LHC-ATLAS Phase-1 upgrade Firmware validation for real time digital processing for new trigger readout system of the Liquid Argon calorimeter

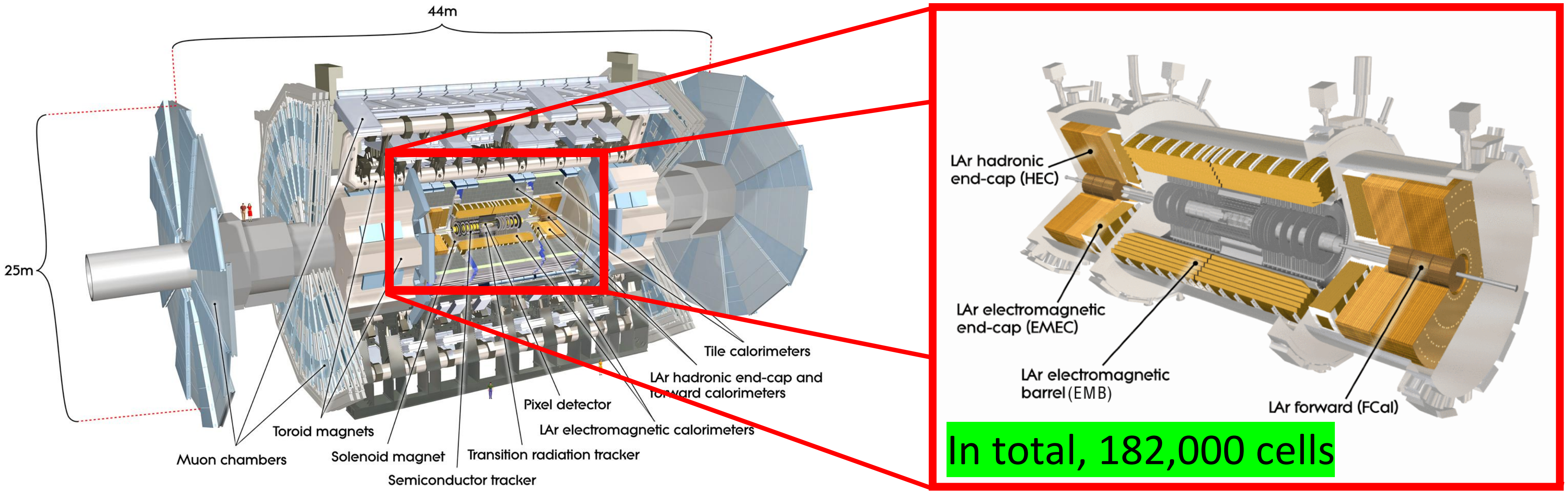
CHEF 25-29 November 2019

Reiyo Oishi

On behalf of ATLAS Liquid Argon Calorimeter collaboration

The University of Tokyo

ATLAS detector and LAr calorimeter



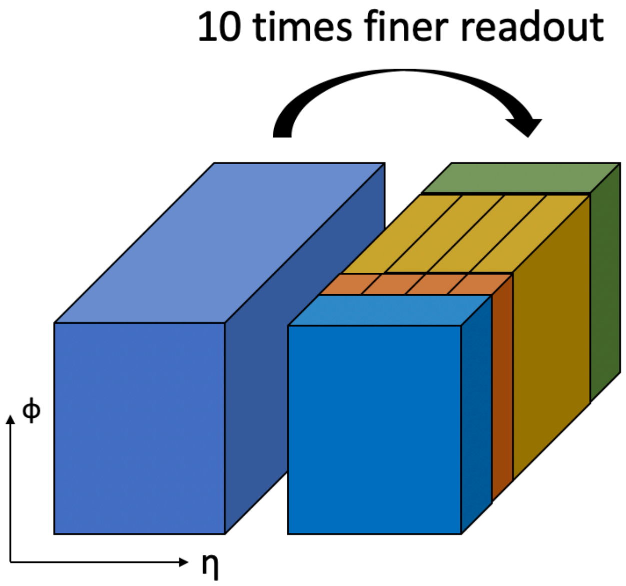
ATLAS detector

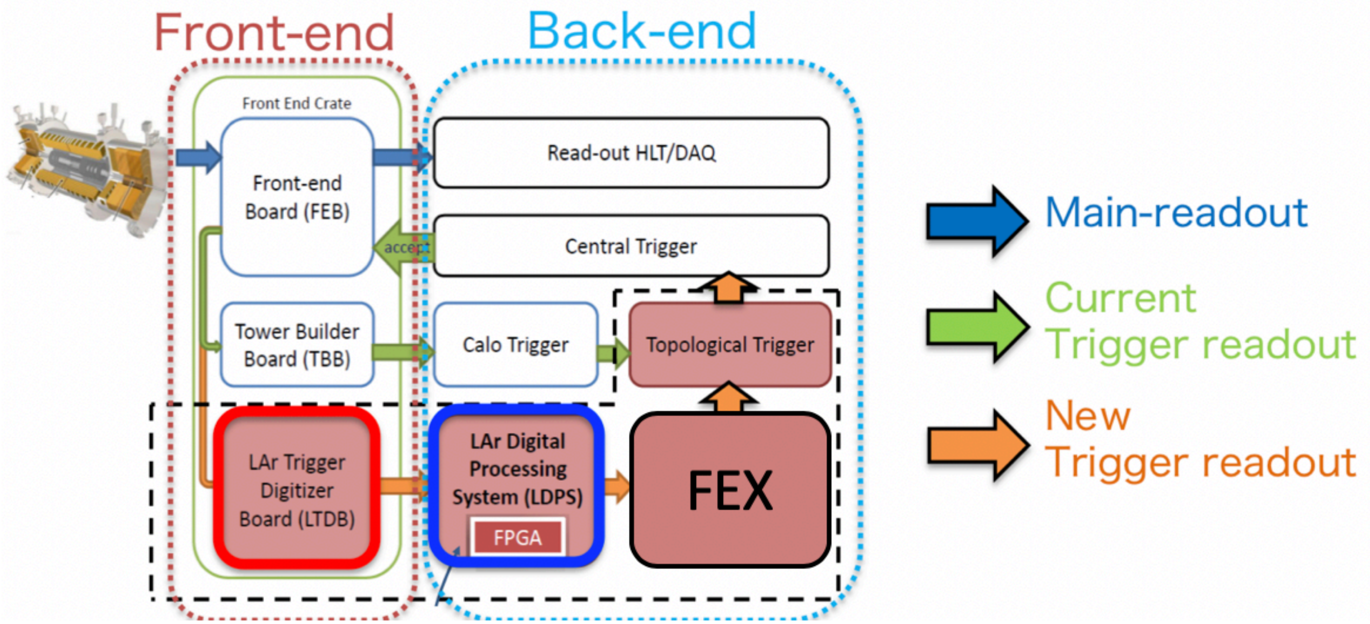
- Multi-purpose detector
- Inner tracker, EM & Hadronic Calorimeter, Muon spectrometer

Liquid Argon (LAr) Calorimeter

- Measure energy and position of electron/photon, hadron
- Sampling calorimeter, 22 radiation lengths
- Cover a region of $|\eta|<4.9$ for EM, $1.5<|\eta|<4.9$ for Hadron
- 3 layers (One more layer in $|\eta|<1.8$ region for EM)

Phase-1 LAr trigger upgrade



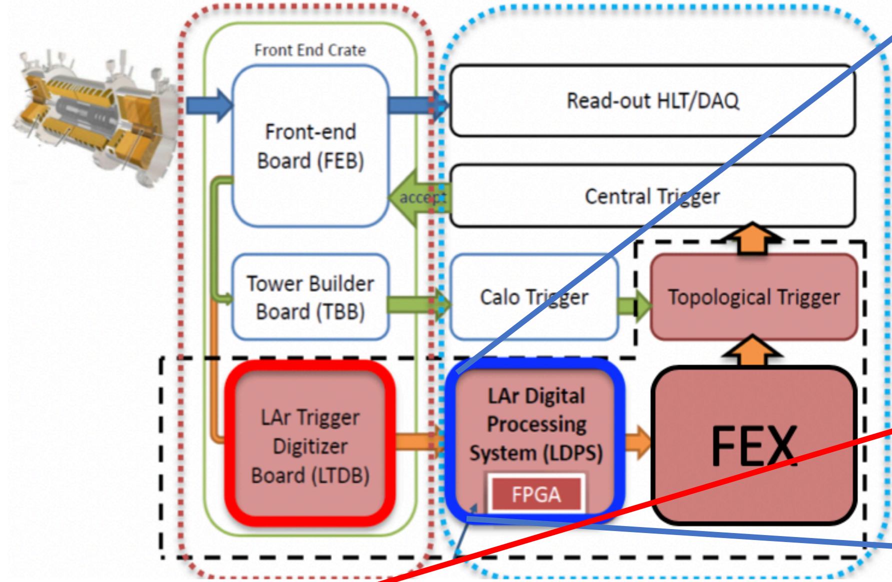


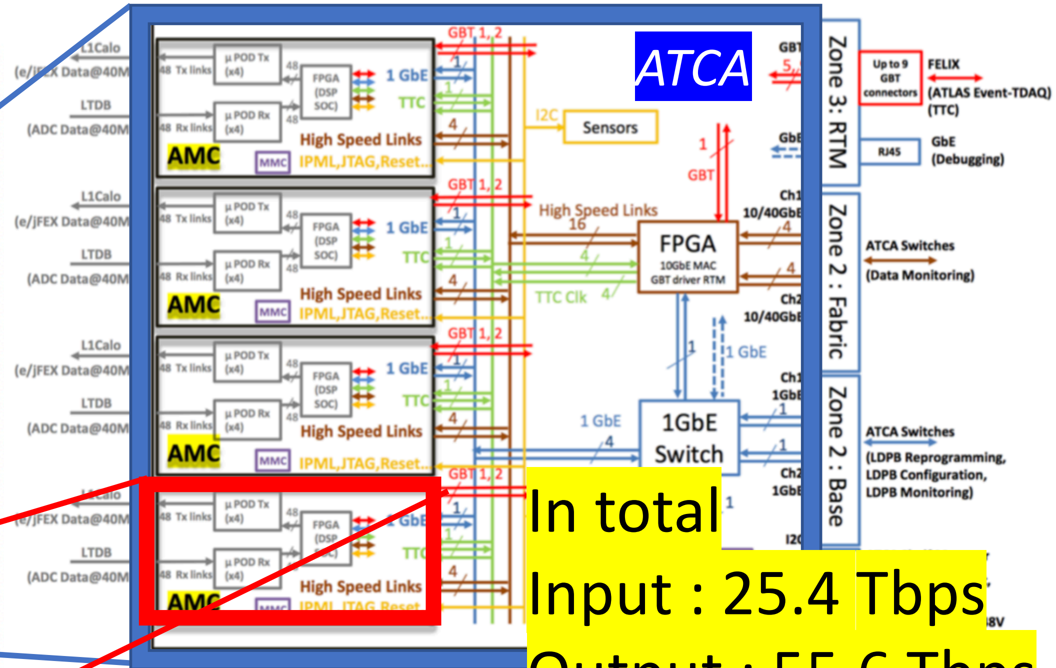
In order to cope high luminosity environment, trigger performance need to be enhanced and some new technologies are introduced.

- 10 times finer readout, so-called Super Cell ($\eta \times \phi = 0.025 \times 0.1$)
- Some new trigger systems will be introduced
  - LTDB : Digitize supercell information every 25 ns
  - LDPS : Convert ADC to energy with a fixed latency
- FEX receives 3 kinds of energy (e, j, g-FEX). e, j, g-FEX identify e/γ, jets, large radius jets, respectively.

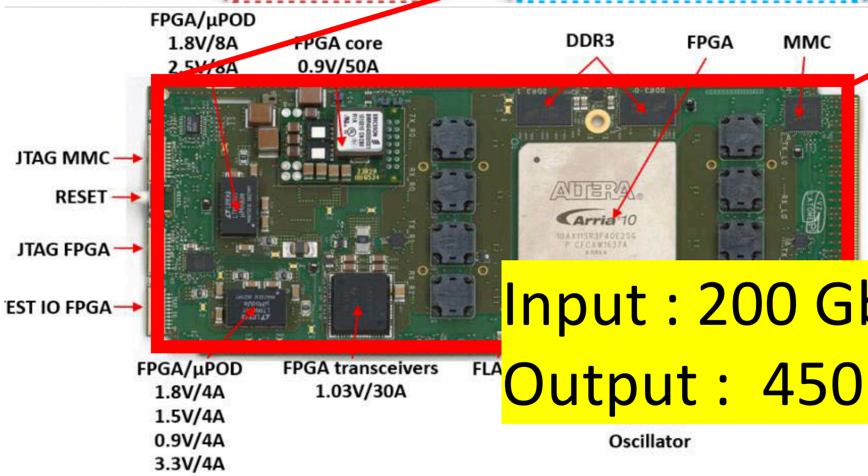
→ Next page, detail of LDPS

LATOME project





In total
Input : 25.4 Tbps
Output : 55.6 Tbps

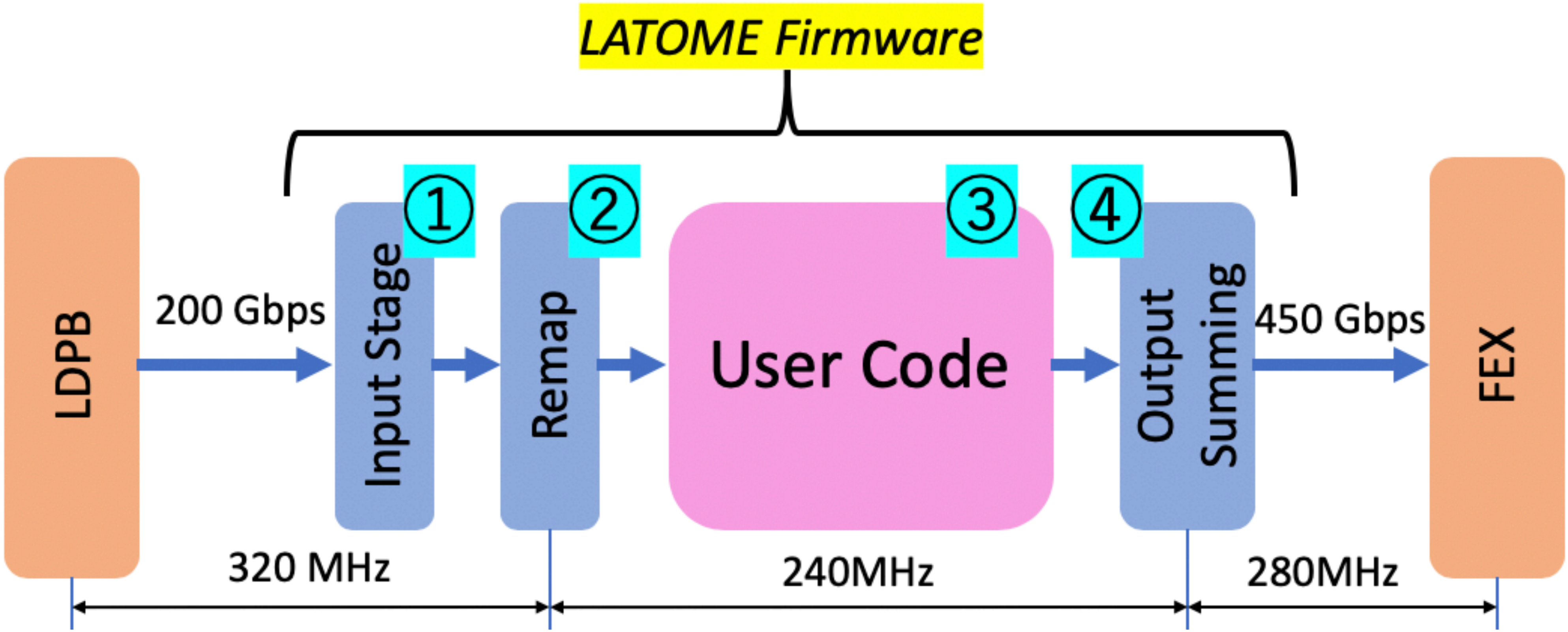


Input : 200 Gbps
Output : 450 Gbps

- Each LDPS has 4 LATOME Boards
- 31 (124) LDPS (LATOME) cover LAr calorimeter region
- Each LATOME Board can process up to 320 supercells
- XAUI for monitoring

Every Super Cells are processed in LATOME Board.
→ Next page, detail of LATOME firmware

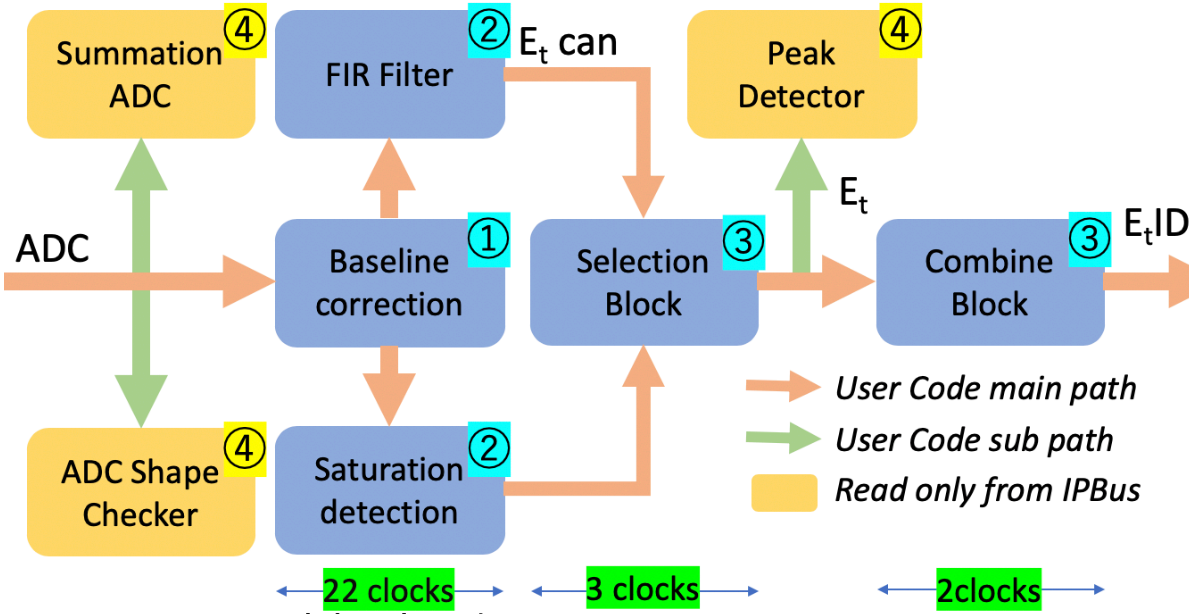
LATOME firmware

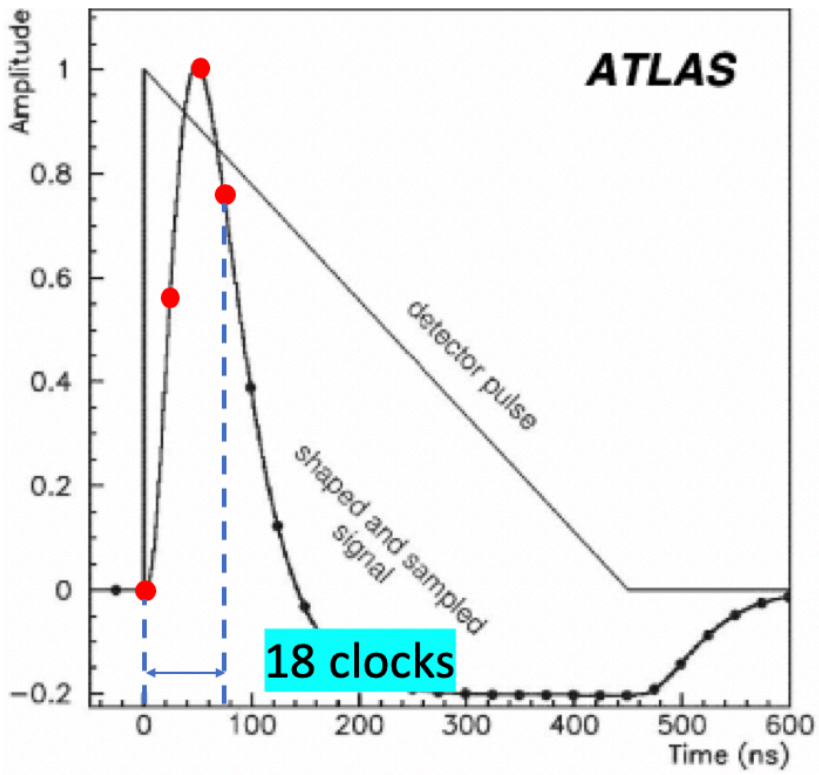


① Receive data from Front End, extract ADC and bunch crossing ID
② Remap channels and decrease clock frequency to compute energy
③ Compute transverse energy from ADC with FIR Filtering
④ Calculate the sums over specific η-ϕ region (η × ϕ = 0.1 × 0.1 & 0.2 × 0.2)
→Transmit 3 kinds of energy to FEX

LATOME firmware base design has been established, currently in validation stage
→ Next page, detail of User Code

User Code





Every blocks drive 240 MHz

Fixed latency 27 clocks

<User Code main path>

① Correct baseline shift and update them every 11 seconds

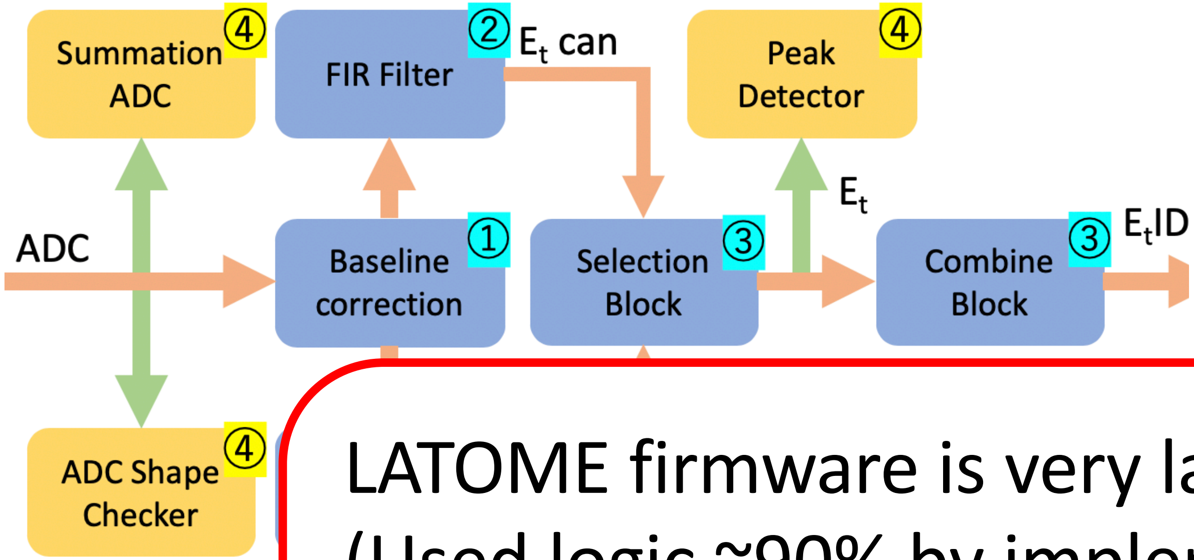
② Energy calculation with 4 ADC samples in 240 MHz (18 clocks are costed to sample 4 ADCs, 4 more clocks are needed to process with DSP blocks)

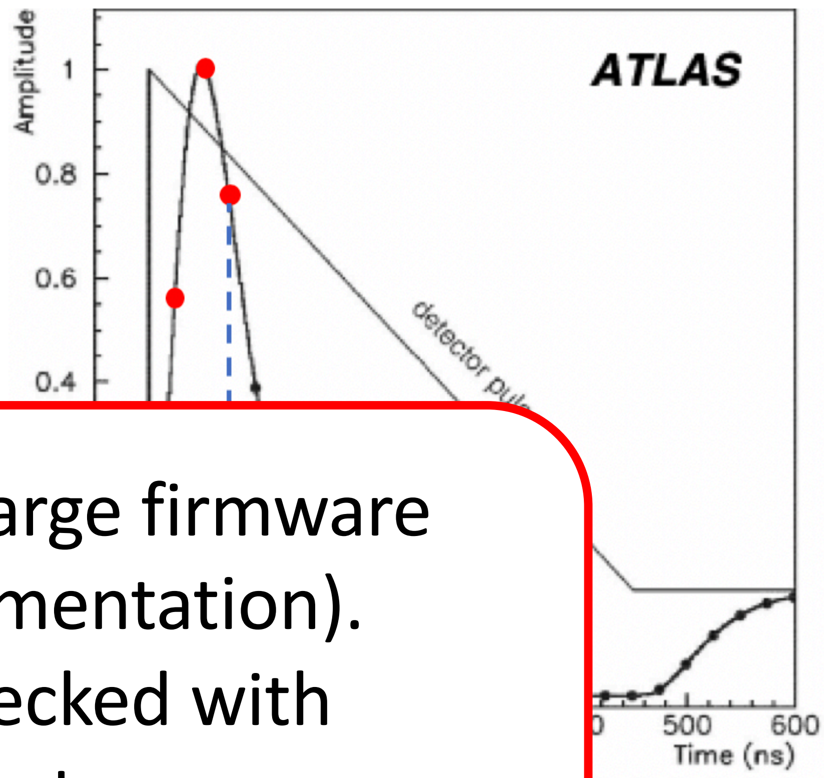
③ Compute energy deposited timing with filtering algorithm (3 clocks)

<User Code sub path>

④ Detect LAr pulses and check sums of 1 LHC cycle ADCs for monitoring

User Code





LATOME firmware is very large firmware (Used logic ~90% by implementation).
Each function has been checked with simulation and on-board test.
→ From now, detail of on-board validation

Every
Fixed

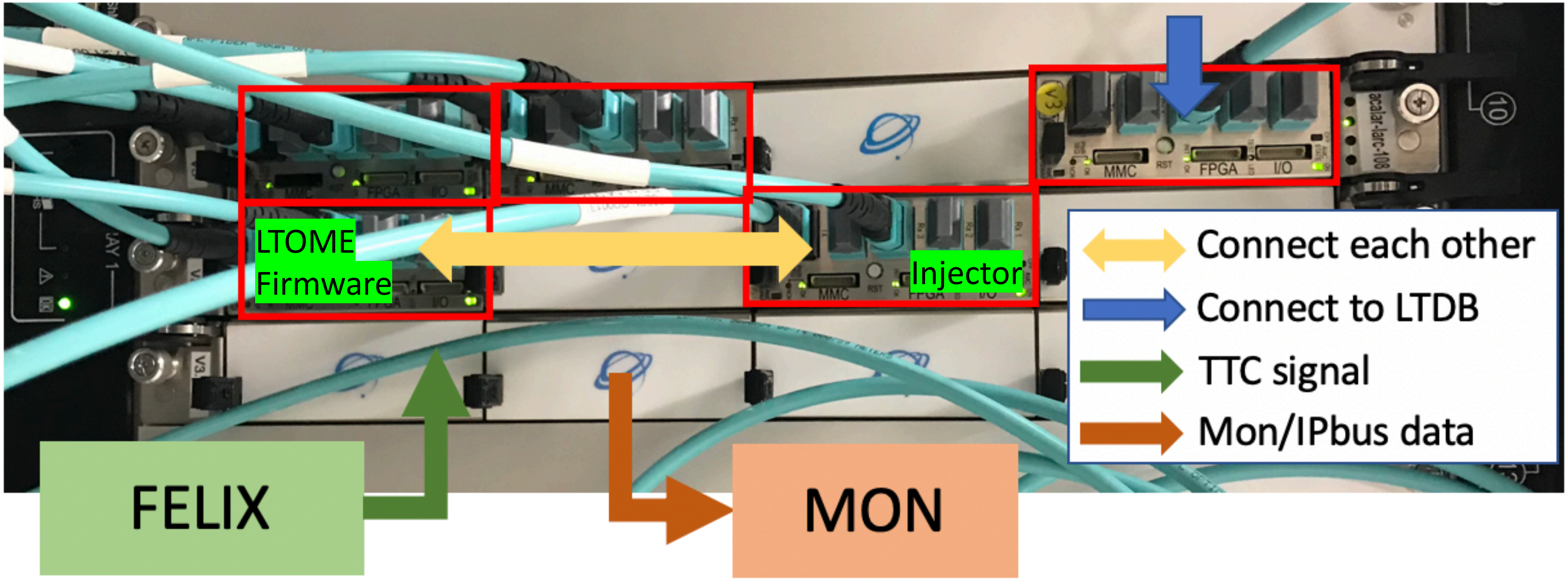
<User Co

① Corre
② Energ ... d to sample 4 ADCs, 4 more clocks are needed to process with DSP blocks)
③ Compute energy deposited timing with filtering algorithm (3 clocks)

<User Code sub path>

④ Detect LAr pulses and check sums of 1 LHC cycle ADCs for monitoring

User Code on board

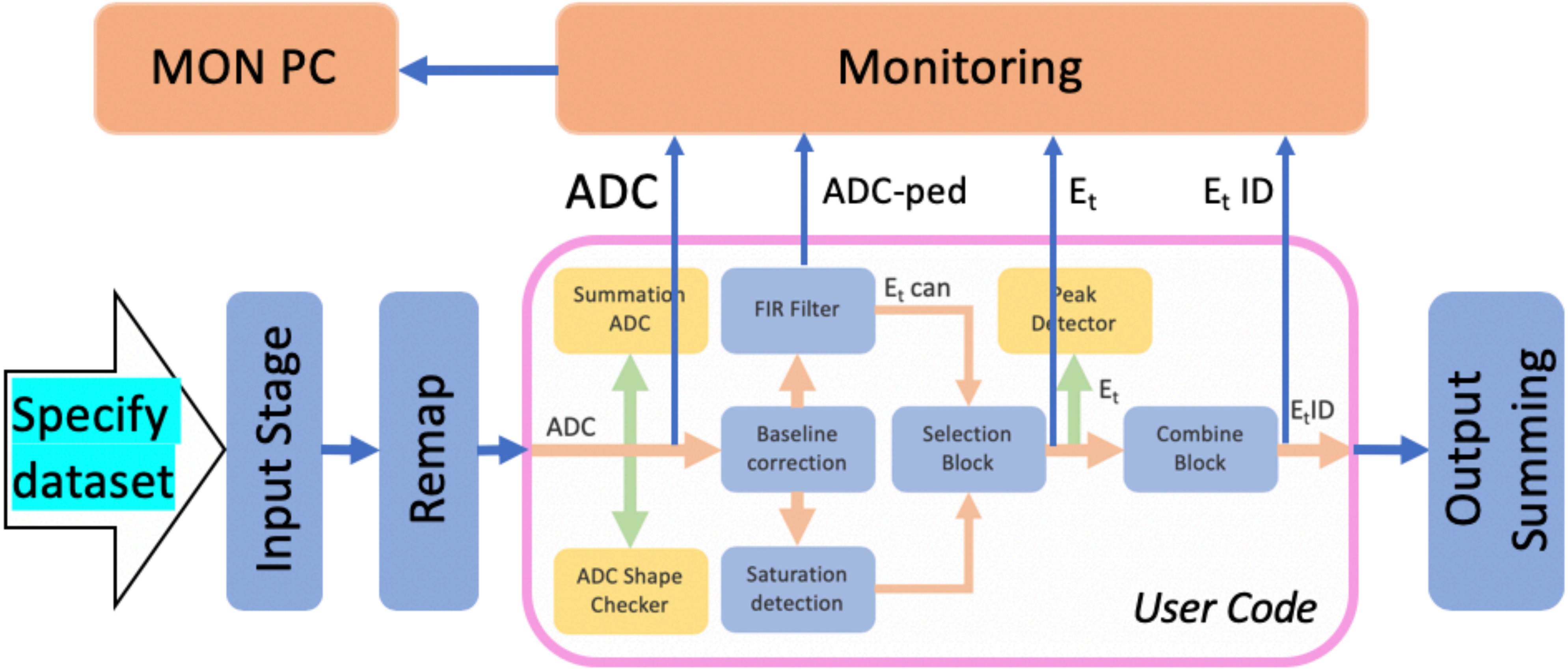


User Code is already confirmed with simulation, then on board.

- Test bench @CERN, 2 LATOME boards validation. One LATOME is injector.
- LATOMEs are connected to pc and controlled (Reset, configuration, calibration …)
- Monitoring data is recorded in monitoring-pc placed at test bench

→ Same condition with RUN3 experiment.

User Code on board

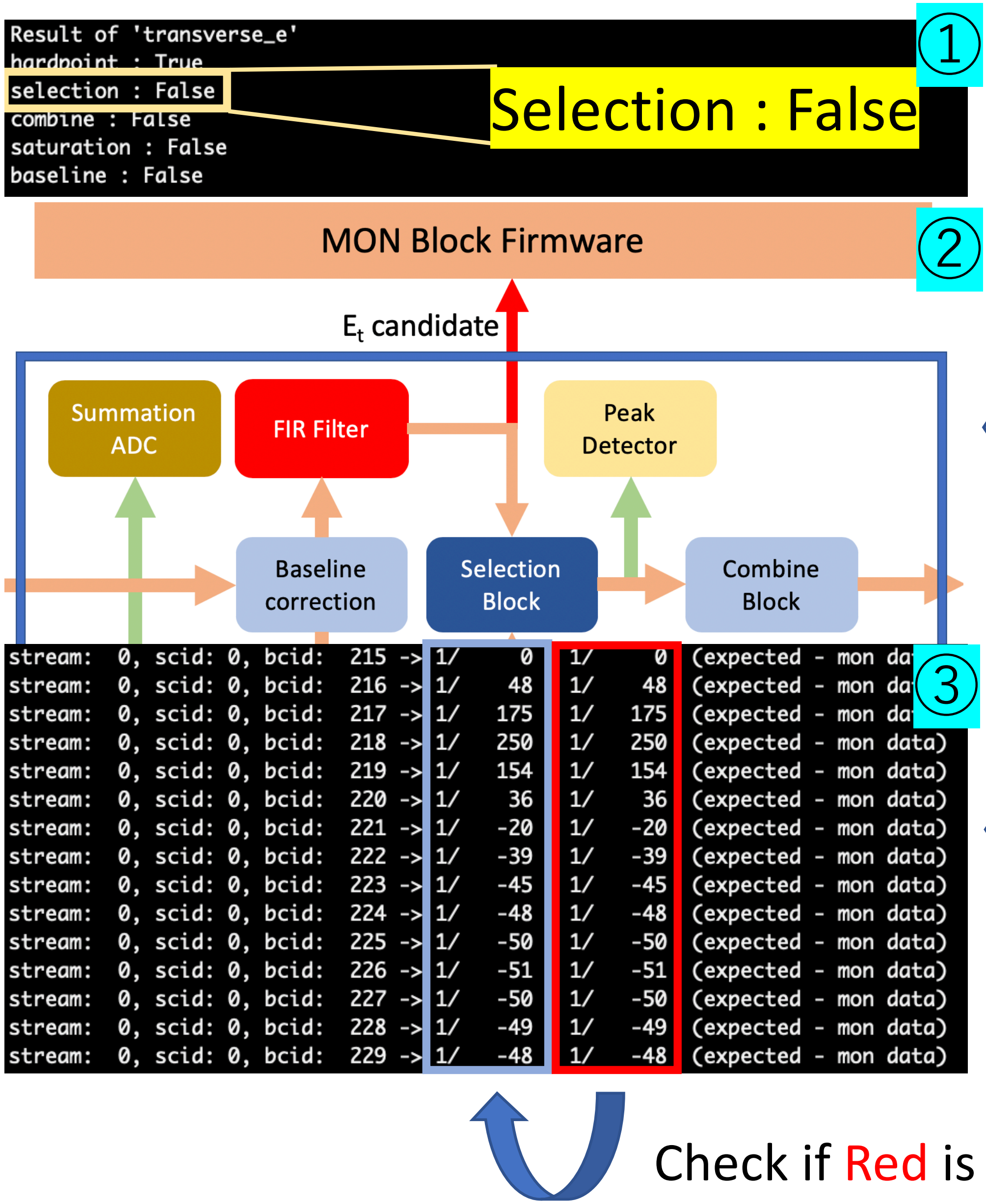


By changing User Code configuration, target block can be validated.

1. Inject data, User Code works depend on its configuration and records outputs
2. User Code model also works with same configuration and prepares predicted value
3. Compare firmware output and model output.
4. Checker reports validation scheme result.

→Next 2 pages show validation result of FIR filter and Selection Block

User Code on board



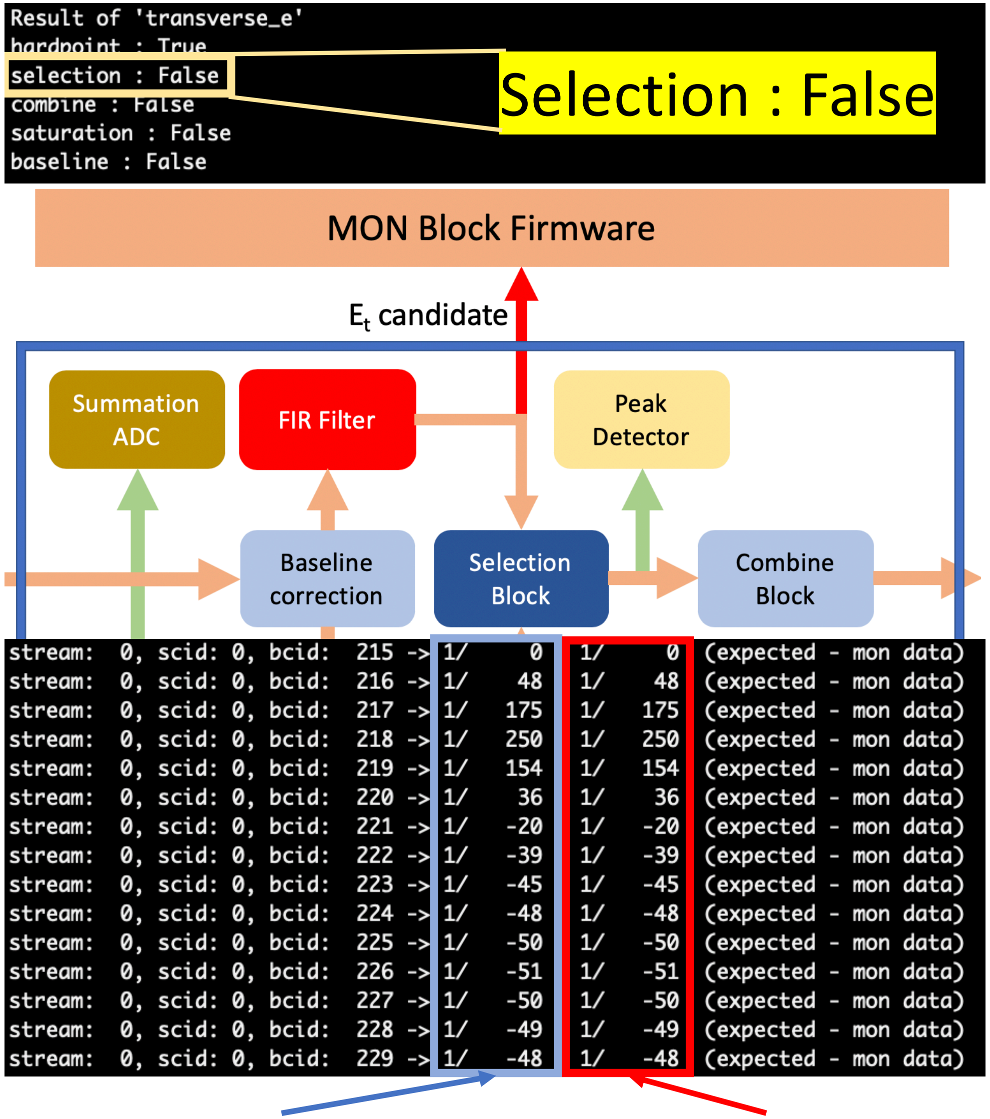
User Code configuration

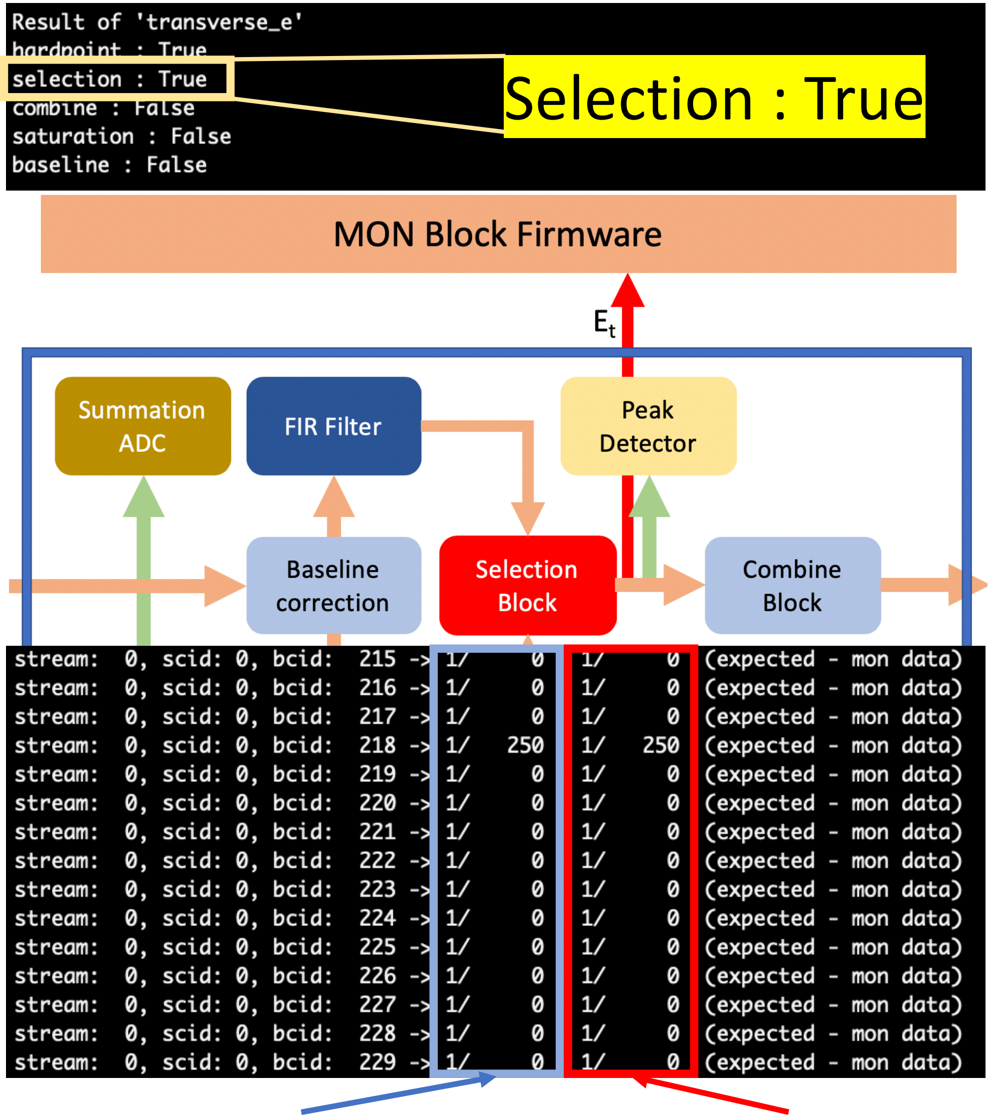
Inside the User Code firmware
FIR filter is target function.

Blue : firmware model predicts values
Red : Real firmware outputs

Check if Red is same with Blue one by one

User Code on board





User Code model output Firmware output User Code model output Firmware output

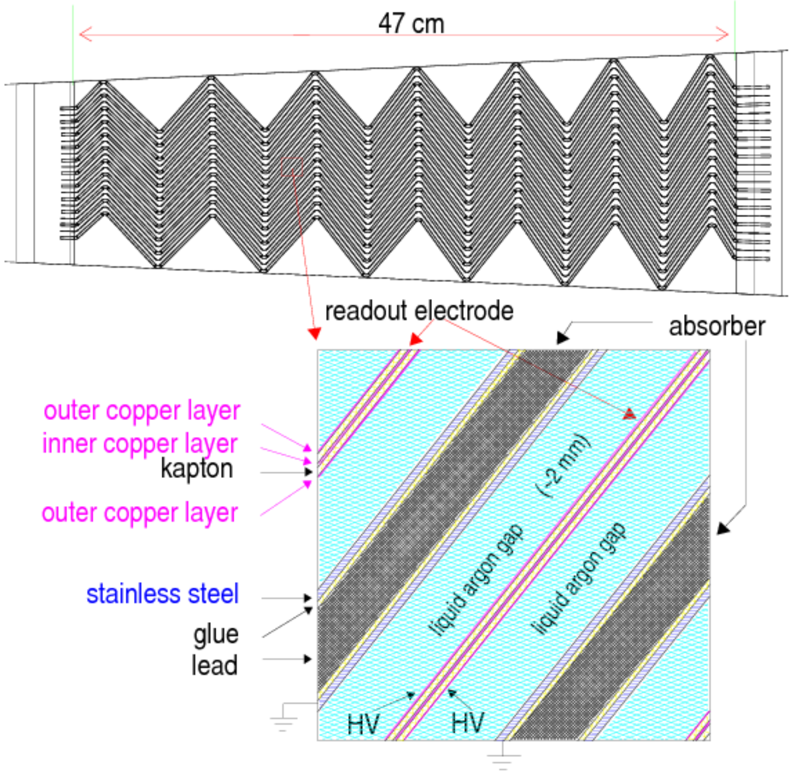
The most significant User Code features, energy calculation and filtering algorithm, are validated with any dataset by selecting proper configuration.

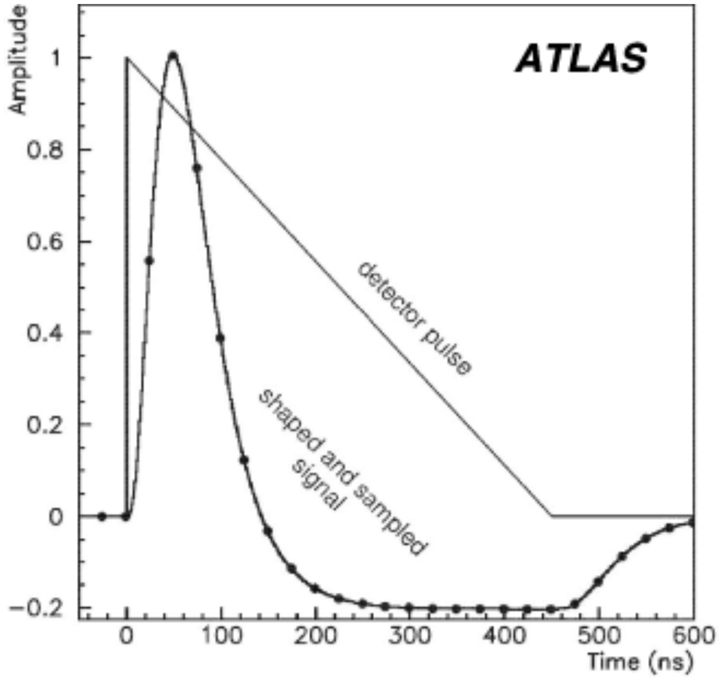
Conclusion and future tasks

- LHC ATLAS RUN3 experiment will start from 2021. To suppress trigger rate, new trigger readout systems and 10 times finely readout (supercell) will be introduced.
- New trigger processing firmware (LATOME firmware) is almost fixed. Each module validation is ongoing.
- Energy computing module User Code verification was success. Each User Code block was verified with simulation.
- On board User Code validation is ongoing at CERN. User Code main functions are validated.
- Need to validate other User Code functions as well.

Back Up

LAr pulse

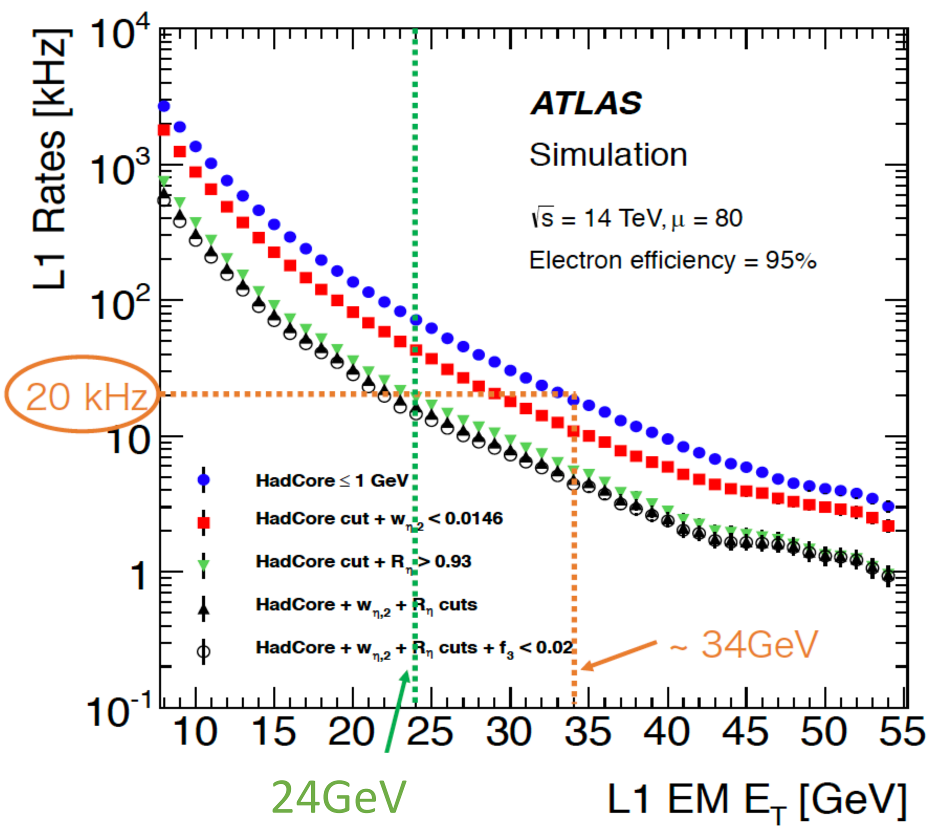




- LAr pulse from detector is triangle pulse. This would be converted to bipolar pulse with CR-(RC)^2 electric circuit.
- About 600 ns pulse length, bipolar shape have good resistance to pileup

Phase-1 upgrade

E_T threshold VS L1 trigger rate for single EM obeject



LAr need to suppress L1 trigger rate up to 20KHz. If same readout with RUN2 will be used, energy threshold will be 34 GeV and many significant signal will be wasted.
To defeat this, new 3 variable will be introduced. These 3 variables are computed by 10 times finer readout system.
By using these new 3 variables and current system, energy threshold will be suppressed to about 24 GeV.

User Code

FIR Filter : Compute transverse energy

Saturation Detection : Compute $E_t$tau

Selection Block : Apply filtering algorithm to E, $E_t$tau

Combine Block : Saturation pulse check

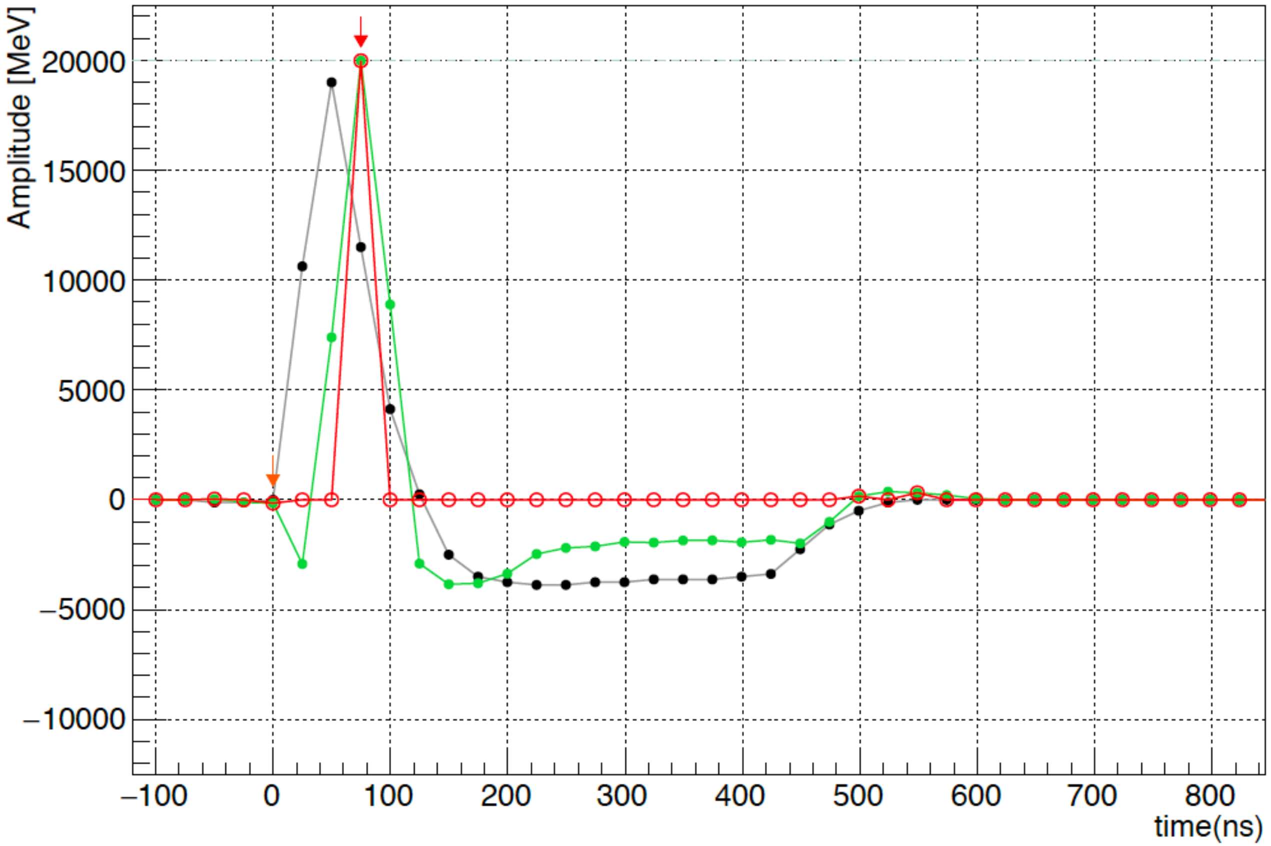
Baseline Correction : Correct baseline shift

ADC Shape Checker : Detect peak by checking ADC

Summation ADC : Sum ADCs of 1 LHC cycle

Peak Detector : Detect peak by checking energy

User Code filtering algorithm

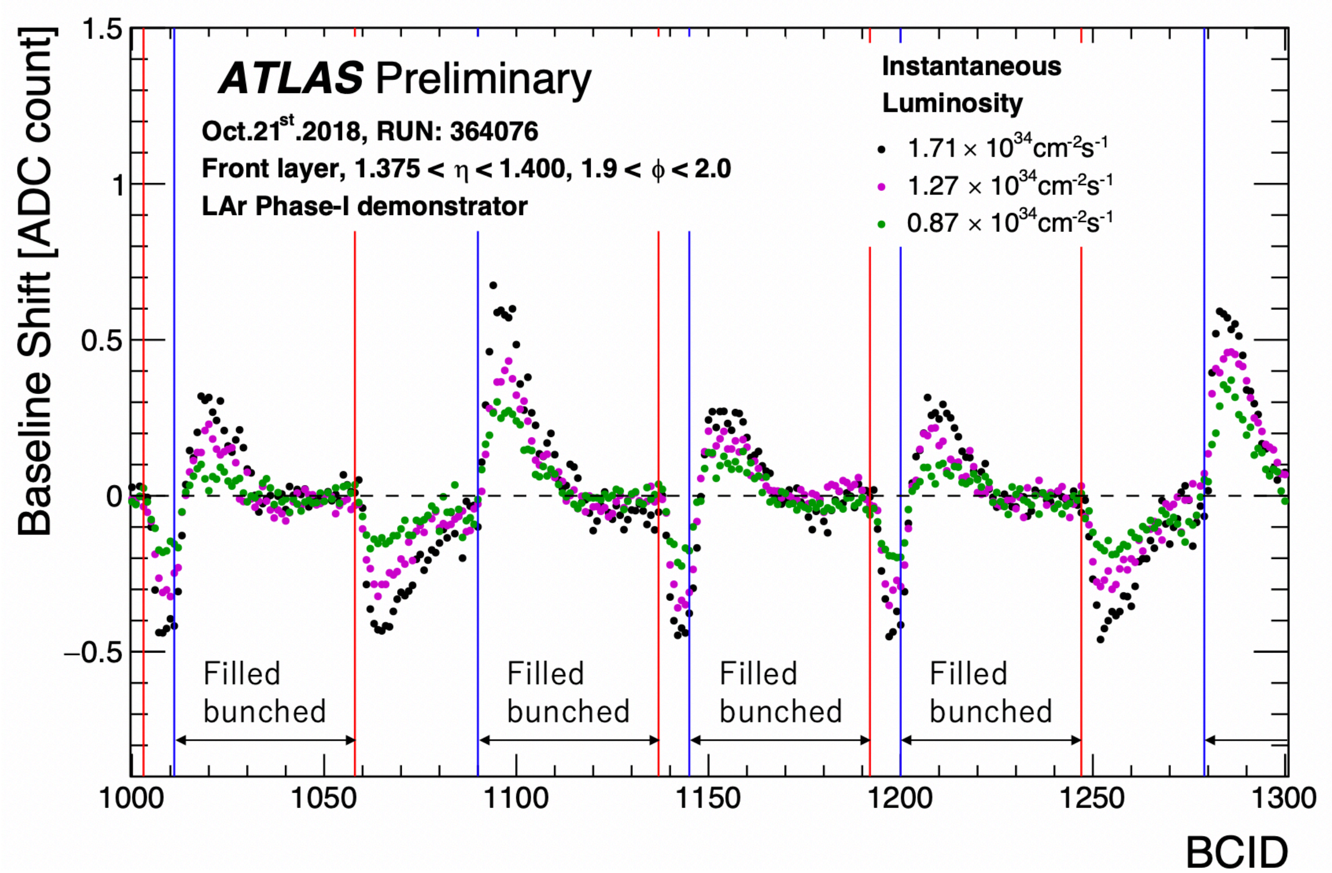


- Sampling ADC
- Energy candidate
- True deposited energy

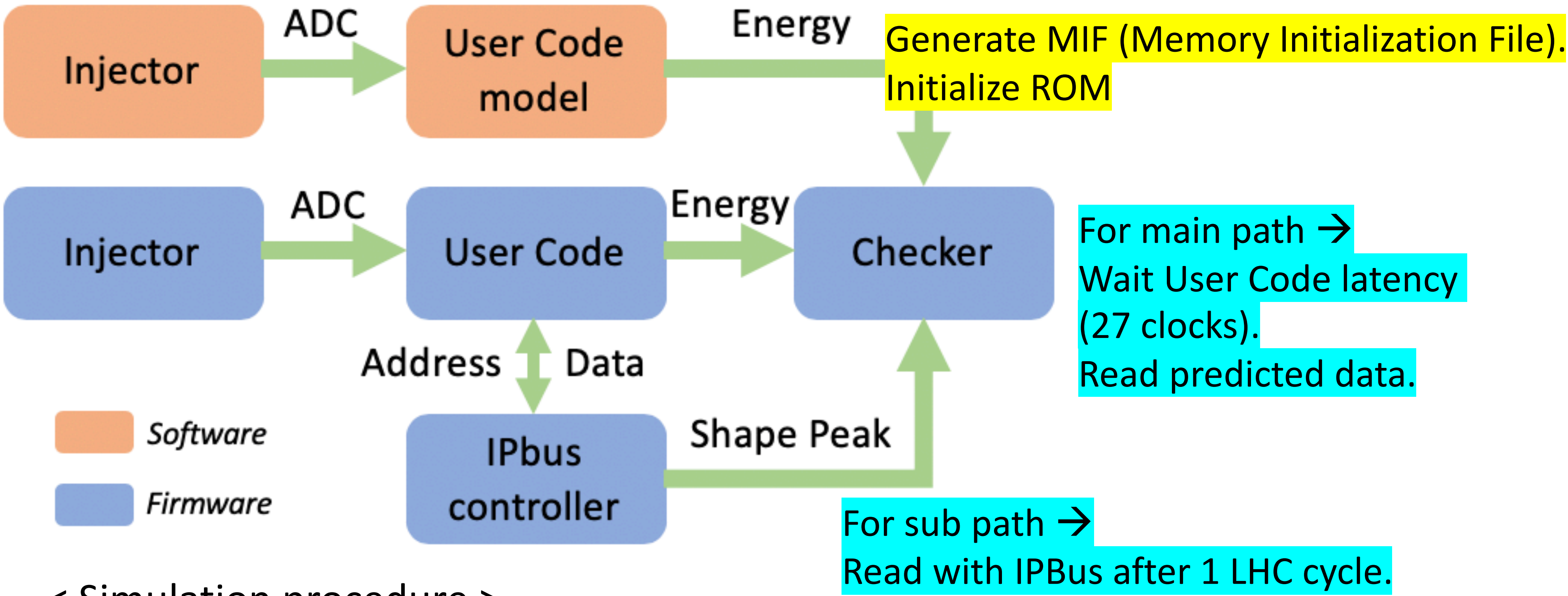
- 1 LAr pulse corresponds to 1 deposited energy
- User Code compute energy candidate every 25 ns for each supercell
- To find out true deposited energy, tau criteria is applied

What is baseline shift ?

- Proton filling scheme affect energy calculation, they depend on instantaneous luminosity. They are updated every 11 second in User Code.



User Code Simulation



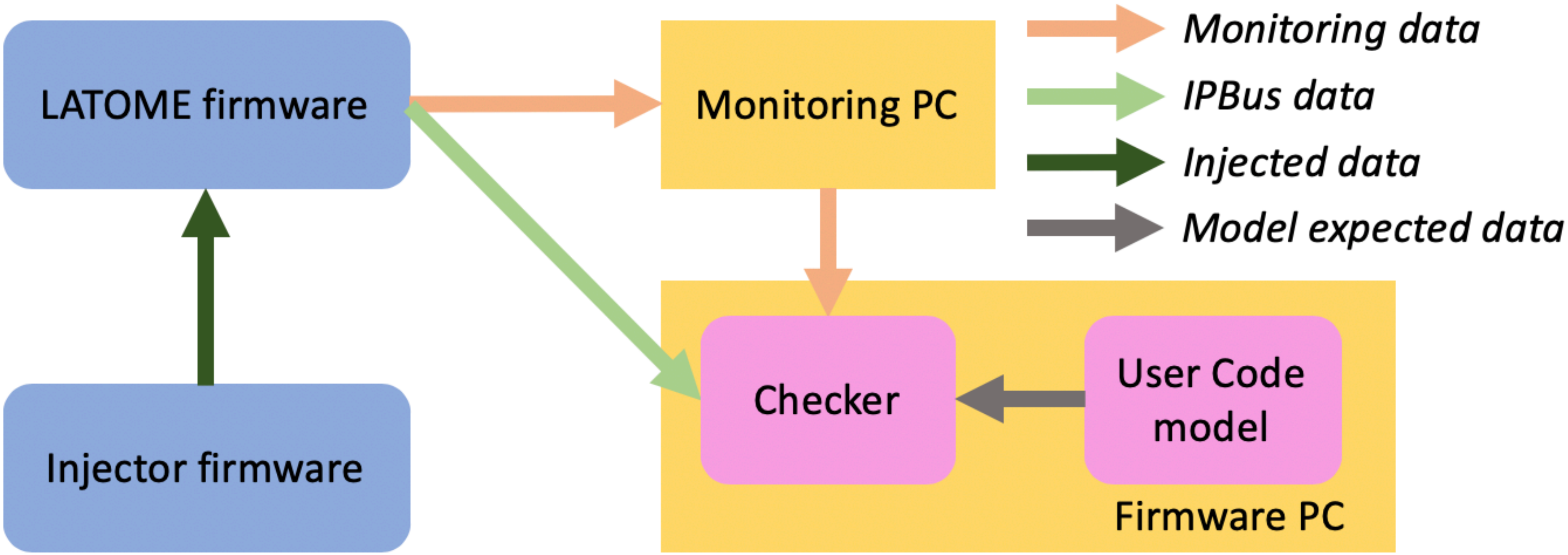
< Simulation procedure >

1. Specify one dataset and inject it to User Code and User Code model
2. User Code model prepare predicted outputs as memory initialization file
3. Read User Code outputs or IPBus data and compare expected data

※ IPBus can read data of register/memory directory. Some register/memory has its unique address.

→ Such testbenches are already constructed for each User Code block. Every User Code blocks and latency of User Code are already verified with this procedure bit by bit.

User Code on board



To validate User Code with LATOME board, 2 kind of procedures are required.

<User Code sub path>

1. Read data with IPBus directory
2. Compare IPBus data and expected data

<User Code main path>

1. Record main path data as monitoring binary data
2. Decode binary data and extract data, data type and so on
3. Compare extracted data and UserCode model outputs

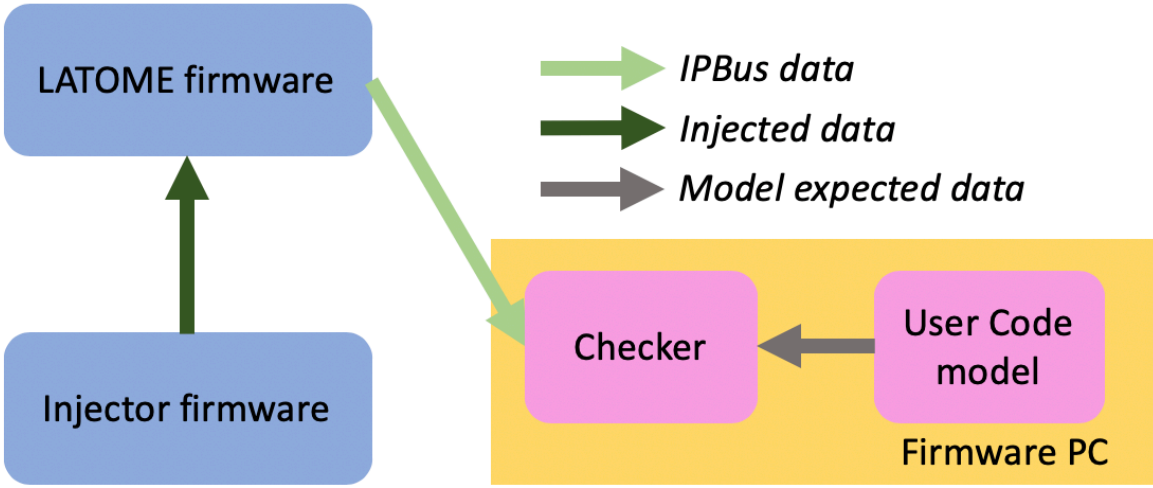
User Code sub path on board

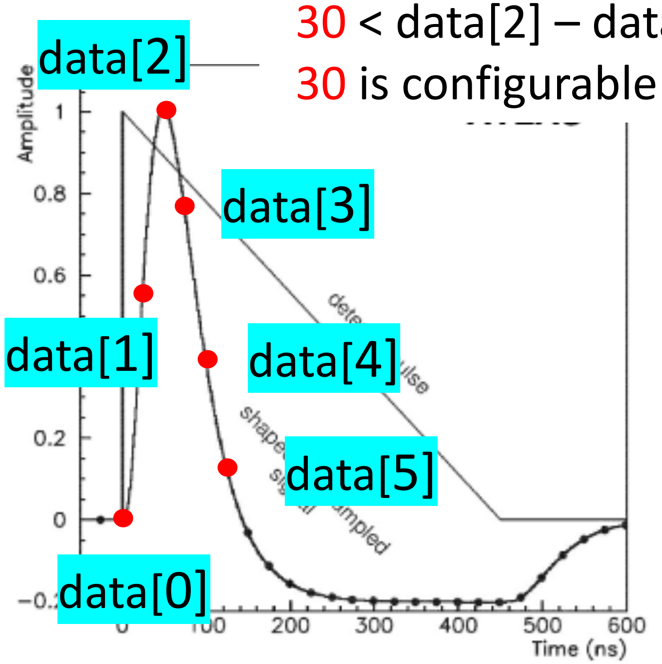
To validate ADC Shape Checker, single pulse data is injected. Each Super Cell has only 1 LAr like pulse.

1. ADC shape checker detects only one LAr shape, records 6 ADCs and BCID
   → Firmware should detect the only one pulse
2. User Code model also detect LAr shape with same conditions which used in firmware.
3. Compare firmware output and model output.
4. Finally checker outputs validation scheme result.

→ Every Super Cell detect one pulse properly.
ADC Shape Checker can be used for mapping check during commissioning

data[0] < data[1]
data[1] < data[2]
data[4] < data[3]
data[5] < data[4]
30 < data[2] – data[2]
30 is configurable value



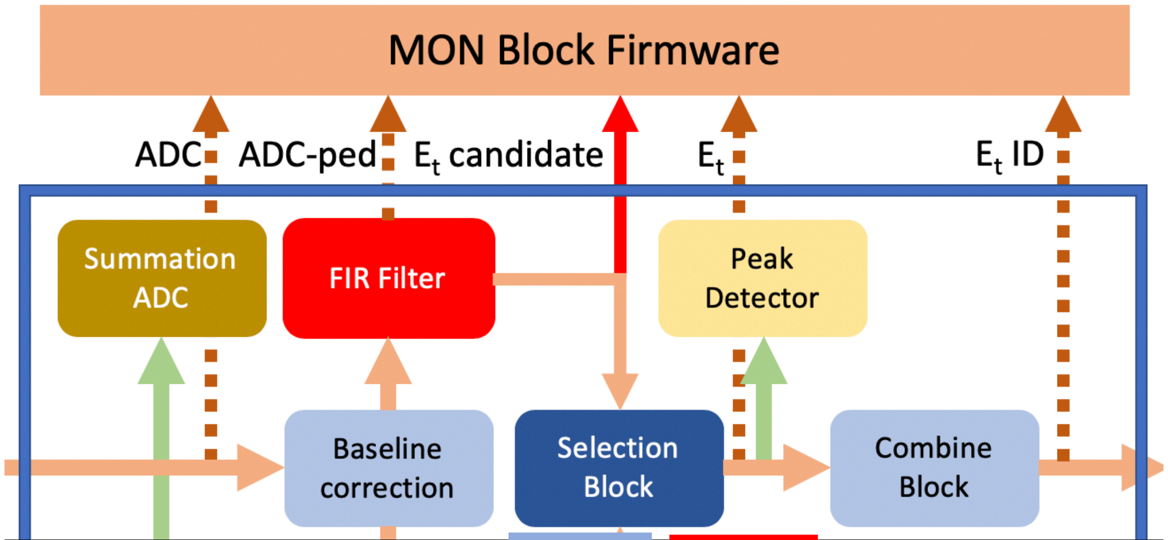


User Code on board

```
Result of 'transverse_e'
hardpoint : True
selection : False
combine : False
saturation : False
baseline : False
```

Configuration

“Transverse energy”
→Selection : False



Due to the selection is false,
E_t is same with E_t candidate
→ Validation for FIR filter

```
stream:  0, scid: 0, bcid:  215 -> 1/    0 - 1/    0 (expected - mon data)
stream:  0, scid: 0, bcid:  216 -> 1/   48 - 1/   48 (expected - mon data)
stream:  0, scid: 0, bcid:  217 -> 1/  175 - 1/  175 (expected - mon data)
stream:  0, scid: 0, bcid:  218 -> 1/  250 - 1/  250 (expected - mon data)
stream:  0, scid: 0, bcid:  219 -> 1/  154 - 1/  154 (expected - mon data)
stream:  0, scid: 0, bcid:  220 -> 1/   36 - 1/   36 (expected - mon data)
stream:  0, scid: 0, bcid:  221 -> 1/  -20 - 1/  -20 (expected - mon data)
stream:  0, scid: 0, bcid:  222 -> 1/  -39 - 1/  -39 (expected - mon data)
stream:  0, scid: 0, bcid:  223 -> 1/  -45 - 1/  -45 (expected - mon data)
stream:  0, scid: 0, bcid:  224 -> 1/  -48 - 1/  -48 (expected - mon data)
stream:  0, scid: 0, bcid:  225 -> 1/  -50 - 1/  -50 (expected - mon data)
stream:  0, scid: 0, bcid:  226 -> 1/  -51 - 1/  -51 (expected - mon data)
stream:  0, scid: 0, bcid:  227 -> 1/  -50 - 1/  -50 (expected - mon data)
stream:  0, scid: 0, bcid:  228 -> 1/  -49 - 1/  -49 (expected - mon data)
stream:  0, scid: 0, bcid:  229 -> 1/  -48 - 1/  -48 (expected - mon data)
```

All the outputs from User Code are consistent with User Code model.

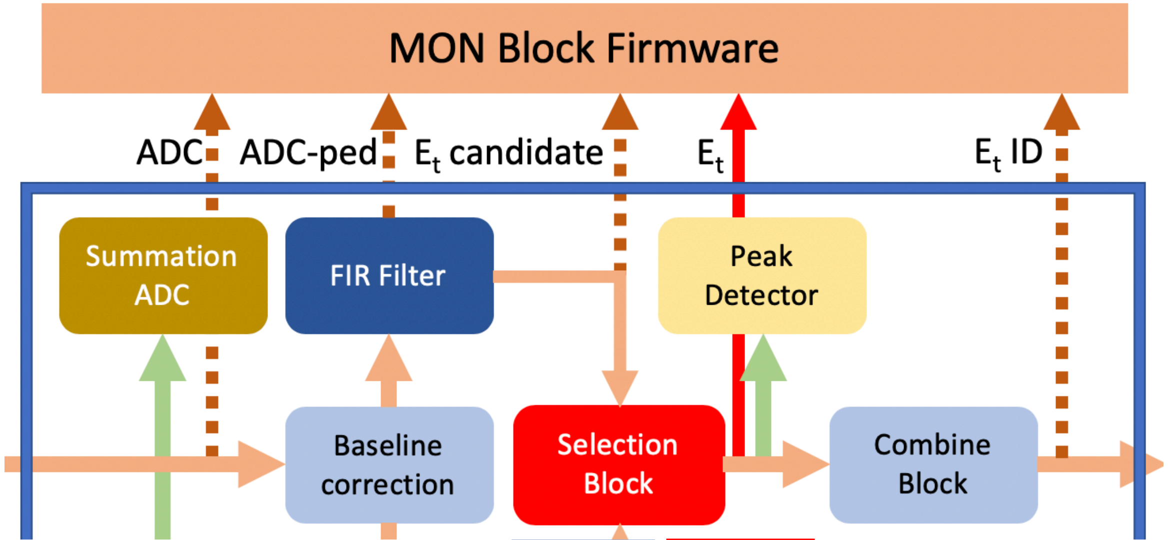
Firmware output

Model output

User Code on board

```
Result of 'transverse_e'
hardpoint : True
selection : True
combine : False
saturation : False
baseline : False
```

Configuration




```
stream:  0, scid: 0, bcid:  215 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  216 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  217 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  218 -> 1/   250 - 1/   250  (expected - mon data)
stream:  0, scid: 0, bcid:  219 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  220 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  221 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  222 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  223 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  224 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  225 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  226 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  227 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  228 -> 1/     0 - 1/     0  (expected - mon data)
stream:  0, scid: 0, bcid:  229 -> 1/     0 - 1/     0  (expected - mon data)
```

Firmware output

Model output

“Transverse energy”
→Selection : True

FIR filter validation was successful. Then apply filtering algorithm and detect true deposited energy.
→ Validation for Selection

Only one energy pass filtering algorithm. And it is consistent with User Code model