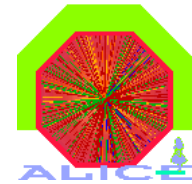


Update on the SPD Offline



Domenico Elia

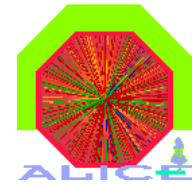
in collaboration with H. Tydesjo, A. Mastroserio

Overview:

- Simulation (strobe window, dead and noisy pixels)
- Reconstruction (dead and noisy pixels, trackleter)
- FastOr trigger (condition data, simulation)
- Online calibration and DAs
- Geometry (→ Mario's talk)



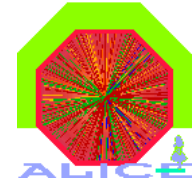
SPD response simulation



- Handling of the time info in Hits->Digits:
 - needed for studying pileup (from different bunch crossings)
 - it basically accounts for SPD strobe (100 nsec window and phase wrt trigger) in the selections of the Hits
 - implemented and tested (Cvetan, me)
 - committed on the trunk (ITS task #2586), release v4-16?

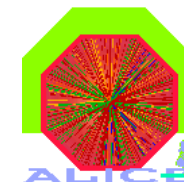
- Further (ongoing) implementations:
 - FastOr simulation connected (see next slides)
 - RemoveDeadPixels(): set via AliITSSimuParam, default=OFF
 - AddNoisyPixels(): new, set via AliITSSimuParam, default=ON

Local reconstruction



- ❑ **Ongoing implementations in ClusterFinderV2SPD:**
 - FastOr simulation connected (see next slides)
 - `GetSPDRemoveDeadFlag()`, via `AliITSRecoParam`, default=ON
 - this feature is **new**
 - generally safer to always remove bad pixels (so ON), but it can be set OFF in case of real data to speed up
 - `GetSPDRemoveNoisyFlag()`, via `AliITSRecoParam`, default=ON
 - as since time, noisy pixels always killed before clustering

Trackleter: $\Delta\phi$ distribution



Savannah bug #46278:
referred to as
“Bug/artifact in the trackleter”

Jan Fiete

Observed “features”:

- smaller dip at $\Delta\phi \approx 0$
- depletion at $\Delta\phi \approx \pm 0.01$

Since when this is there:

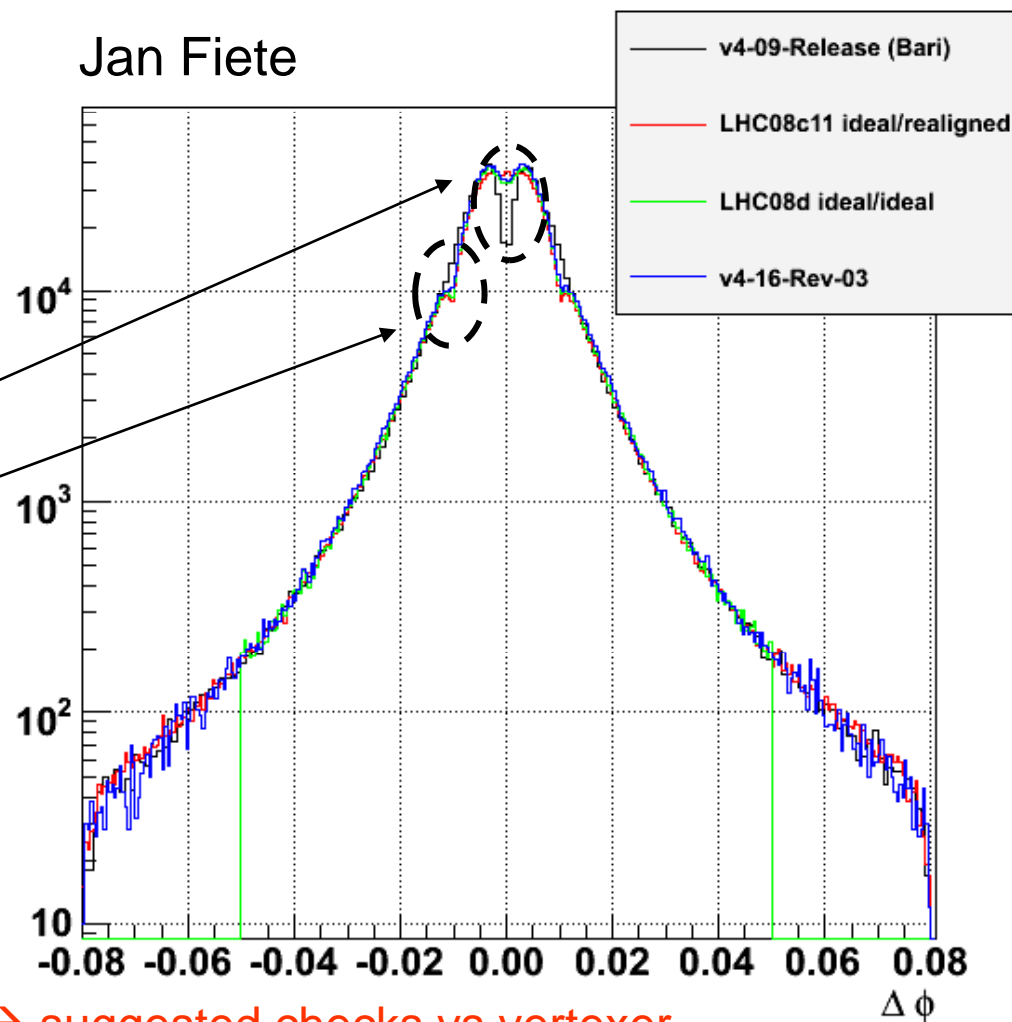
- for p-p:

AliRoot > v4-09-Release

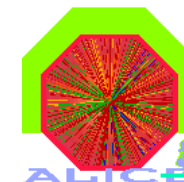
- for Pb-Pb:

fine, features not observed

(checked with v4-16-Rev-03) → suggested checks vs vertexer

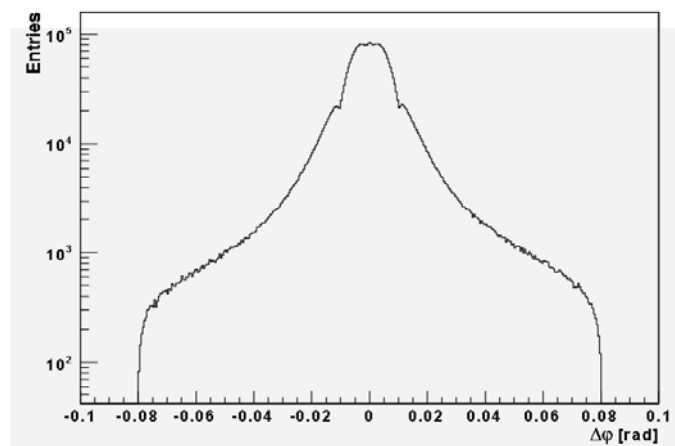


Checks vs vertexer (I)

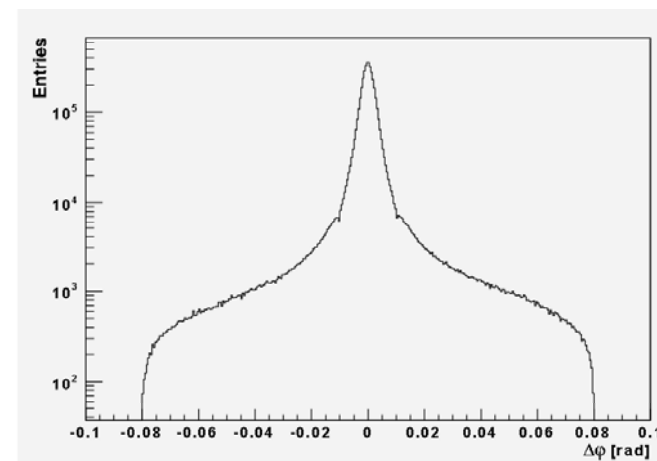


CAF data: vertex generated with usual spread in x,y

All events

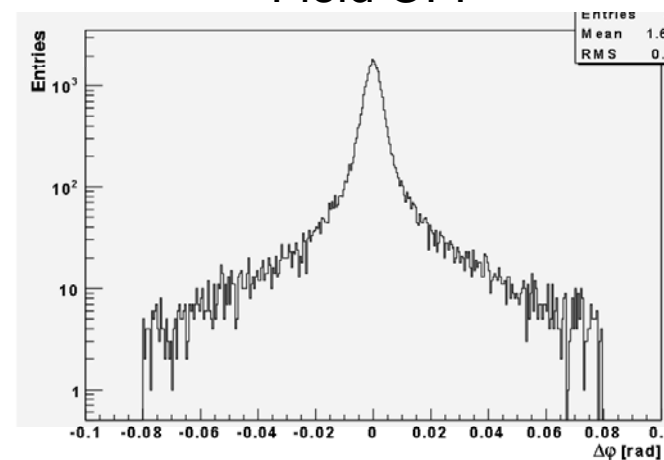
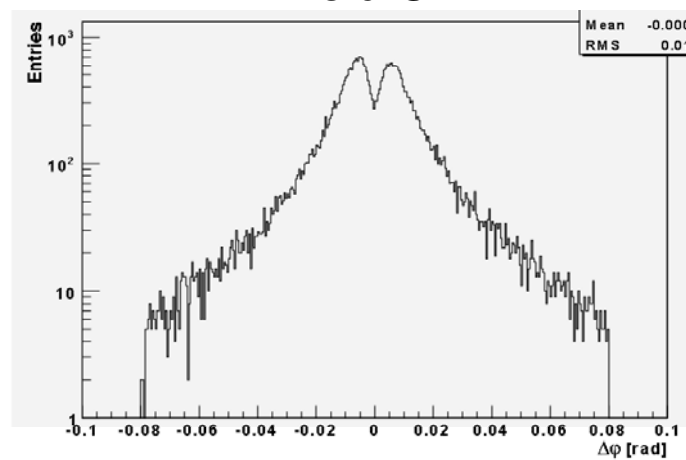


Field ON

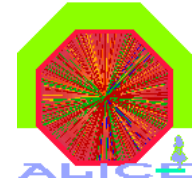


Field OFF

Events with vtx from vertexerZ

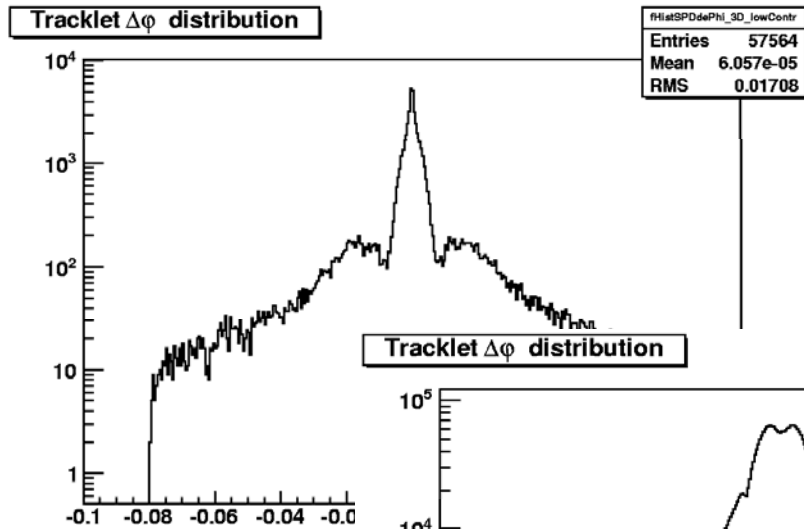


Checks vs vertexer (I)

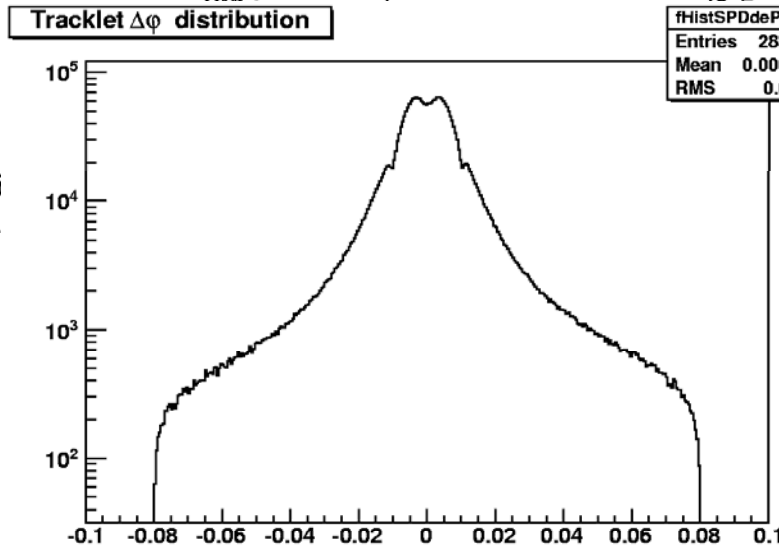
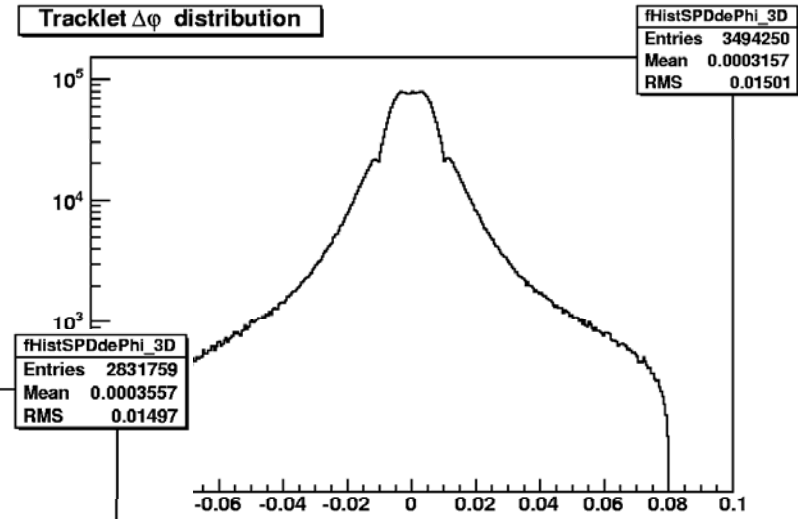


CAF data: vertex generated with usual spread in x,y

vtx3D with Ncontrib = 2



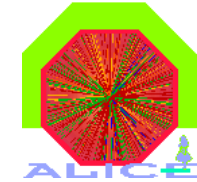
vtx3D with Ncontrib > 2



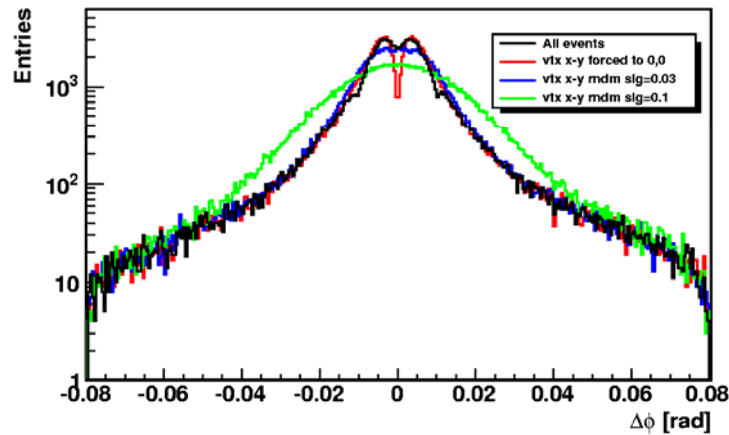
vtx3D with Ncontrib > 10



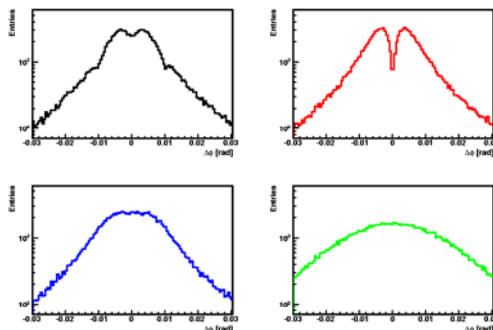
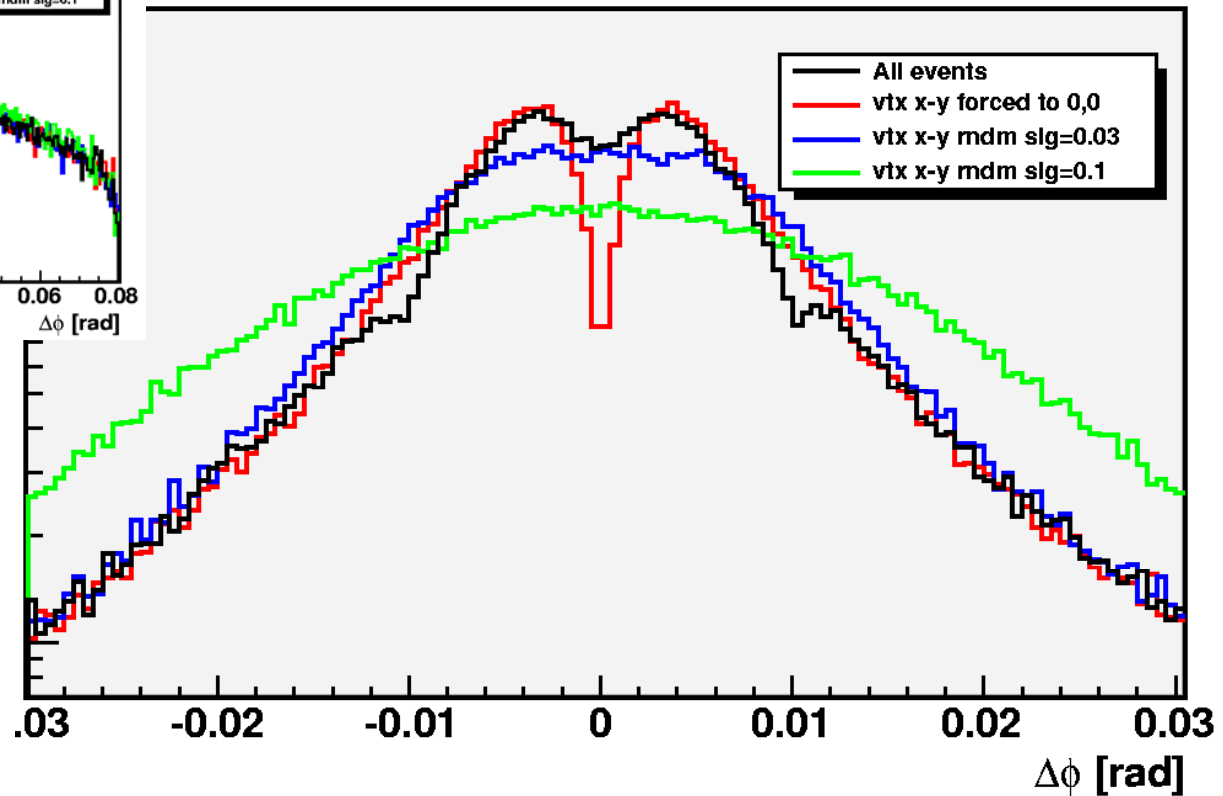
Checks vs vertexer (II)



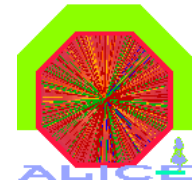
Test production: 10K p-p events, v4-16-Rev-03
vertex generated in $x,y = (0,0)$



→ further investigation in vtxer3D required



FastOr trigger issues



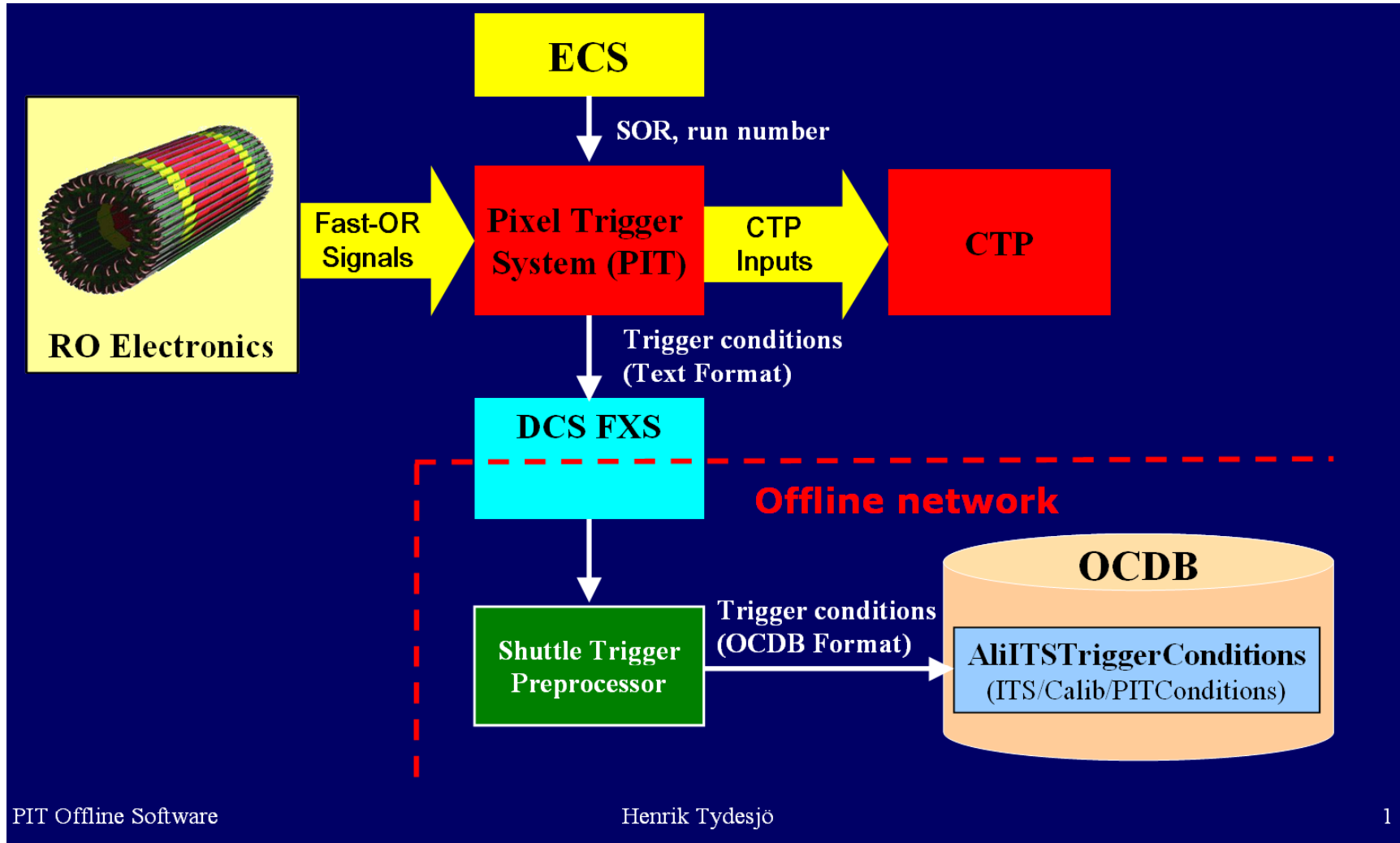
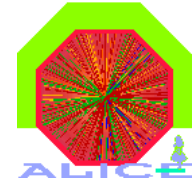
❑ Main offline tasks related to FastOr:

- storing the pixel trigger condition in OCDB for each run
- simulating the generation of the FastOr signals from SPD
- simulating the response of the PIT system to the FastOr signals

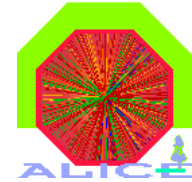
❑ New/Modified classes to implement the procedure:

- schematic picture in the following slides, main benefits:
 - simulated FastOr signals saved in ITS.Digits.root
 - simulated FastOr signals propagated to sim rawdata file as real data
 - FastOr signals from active chips saved to ESD
- code available at http://tydes.home.cern.ch/tydes/ITS_NEW.tgz
- **further tests required before commit!**

FastOr conditions transfer scheme



FastOr based algorithms



1	0SMB	Minimum Bias	$(I+O) \geq th_{IO,mb}$ and $I \geq th_{I,mb}$ and $O \geq th_{O,mb}$
2	0SH1	High Multiplicity 1	$I \geq th_{I,hm1}$ and $O \geq th_{O,hm1}$
3	0SH2	High Multiplicity 2	$I \geq th_{I,hm2}$ and $O \geq th_{O,hm2}$
4	0SH3	High Multiplicity 3	$I \geq th_{I,hm3}$ and $O \geq th_{O,hm3}$
5	0SH4	High Multiplicity 4	$I \geq th_{I,hm4}$ and $O \geq th_{O,hm4}$
6	0SPF	Past Future Prot	$(I+O) \geq th_{IO,pfp}$ and $I \geq th_{I,pfp}$ and $O \geq th_{O,pfp}$
7	0SBK	Background(0)	$I \geq O + offset_I$
8	0SX1	Background(1)	$O \geq I + offset_O$
9	0SX2	Background(2)	$(I+O) \geq th_{(I+O),bnd}$
10	0SCO	Cosmic	<i>Selectable coincidence</i>

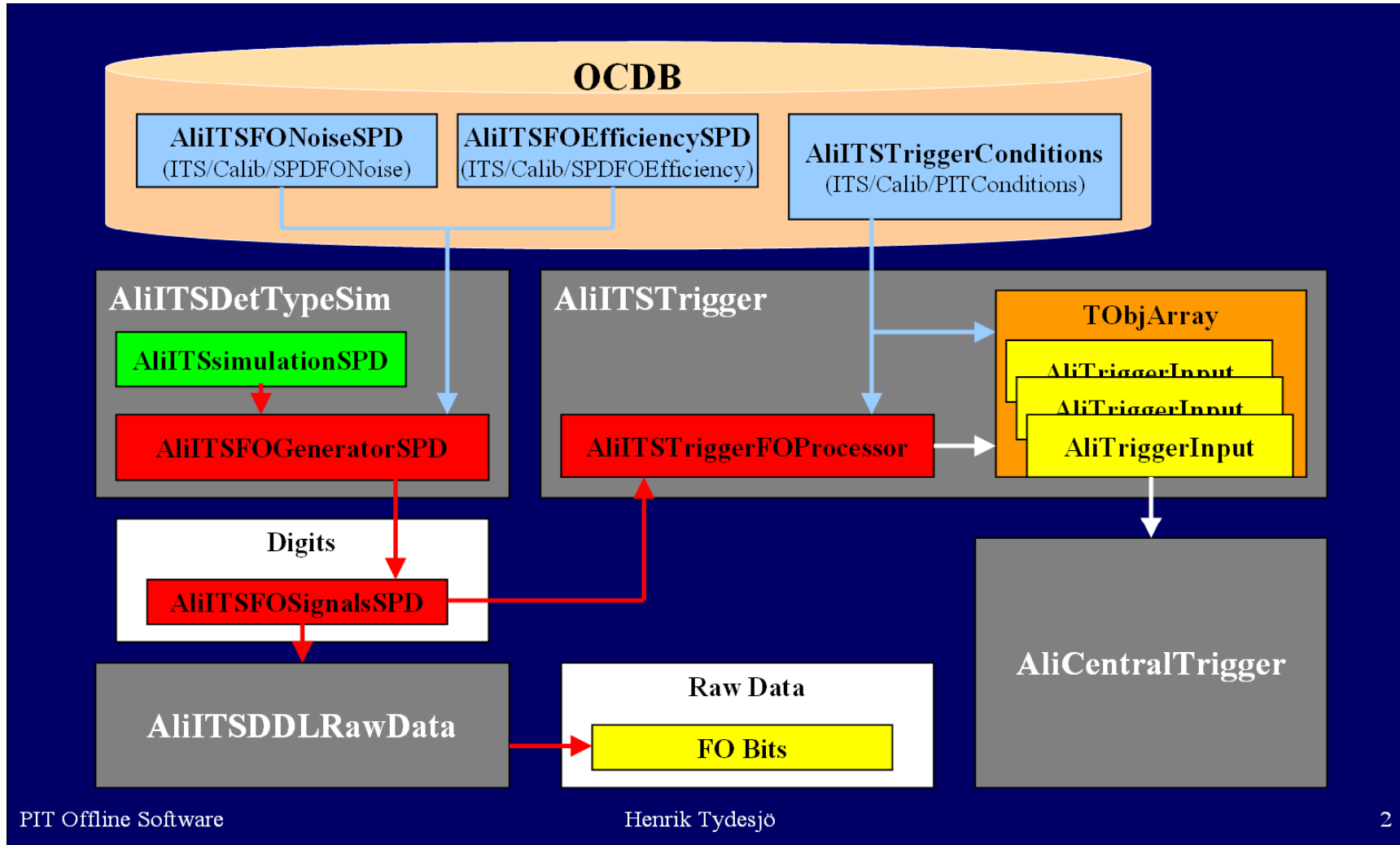
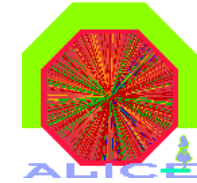


**Implement algorithms
for each label in
AliITSTriggerFOProcessor**

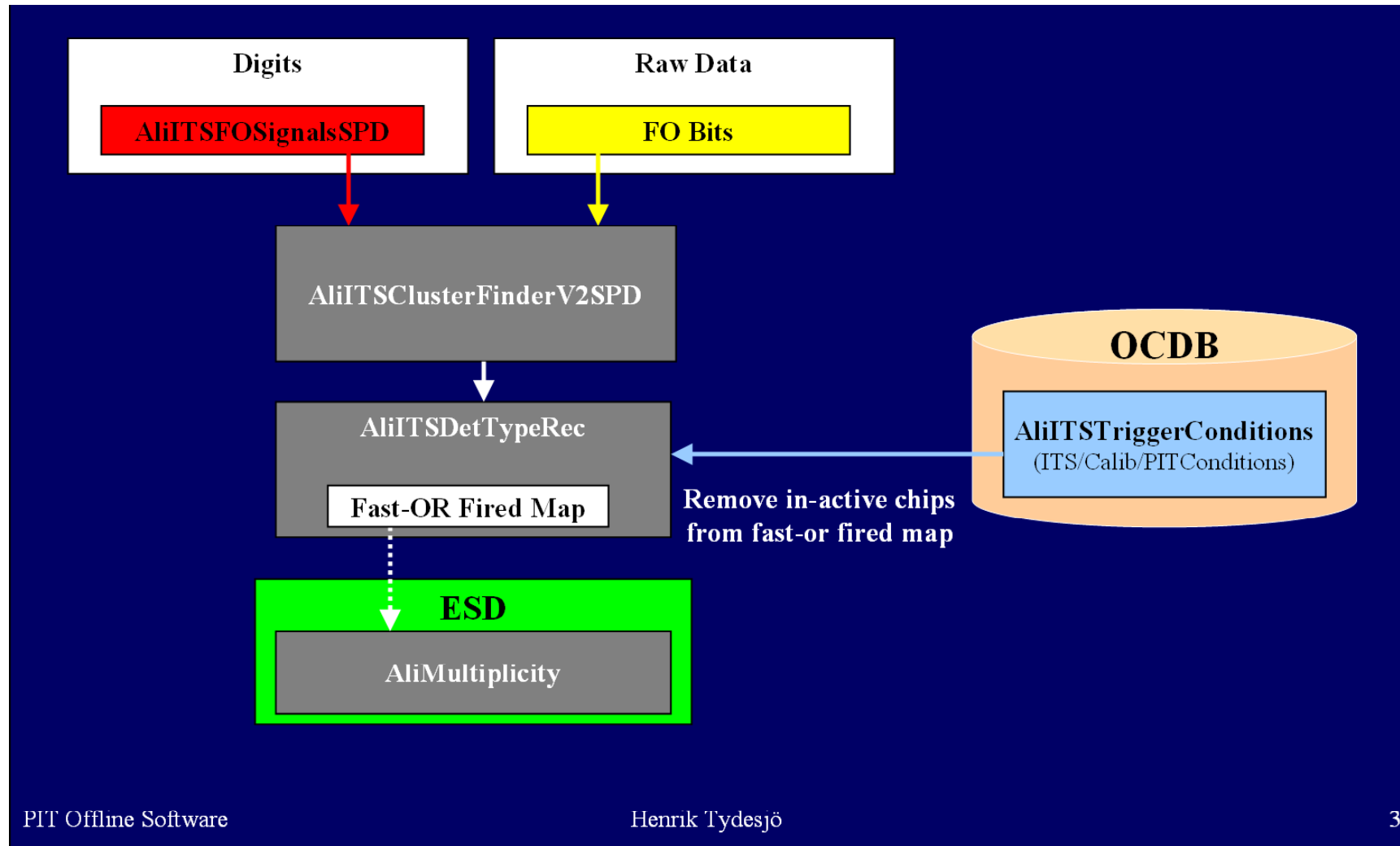
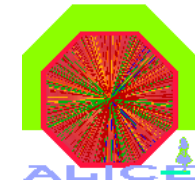
I := number of active FO on Inner layer

O := number of active FO on Outer layer

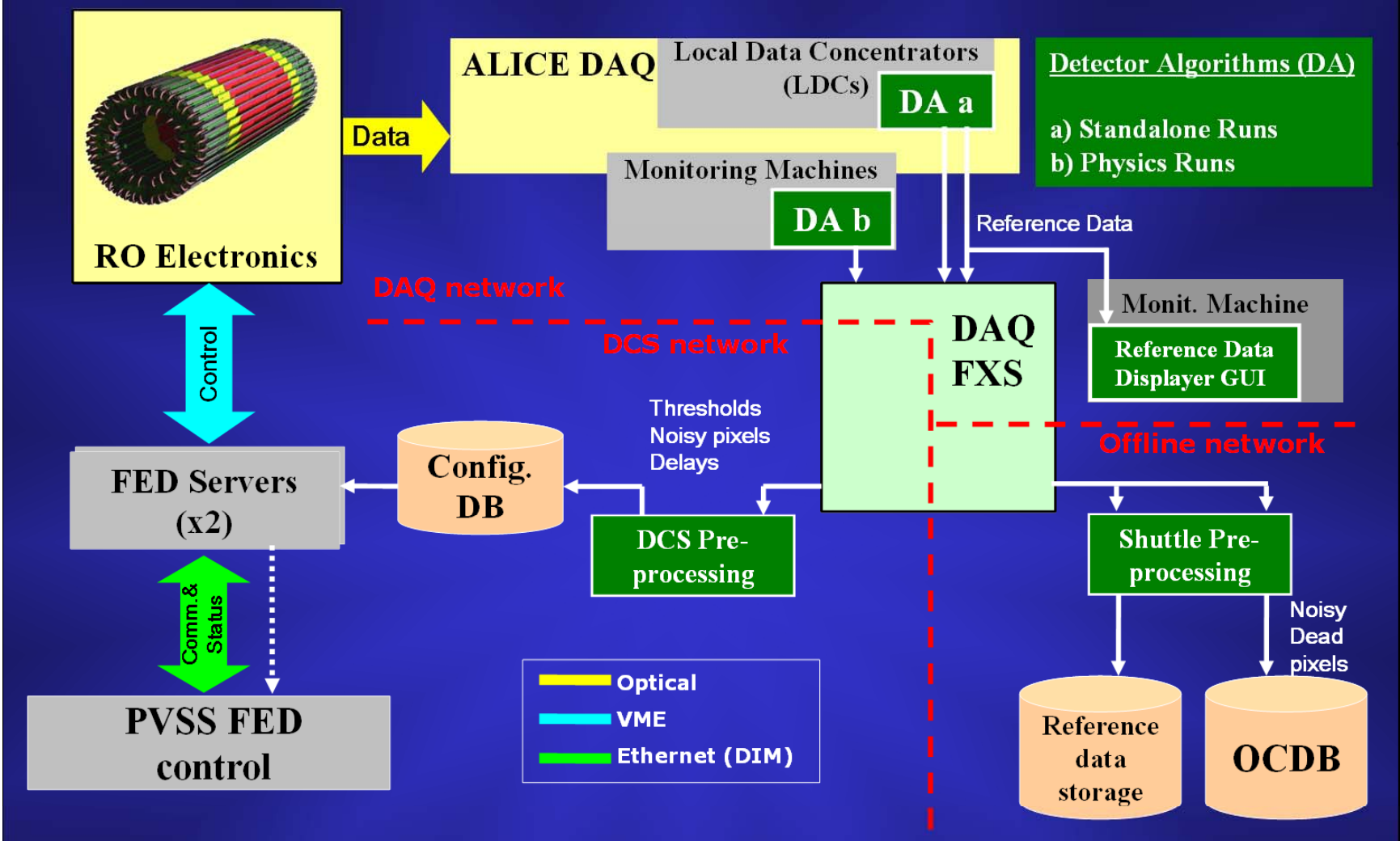
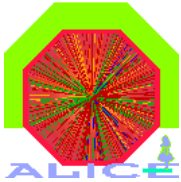
FastOr simulation scheme



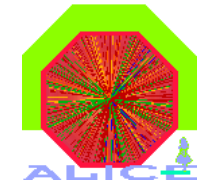
FastOr bits reconstruction



Online calibration scheme



Online calibration readiness



Run Type	Front End Actions	Detector Algorithm	Offline Preprocessor	Configuration DB Update
ITSSPDSCANda (LDC end-of-run processing):				
DAQ_MIN_TH_SCAN				
DAQ_MEAN_TH_SCAN				
DAQ_UNIFORMITY_SCAN				
DAQ_NOISY_PIX_SCAN				
DAQ_PIX_DELAY_SCAN				
DAQ_FO_UNIF_SCAN				
ITSSPDPHYSda (online monitoring):				
PHYSICS				
ITSSPDVertexDiamondda (online monitoring):				
PHYSICS				

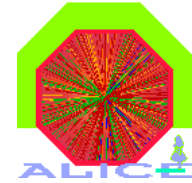
	Working
	Under test

	Nothing to be done
--	--------------------

recently finalized (Annalisa)



FastOr online calibration



- Aim: to find a set of DACs for uniform response of the FO output within the chip
- Run type: DAQ_FO_UNIF_SCAN
- DA to be executed at the very end of the calibration procedure.
- Expected to run few times (once steady running conditions are achieved)

FO Uniformity scan: the procedure

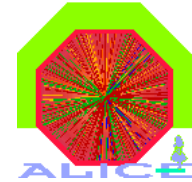
Given a predefined number of trigger inputs: the FO output counts are recorded for all DAC permutations and per specific pixel configuration (matrix ID) within the chip



- Matrix ID 0 (no active pixel)
- Matrix ID 1 (1 pixel)
- Matrix ID 2 (1 pixel)
- Matrix ID 3 (1 pixel)
- Matrix ID 4 (1 pixel)
- Matrix ID 5 (1 pixel)
- Matrix ID 6 (>>1 pixels)



FastOr DA related classes

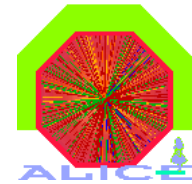


-> optimize the data size and the processing time.

- **AliITSONlineSPDfoChipConfig**
 - Simple container: matrix ID, counts, coordinates (when applicable)
- **AliITSONlineSPDfoChip**
 - Collects the n-matrices used in the chip
- **AliITSONlineSPDfoInfo**
 - Contains general information on the scan (#trigger inputs, DAC unique identifiers, etc.)
- **AliITSONlineSPDfo**
 - Manager class: it knows the data structure, how to store the data, how to read them, how to get information
- **AliITSONlineSPDfoAnalyzer**
 - Performs the algorithm to choose the best N-DAC configuration and assigns to the n-tuple a quality flag (efficiency dependent)



FastOr DA output



- One file per equipment where the DAC permutations and the matrix ID counts are stored.

Such files (20 files in total) will be shipped to the offline Reference Data (preprocessor):

- Output file size depends on the number of DACs and the step size (= permutations):
 - e.g.: 4 DACs, 200 permutations, 6 matrices and full equipment on, retrieves 206k
- A txt file where a set of chosen DACs and the relative quality flag are listed to be sent to the DCS FXS

Status and further plans

- Code committed on svn → SPD task #2084 done (revision 31189)
- Tested with data in DSF
- To be tested/validated at P2 (Annalisa, Sylvain)

