A SEED FILTER + LINEAR REGRESSION FITTER FOR THE TIME MULTIPLEXED TRACK TRIGGER

Davide Cieri* on behalf of the TMTT collaboration

*Supported by the EU FP7-PEOPLE-2012-ITN project nr. 317446, INFIERI, "Intelligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Reasearch and Industry"

davide.cieri@stfc.ac.uk

IV INFIERI Summer School

USP, Brazil • 23 Jan - 3 Feb 2017

The Time Multiplexed Track Trigger

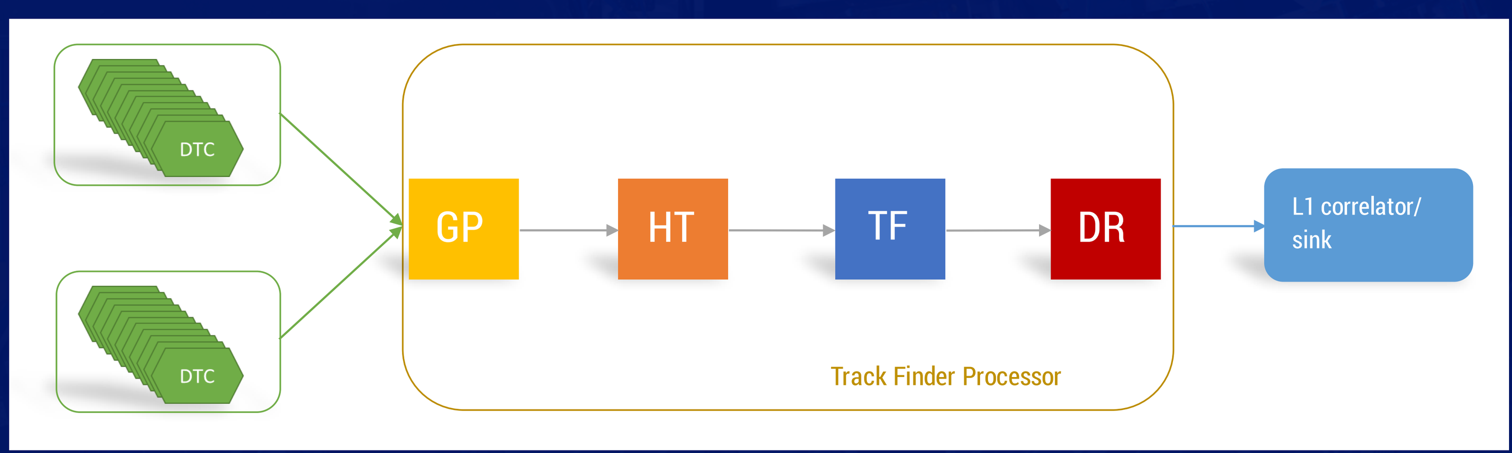
The **Time Multiplexed Track Trigger (TMTT)** is one of the main proposal for the architecture of the future L1 Track Trigger system for the **Phase II upgrade** of the Compact Muon Solenoid (**CMS**) experiment.

Goal of the Track Trigger is to **find track candidates** from a series of hits (stubs) in the outer tracker detector.

In the TMTT approach the L1 Track Finding task is achieved by means of **FPGA circuits**. To handle with the high bandwitdh coming out from the tracker detector, this has been divided in 8 segments (**octants**) along the azimuthal angle Φ. Data from each octant are analysed by a single **L1 Track Finder Processor (TFP)**, where the L1 Track Finding algorithm has been implemented.

The algorithm has been divided in **several blocks**, each of them running on a separate FPGA for the purpose of the hardware demonstrator. Tracker data are first organised and converted in the **Geometric Processor (GP)**, then candidate tracks in the RΦ are found using the **Hough Transform (HT)** technique . Downstream HT candidates are filtered and fitted in the **Track Fitter processor (TF)**. Finally a **duplicate removal (DR)** step get rids of the surviving duplicate tracks.

For more information about the TMTT demonstrator see L. Calligaris' poster.



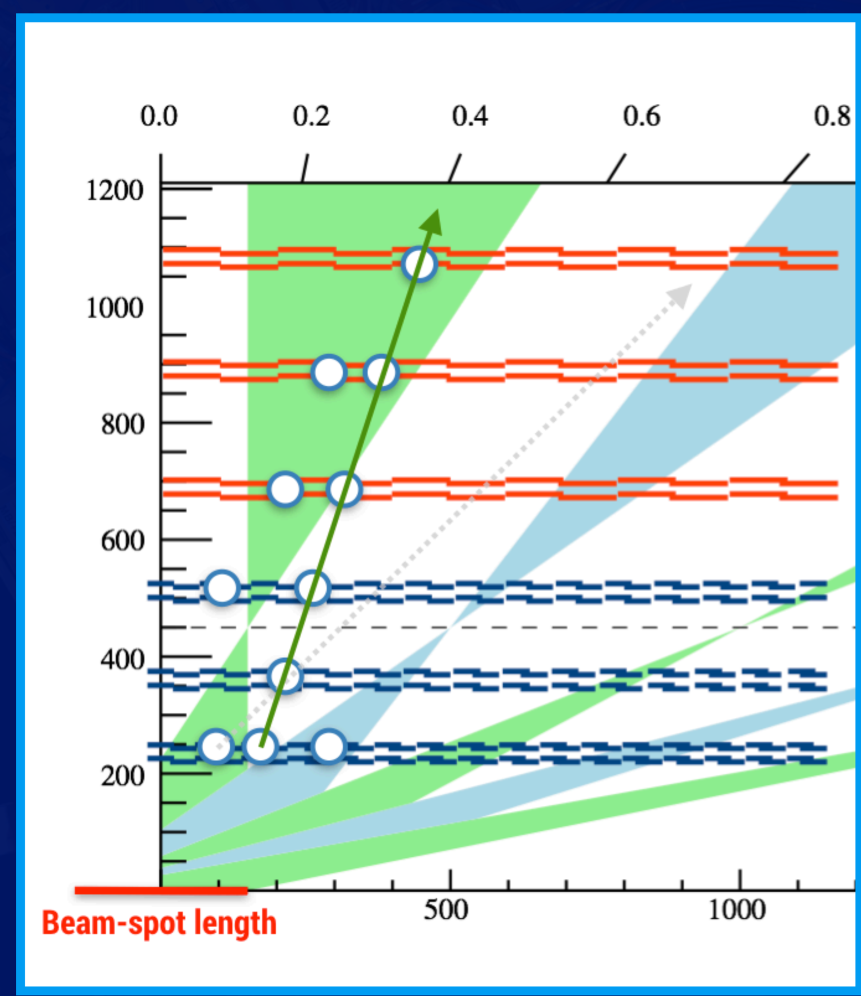
The standard configuration of the TMTT demonstrator adopts the **Kalman filter** technique to perform the track fitting stage. An alternative approach implements a filter stage (**Seed Filter**) to reject tracks not compatible with a straight line in the RZ plane, before doing the fitting using the **Linear Regression (LR)** method.

	Hough transform	R-Z seed filter	LR track fit
Total number tracks/event	252	174	142
Tracking efficiency	96.0%	94.5%	92.6%
Fraction of tracks that are genuine	72.6%	92.6%	92.0%
Fraction of genuine tracks that are unwanted duplicates	47.0%	56.7%	48.9%
Fraction of genuine tracks where all stubs are correct	44.2%	87.8%	100.0%

The Seed Filter Algorithm

The Seed Filter identifies and eliminates stubs assigned to tracks by the HT that lie tens of centimetres from the track in the RZ plane

1. Collects pairs of stubs, which belong to different **PS layers (blue)**
2. Computes lines passing through those stubs **(seeds)**
3. **Discards** seeds that would correspond to tracks out of the **beam-spot** and **sector definition**
4. **Extrapolates** surviving seeds to other tracker layers, keeping only **one stub per layer**
5. Only tracks that still contain **enough** stubs in different **layers** are kept



The Linear Regression Fitter

Tracks with sufficient pT should draw a **straight line on both RΦ and RZ** planes. The Linear Regression Fitter fits the helix parameters using independent straight line fit in the two planes with Linear Regression technique.

1. Calculates the **helix parameters** on the **RΦ** plane
2. **Remove** from the track the stubs with the **largest residual**, keeping at least **two** stubs in the **PS layers**
3. Calculates again the **helix parameters**, also in the **RZ** plane, using only stubs in the PS layers
4. **Computes the χ2** and rejects tracks with a χ2 larger than a predefined cut

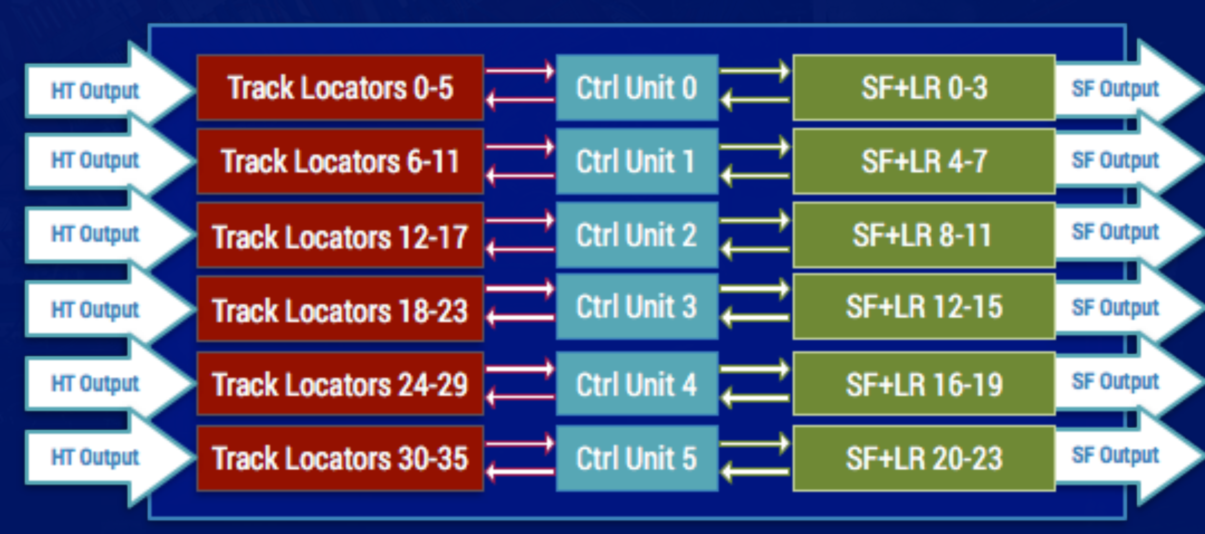
$$m_{R\phi} = q/p_T = \frac{n\overline{R\phi} - \bar{R}\bar{\phi}}{n\overline{R^2} - \bar{R}^2} \qquad m_{RZ} = \tan\lambda = \frac{n\overline{RZ} - \bar{R}\bar{Z}}{n\overline{R^2} - \bar{R}^2}$$

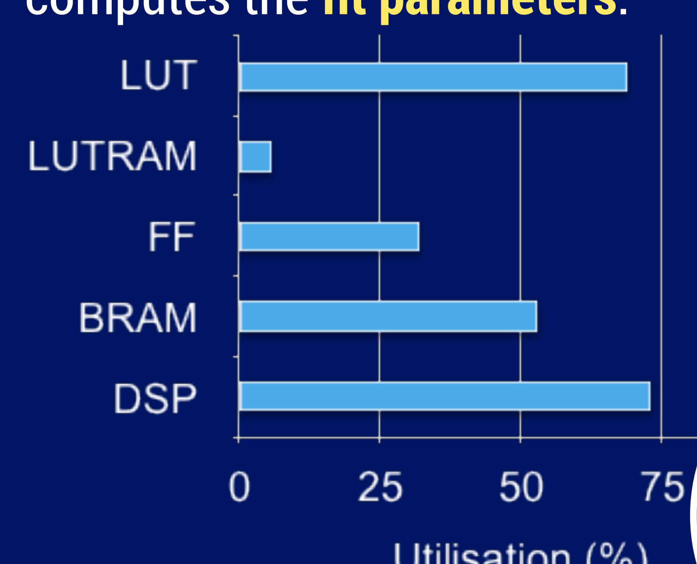
$$c_{R\phi} = \phi_0 = \frac{\overline{R^2}\bar{\phi} - \bar{R}\overline{R\phi}}{n\overline{R^2} - \bar{R}^2} \qquad c_{RZ} = Z_0 = \frac{\overline{R^2}\bar{Z} - \bar{R}\overline{RZ}}{n\overline{R^2} - \bar{R}^2}$$

Firmware Implementation

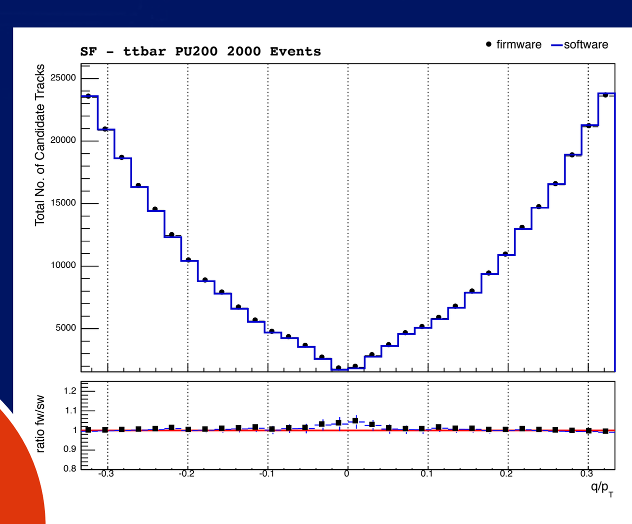
The algorithm has been implemented in a **Virtex 7 FPGA**, mounted on a **μTCA MP7** board. Stubs from each HT output channel are stored in a **separate FIFO (Track Locator)**. The **Control Unit (CU)** reads out stubs, belonging to the same track candidate, from the Track Locators and send them to the first available **Seed Filter + Linear Regression (SF+LR)** module. Stubs from the next candidate are sent to the next free module and so on. Each CU handles 6 Track Locator and 4 SF+LR block.

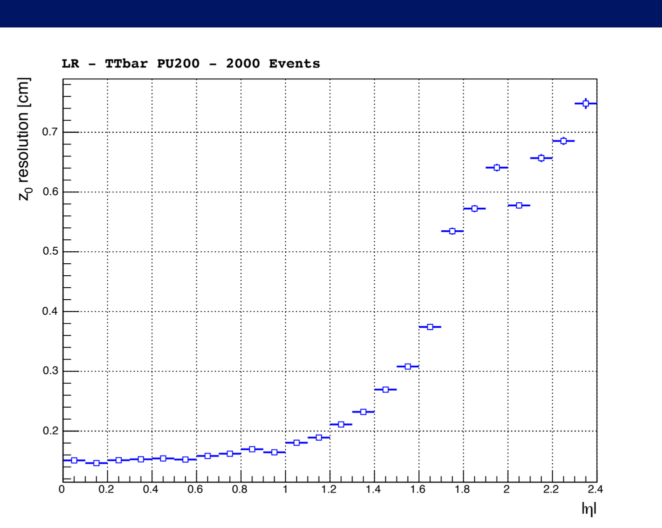
The filter algorithm is implemented in the **SF block**, which processes a **HT candidate at the time**. If the candidate passes the filter, the **surviving stubs** are sent to the **downstream LR module**, where eventually a **pipelined chain** computes the **fit parameters**.

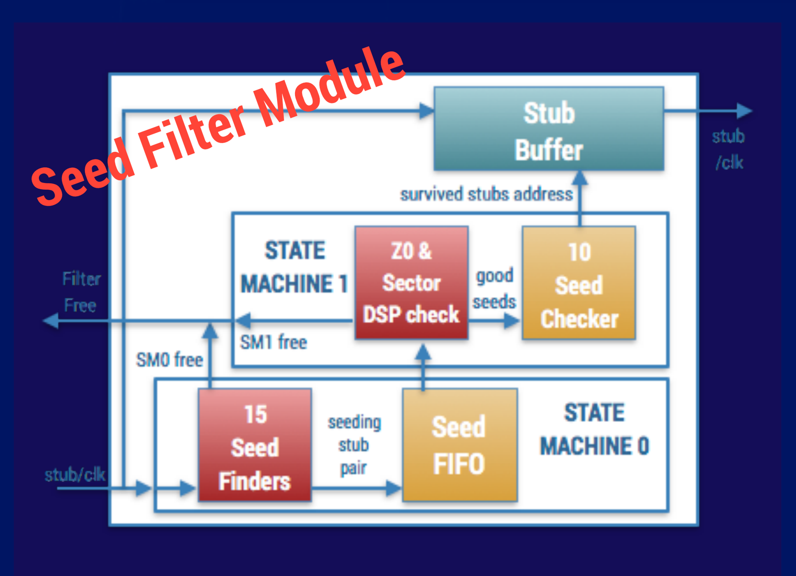


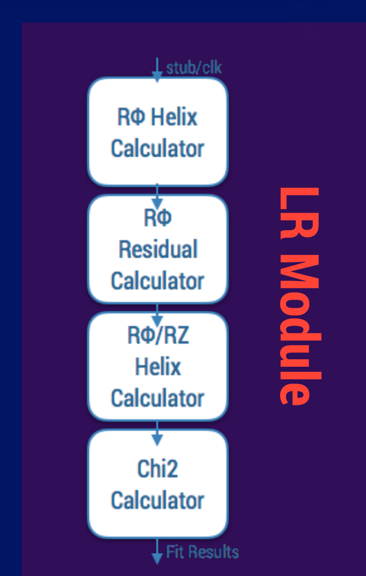


Latency ~635 ns @ 240 MHz









The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA agreeement n. [317446] INFIERI "Intelligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Research and Industry"