Recognition Mezzanine (PRM)
- **2 Second Stage Tracking Processors (SSTPs)**
  - TP carrier board (the same)
  - Two Track Fitting Mezzanines (TFM) cards



Tracks reconstruction



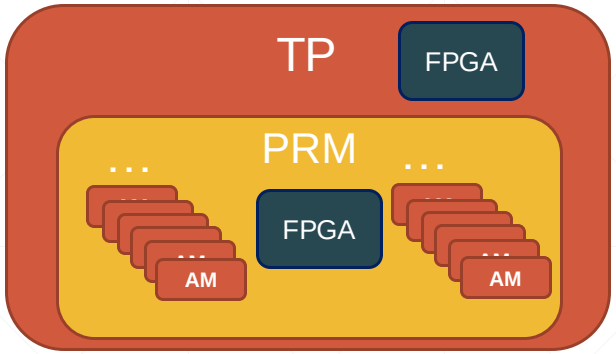


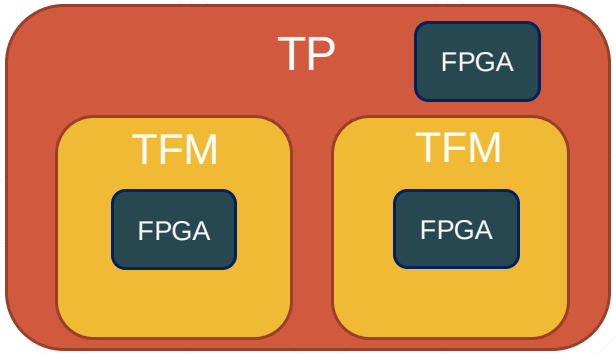
From Itk detector: hit position, per each layer.

Track reconstruction: performed with either 8 or 13 ITk layers.

Carrier board and mezzanines

AMTP





SSTP

First Stage Fitting

- TP mezzanine forms clusters from the data coming from the ITk detector and send them to PRM.
- PRM does pattern recognition on the cluster from 8 layers to preliminary identify candidates of tracks.
- Chi square selection and parameter calculation then performed.
- Tracks and parameters are sent back to TP.

Second Stage Fitting

- Tracks and parameters can be sent to TFM.
- TFM performs extrapolation on the remaining 5 layers
- Chi square selection and parameter calculation performed on the complete tracks
- Tracks and parameters are sent back to TP.

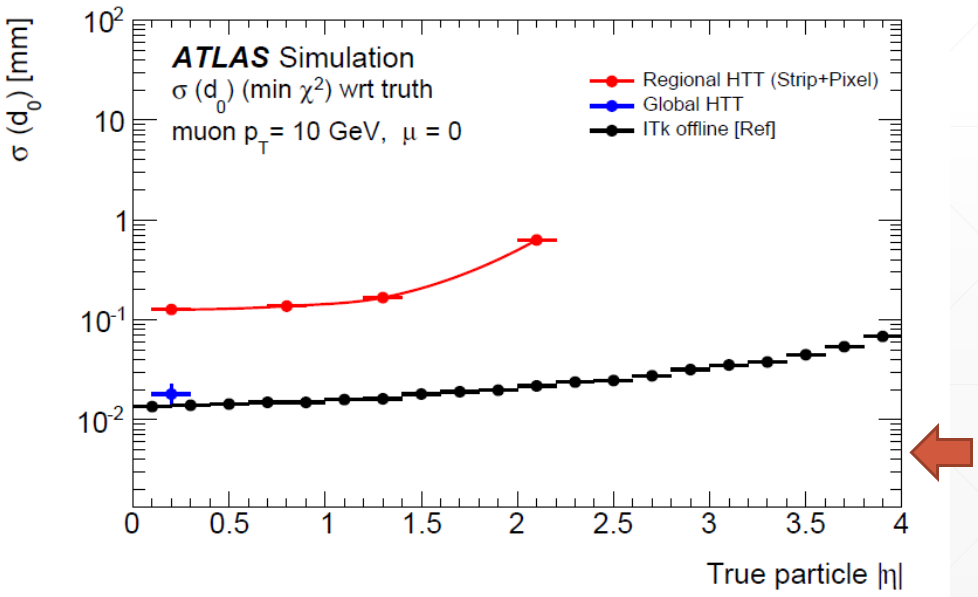
Two-step tracking in HTT

Regional HTT (rHTT)

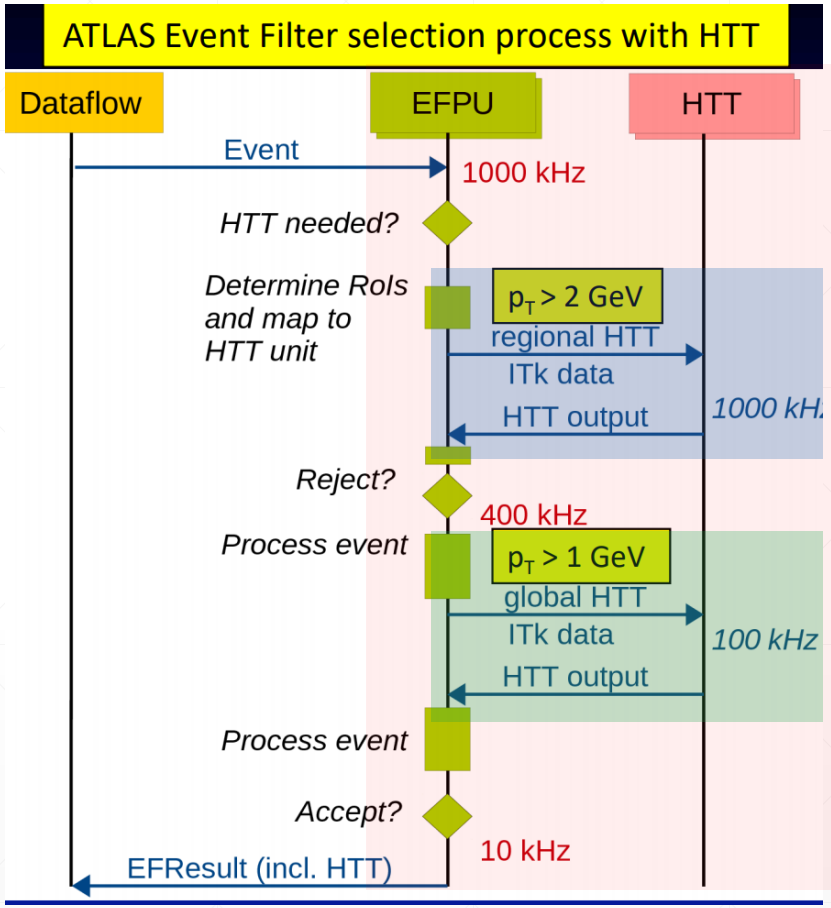
- **Event** from DAQ (Dataflow) to EF Processing Units (EFPU)
- **Partial tracker event** data (8 outermost layers) from EFPU to HTT (pT > 2 GeV)
- EFPU uses **rHTT tracks** in combination with Global Trigger objects to reduce the rate to **400 kHz**

Global HTT (gHTT)

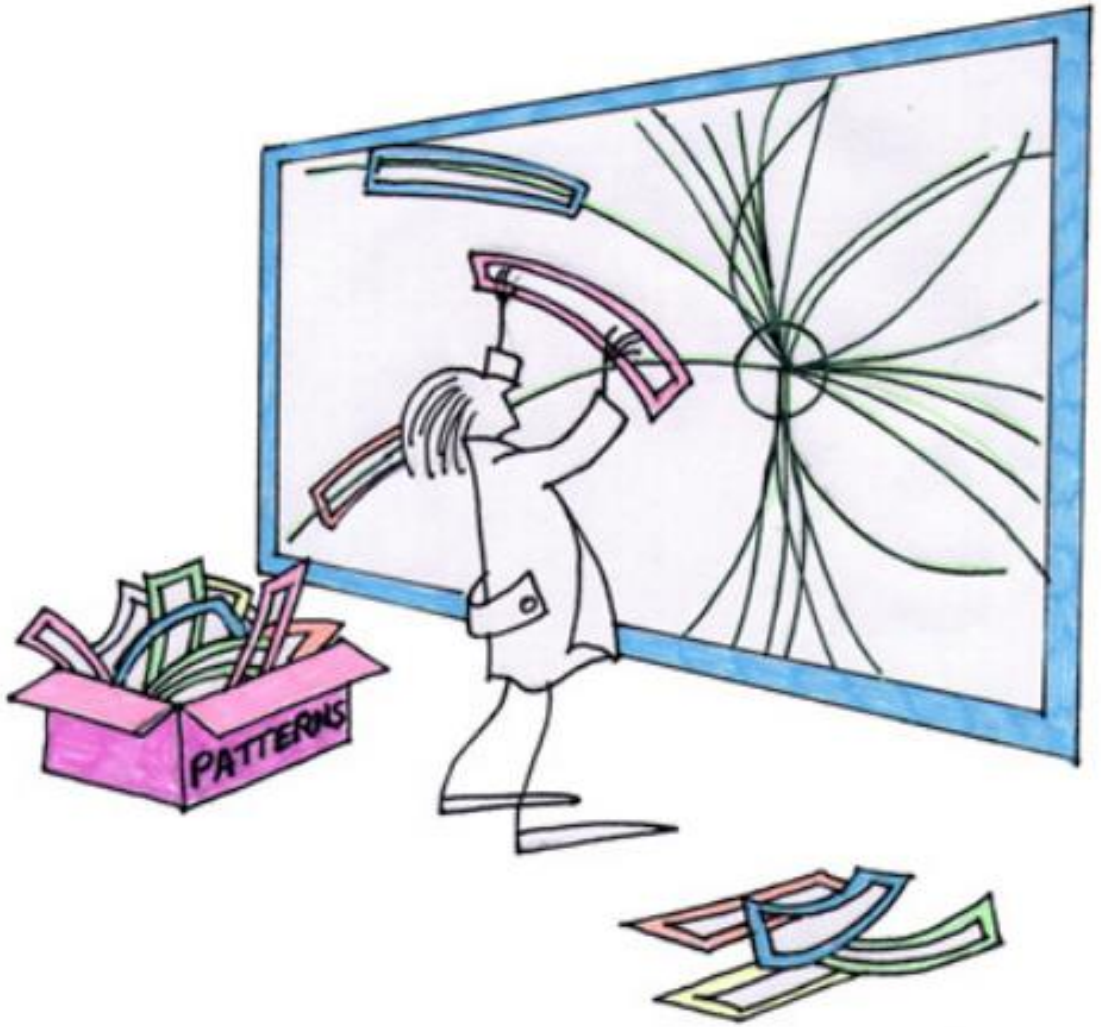
- If required, **full hit data** (strip and pixel) from EFPU to HTT
- **Full tracking** (pT > 1 GeV) performed
- **EFPU uses gHTT tracks**, in combination with offline-like analysis, to the reduce the rate to **10 kHz**



d0 resolution for 10 GeV muons as a function of η for the HTT first and second stage processing and offline tracking resolution.



Pattern Recognition Mezzanine

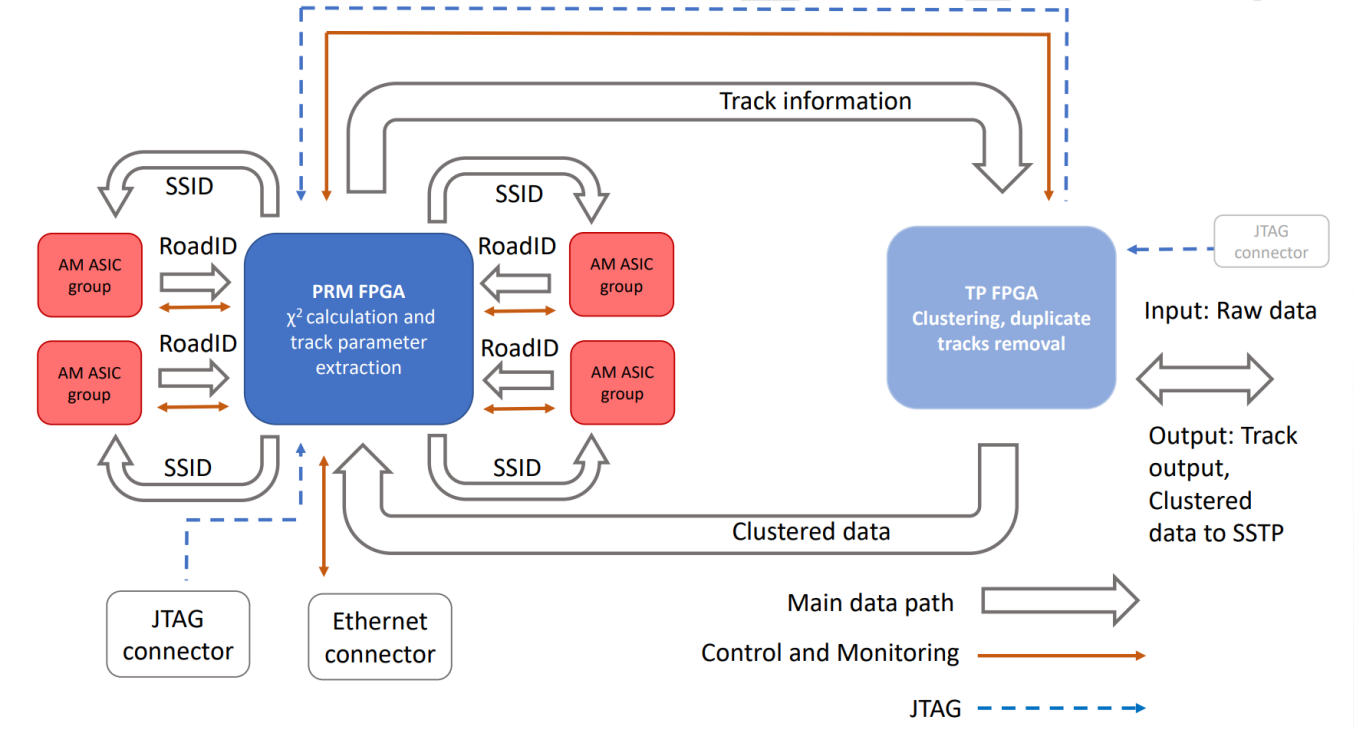


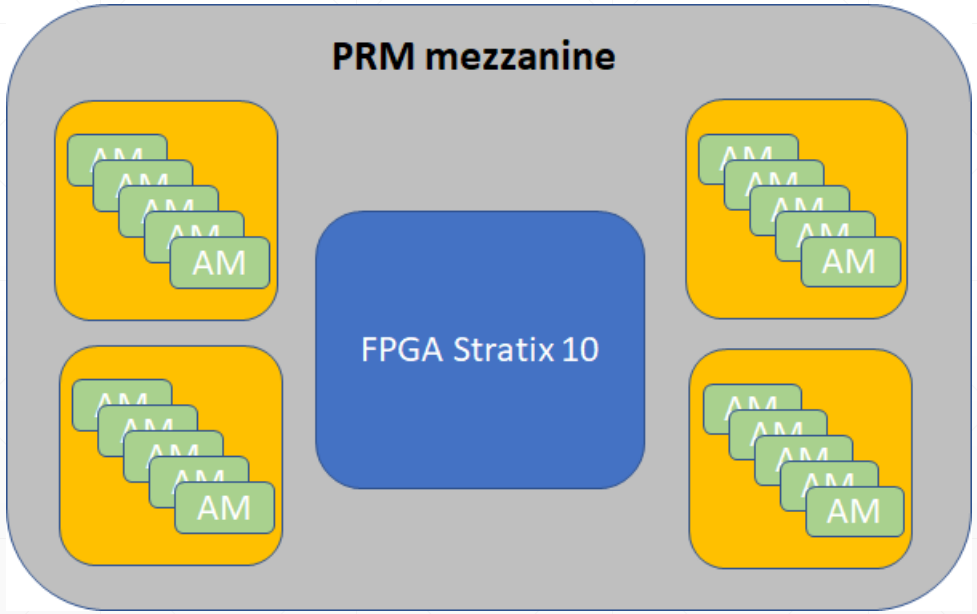
Pattern Recognition Mezzanine

Single mezzanine of about 17 x 30 cm, performing pattern recognition and track fitting.

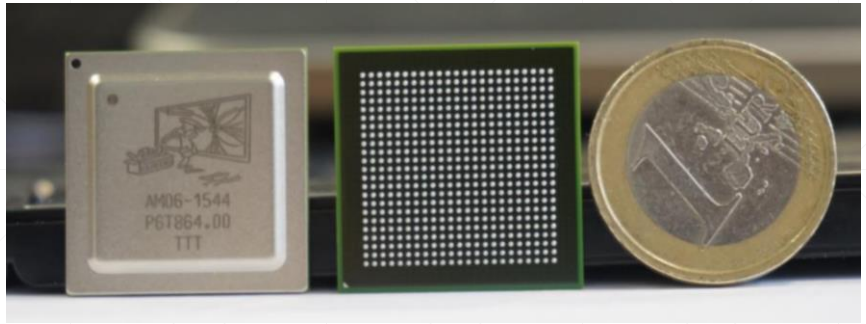
It mounts:

- **Intel Stratix 10 MX FPGA** and **AM ASICs**





What is an Associative Memory

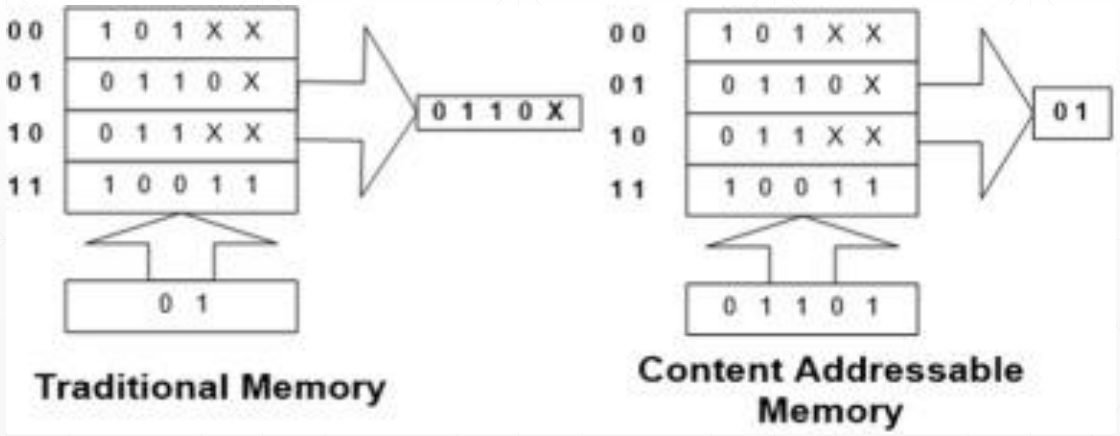


Random-Access Memory (RAM) is a standard type of memory:

- The search is done by address

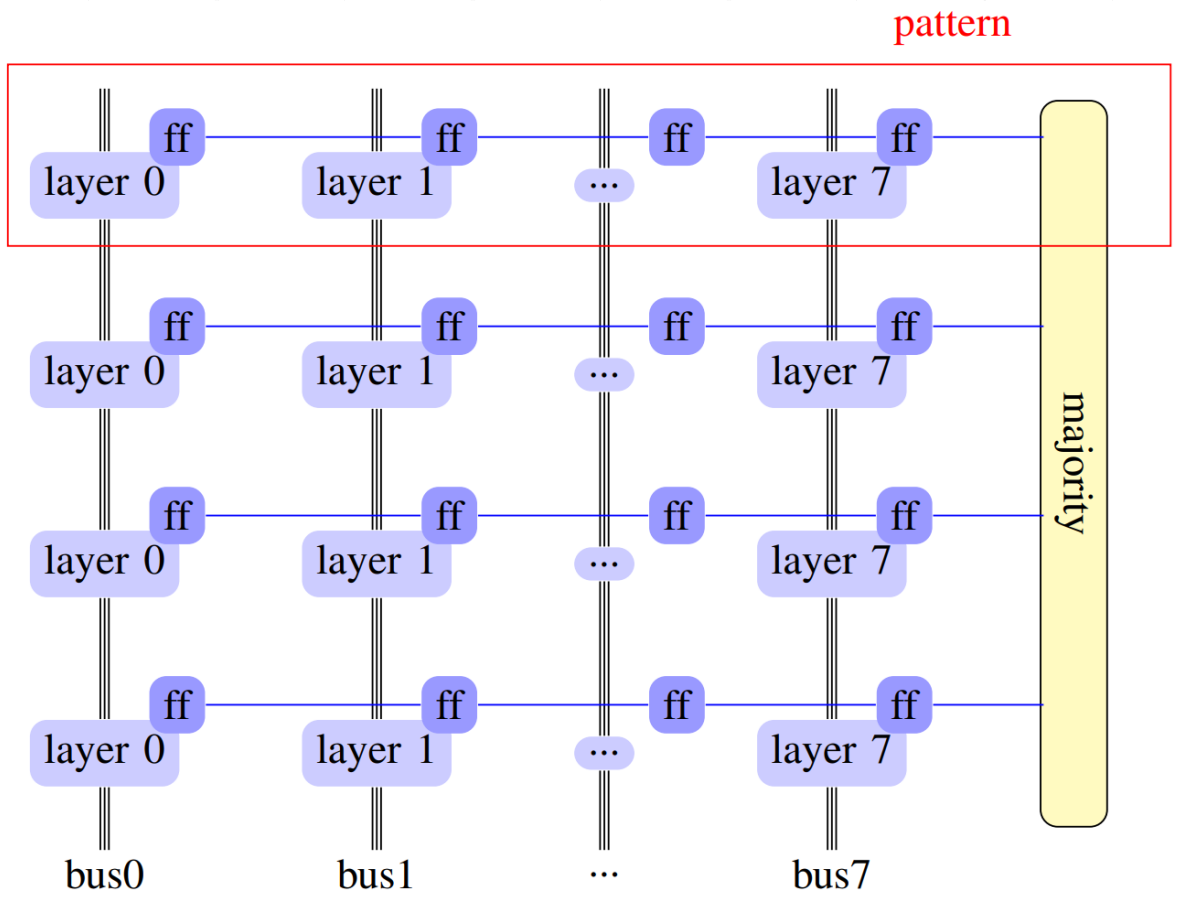
Content-Addressable Memory (CAM) is a special type of memory (also known as associative memory or associative storage):

- The search is done by content



How an AM ASIC works

- 8 bus_layers: corresponding to the detector layers
- Cluster information is sent along the bus
- When the content of the AM matches the hit the Flip Flop (FF) is set to 1.
- When more than >= 6 hits are matched along one line, a track is found

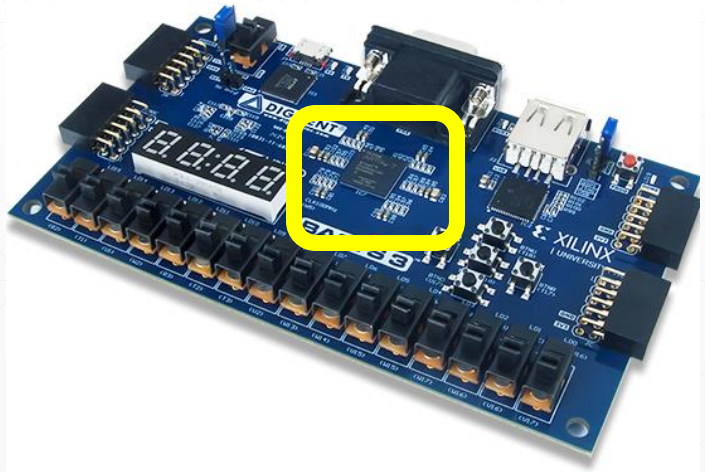


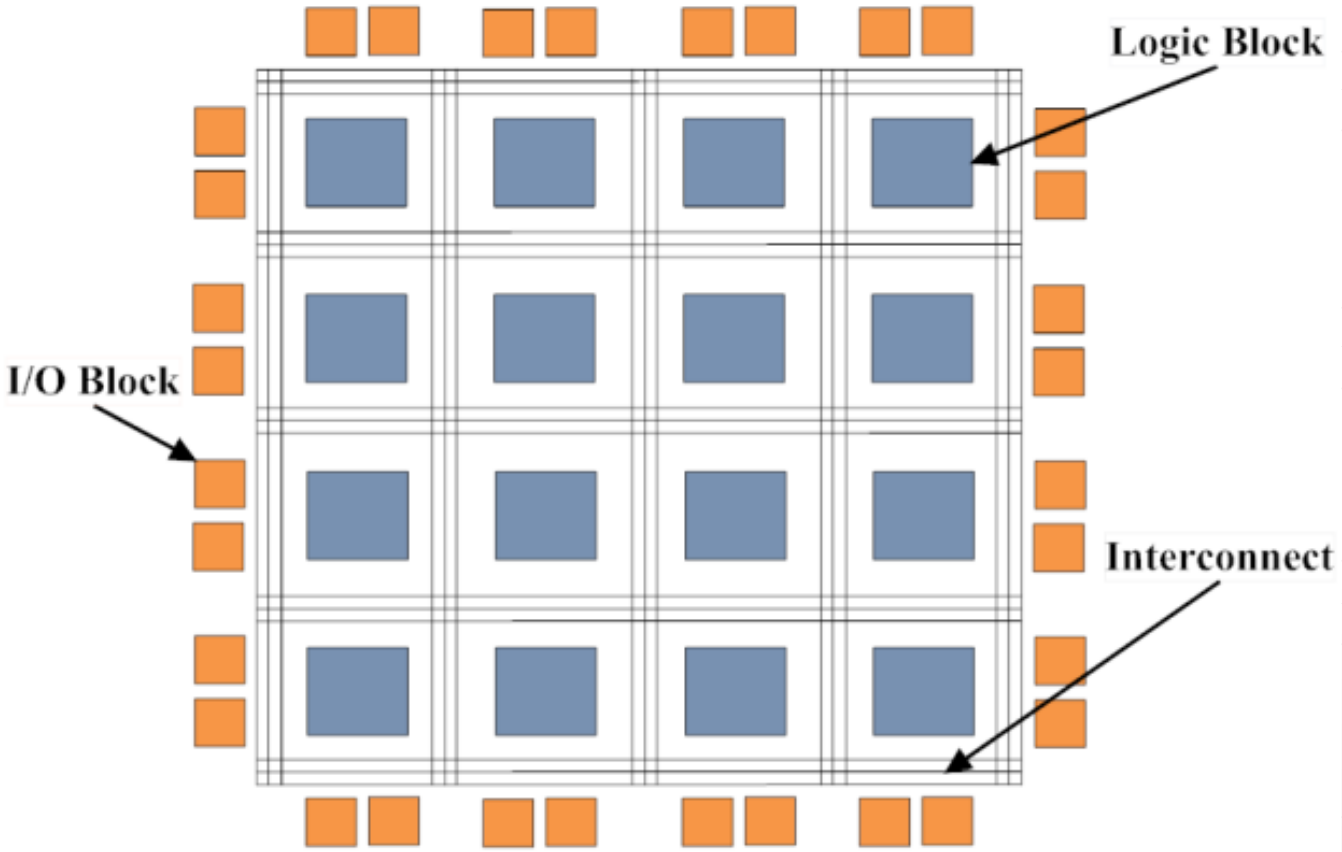
One custom AM ASIC version 09 will be able to store 384k patterns.

What is an FPGA

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based on a matrix of configurable **logic blocks** connected via programmable **interconnects**, surrounded by **IO blocks**.

FPGAs can be **reprogrammed to desired application** or functionality requirements after manufacturing





Where does it sit? On a board, e.g. a development kit. (Xilinx and Intel are the two major brands)

The language

Hardware description languages are used to design a digital circuit in FPGAs: e.g. VHDL and Verilog.

VHDL stands for VHSIC-HDL, that stands for Very High Speed Integrated Circuit Hardware Description Language.

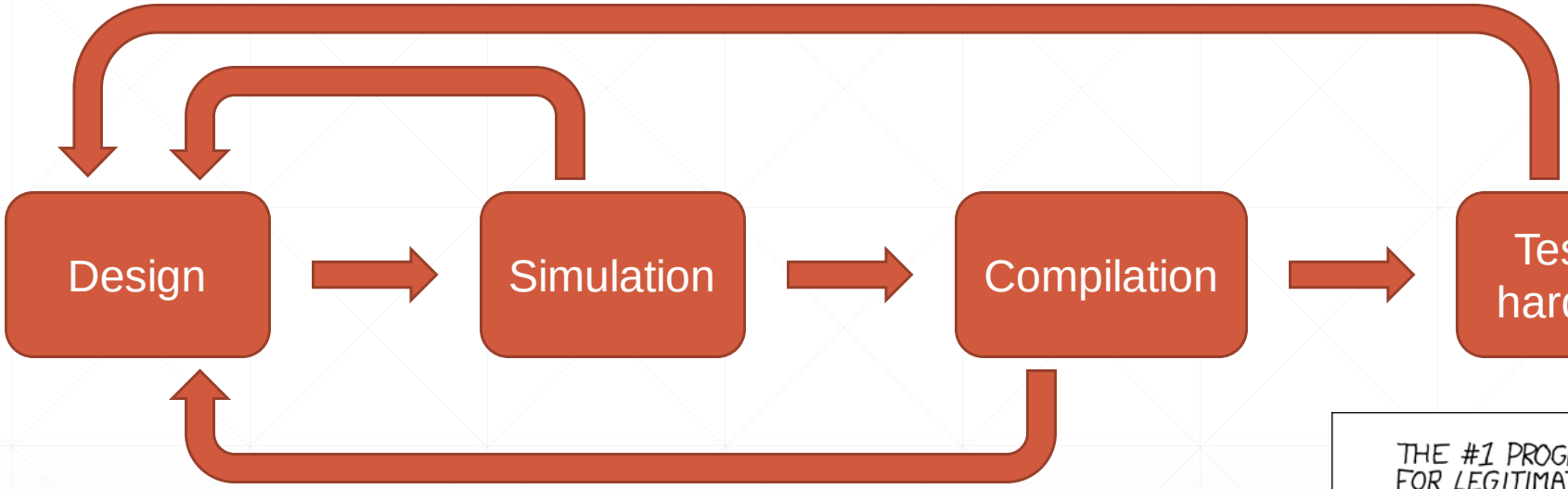
It's a **parallel language**, but it also contains constructs to execute serial commands (processes).

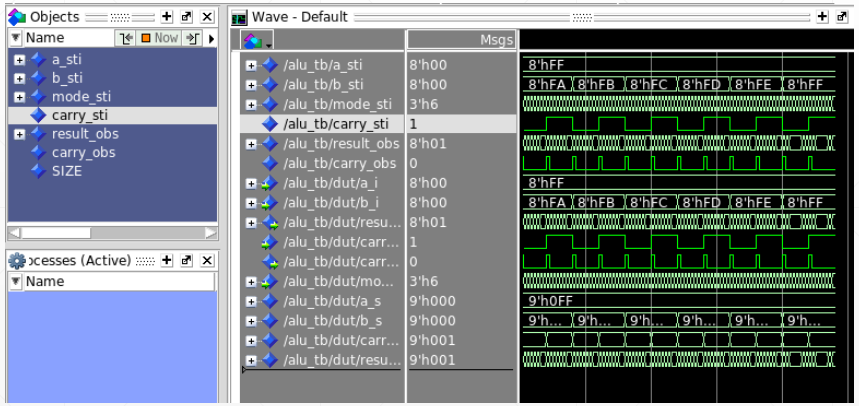
Within each **module** there is a procedural flow of instructions that looks somewhat like a small, self-contained software program - with variables, if statements, loops.
Each **module** can have one or more inputs as well as one or more outputs.
The inputs and output are specified within a structure called an **entity**, and the self-contained logic is defined in an **architecture**.

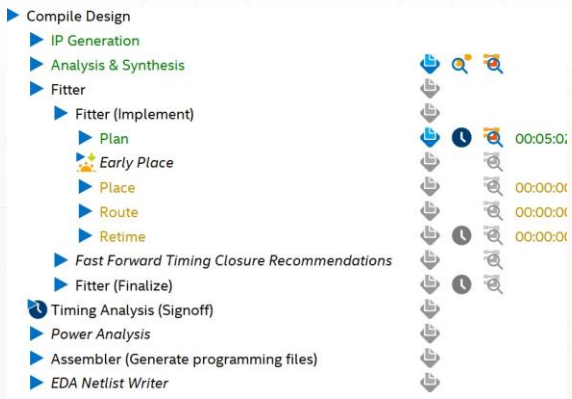
Module:
Counter FPGA Design in VHDL

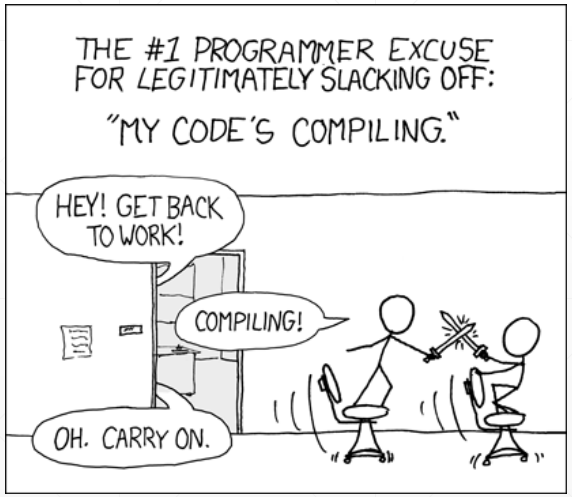
```
 1 library ieee;
 2 use ieee.std_logic_1164.all;
 3 use ieee.numeric_std.all;
 4
 5 entity signed_adder is
 6   port
 7   (
 8     aclr : in   std_logic;
 9     clk  : in   std_logic;
10     a    : in   std_logic_vector;
11     b    : in   std_logic_vector;
12     q    : out  std_logic_vector
13   );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27     begin
28       if (aclr = '1') then
29         q_s <= (others => '0');
30       elsif rising_edge(clk) then
31         q_s <= ('0'&signed(a)) + ('0'&signed(b));
32       end if; -- clk'd
33     end process;
34
35 end signed_adder_arch;
```

Design flow FPGA

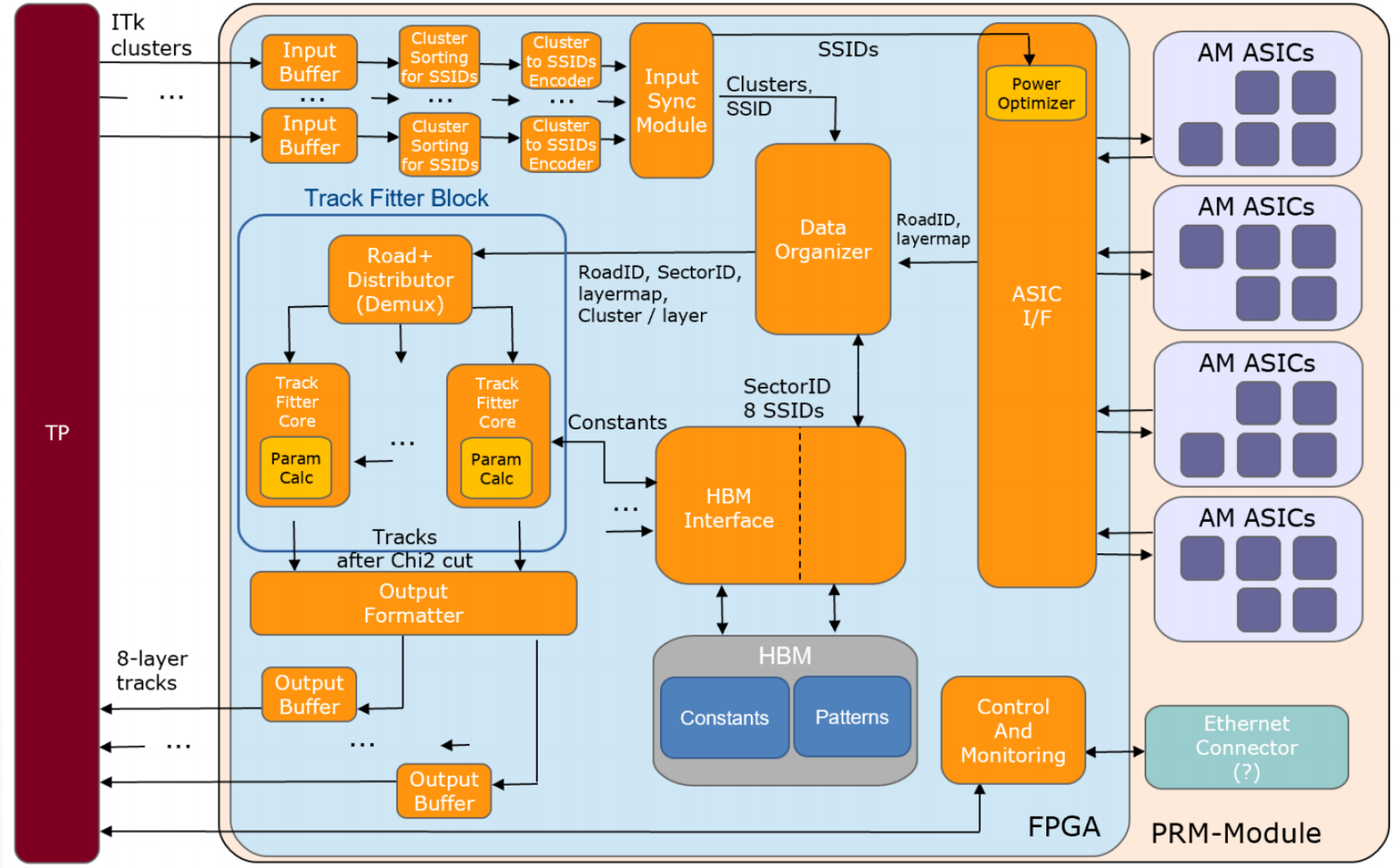






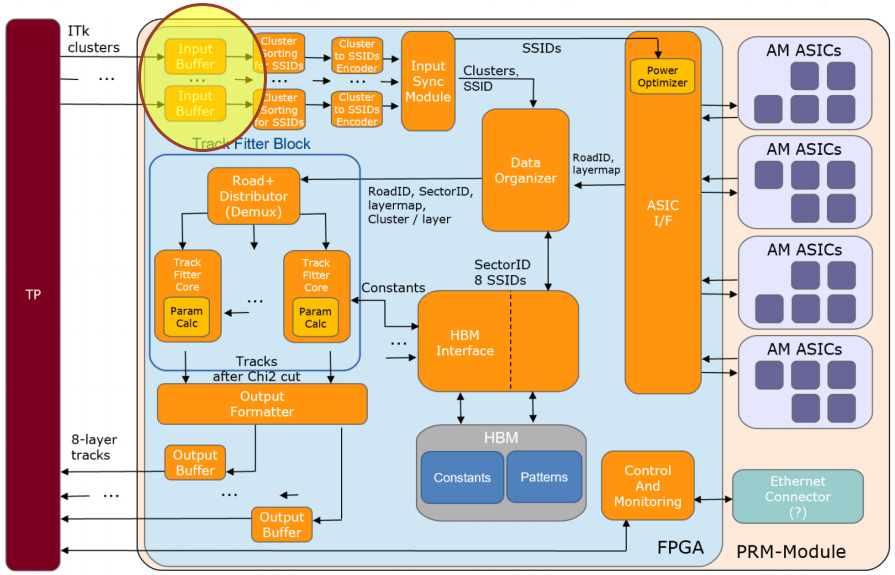


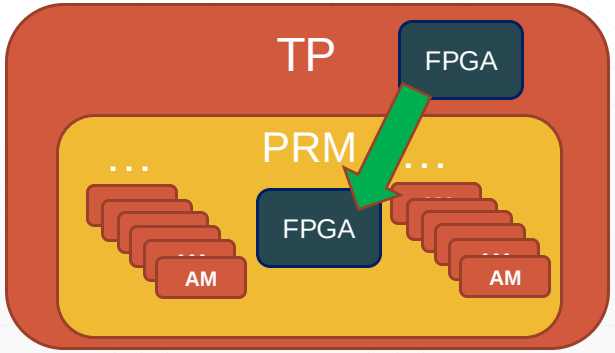
PRM firmware



Step by step

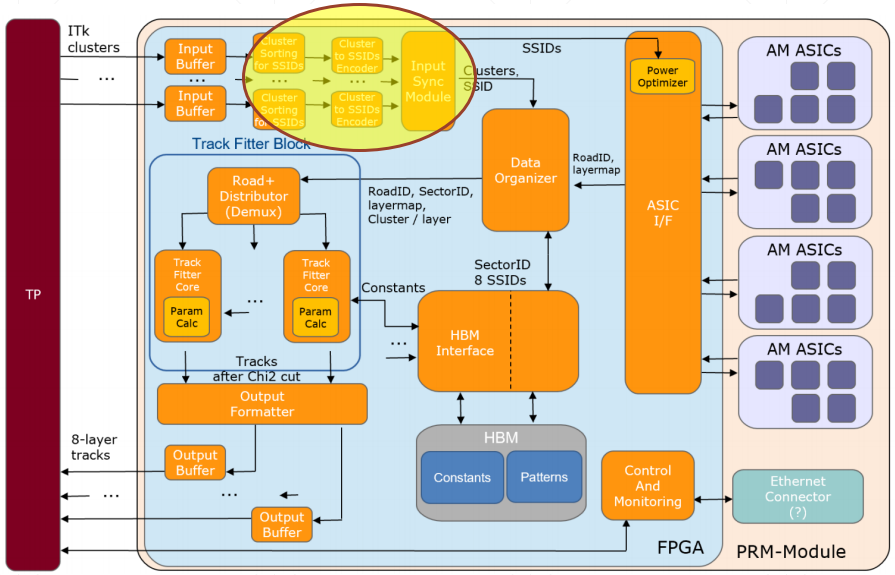
- Clusters of hits are sent from TP to PRM

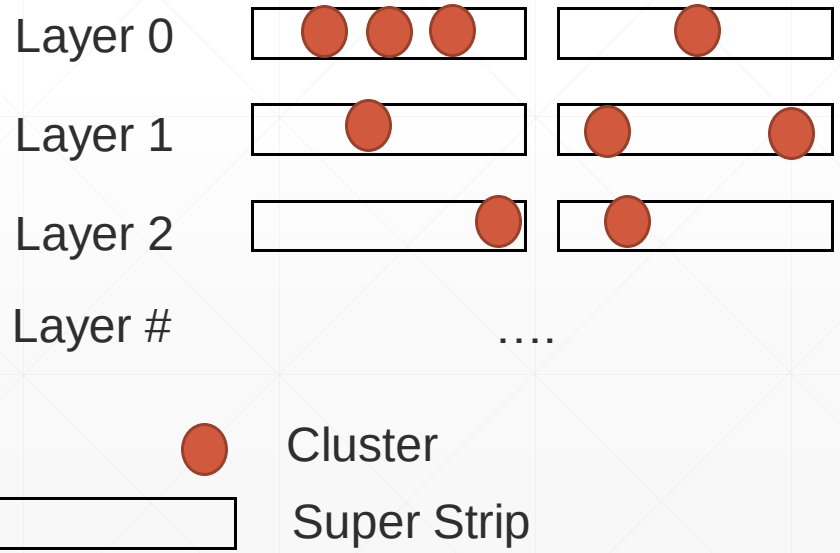




Step by step

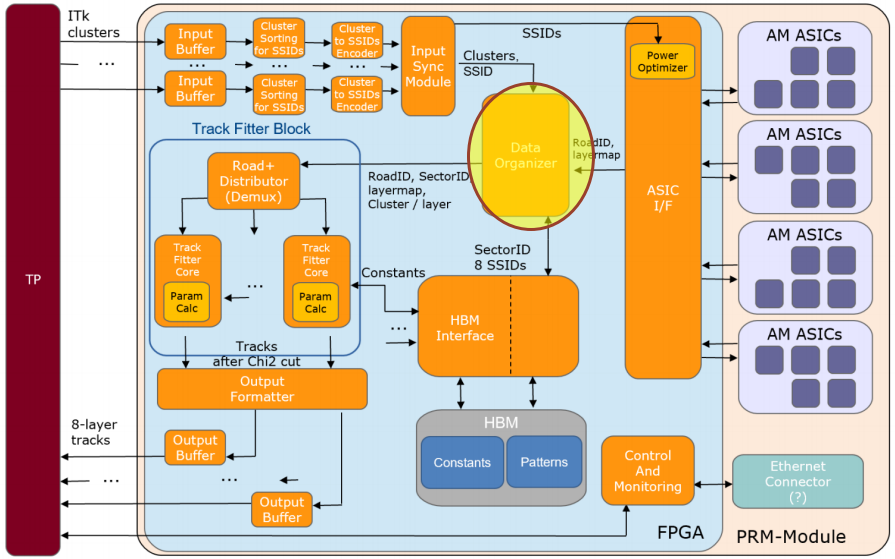
- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)

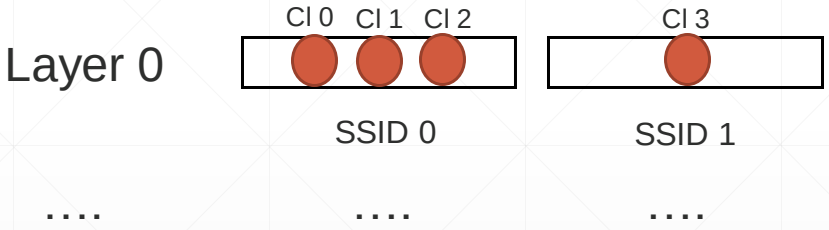




Step by step

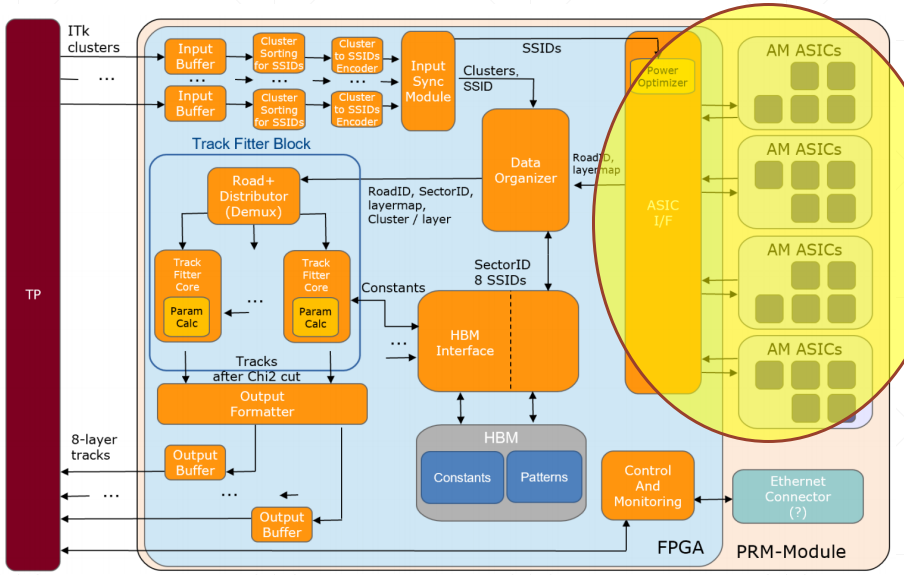
- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories

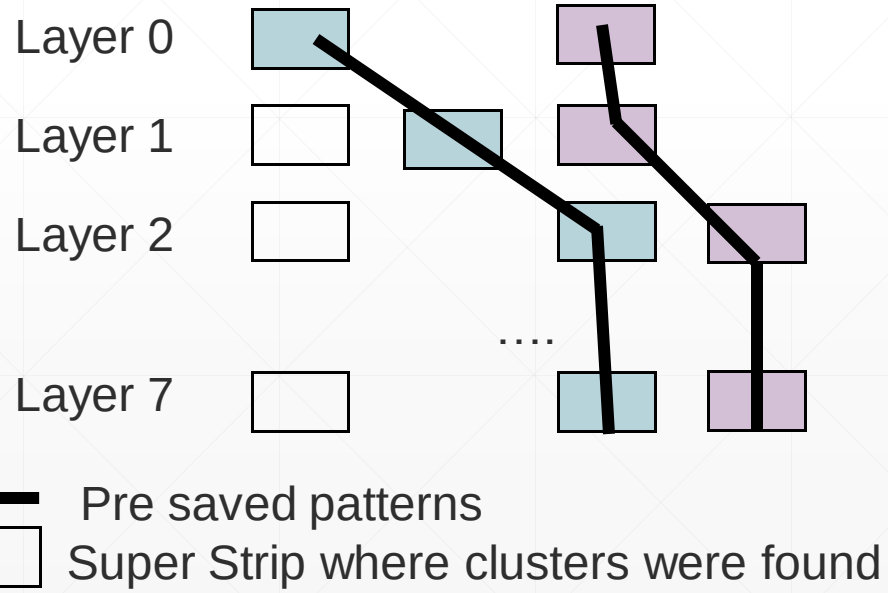




Step by step

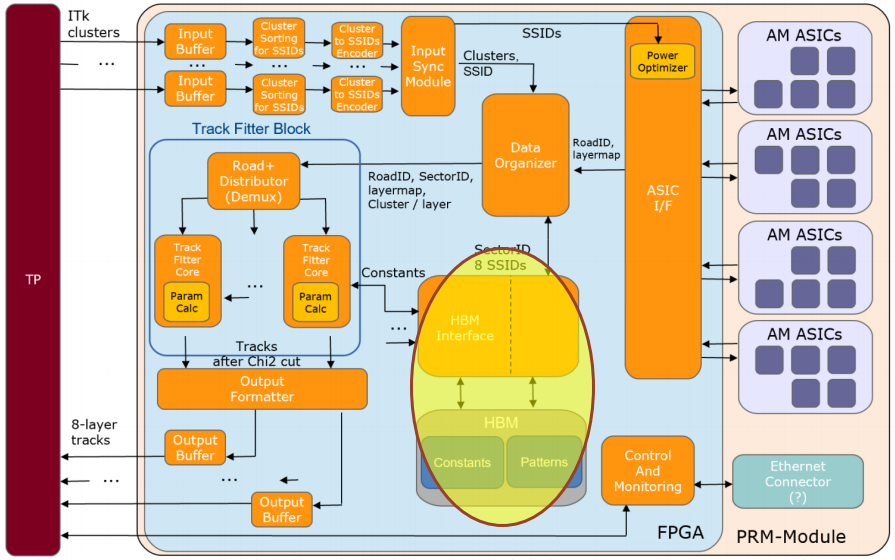
- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition

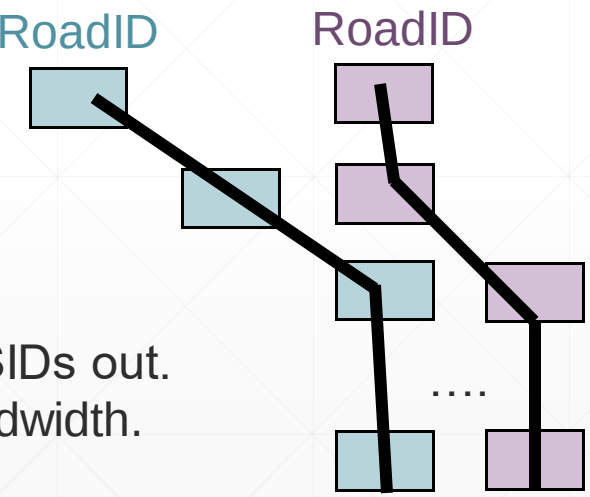




Step by step

- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)



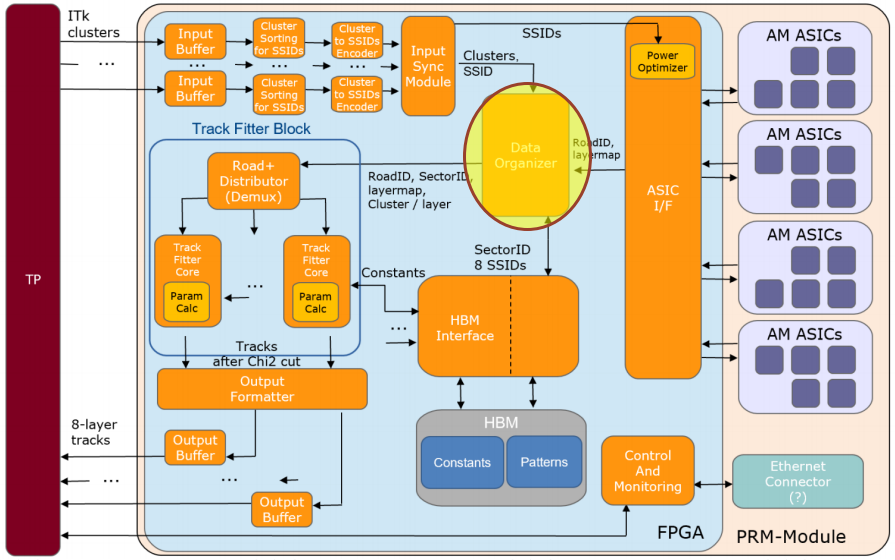


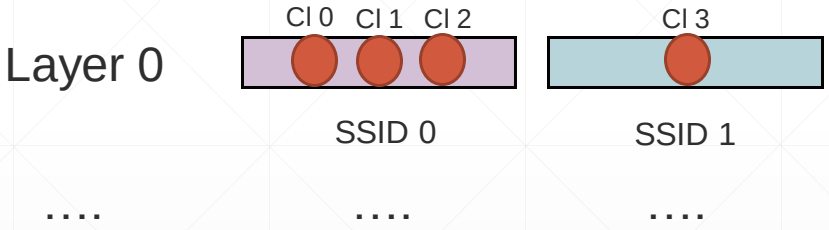
AM ASIC can't send all the SSIDs out.
It would require too much bandwidth.

Easier to send out an identifier and retrieve the list of SSID from a memory.

Step by step

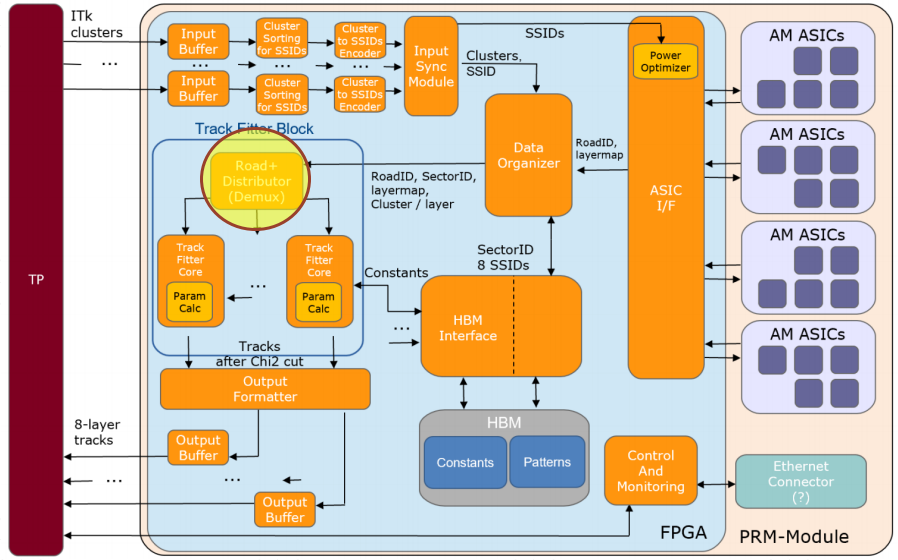
- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)
- Clusters corresponding to a specific SSID are retrieved from the memory

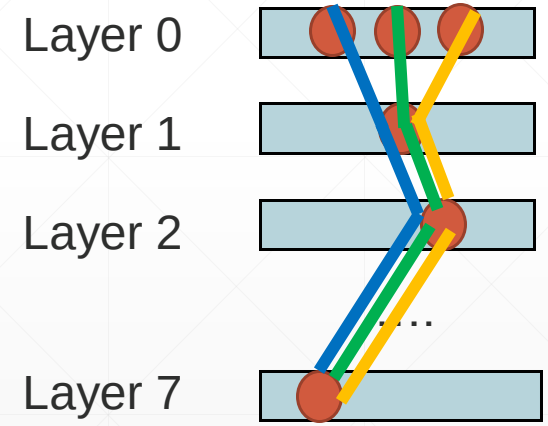




Step by step

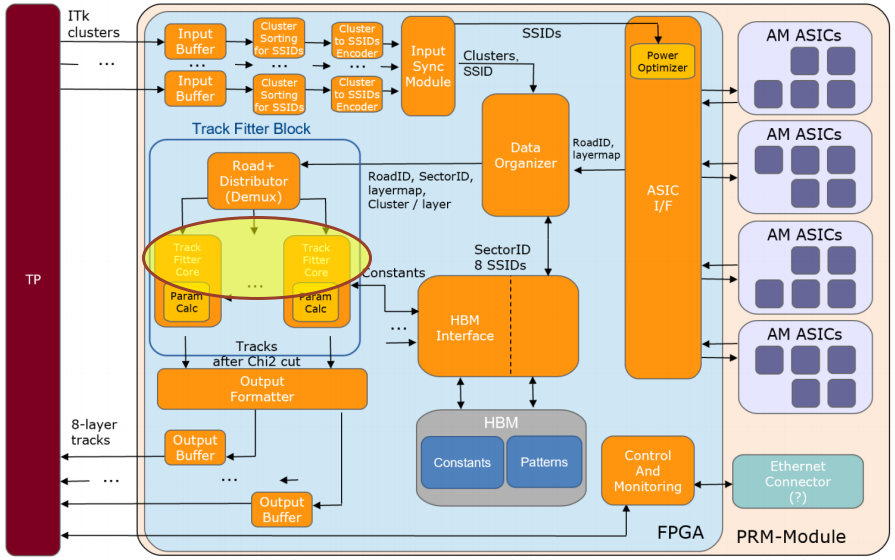
- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)
- Clusters corresponding to a specific SSID are retrieved from the memory
- Clusters from the different layer are sent to the road distributor to form candidates





Step by step

- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)
- Clusters corresponding to a specific SSID are retrieved from the memory
- Clusters from the different layer are sent to the road distributor to form candidates
- Chi square is computed per each candidate (a threshold is then applied)

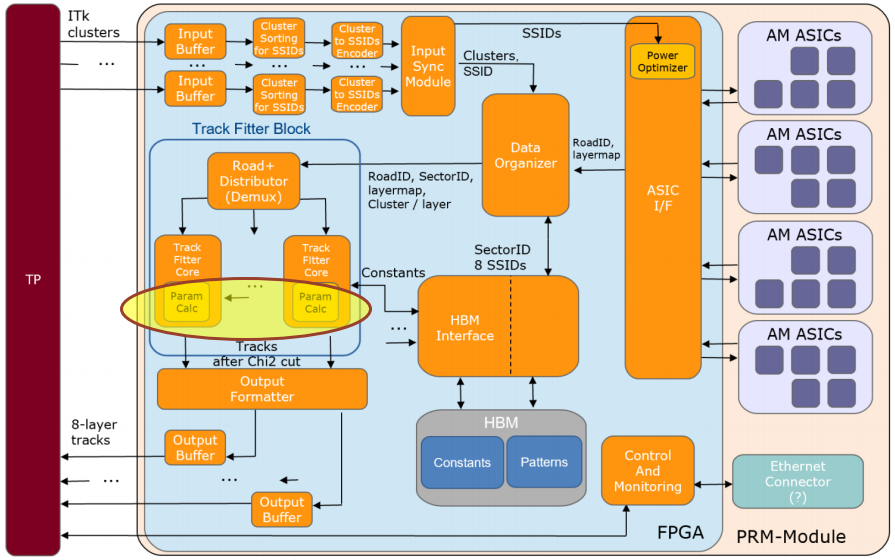


$$\chi^2 = \sum_{i=1}^{N_{\text{dof}}} \left[\left(\sum_{j=1}^{N_{\text{coo}}} S_{ij} x_j \right) + h_i \right]^2$$

Name	Description
N_{pix}	Number of pixel layers used
N_{str}	Number of strip layers used
N_{coo}	Total number of local coordinates for each cluster: two for each pixel, one for each strip
N_{par}	Number of track helix parameters: d_0, z_0, ϕ_0, q/p, $cot(\theta_0)$ (or equivalent)
N_{dof}	Number of χ^2 degrees of freedom

Step by step

- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)
- Clusters corresponding to a specific SSID are retrieved from the memory
- Clusters from the different layer are sent to the road distributor to form candidates
- Chi square is computed per each candidate (a threshold is then applied)
- Track parameters for the candidates that have passed the selection are computed

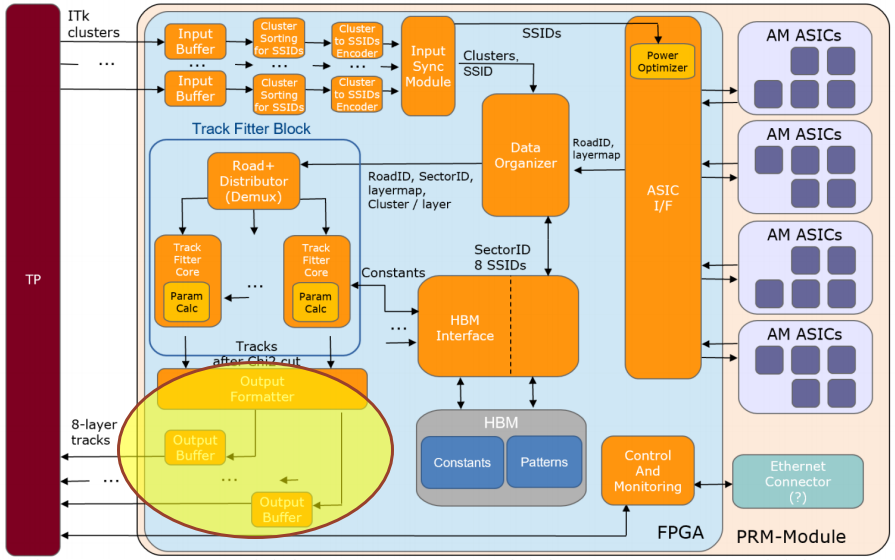


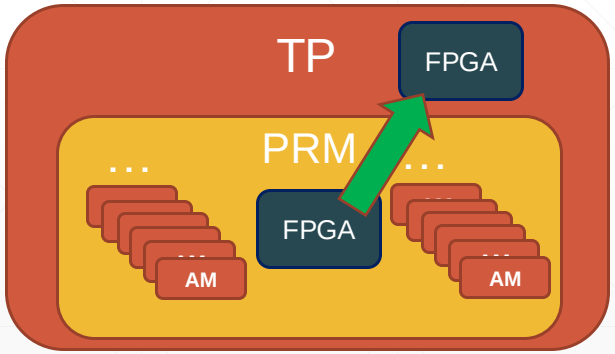
$$p_i = \sum_{j=1}^{N_{coo}} \left(C_{ij} x_j + q_i \right)$$

Name	Description
N_{pix}	Number of pixel layers used
N_{str}	Number of strip layers used
N_{coo}	Total number of local coordinates for each cluster: two for each pixel, one for each strip
N_{par}	Number of track helix parameters: d_0, z_0, ϕ_0, q/p, $cot(\theta_0)$ (or equivalent)
N_{dof}	Number of χ^2 degrees of freedom

Step by step

- Clusters of hits are sent from TP to PRM
- Clusters are converted in a coarser information called Super Strip Identifier (SSID)
- Clusters and the corresponding SSID are saved in memories
- SSID are sent to the ASIC interface and then to the AM ASIC for pattern recognition
- AM ASICs give back a Road Identifiers that correspond to a list of SSID (which is pre-saved in a memory and is pattern dependent)
- Clusters corresponding to a specific SSID are retrieved from the memory
- Clusters from the different layer are sent to the road distributor to form candidate
- Chi square is computed per each candidate (a threshold is then applied)
- Track parameters for the candidates that have passed the selection are computed
- Data are sent back to the TP





Few examples from the current development

Chi square implementation

Implementation	fmax(@200MHz)	Latency(in cycles)	DSP's	ALM's	Register's	Pipelined
Parallel	418	22	188	0	0	YES
Optimized for DSP	188	84	2	24	12	NO
Cascading	243	22	13	19	-	NO
Cascading Pipelined	Approx. 225	22	91	249	+1000	YES

Possible circuits | Maximum allowed speed | Processing Time | Resource Usage | Feature

FPGA pinout definition

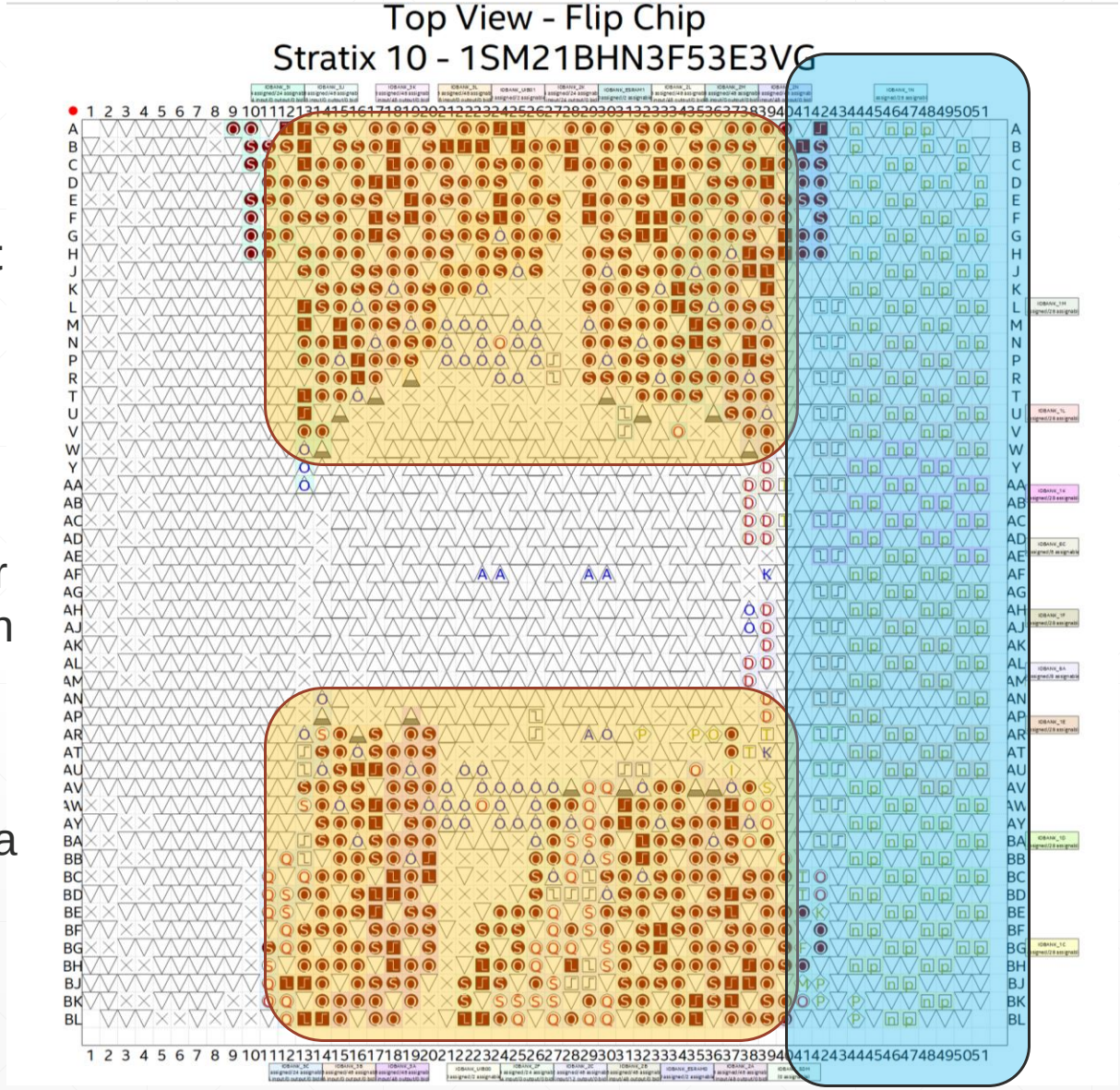
We have to connect to the FPGA pins **all the important signals** for both data transmission and monitoring.

This is very much related to the **board design**!!

In yellow are the general purpose pins: used e.g. for the I2C lines for communication with temperature or monitor sensors but also for the LVDS pairs to communicate with the Associative Memory custom ASICs.

In blues are the transceiver pins: used for fast communication, e.g. to receive and transmit the ITK data and the reconstructed tracks from and to the TP board.

Most of the assignment is done **manually**.



FPGA monitoring

How to monitor the behavior (temperature, voltage..) of the FPGA on the board?
How to save some of the data passing through it?
How to change some settings on the fly?
How to reset the FPGA from far away?

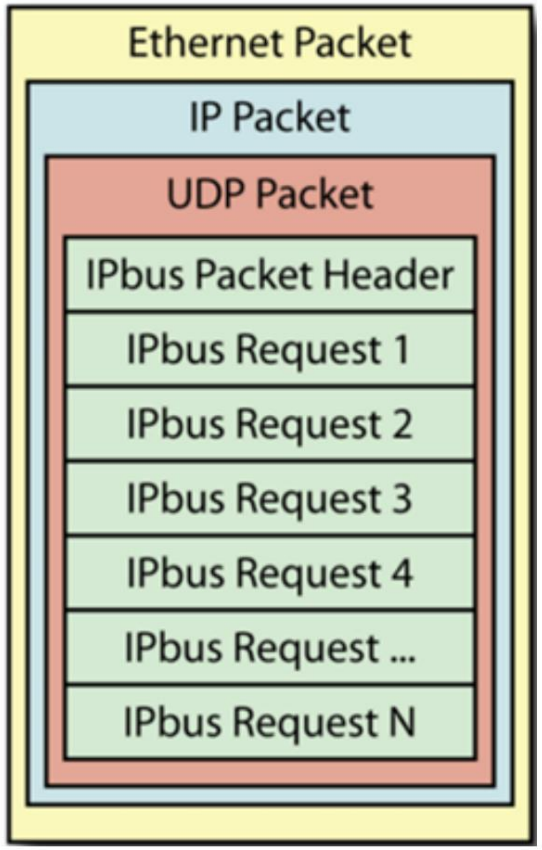
Firmware on the FPGA and software on a computer need to communicate.

The most common way of doing it via Ethernet:
sending and receiving packets to and from the FPGA.

Ethernet and IPbus protocols

Structure of the Ethernet protocol.

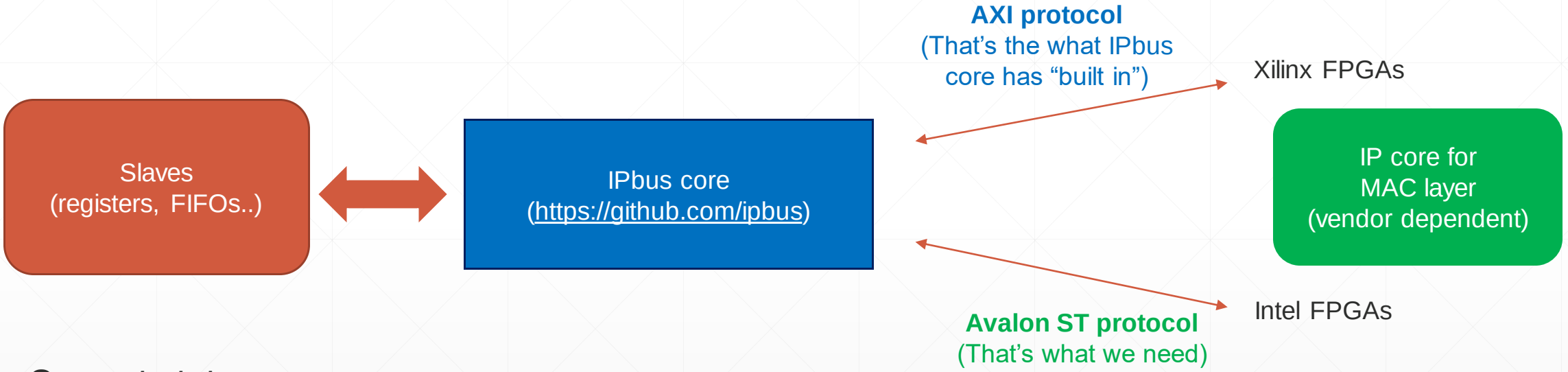
IPbus wrapped Ethernet packet



Each Ethernet frame starts with an Ethernet header, which contains destination and source MAC addresses as its first two fields. The middle section of the frame is payload data including any headers for other protocols (for example, Internet Protocol) carried in the frame. The frame ends with a 32-bit cyclic redundancy check (CRC) used to detect any in-transit corruption of data.

IPbus for monitoring (2)

IPbus core has been prepared for Xilinx FPGAs.
But our project (like many others) uses Intel FPGAs.



Current status:

- The project compiles
- Simulation is under preparation

Using a **converter** from AXI to Avalon ST

Conclusions

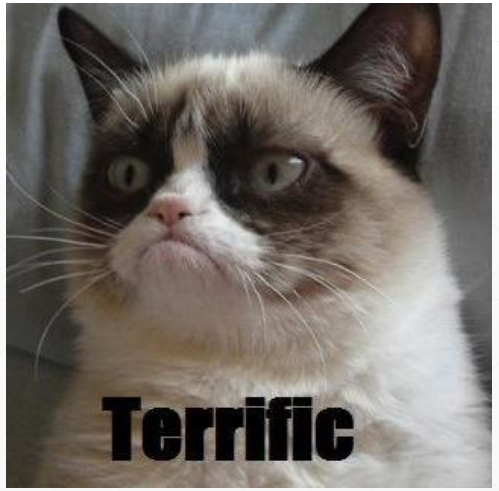
Conclusions

Plans for the ATLAS Trigger and Data Acquisition systems for the High-Luminosity Upgrades are detailed in a Technical Design Report.

The **Hardware Track Trigger** is crucial for the track reconstruction in TDAQ.

The use of **state-of-the-art FPGA and dedicated algorithm** will allow HTT to successfully operate in the challenging scenario foreseen for HL- LHC.

The **Pattern Recognition Mezzanine** is a key element for HTT.
The development of performant hardware and firmware will be necessary to achieve the ATLAS physics goals.



Thanks for your attention!

Backup

Trigger menu

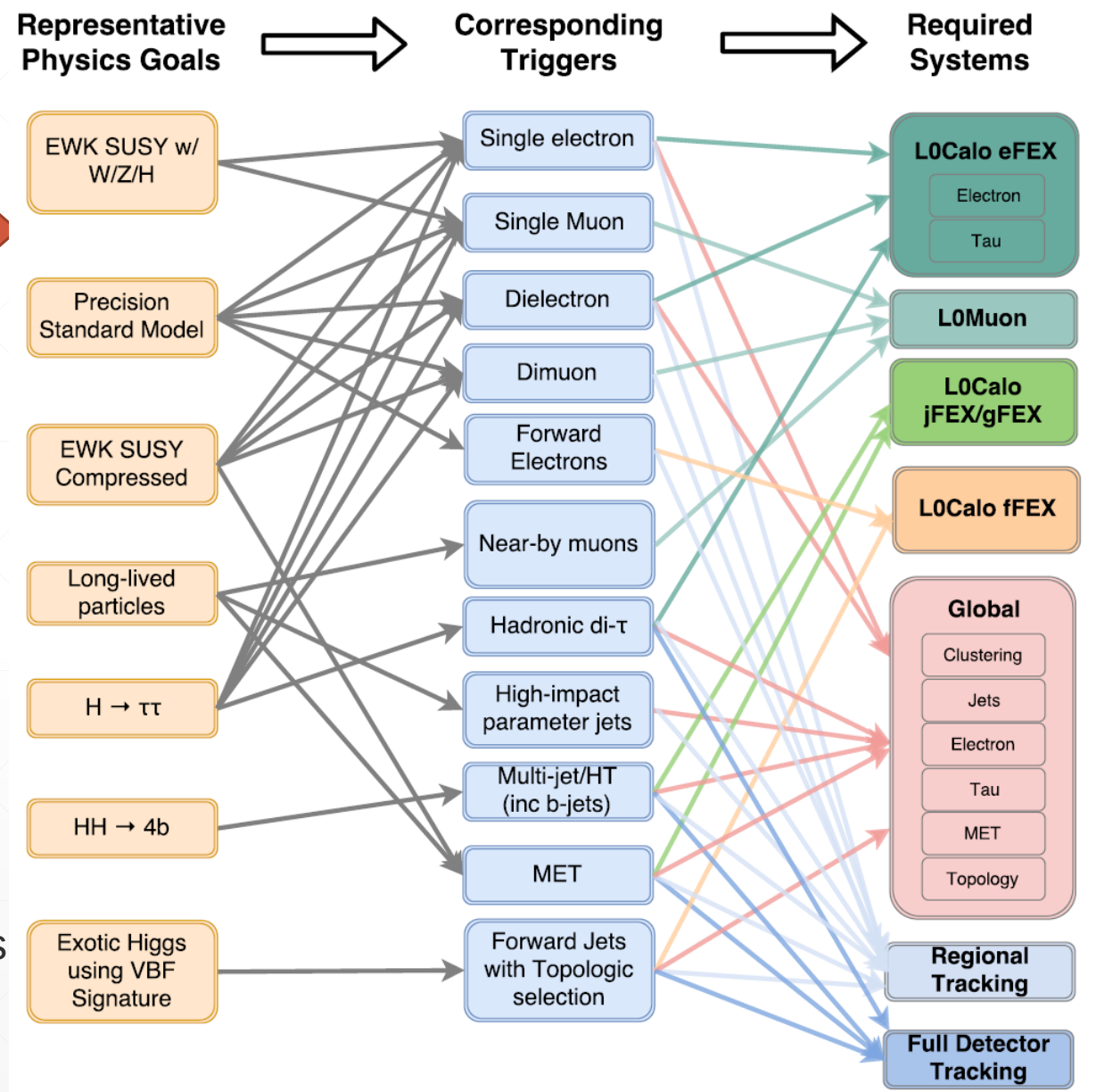
Schematic flow from the representative set of physics goals to the hardware systems needed to achieve them. The middle column lists the **corresponding triggers required.**

E.g.:

1. Global Trigger enables a low-pT electron trigger at Level-0 and then regional-tracking reduces the high rate early in the Event Filter processing.

2. Global Trigger enables low thresholds for multi-jet and Et(miss) triggers at Level-0, then regional tracking and full-detector tracking reduce the background acceptance rate while preserving the physics acceptance.

Definite plans for the Run 4 trigger menu will come towards the end of Run 3.



Phase-II TDAQ upgrade

Three main systems of the TDAQ Phase-II upgrade architecture:

- Level-0 Trigger
- DAQ (Readout and Dataflow subsystems)
- Event Filter

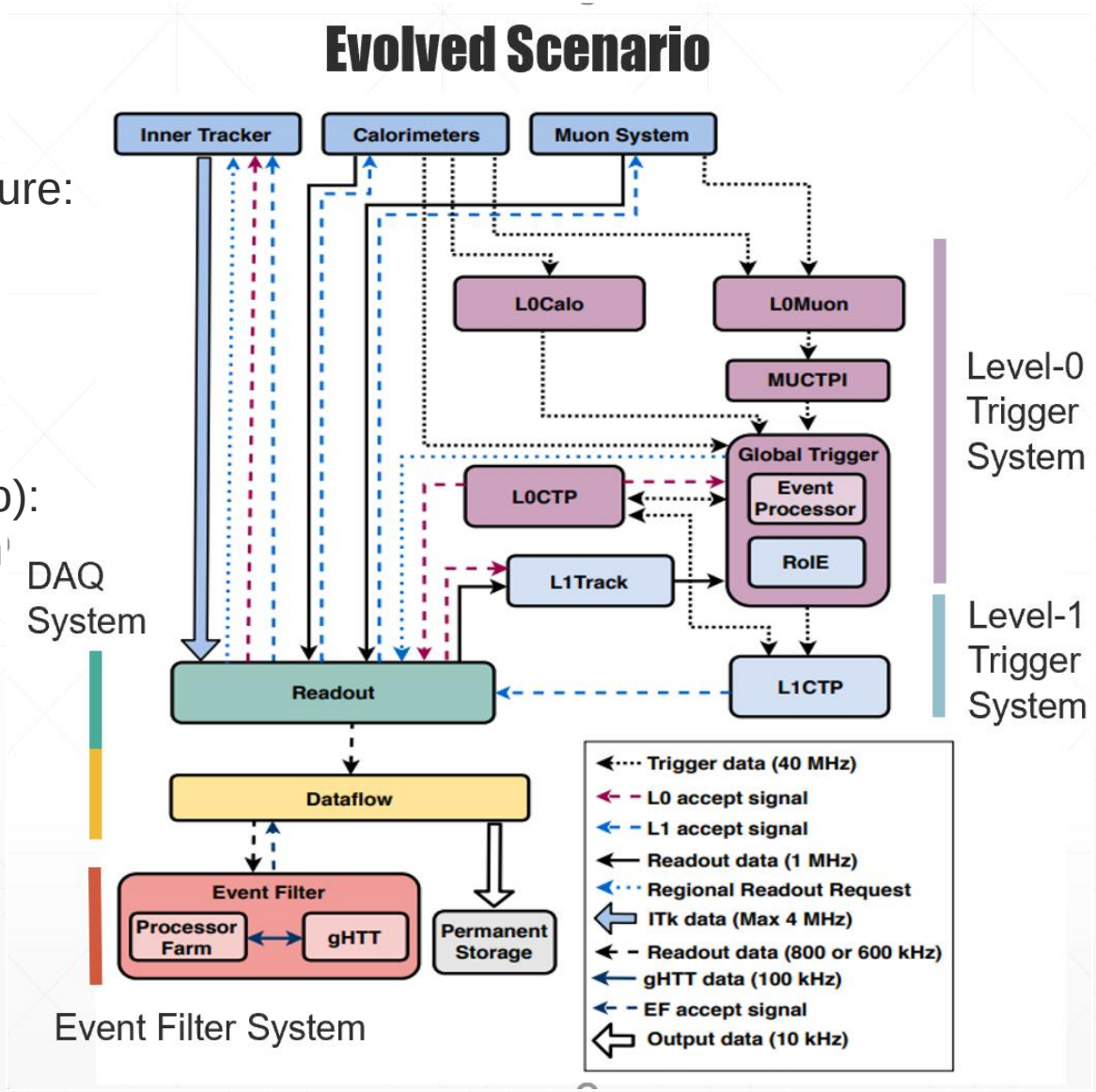
Single-hardware-level trigger architecture (baseline scenario):

➢ capable of evolving into a two-level hardware trigger system (evolved scenario)

The two main **criteria for** an evolution to the **split-level hardware trigger** configuration:

- the hadronic trigger rates
- the inner pixel detector layer occupancies

If either or both are higher than expected, the baseline TDAQ architecture would restrict the trigger menu at the ultimate HL-LHC running conditions.



Reference [1]

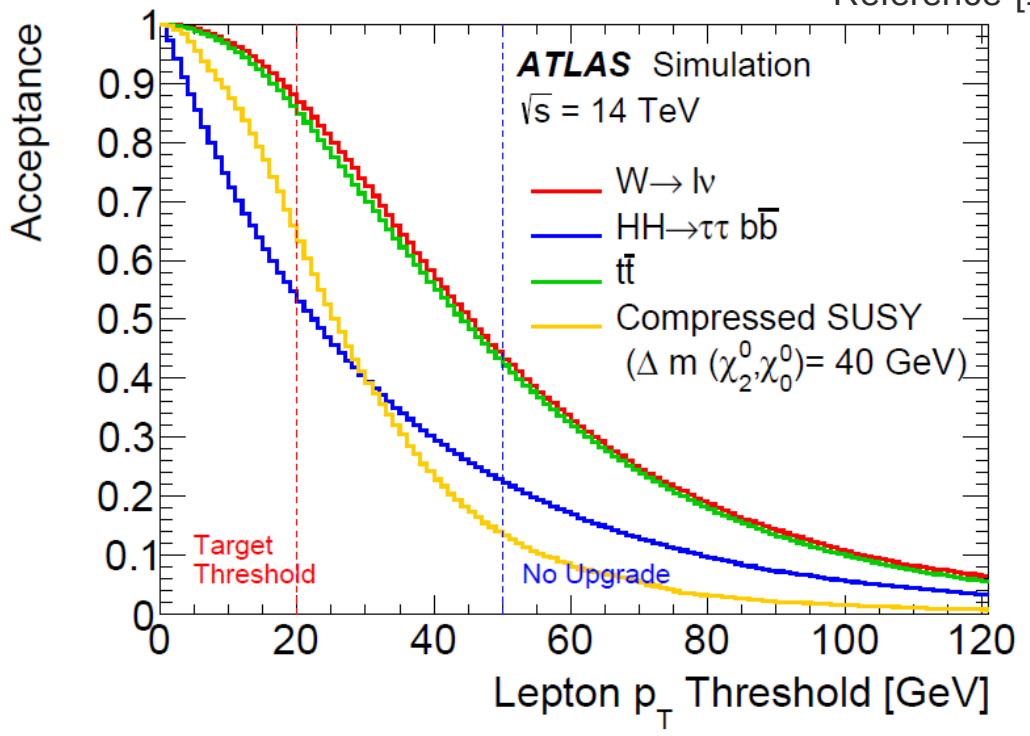
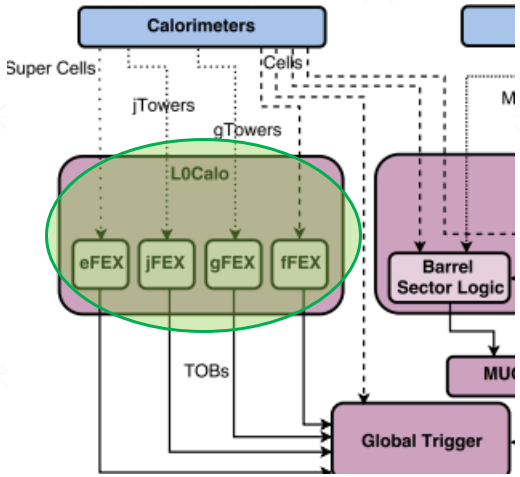


Figure 2.2: *The integrated acceptance as a function of the single lepton p_T threshold for four representative channels: $W \to \ell\nu$, $H \to \tau\tau b\bar{b}$, $t\bar{t}$, and a compressed spectrum SUSY model relevant for "Well-tempered Neutralino" motivated models. The Phase-II TDAQ upgrade would enable lowering the single lepton Level-0 threshold to 20 GeV from 50 GeV, the projected threshold without the upgrade.*

Level-0 Calo



Calorimeters data with coarse granularity are sent to the Feature Extractors (FEXs).

- **eFEX**(1): electron and photon object identification
- **jFEX**(1): single jets identification
- **gFEX**(1): large-R (or multi-jet) triggers identification and global quantities calculation
- **fFEX**(2): forward electromagnetic (forward jet) trigger objects reconstruction at high η

The **Global Trigger refines the identification algorithm** by taking advantage of the transmission of fine-granularity cells

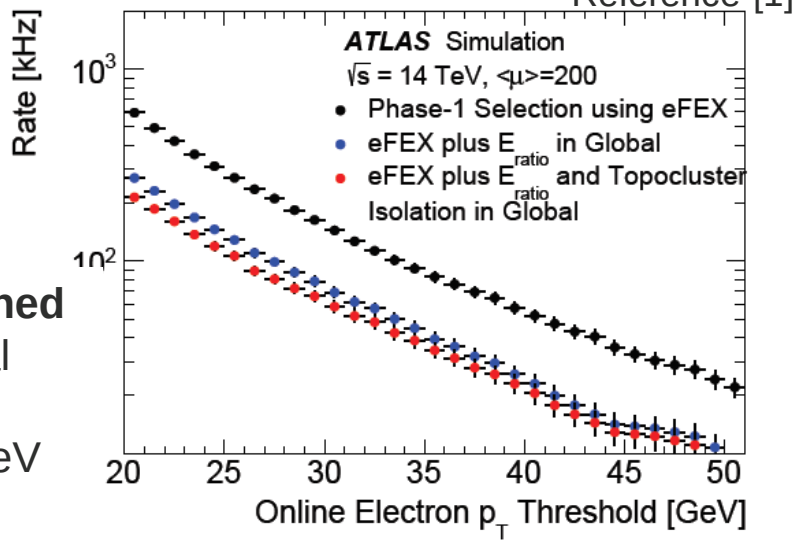
- trigger thresholds can be chosen reasonably low to cover the whole physics program

Subsystem	Trigger Object	Approximate Granularity	Coverage $\|\eta\|$
eFEX	$e/\gamma, \tau$	Super Cells (10 in 0.1×0.1)	< 2.5
jFEX	τ, jet, E_T^{miss}	0.1×0.1	< 2.5
jFEX	τ, jet, E_T^{miss}	0.2×0.2	$2.5 - 3.2$
jFEX	τ, jet, E_T^{miss}	0.4×0.4	$3.2 - 4.9$
gFEX	Large-R jet, E_T^{miss}	0.2×0.2	< 4.9
fFEX	e/γ	Full detector EMEC, HEC, FCal	$2.5 - 4.9$
fFEX	jet	Full detector FCal	$3.2 - 4.9$

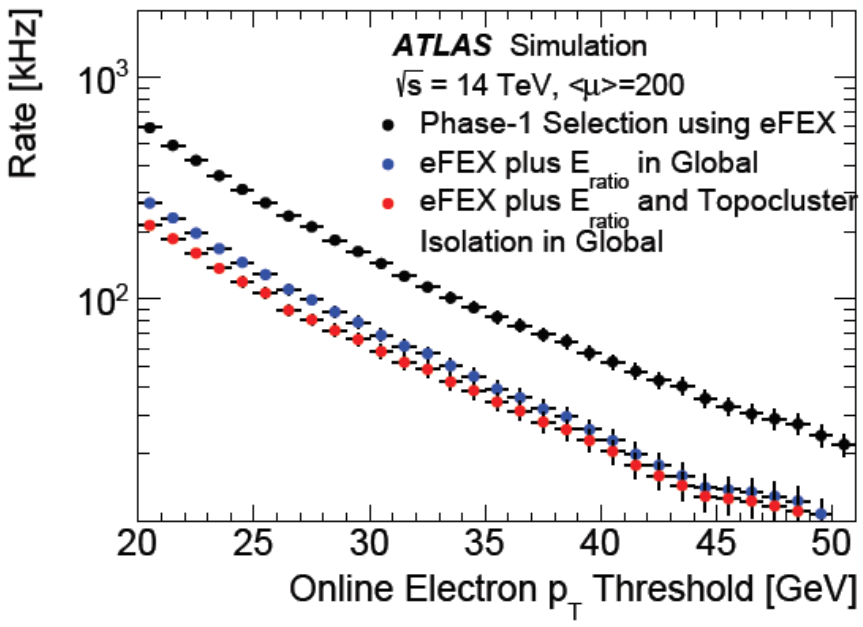
(1) Phase-I Upgrade hardware, firmware upgrade foreseen for Phase-II
(2) New Phase-II system

The **performance of the combined system** (Level-0 Calo and Global Trigger) allows to set the single electron threshold at about 20 GeV

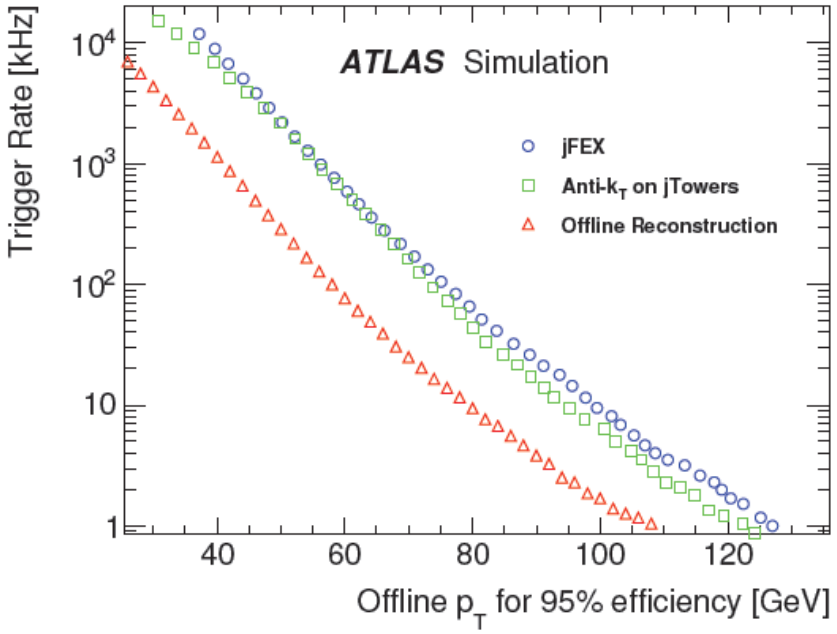
Reference [1]



Reference [1]



(a)



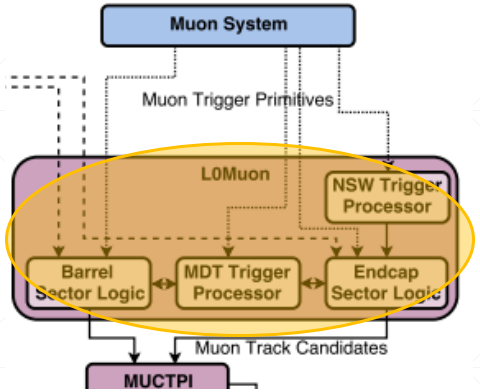
(b)

Figure 7.1: *a) Level-0 trigger rates for electrons. The different curves are for the successive application of veto conditions. b) The forward ($|\eta| > 3.2$) single-jet trigger rate vs. offline p_T thresholds for jets reconstructed in the jFEX. The efficiency is evaluated using $HH \rightarrow bbbb$ signal events, and the trigger rate is evaluated based on minimum bias background events at $\langle\mu\rangle \simeq 200$. The jFEX algorithm, the offline anti-k_t algorithm (run over $\eta \times \phi = 0.1 \times 0.1$ towers), and the full offline reconstruction are compared.*

Level-0 Muon

Based on the data of the upgraded muon spectrometer and the Tile calorimeter

- Improvements in trigger performance will be achieved by increasing
  - detector acceptance
  - momentum resolution (by including new MDT(3) chamber data)



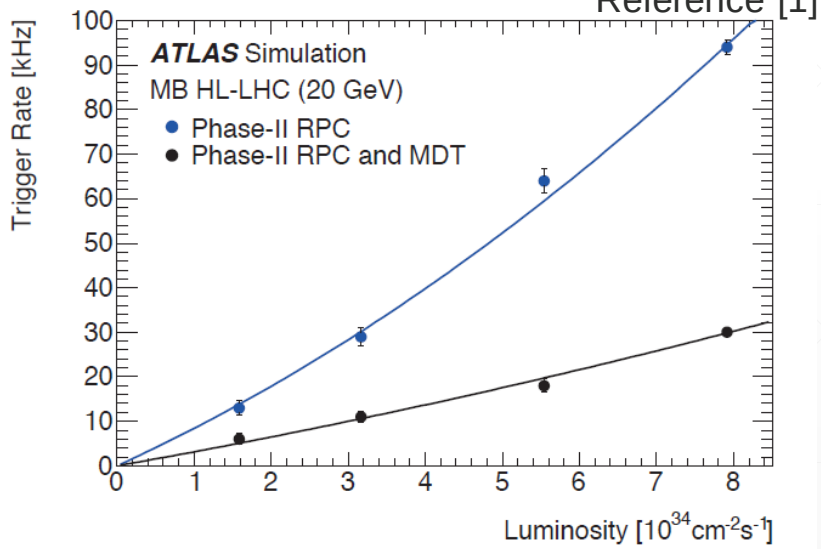
Different detector technologies provide different angular coverage.

Subsystem	Granularity	Coverage $\|\eta\|$
NSW processor (1)	Full NSW detector	$1.3 - 2.4$
MDT processor (2)	Full MDT detector	< 2.4
Barrel Sector Logic (1)	Full RPC and Tile, MDT	< 1.05
Endcap Sector Logic (1)	Full TGC, Tile, RPC, NSW, MDT	$1.05 - 2.4$

Selectivity of the current Level-1 muon trigger is limited by the moderate spatial resolution of RPC (3) and TGC (3).

- MDT chambers will be included in L0 Muon and will provide:
  - better spatial resolution
  - pT resolution close to that of the offline reconstruction

Reference [1]



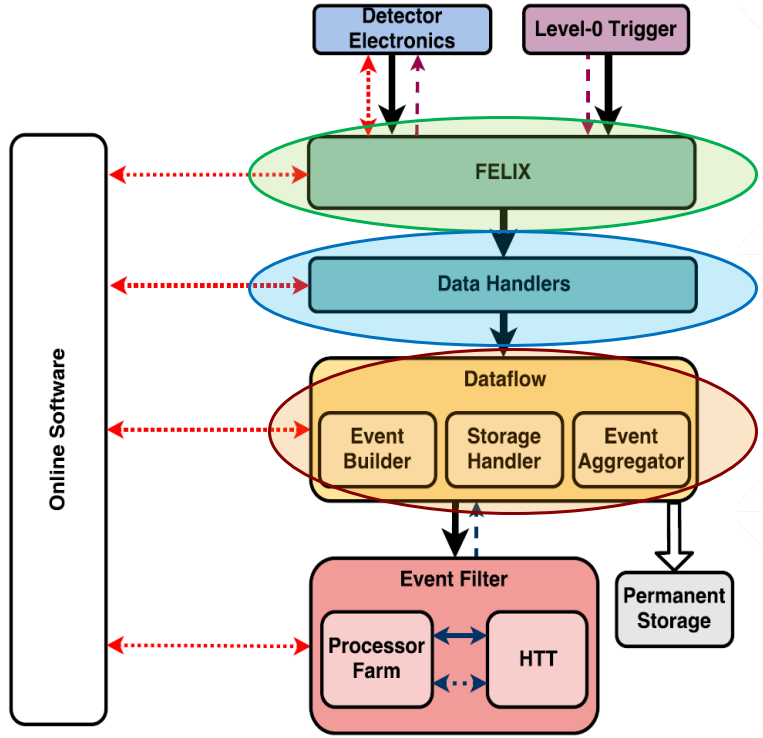
Rate of Level-0 single-muon trigger based on RPC only and RPC plus MDT for the barrel region | η | < 1.05.

(1) Phase-I hardware will be upgraded for Phase-II
(2) New Phase-II system
(3) RPC = Resistive Plate Chambers, TGC = Thin Gap Chambers, NSW = New Small Well, MDT = Monitored Drift Tube

DAQ details

Readout:

- receives data from the ATLAS detector front-end electronics
  - at the L0-trigger rate (1 MHz)
- performs basics processing
- sends them to the Dataflow system



FELIX (Front-End LInk eXchange):

- Custom cards PCIe based
- The new interface to detector-specific electronics including limited detector-specific firmware

Data Handlers:

Servers running a software application that :

- receives event fragments from FELIX
- performs detector-specific formatting and monitoring tasks:

Dataflow:

- Buffers data before, during and after the Event Filter decision
- Provides partial and full event access as needed and transfers data to permanent storage
- Managed by commodity software

Storage Handler

- buffering event data before and during Event Filter processing

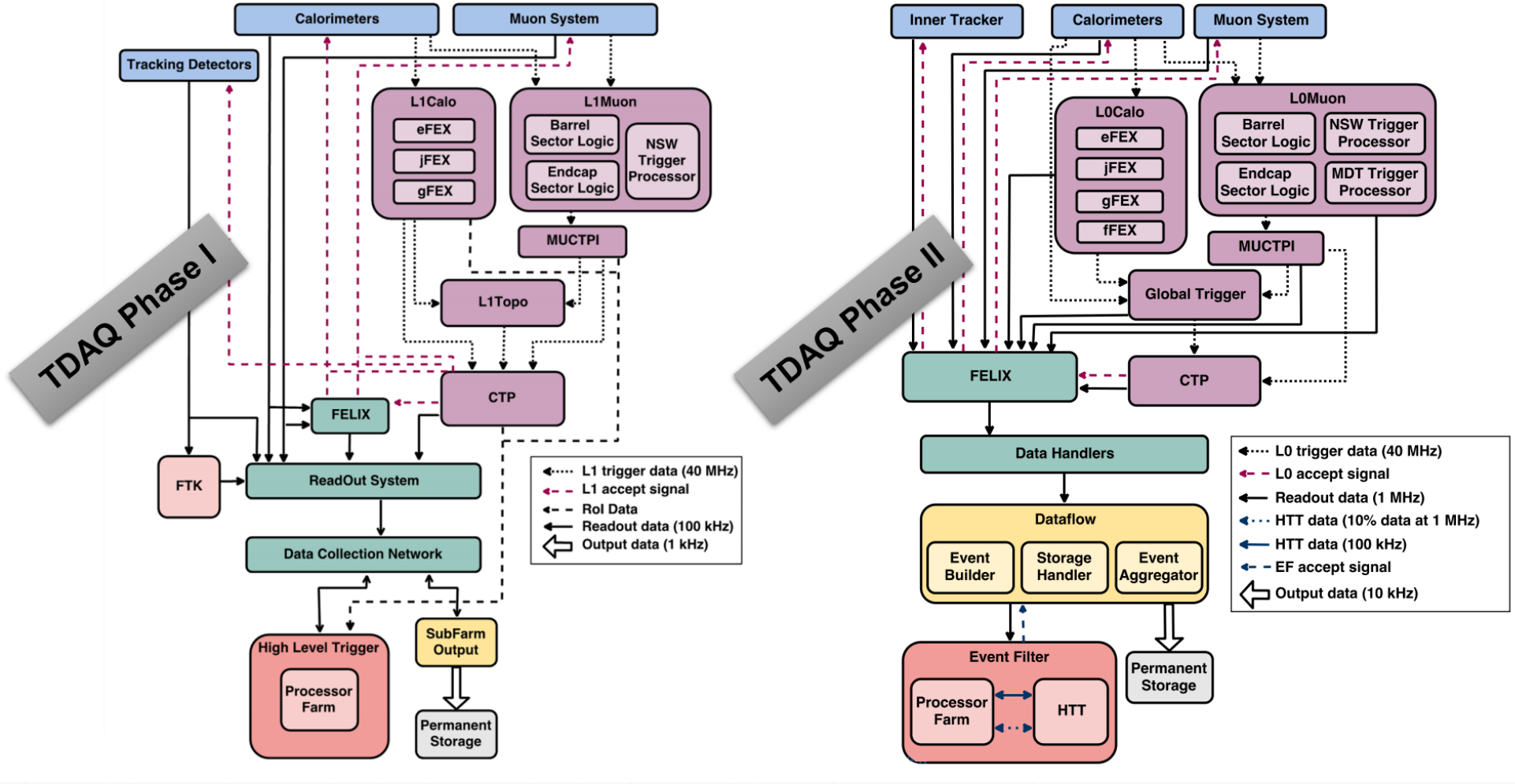
Event Builder

- interface of the dataflow to Data Handlers and Event Filter

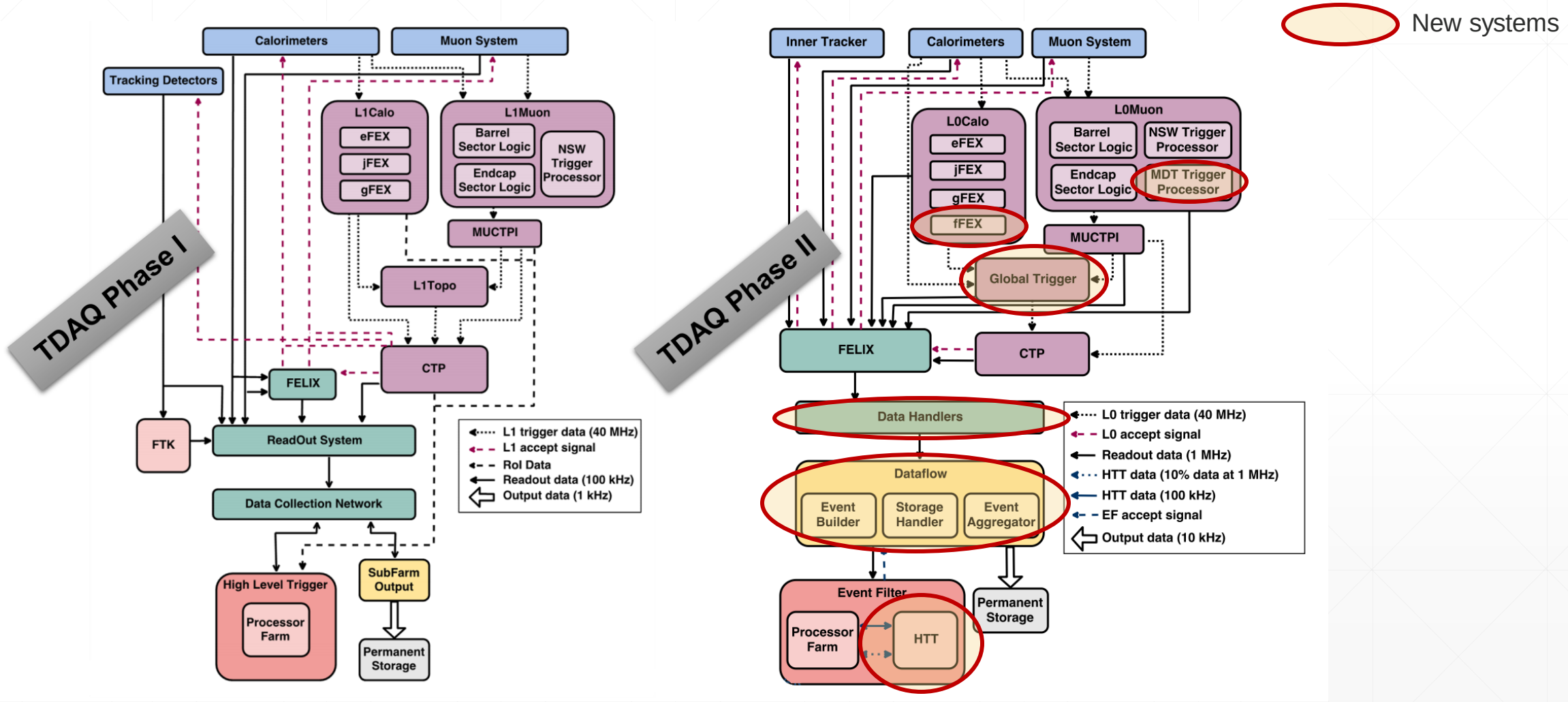
Event Aggregator

- receives the selected events from the Event Filter and groups/compress them before sending them to Tier-0

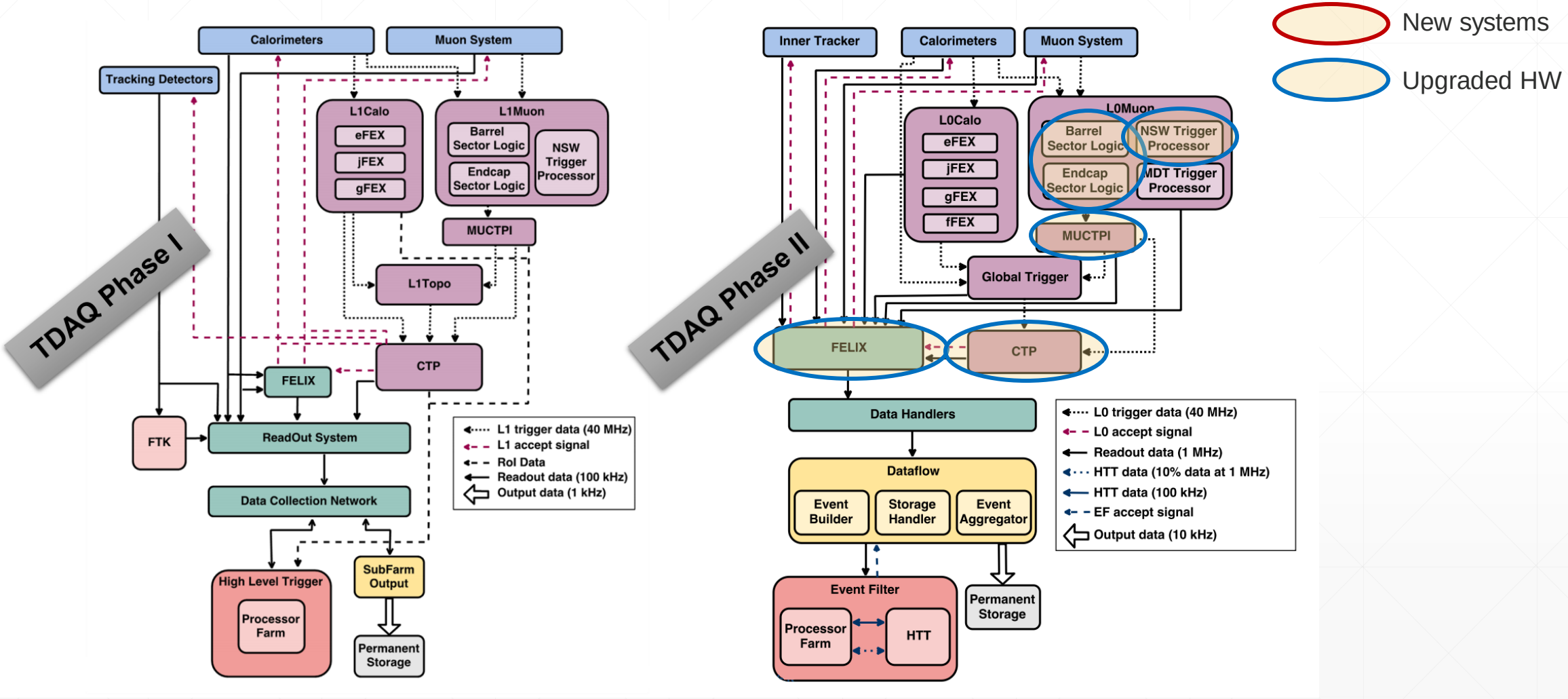
Overview of architecture upgrade



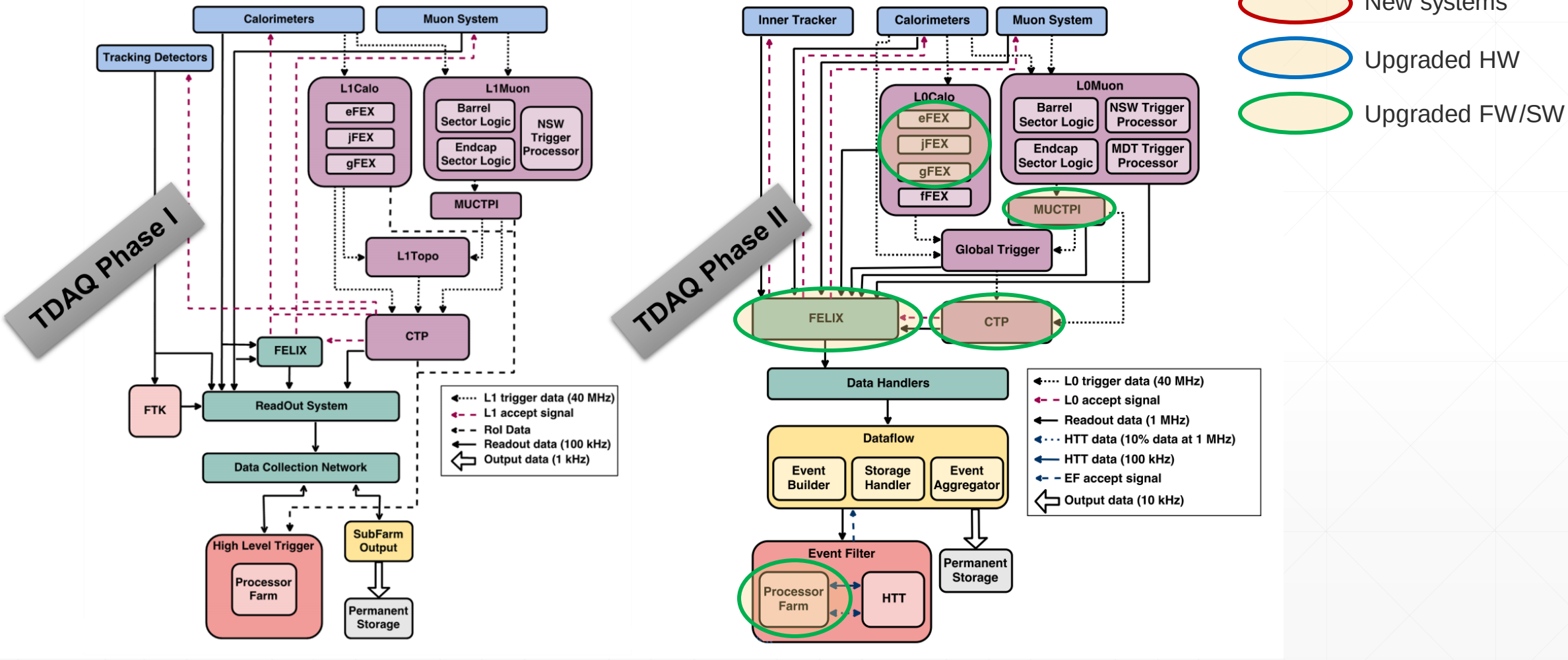
Overview of architecture upgrade



Overview of architecture upgrade



Overview of architecture upgrade



Reference [1]

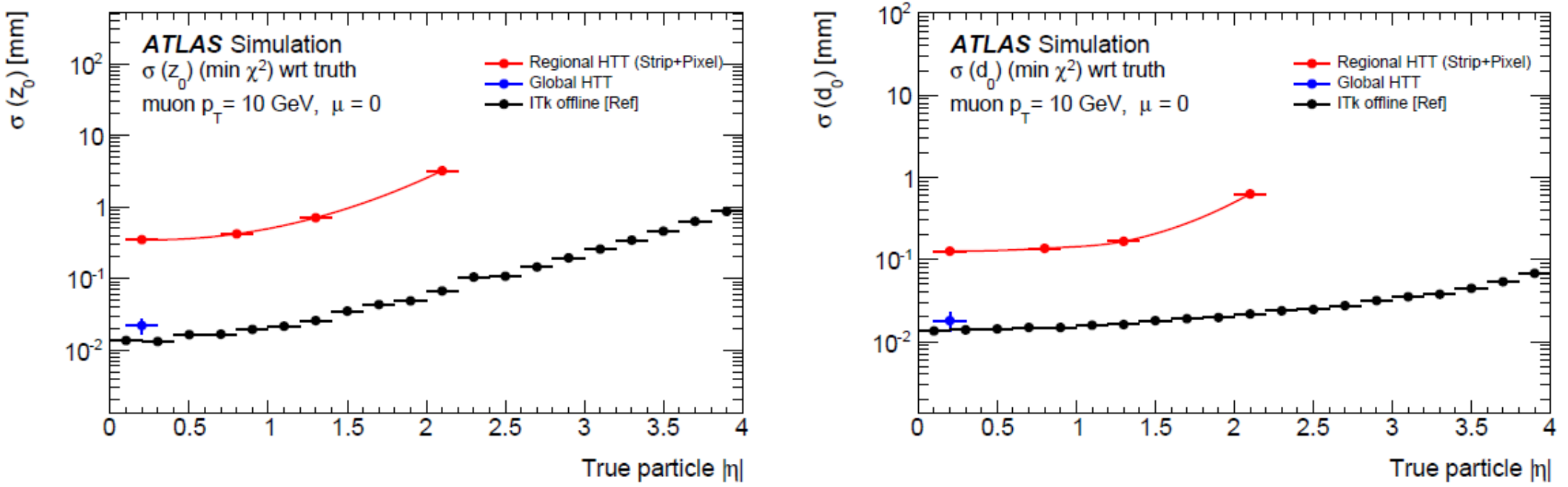


Figure 13.9: *Comparison of the z_0 (left) and d_0 (right) resolution for first- and second-stage fitting and offline.*

DO behaviour

With the assumption of having grouped the clusters per SSID

SSID = 0005
Clusters associated to this SSID
87643AB0
87643AA0
87643AB3

CLM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
CLP	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
CCM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6

DO behaviour

With the assumption of having grouped the clusters per SSID

SSID = 0005
Clusters associated to this SSID
87643AB0
87643AA0
87643AB3

CLM	**Addr 0**	**Addr 1**	**Addr 2**	**Addr 3**	**Addr 4**	**Addr 5**	**Addr 6**
	87643AB0	87643AA0	87643AB3				
CLP	**Addr 0**	**Addr 1**	**Addr 2**	**Addr 3**	**Addr 4**	**Addr 5**	**Addr 6**
CCM	**Addr 0**	**Addr 1**	**Addr 2**	**Addr 3**	**Addr 4**	**Addr 5**	**Addr 6**

DO behaviour

With the assumption of having grouped the clusters per SSID

SSID = 0005
Clusters associated to this SSID
87643AB0
87643AA0
87643AB3

CLM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
	87643AB0	87643AA0	87643AB3				

CLP	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
						2	

CCM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
						3	

SSID used as address

# of clusters to be retrieved

DO behaviour

With the assumption of having grouped the clusters per SSID

SSID = 0005
Clusters associated to this SSID
87643AB0
87643AA0
87643AB3

CLM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
	87643AB0	87643AA0	87643AB3	The next set of clusters		will continue from here	

CLP	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
						2	

CCM	Addr 0	Addr 1	Addr 2	Addr 3	Addr 4	Addr 5	Addr 6
						3	