A novel 4D fast tracking pixel detector

M. Petruzzo\textsuperscript{1,2}, M.Citterio\textsuperscript{1}, J.Fu\textsuperscript{1}, P.Gandini\textsuperscript{1}, A. Lai\textsuperscript{3}, V. Liberali\textsuperscript{1,2}, N.Neri\textsuperscript{1,2}, S.Riboldi\textsuperscript{1,2}

\textsuperscript{1}INFN, Sezione di Milano
\textsuperscript{2}Università degli Studi, Milano
\textsuperscript{3}INFN, Sezione di Cagliari

Connecting the Dots and Workshop on Intelligent Trackers

April 5, 2019
LHCb Upgrade II

- **Major detector upgrade** during Long Shutdown 4 in 2030
- **Luminosity up to** $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
  - 10x w.r.t to Upgrade I
  - ~50 visible interactions per crossing
- **Aims at collecting** $> 300 \text{ fb}^{-1}$

**Big detector challenge:**
- Tracking: 1500-3500 charged particles per bunch crossing
- PID: higher occupancy, increase the momentum $< 10 \text{ GeV}/c$ and $> 100 \text{ GeV}/c$
- ECAL: sustain radiation dose $\leq 200$ Mrad, energy resolution $\sigma(E)/E \sim 10\%\sqrt{E} \oplus 1-2\%$, reduce Moliere radius
- Radiation damage: greater concern for certain sub-detectors
Track reconstruction challenge

- Upgrade I VELO detector cannot sustain the increased luminosity of Upgrade II:
  - 10x luminosity up to $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, 40% ghost rate, PV merging, $\sim 13\%$ mis-association.

Possible solution in VELO Upgrade II:

- Reduce pixel size $55\mu m \rightarrow 27.5\mu m$
  - Ghost rate reduction to acceptable level
  - PV-mis association still present
- Timing information of the VELO hits:
  - $\sim 200$ ps resolution
  - $\sim 13\% \rightarrow \sim 1\%$ PV mismatch reduction
Role of timing

- **Timing in track reconstruction:**
  - *Hits distributed in time* with RMS ~ 170ps
  - Need ~30ps hit resolution to discriminate hits with similar positions, based on time
  - **Use only compatible hits** in the pattern recognition

- **Timing in PV-association:**
  - Additional power to select the correct primary vertex
  - Hit time resolution needed: ~200ps
  - Enough to separate tracks with similar spatial parameters

- **Others:**
  - Precise measurement of time of primary vertices
  - Track time stamping for better association of track upstream and downstream the magnet
  - Timing in PID and calorimetry
  - **Reduce data to process in High Level Trigger** by selecting only interesting pp interactions
TIMESPOT R&D INFN project

- **TIME & SPace real-time Operating Tracker:**
  - 1 M€ three years project (from 2018), funded by INFN
  - Development of a silicon and diamond 3D tracker with timing facilities
  - Construction of a demonstrator integrating sensors, electronics, real-time processors

- **Objectives:**
  - Pixel pitch (55 × 55)μm²
  - Time resolution on single hit < 100 ps
  - Radiation resistance: $10^{16}$ to $10^{17}$ n.eq/cm²

- **Work Packages and coordinators:**
  - 3D silicon sensors: development and characterization – G.F. Dalla Betta
  - 3D diamond sensors: development and characterization – S. Sciortino
  - Design and test of pixel front-end – V. Liberali
  - Design and implementation of fast tracking devices – N. Neri
  - Design and implementation of high speed readout boards – A. Gabrielli
  - System integrations and tests – A. Cardini

Principal investigator: A. Lai
4D stub based tracking

• **Stub approach:**
  - Stubs are identified as *doublets of hits* in adjacent planes
  - A stub provides a “**track hint**” with no assumptions on the particle origin
  - **Geomtrical cuts** are applied to filter stubs not compatible with tracks from the luminous region
  - Stubs used as **track seeds to simplify pattern recognition** and track finding
  - Tracks identified from clusters of stubs with similar parameters

[N. Neri et al., JINST 11 (2016) no.11, C11040 ]

• **Stubs + Timing:**
  - Stub creation is an important step: need to maintain low fake stub rate
  - **Fake stubs can survive the geometrical cuts**
  - **Time** information allows further combinatoric suppression and fake stub rate reduction
  - Particle velocity is required to be compatible with the speed of light
Stub approach

Measured hits from multiple tracks

Candidate stubs from hits belonging to different tracks that don’t point back to the interaction region → rejected

Real stubs that point back to the interaction region → accepted

Candidate stub from hits belonging to different tracks that points back to the interaction region → NOT rejected
Stub approach

Measured hits from multiple tracks

Identified stubs after geometrical filtering. Some fake stubs are still present

Measured candidate stub velocity is required to be compatible with the speed of light.

\[ \frac{v}{c} = \frac{|\vec{x}_2 - \vec{x}_1|}{c(t_2 - t_1)} \approx 1 \]

Identified stubs after geometrical and timing filtering. Timing provides further combinatoric suppression.
• **Stubs are identified**, filtered with geom+time cuts.

• Stubs are **projected to 2D reference plane**

• A grid of “ Engines” is distributed in the reference plane
  - The Engines identify tracks from groups of stubs with similar projections
  - The pattern recognition is based on **only two spatial parameters**
  - The **track parameters** are evaluated by **average of the stubs’ parameters**
Algorithm detail

Coordinates system:

\[(x_f, y_f, z_f), (x_l, y_l, z_l)\]

: intersections of the track with first and last tracking plane

\[x_\pm = (x_f \pm x_l)/2\]
\[y_\pm = (y_f \pm y_l)/2\]
\[z_\pm = (z_f \pm z_l)/2\]

Stub parameters:

\[
\begin{pmatrix}
    x_-
n_s
    x_+
n_s
    y_-
n_s
    y_+
n_s
    t
\end{pmatrix}_{stub} = 
\begin{pmatrix}
    \frac{x_1 z_- - x_2 z_-}{z_1 - z_2} \\
    \frac{x_1 (z_- - z_2)}{z_1 - z_2} - \frac{x_2 (z_- - z_1)}{z_1 - z_2} \\
    \frac{y_1 z_- - y_2 z_-}{z_1 - z_2} \\
    \frac{y_1 (z_- - z_2)}{z_1 - z_2} - \frac{y_2 (z_- - z_1)}{z_1 - z_2} \\
    \frac{t_1 + t_2}{2} - \frac{z_1 - z_2}{2 c \sqrt{1 + (x_-/z_-)^2 + (y_-/z_-)^2}}
\end{pmatrix}
\]
Algorithm detail

- **A grid of engines** is distributed in a reference plane \((x^+,y^+)\) at \(z=z^+\).

  Each engine evaluates a **Gaussian response** according to its distance \((s_{ijk})\) from the stubs projections:

  \[
  W_{ijk} = \begin{cases} 
  \exp \left( -\frac{s_{ijk}^2}{2\sigma^2} \right) & \text{if } s_{ijk} < 2\sigma \\
  0 & \text{otherwise}
  \end{cases}
  \]

- The **total response** is evaluated as:

  \[
  W_{ij} = \frac{1}{N_{ij}} \sum_k W_{ijk}
  \]

- Each **local maximum** identifies a **candidate track**

  - The \((x^+,y^+)\) track parameters are recovered by **interpolation** of the response function

  - \((x^-,y^-,t)\) information not used in the pattern recognition

  - The \((x^-,y^-,t)\) track parameters are obtained by **average of the stub values** belonging to the candidate track

  \[
  x_{-ij} = \frac{1}{N_{ij}} \sum_k x_{-ijk} \\
  y_{-ij} = \frac{1}{N_{ij}} \sum_k y_{-ijk} \\
  t_{ij} = \frac{1}{N_{ij}} \sum_k t_{ijk}
  \]

Typical response of the grid of engines to an event with multiple tracks.
**Simulation of the algorithm**

**Sensor area = 6x6cm²**
- Pixel size = 55x55µm²
- Thickness = 200 µm
- Time res $\sigma_t=30$ ps

**VELO-like tracking device:**
- 12 planes in the forward region
- Pile-up ~40, ~1200 tracks/event

**Lumi. region:** $\sigma_z=5$cm, $\sigma_t=167$ps

**90'000 engines** in [-2,2]x[-2,2]cm² reference plane

*Top: track parameters resolution $(x^+, x^-)$ without using timing information*

<table>
<thead>
<tr>
<th>$\sigma_x$</th>
<th>Entries(10 µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.8 µm</td>
<td>Constant 1 2923</td>
</tr>
<tr>
<td></td>
<td>Mean 1 -0.4355</td>
</tr>
<tr>
<td></td>
<td>Sigma 1 53.09</td>
</tr>
<tr>
<td></td>
<td>Constant 2 1.106e+04</td>
</tr>
<tr>
<td></td>
<td>Mean 2 0.1783</td>
</tr>
<tr>
<td></td>
<td>Sigma 2 26.25</td>
</tr>
</tbody>
</table>

*Bottom: track parameters resolution $(x^+, y^+, t)$ including timing information*

<table>
<thead>
<tr>
<th>$\sigma_x$</th>
<th>Entries(10 µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>35 µm</td>
<td>Constant 1 2708</td>
</tr>
<tr>
<td></td>
<td>Mean 1 -0.9024</td>
</tr>
<tr>
<td></td>
<td>Sigma 1 51.18</td>
</tr>
<tr>
<td></td>
<td>Constant 2 1.242e+04</td>
</tr>
<tr>
<td></td>
<td>Mean 2 0.274</td>
</tr>
<tr>
<td></td>
<td>Sigma 2 24.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma_x$</th>
<th>Entries(10 µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>73 µm</td>
<td>Constant 1 1.042e+04</td>
</tr>
<tr>
<td></td>
<td>Mean 1 -0.00755</td>
</tr>
<tr>
<td></td>
<td>Sigma 1 18.92</td>
</tr>
<tr>
<td></td>
<td>Constant 2 1434</td>
</tr>
<tr>
<td></td>
<td>Mean 2 0.6551</td>
</tr>
<tr>
<td></td>
<td>Sigma 2 108.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma_t$</th>
<th>Entries(8 ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.1 ps</td>
<td>Constant 1 2.286e+04</td>
</tr>
<tr>
<td></td>
<td>Mean 1 -0.4332</td>
</tr>
<tr>
<td></td>
<td>Sigma 1 13.96</td>
</tr>
<tr>
<td></td>
<td>Constant 2 978.8</td>
</tr>
<tr>
<td></td>
<td>Mean 2 6.017</td>
</tr>
<tr>
<td></td>
<td>Sigma 2 69.52</td>
</tr>
</tbody>
</table>
**Simulation of the algorithm**

- The track parameters resolution improves when including the time information
- The reconstruction **efficiency is stable**
- The tracks **purity improves**

- **Inclusion of timing information of the hits allows to evaluate the time of the track**, to be used to better associate tracks to their primary interaction
- **Track mis-association** >10% (no time information)
- **<1% using precise time** information of the hit in offline reconstruction

---

**Graphs:**
- **Purity vs. Hit time resolution**
- **Efficiency vs. Time**
  - Efficiency = 99%
  - Purity = 64%
  - Time: 30ps per hit
  - Efficiency = 98.5%
  - Purity = 82%
- **Vertex time resolution vs. Misassociation**
General architecture

- The device **receives the hits and provides the reconstructed tracks** as output, to be used for further processing.

- **Key points:**
  - **Highly parallelized** architecture
  - **Pipelined** architecture
  - **Hold logic** implementation for **data flow management** with guaranteed minimum latency and **reduced data serialization**
  - All the modules are optimized for low latency: 
    - $<1\mu$s total latency

---

**Stub Constructor**
- Evaluates and filters the combinations of hits in adjacent detectors
- Implemented as full-mesh network, delivers the stub data to the engines
- Organized in “regions”, receive the stubs, identify and reconstruct the tracks
- Collect the track results from the Engines

**Switch**

**Engines**

**Fan In**

**Tracks**

**Hit data**

**Stub data**
• **One dedicated Stub Constructor for each plane doublet:**
  - Implemented in dedicated or front-end FPGA
  - The Stub Constructor modules are completely independent from each others

• **Engines** in the 2D tracking reference plane are organized into multiple independent areas.
  - Each Area comprises a dedicated Stub Switch and a pool of Engines.
  - Absence of “lateral” communication between different Areas makes the system modular and scalable
Stub Constructor

- One module for each couple of sensors
- **Sensors are divided in exclusive regions.** Only compatible regions on first and second sensor are processed by dedicated **Stub Makers**.
- **Sensor regions** are optimized to have uniformly populated **Stub Makers** and reduce the total latency.
- Hit data are delivered to the **Stub Makers** by dedicated **Hit Switches** based on the hit coordinates \((r,\phi)\) and pre-computed patterns.

Each **Stub Maker** evaluates the combination of hits from the first and second sensors:

- Incoming hit data are stored in **two separate buffers**
- When and “End Of Event” signal is received at both inputs, the hit combinations are evaluated with a double loop on the buffer positions.
- Each hit combination forms a stub candidate.
- Geometrical and timing cuts are applied to suppress fake stubs
Switch

- **Full-mesh network** of Sorters (2x2 in the example) arranged in multiple layers.
- The **Sorter distributes the incoming Stub data** to one or multiple outputs based on the radial and azimuthal coordinates of the stub projection to the reference plane.
- An **hold logic is implemented to pause the data flow** when the data can not be accepted by the following items in the distribution chain.
The **Engine** is associated to **9 cells in the reference plane** (1 central + 8 lateral).

- It receives the **stub data with 5 parameters** (4 spatial + 1 timing): the coordinates of the stub projection in the ref.plane are **used to evaluate in which cell the stub lies**. A counter associated to the identified cell is increased.

- The **5 stub parameters are separately accumulated** in order to evaluate the candidate track parameters.

- The **process is repeated** for all the incoming stubs until an “End Of Event” signal.

- **If the central cell is a local maximum** (compared to the lateral) a track candidate is identified: the accumulated stub parameters and the total counter are provided as output.
• The 4D fast tracking algorithm is being **implemented in FPGA** on a custom board.

• gFEX board, developed at BNL for ATLAS calorimeter:
  - Two **Xilinx Virtex Ultrascale** FPGAs
  - High-speed optical transceivers → ~**1.6 Tbps input data rate**
  - One Xilinx Zynq FPGA

<table>
<thead>
<tr>
<th>System Logic Cells (K)</th>
<th>1,176</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP Slices</td>
<td>768</td>
</tr>
<tr>
<td>Memory (Mb)</td>
<td>60.8</td>
</tr>
<tr>
<td>GTH 16.3 Gb/s Transceivers</td>
<td>32</td>
</tr>
<tr>
<td>GTY 30.5 Gb/s Transceivers</td>
<td>32</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>832</td>
</tr>
</tbody>
</table>

**“Processor B”:**
Xilinx Virtex UltraScale
xcvu095-ffvc2104-3-e

8 x 12.8Gpbs TX
64 x 12.8Gpbs RX
4 x 12.8Gpbs TX
64 x 12.8Gpbs RX

**“Processor A”:**
Xilinx Virtex UltraScale
xcvu095-ffvc2104-3-e

8 x 12.8Gpbs TX
64 x 12.8Gpbs RX
4 x 12.8Gpbs TX
64 x 12.8Gpbs RX

• **Data communication** using high speed serial links is implemented

• **Stub Constructor** partially implemented (see next slides)

• **Switch + Engines + Fan Ins** fully implemented
Communication test

- **Bit Error Ratio** tests successful with GTY/GTH links speed of 12.8Gbps
- **32b/40b** data communication between FPGAs implemented and tested
- **64b/66b** data communication between FPGAs implemented and tested

Setup of the prototype board with optical fibers at INFN-Milano

Example of eye diagram performed on one optical link
Simulated sector of VELO-like det.

- The main space of track parameters (radial and azimuthal coords of stub projection) has been divided in **8x8 (r,phi)-sectors**.
  - Already filtered stubs, evaluated in software simulation, have been used to perform the following test

- The system is modular and scalable thanks to the absence of “lateral communication”
  - Results from one sector test can be extended to the full system
Switch + Engines implementation

- Implemented firmware corresponds to the red rectangle, w.r.t. general scheme:
  - A 64x64 Stub Switch
  - 64 Engine Regions with 16 Engines within each region: total of 1024 engines
  - 16 Fan Ins collecting the output track data

- 1024 Engines cover 1/64 of the full space of track parameters

- Implemented in a single Xilinx Virtex UltraScale FPGA (2 available on the gFEX board)

- Resources utilization about 50% of one FPGA
Switch + Engines test

- Architecture test, implemented in the same FPGA.
- System clock: 320 MHz
- 3 Blocks:
  - Data generator
  - Device Under Test (Switch+Engines+Fanins)
  - Data checker
- Data Generator provides stub data stored in a ROM, filled with data associated to 398 simulated events in a 1/64 sector of the whole detector
- Data are continuously provided to the DUT; the output is provided to the Data Checker
- The Data Checker compares the received track data with the values stored in a ROM, populated with the expected results
Switch + Engines test (2)

- Data processed in hardware are compatible with data stored in the Output ROM, evaluated from behavioral simulation.

- **Rate test:**
  - The Input Data ROM has a depth of 1024 and contains data from 398 events.
  - The ROM is read at 320 MHz ref. clock, then can provide a max. event rate of 124.375 MHz.
  - If the DUT provides the hold signal the reading is paused until the hold is released, reducing the rate of events provided to the DUT.
  - At the maximum reading rate of 124.375 MHz the DUT can accept ~1/3 of the data, resulting in a processing throughput of 40.9 MHz.
  - (With a system clock of 400 MHz this value increases up to 51.1 MHz → not yet tested)
Conferences


- M.Petruzzo: “Gains from timing”, March 22, 2018, Talk at 3rd Workshop on LHCb Upgrade II

  A novel 4D fast track finding system using precise space and time information of the hit,
  Poster contribution, M. Citterio, P. Gandini, J. Fu, N. Neri, M. Petruzzo, S. Riboldi

- VCI 2019, 18-22 Feb 2019, Vienna, Austria
  A novel 4D fast track finding system on FPGA,
  Poster contribution, M. Citterio, P. Gandini, J. Fu, N. Neri, M. Petruzzo, S. Riboldi
Conclusions

- **An new device for fast track finding** has been studied for application to a **VELO-like detector**
  - can be applied to other tracking detectors

- **The system has been implemented and tested in FPGA:**
  - Stub identification and construction strategy tested in high-level simulation
  - Switch + Engine tested in low-level simulation and hardware

- **Simplified test performed to validate the correct behavior of the system:**
  - Loop on a subset of events and bit-to-bit comparison between results from simulation and results from hardware processed data
  - 1/64 of VELO-like detector processed in one FPGA
  - 40 MHz throughput reached

- **Plans for the future:**
  - Systematic tests of the fast-tracking device with simulated data generated by external FPGAs
  - Optimize the system and the device architecture
Simulation of the algorithm

Track efficiency vs track parameters $r_+, \phi, d_0, z_0, \eta$:

- Track efficiency $\sim 98\%$
- Track purity $>80\%$ with 1200 tracks per event
- Track purity $\sim 60\%$ (without timing)
Optimization for stub construction

- Sensors are divided in exclusive in \((r,\phi)\) regions to have uniformly populated bins.

- Correlation between regions in first and second sensor (of each couple) is evaluated from simulation → not correlated regions are not considered in the stub search.

- A Stub Maker is instantiated for each possible combination of regions.

Distribution of the radial coordinate of the hits in the sensors of a VELO-like detector.

Not uniform, as expected

Angular distribution is uniform, as expected (not shown here)
The tracking ref. plane is divided in (r, phi) regions in order to obtain uniformly populated regions.

- Distribution w.r.t. phi is flat as expected.
- Distribution w.r.t. radius is not flat.
- Radius is normalized and Engines are distributed in [f(r), phi] space.
Occupancy of the Stub Makers

- **Dots**: distribution of number of candidate stubs to process in each Stub Maker
  - the value is proportional to the latency required for producing the hit combinations
- **Dashed**: distribution of number of filtered stubs provided by each Stub Maker

- The latency budget produced by the Stub Constructor can be reduced, if needed, by incrementing the number of Stub Makers

---

March 29, 2019

Real Time Analysis General Meeting

31
Development of Stub Constructor

- **Status of Stub Constructor** implementation:
  - **Hit Switch: already implemented**
    The Hit Switch is equivalent to the Stub Switch, already tested
  - **Stub Maker: implemented the combinatorial processing of hits**, behaviorally tested.
  - Need to implement the stub formatting to generate the stub from a combination of hits
  - Need to implement the filtering of stubs based on geometrical and timing cuts