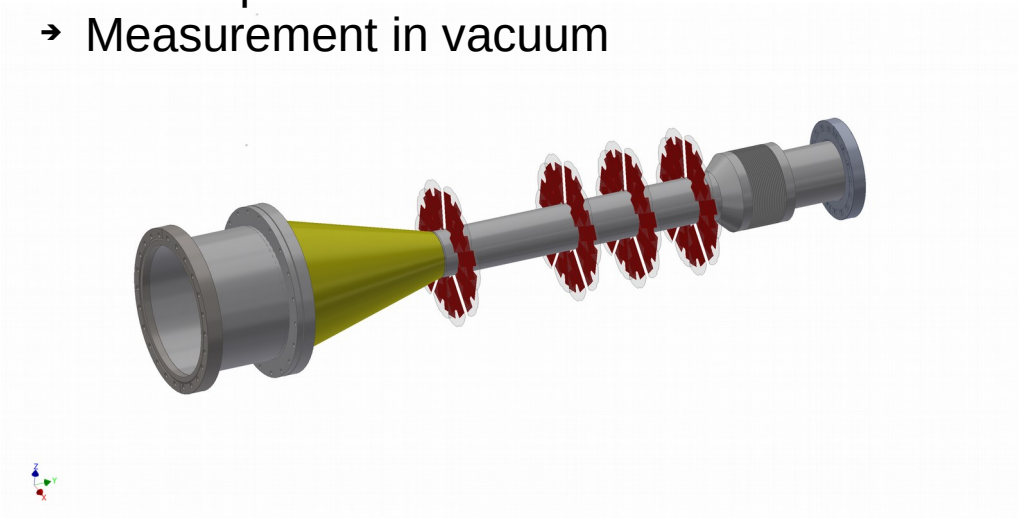
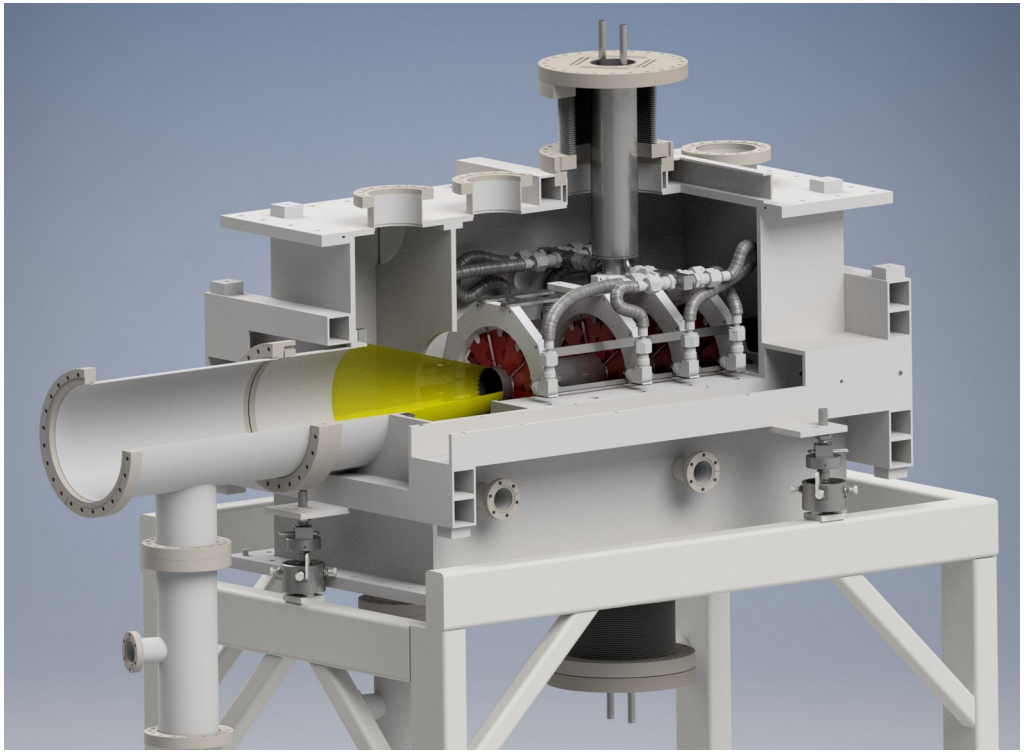
Status: Luminosity Detector Online Track Finding

Luminosity Detector

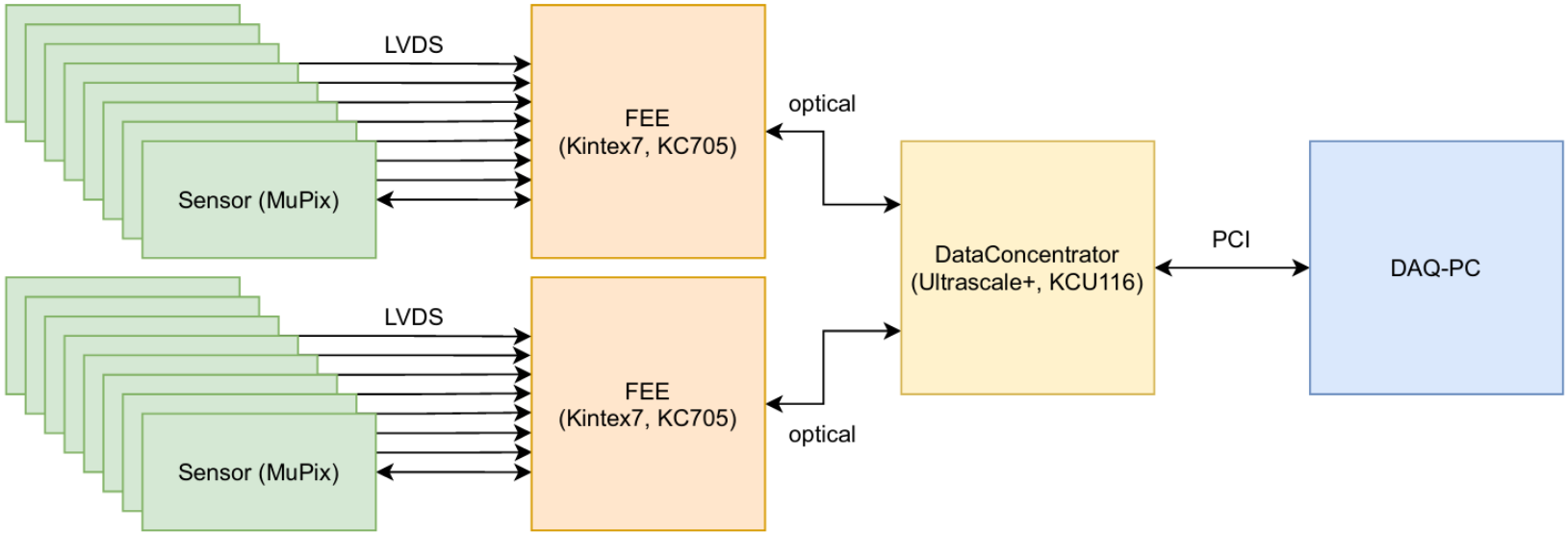
- Tracks must be measured precisely
- Prevent multiple scattering
  - ➔ Silicon pixel sensors
  - ➔ Measurement in vacuum





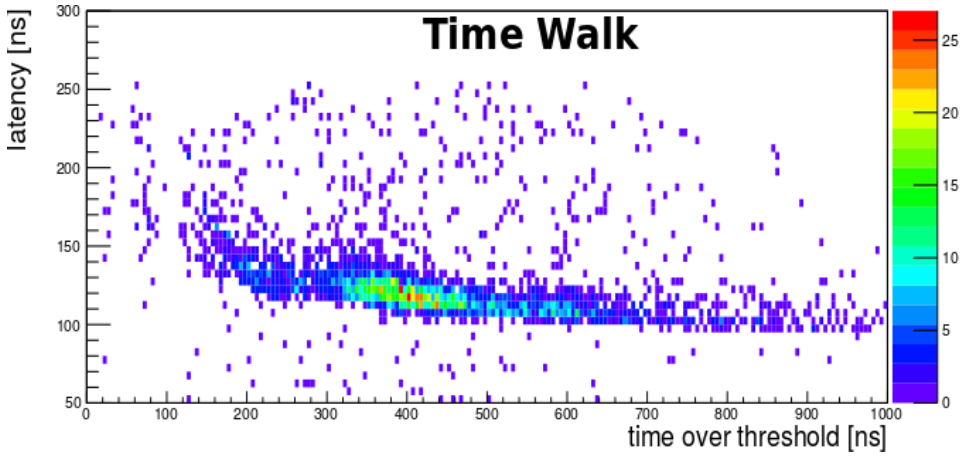
- Data rate of > 5,5 TB per day expected
- Save data after selection
- Selection in real time necessary

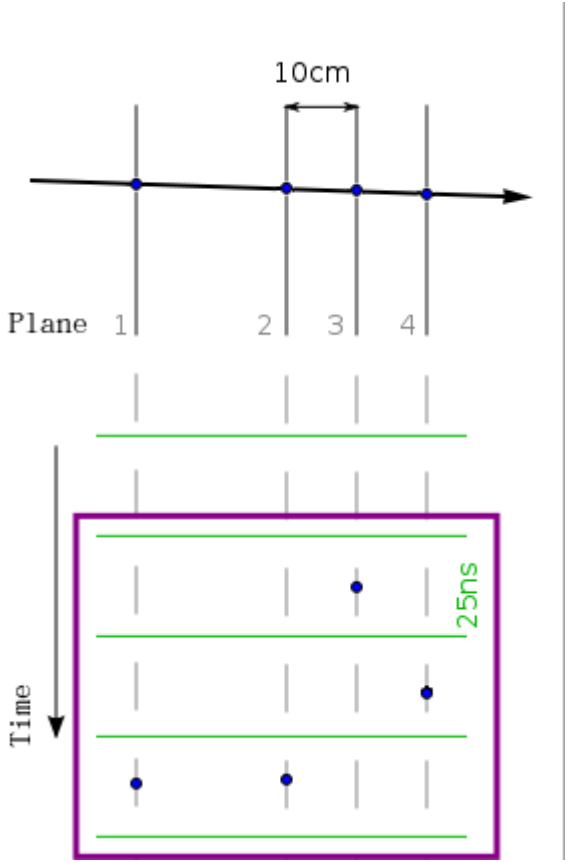
LMD Prototype DAQ



- Hit data from MuPix via LVDS links
- Data from FEE to DataConcentrator via optical links
- Track finding on DataConcentrator

Hit distribution in Time

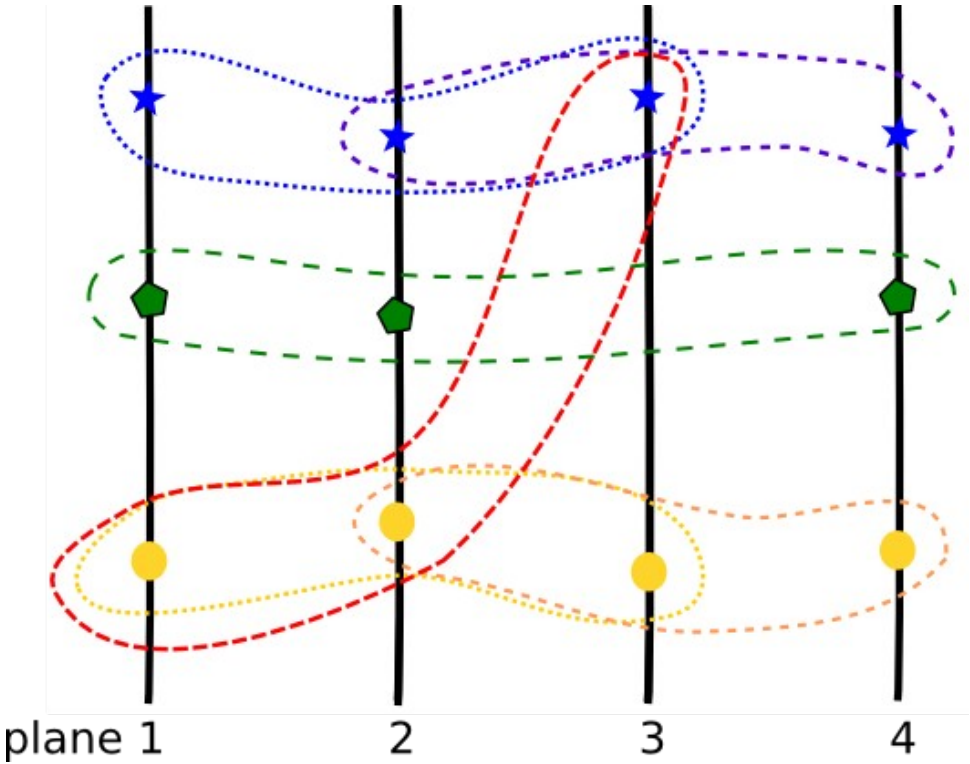




- Different charge depositions leads to different latencies Δt=75 ns
- Timestamp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ “Moving” window to get all tracks
- Avoid duplication by requiring hit in first timestamp of timeframe

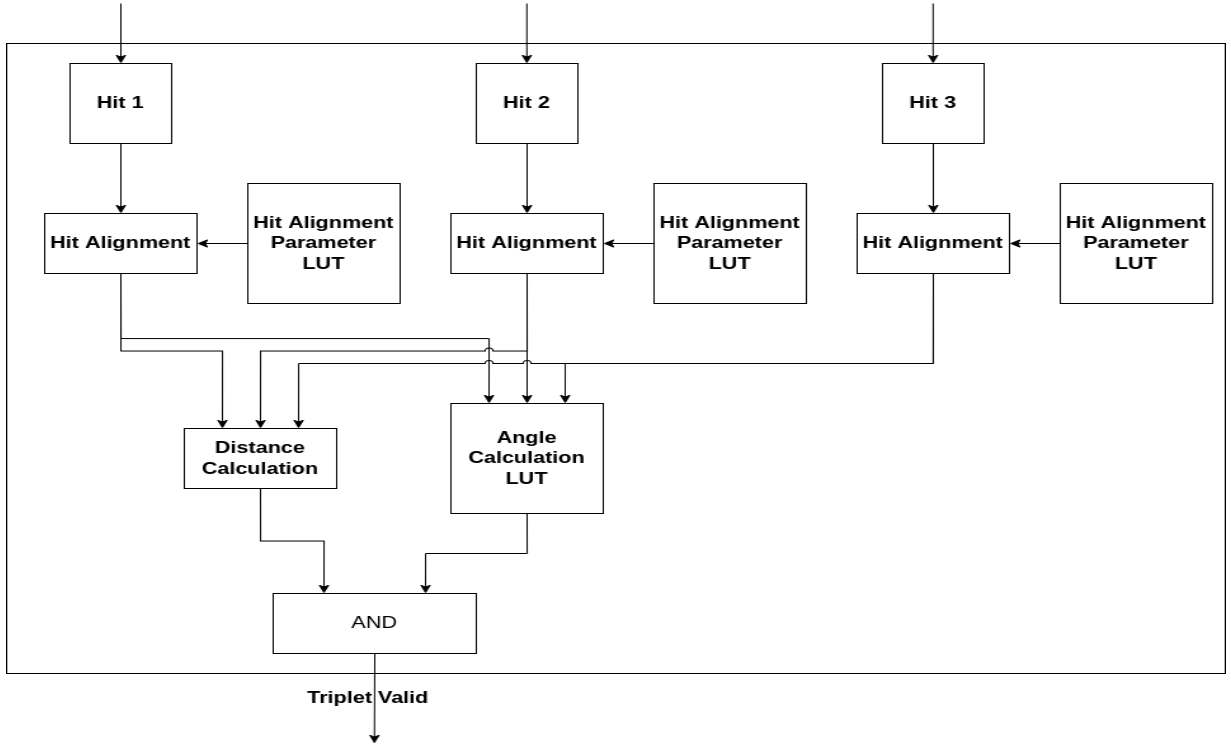
Triplet

- Three Hits form a triplet
- Angle between segments determines correctness of triplet
- Combine Triplets to Tracks
- One missing hit allowed



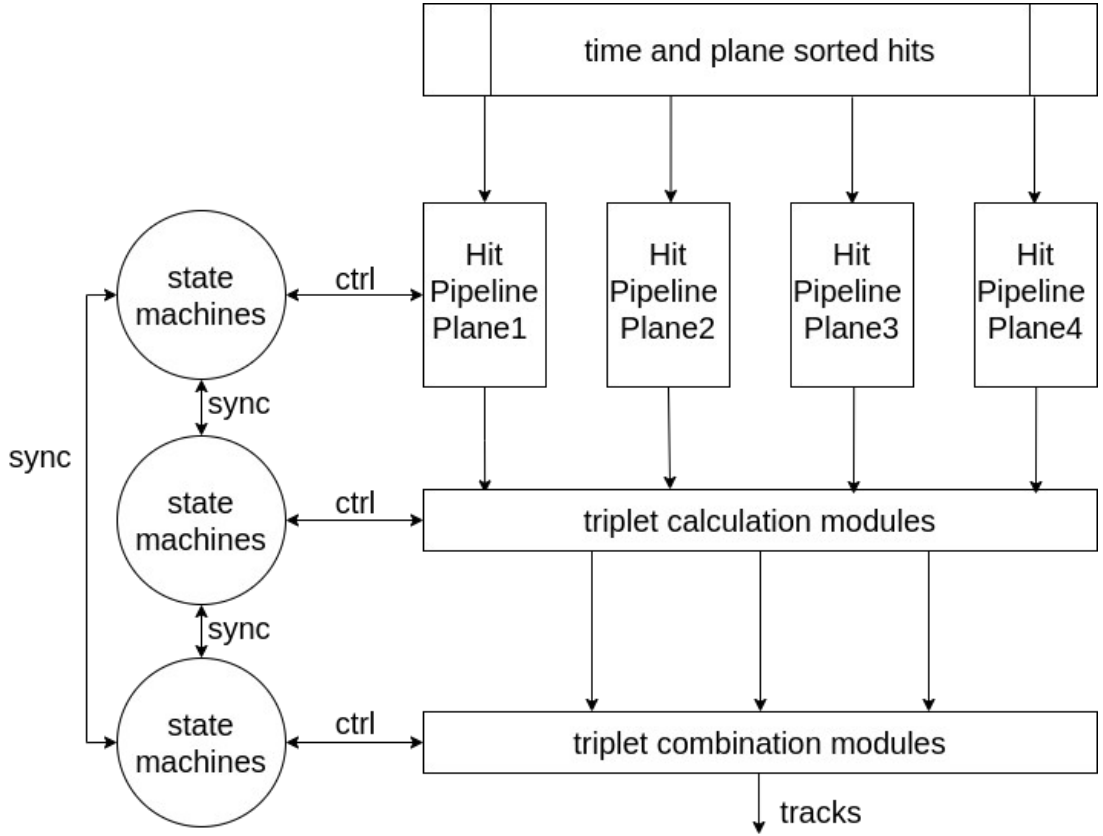
Triplet Check

- Hit alignment only for triplet angle calculation
- Hit alignment parameters stored as LUT (precalculated before implementation)
- Angle between triplet segments precalculated
- LUT only stores if angle is smaller than predefine value
- Distance calculation between hits to allow for smaller angle LUT

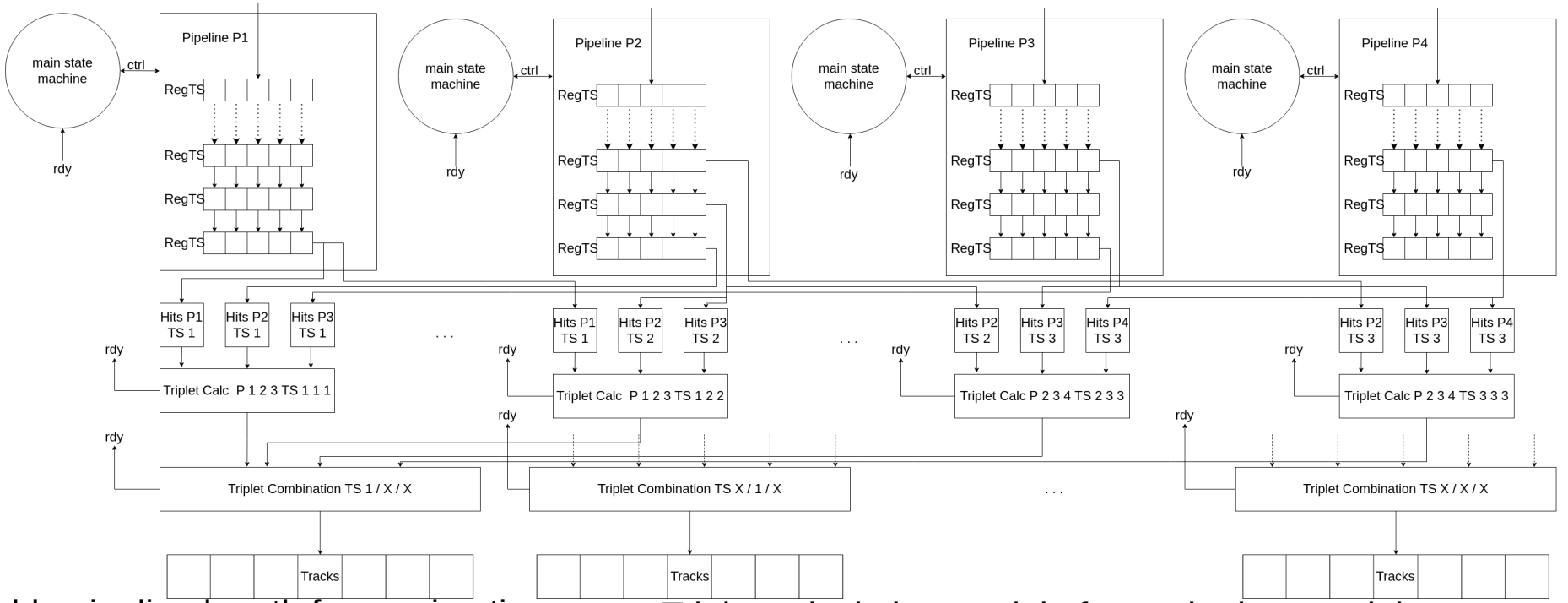


Tracking module

- Assumes hits sorted by time and detector plane
- Pipeline for each detector plane collects hits for each time stamp concurrently
- Each layer works simultaneously and iterates through specific amount of data combinations
  ➔ Clearly defined maximum time
- State machines control and synchronize individual modules and layers



Tracking module



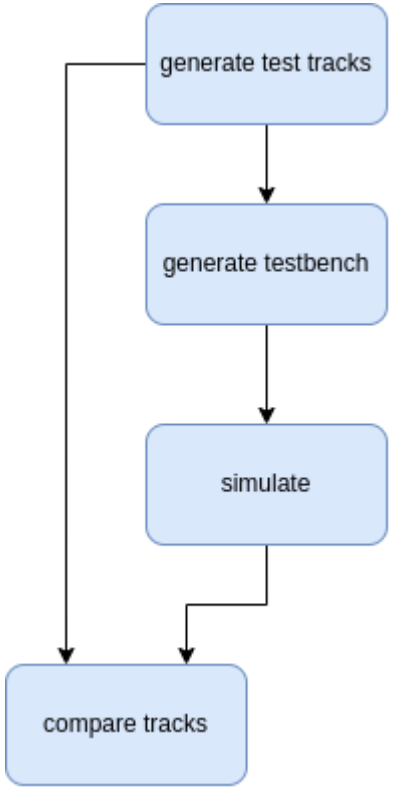
- Adjustable pipeline length for moving time window size
- Modules iterate through all hit/triplet combinations
- Triplet calculation module for each plane and time stamp combination
- Triplet combination module for multiple time stamp combinations but fixed first timestamp

Resource utilization

- Size of current small module (only 1 triplet combination module)
  - 45313 LUTs (~21% of device) / 34506 Flip Flops (~8 %)
- Moving window can’t grow to large or combinatorics explode
- If module turns to large there are multiple options
  - Single time stamp set triplet calculation → multi time stamp set triplet calculation
- If module turns out to slow
  - Decrease potential number of hits/triplets in each module
  - Multiply complete tracking module and run in parallel for different detector segments

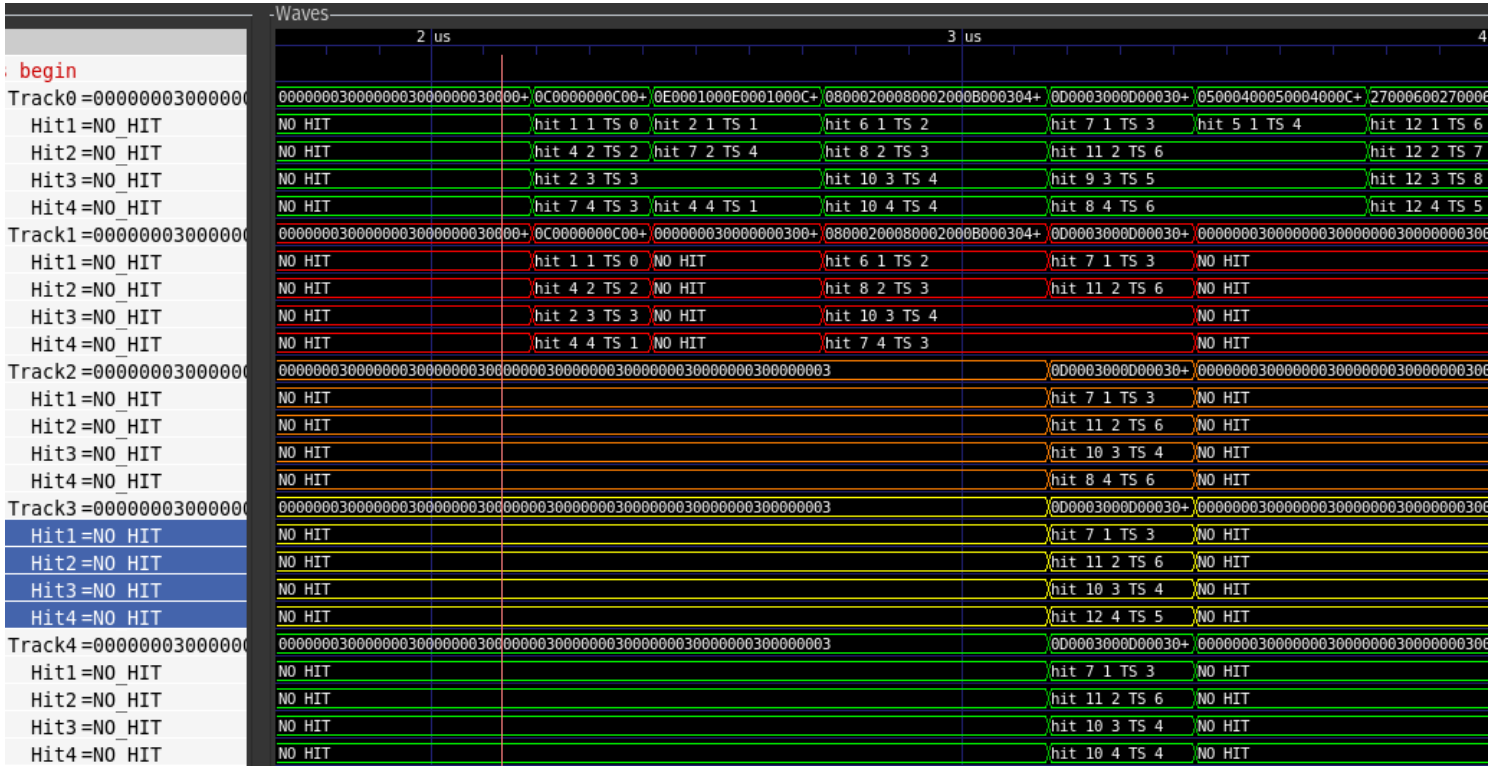
“Automated” Testing

- Python Script to generate
  - Testbench with random test tracks
  - JSON file with used test tracks
- Simulate Testbench
- Python Script to read and compare
  - JSON file with used test tracks
  - Tracks resulting from Simulation

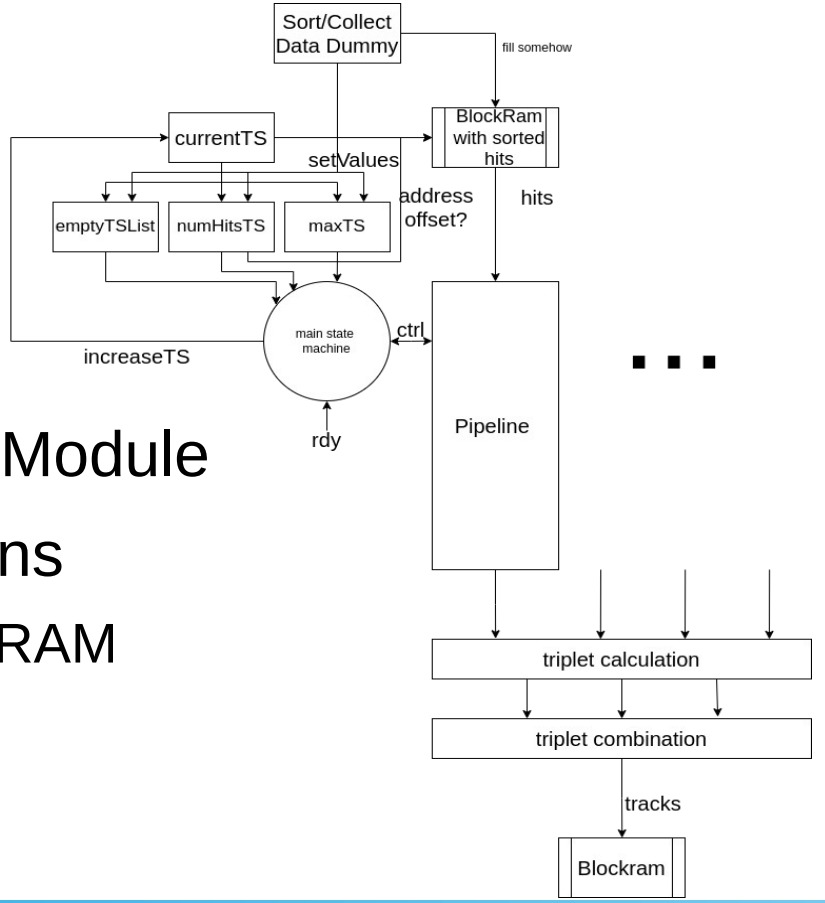


Behavioural Simulation

- Debugging ongoing
- Test tracks to close together
  - ➔ Many ghost tracks
- Synchronization issues
  - ➔ Triplet combination doesn’t start for to few triplets



Simulation Test Structure



- Not enough FPGA Pins to Map Tracking Module
- Mapping needed for Advanced Simulations
  - ➔ Implement Test Structure Based On Block RAM

Progress

- Basically Completed
  - Tracking Module Design
- In Progress
  - Debugging Behavioural Simulation
- To Do
  - Implement Test Structure For Post-Synthesis and -Implementation
  - Evaluate Resource Usage and Timing
  - Post-Synthesis and -Implementation Simulation
  - Expand Test Structure for Test on FPGA
  - Test on FPGA