

# 3D tracker firmware

Jae-Bak Kim\*, Eunil Won

Elementary Particle Physics Laboratory  
Korea University

2016 September 5th

2016 Trigger/Daq Workshop

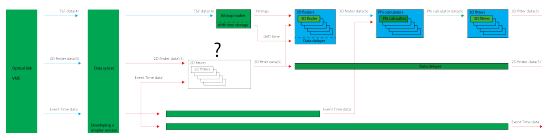
# Contents

- Review of last B2GM
- New Finder3D
- Tracker3D firmware

## Slide from last B2GM

## 3D tracker board firmware status

## Status of the 3D tracker board



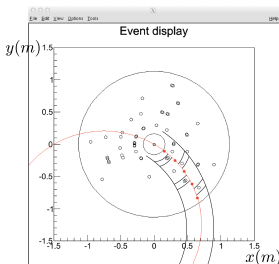
Green: Finished, Red: Not ready, White: Unknown, Blue: Goals

- 2nd Goals
  - Connect the red lines.
  - Add 2D fitter to firmware.
- Difficulties
  - Debugging will be difficult because firmware TSIM is not developed enough. (No 2D finder, ETF)

# Limits of previous Finder3D

- Was developed at 2013.03.04 (3 years ago)
  - At that time, handling CDC's TDC information was not considered.
- Modifying previous code is very difficult.
  - The VHDL code and C++ code were developed independently.
    - Very stiff coding style
    - Many parameters to optimize for firmware.
  - There were separate scripts to generate LUT (look-up-table) data for firmware.

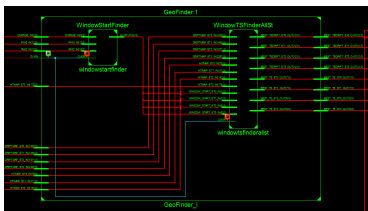
# New Finder3D



Example of the Finder3D

- Uses C++ classes used in Fitter3D to generate part of the Finder3D VHDL code.
  - This makes modifying logic much easier.
- Handles CDC's TDC information.

# Components



RTL of Finder3D

- WindowStartFinder: Calculates the starting position of the window.  
(=  $\phi_{\text{start}}$ )
  - Used the C++ classes used in Fitter3D.
- WindowTSFinder: Selects a TS in the TS window. Get's the TDC information for the selected TS.
  - Developed C++ and VHDL independently. No LUTs. Very few parameters to modify.

# Resources

- Finder3D resource estimate for one track.

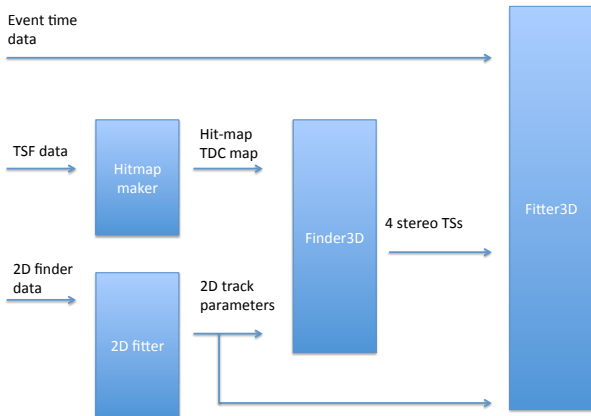
Device Utilization Summary (estimated values)				[ - ]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	8789	708480		1%
Number of Slice LUTs	27637	354240		7%
Number of fully used LUT-FF pairs	8433	27993		30%
Number of bonded IOBs	7289	720		1012%
Number of BUFG/BUFGCTRLs	1	32		3%

# Introduction to modules in Tracker3D

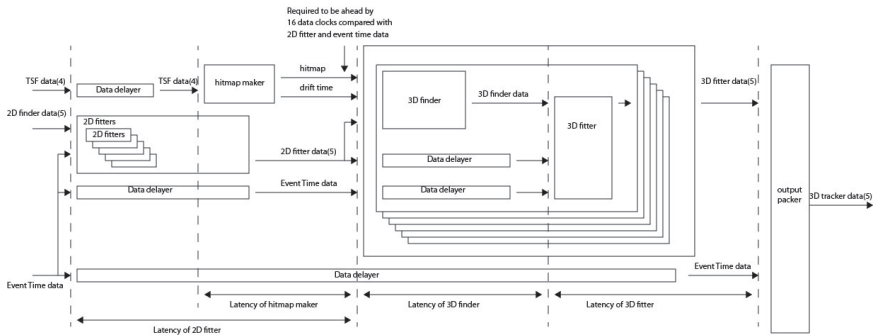
- Tracker3D has 4 main components
  - Fitter2D: Tsu-An is developing this module.
  - Finder3D: Selects stereo TSs to fit.
  - Fitter3D: Fits the track to find  $z_0$  and  $\lambda$  track parameters.
  - Hit-map maker: Makes TSF data suitable for Finder3D.
- Tracker3D will process 6 tracks for every 32 ns.



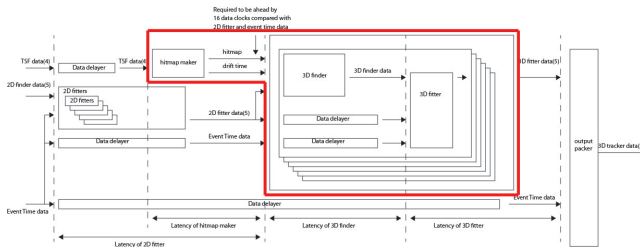
# Connections between modules



# Timing of modules



# Resource problems



Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	93782	708480	13%	
Number of Slice LUTs	253258	354240	71%	
Number of fully used LUT-FF pairs	74375	272665	27%	
Number of bonded IOBs	2460	720	341%	
Number of Block RAM/FIFO	147	912	16%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	336	864	38%	

- Try attaching I/O and check if there is a timing constraint error.

# Testing problem

- To test the firmware, a input and expected output data set is required.
  - TSIM firmware for EventTimeFinder and Finder2D is being developed.
  - Very difficult to test firmware with no TSIM.
  - Waiting for upstream to be developed in TSIM.
- To test the firmware, it is needed to send data to UT3 board through optical links.
  - Currently the only setup is in E-Hut.
  - Finder2D is using the setup.
  - Next EventTimeFinder will use the setup.
  - Waiting for upstream to debug their firmware.

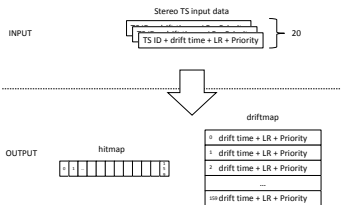
# Summary

- A new Finder3D was developed.
  - Easier to modify
- Combined most of the logics for Tracker3D.
  - Hitmap maker, 6 × 3D finders, 6 × 3D fitters.
- Resources could be a problem with current firmware.

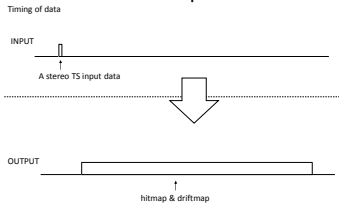
# Backup

# Introduction to Hit-map maker

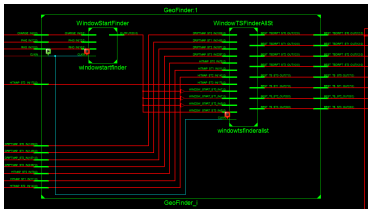
## About hitmapMaker



## About hitmapMaker



# Components



- WindowStartFinder: Calculates the starting position of the window.  
(=  $\phi_{\text{start}}$ )
  - $\phi_{\text{start}} = \pm \arccos\left(\frac{r}{2R}\right) + \phi_{\text{cc}}$
  - $r$  is radius of CDC SuperLayer,  $R$  is radius of track,  $\phi_{\text{cc}}$  is phi of circle's center.
  - Used the C++ classes used in Fitter3D.
- WindowTSPFinder: Selects a TS in the TS window. Get's the TDC information for the selected TS.
  - Developed C++ and VHDL independently. No LUTs. Very few parameters to modify.