Hardware Based Tracker Track Finders for Triggering at HL-LHC

Mark Pesaresi on behalf of the CMS and ATLAS collaborations

with thanks to Nikos Konstantinidis, Alberto Annovi, Peter Wittich, Kristian Hahn, Fabrizio Palla, Tom James, Giacomo Fedi

ACES 2018 - Sixth Common ATLAS CMS Electronics Workshop for LHC Upgrades

25th April 2018
HL-LHC will deliver higher luminosity than ever
- Ultimate luminosity of $7.5 \times 10^{34}$ Hz/cm$^2$
⇒ 200 pileup (PU) interactions per 25ns crossing

Isn’t tracker data already use in the trigger?
- CMS & ATLAS: In the CPU-based HLT processing farm (100 kHz)
- ATLAS: Also, Fast TracKer (FTK) hardware tracking pre-processor now coming on-line (100 kHz rate) -> frees significant CPU time

Why might tracker data needed earlier in the trigger?
- Detectors will be triggered $O(10)$ more often than present [e.g. ATLAS 100 kHz -> 1 MHz]
- Huge amounts of data will be produced [e.g. CMS 2 Tb/s -> 50 Tb/s]

All new silicon trackers for ATLAS & CMS at HL-LHC
- Higher granularities to cope with increased fluence [e.g. CMS ~75 million channels -> ~2 billion channels]

Extended performance at high pileup is important
- But just surviving will be difficult enough!
[SIMPLIFIED] HL-LHC TRIGGER ARCHITECTURES

**ATLAS** => **L0**
**CMS** => **L1**

- **L1 accept (Latency < 12.5μs)**
- **L0 accept (Latency < 10μs)**

**Readout Boards (subdet specific)**
- **Track Finder**
- **Cluster Finders**
- **μ Track Finders**

**Global Trigger**
- **Correlator Trigger**

**Readout Data (< 750 kHz)**

**DAQ Interfaces**

- **High Level Trigger Farm**
- **Output Data (7.5 kHz)**
- **Permanent Storage**

**FELIX Readout**
- **Event Filter Farm**
- **Permanent Storage**

**Tracker (ITK)**
- **Cluster Finders**
- **μ Track Finders**

**Central Trigger Processor**

**Calo Systems**

**Muon Systems**

NB: First Level Trigger
Necessary to include tracking information at first level of triggering

Rate w/o tracking approaches 4 MHz at 200PU!

Many trigger thresholds can be either maintained or improved – typically ~ x10 reduction

Vertexing a new handle at L1

NB: First Level ATLAS & CMS =
FTK would not scale at HL-LHC
- Factor of 20 input bandwidth increase
- Maximum processing rate would fall from 100kHz to 5kHz
- Alternatively reconstruction $p_T$ threshold must be raised from 2 to 10 GeV/c

CPU time for software based track reconstruction scales geometrically with PU

HLT needs to be able to maintain trigger thresholds/rejection
These describe **baseline designs** for ATLAS and CMS

ATLAS also considers an ‘evolved’ design
- Two stage trigger (L0 & L1) before full data readout
- Based on Regions of Interest
- Regional Tracking at 4 or 2 MHz, vs 1 MHz

Not discussed further
- Track Finder implications minimal
- More details in backup slides

ATLAS also considers an ‘evolved’ design
- Based on Regions of Interest
- Regional Tracking at 4 or 2 MHz, vs 1 MHz
Data arrives at Level0 rate of 1 MHz
-> online Storage volume

Event Filter (EF) computing farm consists of Processing Units
- Must reduce event rate from **1 MHz to 10 kHz** for offline storage
- Needs to **reconstruct tracks from Tracker**

Baseline solution -> **custom hardware-based Track Finding co-processor, known as the Hardware Track Trigger (HTT)**
- x10 reduction in CPU power required (30,000 -> 3,000 dual-socket servers)
TWO-STEP TRIGGERING WITH THE HTT

**Level0 information provides trigger path & Regions of Interest (RoIs)**

- Partial Tracker event data pulled from Storage (10% on average)
  - Hits from 8 outermost layers only

**Regional tracking \((p_T > 2 \text{ GeV/c})\) performed by HTT -> rHTT**

**EFPU makes use of rHTT tracks, in combination with L0 information**
- helps reduce rate to **400 kHz**

**Step 1**

**Step 2**

**Full hit data** (strip + pixel) pulled from Storage if required

**Full tracking \((p_T > 1 \text{ GeV/c})\) performed by HTT -> gHTT**

**EFPU uses gHTT tracks, in combination with ‘offline-like’ analyses**
- helps reduce rate to requisite **10 kHz**
rHTT & gHTT implemented by the same ATCA-based hardware system
- Builds on the present FTK experience

672 Tracking Processor (TP) blades
- 56 ATCA shelves, full-mesh backplane

Each TP handles small region in φ - η space
- Pre-duplication of hit data performed mainly by Event Filter network via HTT Interface (HTTIF)
Two TP stages depending on trigger path:

**rHTT -> 1st Stage TP only**
- Tracks \( p_T > 2\text{GeV/c} \) passed back to EF

**gHTT -> both 1st and 2nd Stage TPs**
- Tracks \( p_T > 1 \text{GeV/c} \) passed 1st -> 2nd stage
- Tracks are refined with full TK hit data
Single base card, differentiated by Mezzanines, implement 1\textsuperscript{st} and 2\textsuperscript{nd} stage TP types

1\textsuperscript{st} Stage - AMTP
- Hit clustering and data organisation
- Uses Associative Memories (AMs) to match roads from clusters in up to 8 layers
- Track Fitting on roads performed in an FPGA
- Pattern Recognition Mezzanines (PRM)

2\textsuperscript{nd} Stage - SSTP
- Second Stage (SS) TP receives 1\textsuperscript{st} stage tracks/clusters from AMTPs, plus full event data from HTTIF
- Hit clustering in new layers
- Track Extrapolation & Re-Fitting performed in FPGAs
- Track Fitting Mezzanines (TFM)
TP ATCA blade includes:
- **2 FPGAs** (targeting Xilinx Kintex Ultrascale)
- 8Gb external RAM per FPGA
- **2 slots** for PRM or TFM mezzanines

**Low demands in terms of I/O**
- ~16 x 10 Gb/s optical link pairs
- RTM for optical interface to HTTIF, AMTP->SSTP link
- Backplane interface to other boards (full-mesh)

**Power per blade estimated at ~300W**

**Complexity hidden in TP firmware**
- Hit data clustering and organisation
- Event synchronisation and management
- Switch matrix for board-to-board communication
- Mezzanine readout/control & management
- Duplicate Track Removal
1152 Production PRMs required for final system
- Each with 12 AM ASICs (6 x 2)
- Processing FPGA; likely Kintex Ultrascale
- External RAM for storing patterns and constants

FPGA responsible for
- Superstrip conversion
- Data Organisation to handle hits & roads
- Track Fitting (see TFM)
- AM configuration & monitoring

Production PRM target: 1.5M -> 4M patterns for $p_T > 1$ GeV/c
Pre-prototype PRMs in hand [INFN]
- Based on AM06
- Otherwise meets expected requirements for production PRM

Demonstration of PRM06 on a Pulsar IIb [FTK Data Formatter board]
- As part of a CMS test stand
- Results documented here: https://cds.cern.ch/record/2272264/files/CMS-TDR-014.pdf
AMTP – PATTERN RECOGNITION MEZZANINE

AM07 evaluation ongoing - first results encouraging
- Verified lower power consumption per comparison matches simulation

Details on first results in dedicated ACES poster by G. Fedi

AM09 requirements vs AM06
- Higher pattern density [384k patterns]
- Faster clock speed/bandwidth [250 MHz core/200 MHz readout]
- Low power [4W @60 MHz cluster rate]
- Better decoupling (vs. AM06)

AM06
65nm
128k patterns
2015
Used in ATLAS FTK

AM07
28nm
16k patterns
2016
Evaluation ASIC

AM08
28nm
16k patterns
Q4 2018
Final Technology Selections

AM09 for HTT
28nm
384k patterns
12 AMs -> 4M patterns/PRM
2019/20
Production ASIC

13,824 AMs in HTT!
**SSTP – TRACK FINDING MEZZANINE**

**TFM must refit tracks** from AMTP using full ITK hit information
- Uses a **linearised track fitter** (same as in PRM -> based on FTK)
- \(O(100)\) million pre-calculated coefficients per mezzanine

Likely design specifications:
- 2 Kintex Ultrascale FPGAs, 11 x 10 Gb/s optical inputs
- DDR3 for storing constants
- 192 needed in final system
A TRACKER DESIGNED FOR TRIGGERING

Tracker provides limited hit data at **full 40 MHz crossing rate**
- No inner pixel tracker hits (Inner Tracker)
- Hits from outer 6 tracking ‘layers’ only (Outer Tracker)
- Only hits compatible with tracks $p_T > \sim 2$ GeV/c read out

$p_T$ discrimination provided by use of special modules
- Pairs of closely spaced silicon sensors, separated 1-4 mm
- Signals from each sensor are correlated
- Only hit pairs compatible with $p_T$ threshold (“stubs”) are forwarded off-detector
- Factor $\sim 10$ data reduction

---

**Strip-Strip (2S) modules**
- Both sensors: $2 \times 1016$ strips
  - 5 cm long, 90 μm

**Pixel-Strip (PS) modules**
- Top sensor: $2 \times 960$ strips
  - 2.4 cm long, 100 μm
- Bottom sensor: $32 \times 960$ pixels
  - 1.5 mm $\times$ 100 μm
Even still, data rate **not insignificant**!
- Average 15,000 stubs every 25ns (200PU)
  -> Stub bandwidth $O(20)$ Tb/s

L1 hardware trigger reduces event rate from **40 MHz to <750 kHz** using calorimeter, muon and tracker primitives
- TK primitives are all tracks ($p_T > 2$-3 GeV/c), from Outer Tracker
- L1 accept triggers all front end buffers to read out to DAQ -> HLT farm

FE L1 latency buffers (including TK) limited to 12.5 $\mu$s

-> Track finding from stubs must be performed in 4 $\mu$s
Two stages of data processing
- DAQ, Trigger and Control (DTC) layer
- Track Finding Processing (TFP) layer
- All-FPGA processing system
- ATCA; CMS standard backplane (dual-star)

Outer Tracker cabled into nonants
- Use of **time-multiplexing** to increase parallelisation

Time-multiplexing directs data from multiple sources to a single processing node
- Employed in CMS L1 calorimeter trigger
- 1 event per processing node
- Fixed passive optical patch panel as MUX

Helps minimise complexities due to physical segmentation and boundary handling

Processors are independent entities – simplifies commissioning and operation

Spare nodes available for redundancy
DTC card must handle
- \(\leq 72\) modules (5G/10G lpGBT opto-links)
- Control/Readout for each module
- Direct L1 stream to central DAQ (16G/25G)
- Direct stub stream to TFPs (16G/25G)

Stub pre-processing includes:
- Local->Global look up, position calibration
- Sorting and pre-duplication
- Time-multiplexing

\(\Rightarrow\) 216 DTC boards, 18 shelves, 1 rack/nonant
TRACK FINDER ARCHITECTURE - TFP

Two stages of data processing
- DAQ, Trigger and Control (DTC) layer
- Track Finding Processing (TFP) layer
- All-FPGA processing system
- ATCA; CMS standard backplane (dual-star)

Outer Tracker cabled into nonants
- Use of **time-multiplexing** to increase parallelisation

TFP card must handle
- Up to 72 DTCs (16G/25G optical links)
- Track Finding from stubs
- Track Fitting
- Transmission to L1 Correlator Trigger

High bandwidth processing card
- Rate to L1 Correlator much lower < 30Gb/s

-> 144 TF boards, 12-18 shelves

~1 Tb/s processing card

overall compact system
(30-36 shelves)
integrating readout & trigger
Two principle algorithms for reconstructing tracks
- Plus a number of hybrids, variations and options

**Tracklet Approach**
- Combinatorial approach using **pairs of stubs as seeds**
- **Extrapolation** to other layers -> hit matching
- **Linearised χ² fit** on candidates
- Uses **full resolution stubs** at earliest stage of processing
- N time-slices x M regions -> 6 x 24, 9 x 18

**Hough Transform + Kalman Filter Approach**
- Uses a **Hough Transform** to detect coarse candidates
- Candidates are filtered and fitted in a single subsequent step using a **Kalman Filter**
- Combinatorial problem pushed to latter stages of processing
- N time-slices x M regions -> 18 x 9
Seeding Step
- Seeds are generated from **pairs of layers**
- Layer pairs selected to give full coverage in $\eta$, including **redundancy**
- Seeding step **massively parallelised** by internal geometrical partitioning & stub organisation (Virtual Modules - VMs)

only ~30% of VM pairs compatible with $p_T > 2 \text{ GeV/c}$
- reduces combinatorics & f/w complexity

Tracklet Projection & Fitting Steps
- Tracklet seeds are projected to other layers -> matched with stubs in VMs
- **Residuals calculated** -> update track parameters
- Track re-fitted at each step using linearised $\chi^2$ fit -> constants tabulated in LUTs
- Duplicate removal step needed due to multiple seeding layer pairs

Demonstration in hardware, verified using emulation software
- MC stubs from PU 0->200 samples passed through slice demonstrator
- Latency verified to be **3.3 $\mu$s**, agrees with latency model
Hough Transform (HT)
- Search for primary tracks in $r$-$\phi$ using parameterisation $(q/p_T, \phi_0)$
- Stub positions correspond to straight lines in Hough Space
- Where 4+ lines intersect -> track candidate
- Internally parallelised into 36 independent $\eta$-$\phi$ regions (HTP), 1 HT array per region

Combinatorial Kalman Filter (KF) & Duplicate Removal (DR)
- Iterative algorithm as used in offline reconstruction, seeded by HT
- Able to fit and select different combinations of stubs on track candidates -> best combination selected on $\chi^2$
- Only constants are hit errors (and scattering angle vs $p_T$ if required)
- Simple DR block removes HT candidates with similar helix parameters after fit

Demonstration in hardware and emulator
- Many samples from PU 0->200 passed through MP7-based slice demonstrator
- Latency verified to be 3.5 $\mu$s
R&D ACTIVITIES

- Evaluation of Xilinx Ultrascale/Ultrascale+ FPGAs

- Testing and evaluation of data transmission at 16->25Gb/s
  - Trialling different optical transceivers and modules (mid-board, edge-mount..), attempting to follow industry
  - PCB signal integrity & design for high speed
  - Protocols & latency (important for L1)

- Evaluation of embedded CPU options, kick-start s/w projects
  - Required for slow control plus on-board processing/calibration
  - Zynq SoC; integrated FPGA/ARM control processor
  - Computer On Module (COM) express; pluggable module based on quad Intel Atom

- [Image of YUGE uTCA platform]
  - Samtec Firefly x2
  - FPGA socket
  - MTCA connector
  - RTM connectors
  - RTM blind-mate optical

- [Image of Mars ZX2]
  - 84mm x 55mm
  - dedicated ACES talk by R. Spiwoks

- [Image of COM express T10 mini]
  - 67 x 30 mm SO-DIMM

- [Image of Aurora 64b/66b demonstration]
  - electrical loopback & board to board
  - via Firefly twinax @ 28Gb/s
  - optical loopback at 14Gb/s verified

- [Image of YUGE uTCA platform]
  - Zynq XC7Z020 (Mars ZX2)
  - Samtec Firefly x2
  - QSFP28 x2
  - Ethernet

- [Image of YUGE uTCA platform]
  - YUGE uTCA platform
  - COM express T10 mini

- [Image of YUGE uTCA platform]
  - Aurora 64b/66b demonstration
  - BER < 1e-16

- [Image of YUGE uTCA platform]
  - 140 ns
  - electrical loopback & board to board
  - via Firefly twinax @ 28Gb/s
  - optical loopback at 14Gb/s verified
- **ATCA infrastructure**
  - Systematic thermal studies - significant concerns about air x-section and impact on opto-lifetime
  - Backplane signal integrity -> important for DAQ/timing

- **Use of interposer technology**
  - Flexibility (e.g. FPGA)
  - Mitigate losses/costs due to yield issues
  - Modularity; separate complex and simpler parts of board design

- **PCB design practices, stackup and material**
  - Build up relationship with manufacturers

ATCA Test Blade

**R&D ACTIVITIES**

ATCA developments discussed in dedicated ACES talk by E. Hazen

- **Serenity ATCA Platform**

FPGA KU115 daughtercard on interposer

recently returned from manufacturer - testing ongoing

Samtec Z-RAY interposer

Samtec Firefly x16 RX/TX pairs

COM express

QSFP28

CERN IPMC

133 x 30 mm

CERN IPMC
SUMMARY & OUTLOOK

Both ATLAS and CMS must include tracking information in their trigger decision logic at HL-LHC
- CMS at L1 (first level)
- ATLAS at the EF (high level)

ATLAS track finder an evolution of the FTK
- More powerful Associative Memory (AM09), key technologies currently under test
- Acts as a co-processor farm, independent nodes receiving data from the network
- More homogeneous hardware solution, common components, interfaces and technology
- Baseline configuration defined Q2 ’19, prototype demonstrators in 2021

CMS track finder is a new experience
- Stringent requirements on latency and performance to keep L1A rate below 750 kHz
- Leveraging advances in FPGA technology, and industry where possible
- Pre-prototype slice demonstrators in 2016 -> prototype demonstrators in 2019
- Final system specification Q4 ’19

Both systems based on ATCA
- Blades will dissipate significant heat
- Requires careful board design to minimise hotspots and protect optics

IpGBT/VL+ common to detector readout.
Other common projects? IPMC, embedded systems?
AM PRINCIPLE

[Diagram showing a pattern with labels 'pattern 0', 'pattern 1', 'pattern 2', 'pattern 3', 'pattern n', 'Bus_Layer<0>', 'Bus_Layer<1>', 'Bus_Layer<2>', '....', 'Bus_Layer<7>', and connections labeled 'HIT', 'READOUT TREE', and 'OUTPUT BUS'].
EVOLVED OPTION

### 4 MHz L0A Scheme vs 2 MHz L0A Scheme

<table>
<thead>
<tr>
<th>Metric</th>
<th>4 MHz L0A Scheme</th>
<th>2 MHz L0A Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Level-1 Latency</td>
<td>30 μs</td>
<td>35 μs</td>
</tr>
<tr>
<td>Fraction of Regional Data</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Effective Regional Readout Rate</td>
<td>400 kHz</td>
<td>200 kHz</td>
</tr>
<tr>
<td>Full ITk Detector Readout Rate</td>
<td>600 kHz</td>
<td>800 kHz</td>
</tr>
<tr>
<td>Total Data Rate</td>
<td>1 MHz</td>
<td>1 MHz</td>
</tr>
</tbody>
</table>
## Evolved Option

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Latency requirement</th>
<th>Level-0 rate [MHz]</th>
<th>Trigger threshold [GeV]</th>
</tr>
</thead>
<tbody>
<tr>
<td>rHTT</td>
<td>No</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>L1Track</td>
<td>6.0 μs</td>
<td>2–4</td>
<td>4</td>
</tr>
</tbody>
</table>

![Diagram showing network switch and ATCA units](image)
FTK in Run II / Run III

Provides full 100kHz tracking as an input to the HLT
- Tracking in $|\eta| < 2.5$
- $p_T > 1$GeV
- Tracking volume is divided in 64 $\eta - \phi$ towers
- All events selected by Level-1 are processed

Very complex hardware system
- About 150 boards of 6 different kinds, each with specialized functions
- Heterogeneous connections between boards (fiber, backplane, connectors, and with different bandwidth / width)

Regional Tracking is done also in CPU software
Optimization

- Found a range of optimal settings
  - Within the range open areas vary by +/- 5% (tolerance)
- Settings closer to IBERT default are chosen
  - RX Termination = 400 mV, TX Pre/Post Cursor Emphases = 2.21/0.00 (dB), TX Driver Swing Control = 924 mV
- Eye diagram reproduced with maximum amount of HORZ/VERT_OFFSET increments
- Wide open area observed

Results

- BER range = [0,1e-12]
- Wide open area observed
- Expected (20%) shrinking in 95% data
  - N.B: chosen a medium amount of ___OFFSET, otherwise ~20 days needed
  - BER < 1e-16 (test duration ~ 4 days)
25Gb/s TESTING

How to start

1. Aurora Example Design (link)
2. Ultrascale FPGA Transceiver Wizard Example Design (link)

Latency Measurement (via 2.)

- **Line rate** = 25 Gbps, **clk frequency** = 390 MHz
- **TX**: 64b packets from the wrapper moved to the channel, ready to be sent over GTY transceivers
- **RX**: Packets received over GTY transceivers, moved to the user interface
- **Watched TX and RX**: **latency ~ 50 ns**
  - extrapolated from 21 clks @ 390 MHz
  - smaller latency than 1. because of the reduced overhead logic
  - smaller than what expected (~70-80 ns) from previous studies (link)...investigating

Latency Measurement (via 1.)

- **Line rate** = 25 Gbps, **clk frequency** = 390 MHz
- **TX data**: 64b packets from the user interface moved to the channel, ready to be sent over GTY transceivers
- **RX data**: Packets received over GTY transceivers, moved to the user interface
- **Watched TX and RX**: **latency ~ 130 ns**
  - from specs (GTH) expected ~55 clks @ 390 MHz = 140 ns

- Imported **IBERT optimal settings** into the design
- No errors observed for ~ 5 days
- **BER < 1e-16**
Power

- FPGAs
- Dual Kintex KU115: **45W – 68W**
- Single VU9P: **90W – 130W**

- Optics Module Power Dissipation
  - 12x 16G, 1.7W nom, 3.6W max
  - 4x 28G, 5W nom

- Dual FPGA Tracker Board
  - x6 16G and x6 (perhaps x3) 28G per FPGA
  - Assume 2x20W over 2x6 modules per FPGA

- 96 Ch Trigger Board
  - x24 28G for single FPGA
  - Assume 2x60W over 2x12 modules

- 64 Ch Trigger Board
  - x16 28G for single FPGA
  - Assume 2x40W over 2x8 modules
THERMAL STUDIES

FPGA Heatsink

- Types
  - Die casting, Extrusion, Stacked fin, Folded fin, Skived fin

- Types
  - Moderate, Sparse
  - Cu or Al

Single Mid-range Ultrascale+

This is best case – Real world performance may be different

Set to high LUT usage, low DSP usage

Two estimates of air speed with 96 dBA in USC55

480 Mhz clock - 130W

240 Mhz clock - 90W

452X592, 157 x 157 x 250
**System Overview**

- Algorithm consists of **11 processing steps** (hand-optimized Verilog)

- Each processing step of the algorithm
  - Separated by **memories** (BRAMs):
    - read in memory → process → write out memory
  - Processes a **new BX every 25 ns** ° TMUX = 25 * 6 = 150 ns currently
    - Has a **fixed time** to produce its first output (latency)
    - ** Pipelined design →** producing outputs until new BX (150 ns)

- **Automated wiring** (python script) of the processing modules ←→ memories
  - Driven by configuration file
  - Python script wires firmware and integer emulation ←→ identical setup
  - Also generates…

**Step 1) Stub Organization**

- Stubs coming from DTC are split by the **Layer Router** into each layer/disk

  ![Layer Router Diagram]

- Stubs in from DTC
  - Header (with BX)
  - Encoded order
  - Each stub
  - Trailer

**Virtual Modules**

- Split the φ sector into smaller φ regions (full length in z)
  - Inner (odd) layers (eg. L1) 24 φ divisions
  - Outer (even) layers (eg. L2) 16 φ divisions

**VM memories** for even layers split into 8 z bins

- Total of 24 * 16 = 384 pairs of VMs, only **120 pairs** consistent with $p_T > 2$ GeV
**TRACKLET**

### Tracklet Formation
- Seed by forming a **tracklet** from pairs of **stubs** in adjacent layers (or disks)
  - Initial track parameters from stubs + IP
  - Must be consistent with $p_T > 2 \text{ GeV}, |z_0| < 15 \text{ cm}$

![Image of Tracklet Formation Diagram](image)

**Built in Redundancy**
- Seed multiple times in parallel
  - good coverage & redundancy
  - Barrel: L1+L2, L2+L3, L3+L4, L5+L6
  - Disk: D1+D2, D3+D4
  - Overlap: L1+D1, L2+D1

**Inverse of radius of curvature:**

$$\rho_1 = \frac{2 \sin \Delta \phi}{r_2 - r_1}$$

Starting point, Taylor expand to translate to calculation for FPGA

Details on integer calculations in DN

**Note:** Using full stub resolution!

### Tracklet Parameter Resolutions
- Even at this stage, tracklets have good track parameter resolutions

![Image of Tracklet Parameter Resolutions](image)

**Step 3) Projections**
- Use tracklet to **project** track to other layers and disks
  - Project both inwards and outwards
  - All projections made in parallel in the **Tracklet Calculator**

![Image of Tracklet Projections](image)

- Projection calculation:
  - Use the average radius ($r$) for barrel (disk)
  - Correct avg. position by derivatives (LUTs) to get exact projection for actual stubs

**NB:** Keep these derivatives for use in matching
Hough Transform

- Pipelined firmware processes one stub per clock cycle. The Book keeper receives stubs and propagates to each $q/p_T$ bin (represents one array column) in turn.
- The $p_T$ estimate of the stub from stacked modules is used to constrain the required $q/p_T$ space.
- Inside the Bin, the corresponding $\phi_0$ of the stub for the column is calculated and the appropriate cell(s) are marked.
- Candidates marked by stubs from > 4 layers propagate back to the Book Keeper and are read out.
Kalman Filter

- 1) Incoming stubs stored in BRAM for later retrieval
- 2) Seed state creator outputs initial state (HT seed)
- 3) State control multiplexes incoming seeds and partially worked states
- 4) Stub state associator retrieves next stub (in increasing radii) from memory
- 5) Kalman filter updates matrices and state with weighted average of previous and new inputs
- 6) Repeat for (a configurable) accumulation period (or until 4 stubs are added to all tracks)
- 7) State filter selects the best state for each candidate ($\chi^2$)
  - Processing latency dominated by matrix math update, 230 ns per iteration
BDT Fake Rejection

- Idea to use a gradient boosted decision tree, implemented in FPGA logic, to select and remove fake tracks after the track fit
- Make a static, fully pipelined implementation of a pre-trained BDT ensemble
- Train ensemble on a CPU (using scikit-learn)
- Export trained ensemble to JSON file
- Read by firmware
- 4 integer features
  - $\chi^2$, $|1/p_T|$, $|\tan \lambda|$, num. skipped layers
- 100 trees, depth 3
- Tuneable on eff. vs fake rate curve
- Latency 12 clocks @ 400 MHz (Stratix V), 30 ns only!

KF, $t\bar{t}$ at 200 PU

### Efficiency vs Fake Rate Curve

<table>
<thead>
<tr>
<th>Efficiency Loss [%]</th>
<th>1.0</th>
<th>0.5</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake Rate Reduction [%]</td>
<td>70</td>
<td>50</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BDT</th>
<th>V7-690</th>
<th>KU-115</th>
<th>VU-9P</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTS [%]</td>
<td>2.24</td>
<td>1.46</td>
<td>0.82</td>
</tr>
<tr>
<td>FFs [%]</td>
<td>1.14</td>
<td>0.75</td>
<td>0.42</td>
</tr>
</tbody>
</table>

**FPGA Resources**

- $n$: number of trees
- $2^n$: maximum depth