CDC cross-talk simulation

4/02/2019

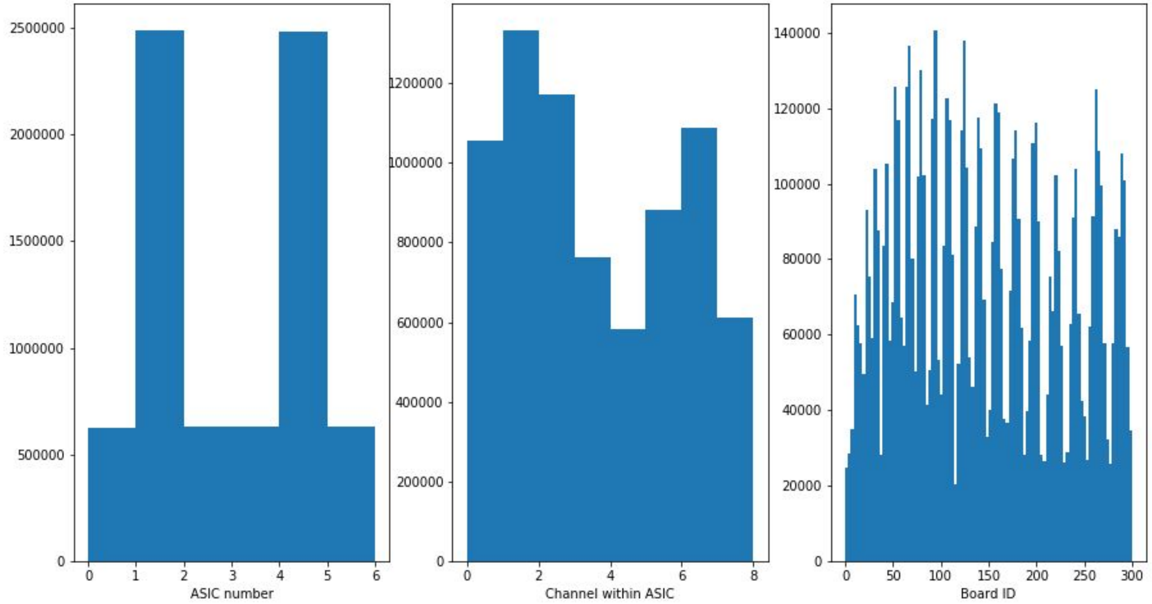
Basic idea

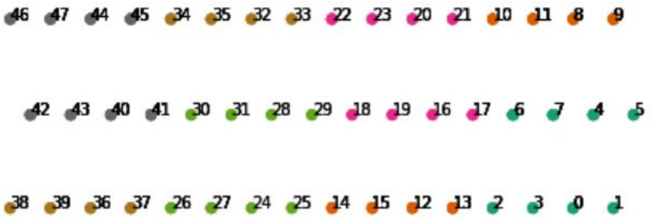
- Start with simulating background inside ASIC, using data
- Select CDC hits, attached to tracks (“signal”)
- Use ASICs with one and only one hit-attached to a track
- Use cosmic data to avoid beam background
- Create a library of ASIC response:

(ADC,TDC,TOT) for each channel as a function of ADCsignal, ASICchannel_signal, …, of the triggering hit.

- Delta-rays to be studied in MC / subtracted to avoid double counting.

Distributions for hits on track (only one hit in ASIC)



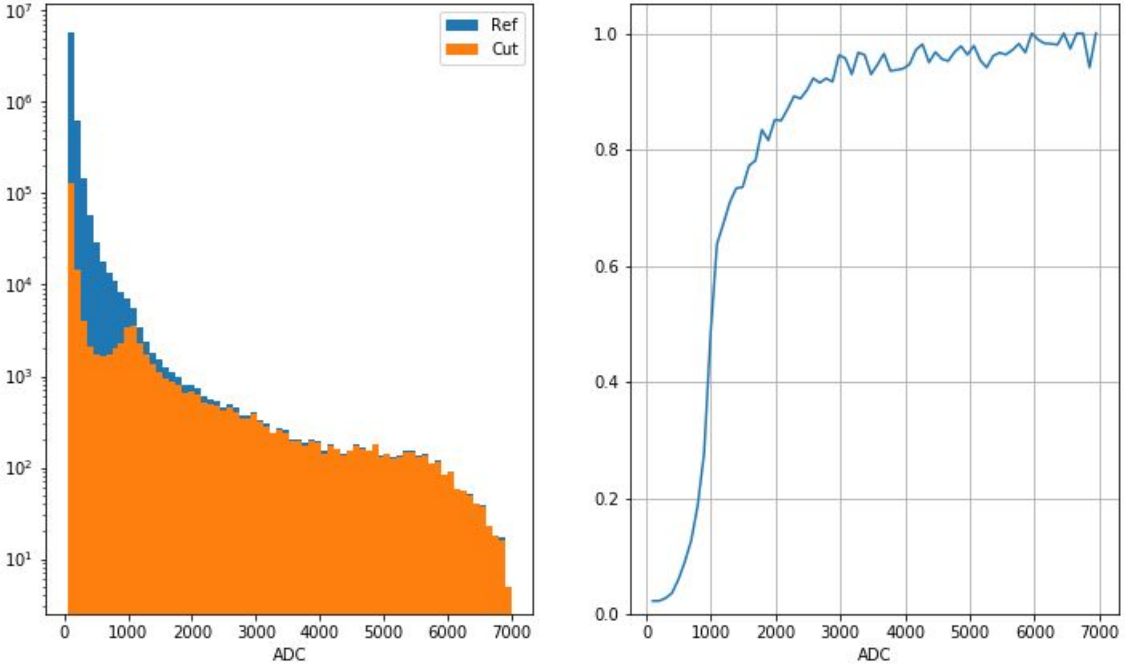


Channel —> wire relation

ASICs 1 and 4 are selected more often since track has higher chance to hit single channel for them. The same is true for channels 1,2 and 6,7 within a given ASIC.

Runs = 342x, exp 8

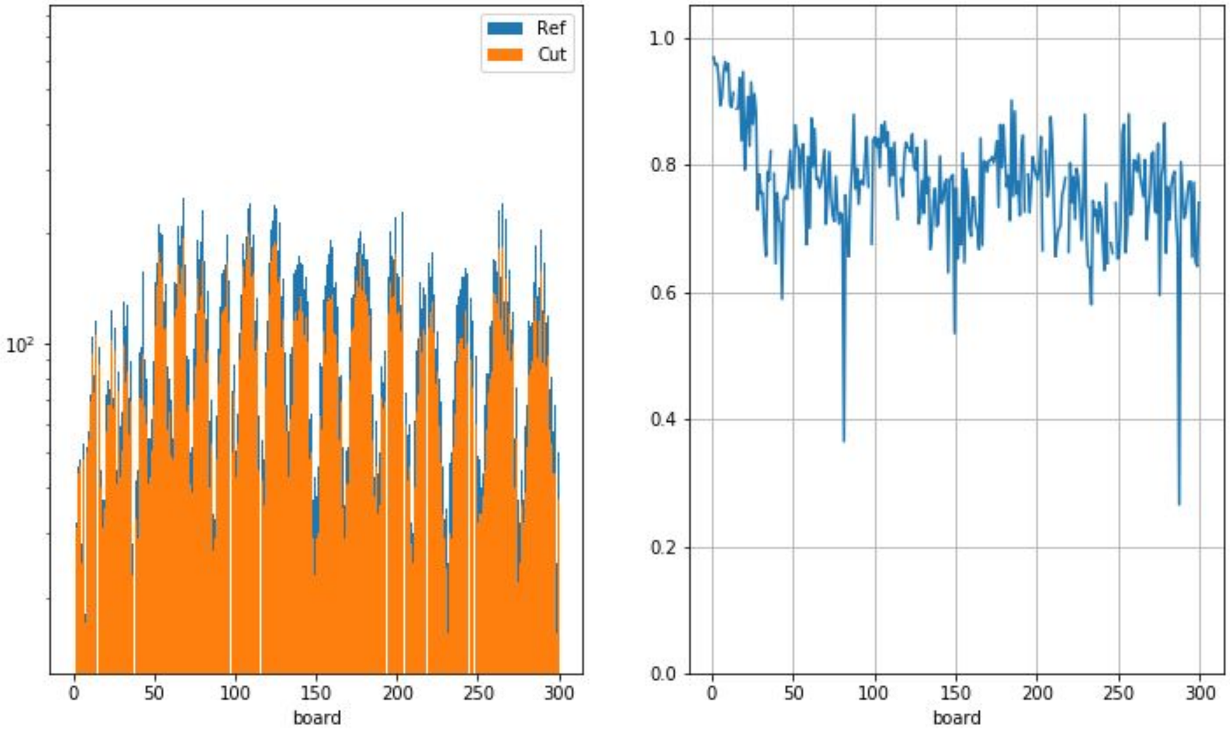
Cross talk probability vs ADC signal



Define ASICs with cross talk if there are at least 3 hits inside it.

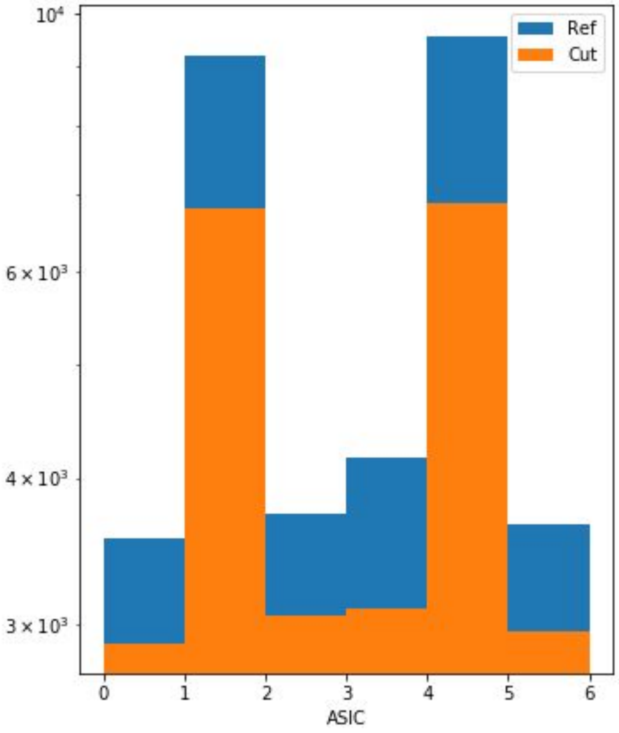
Strong dependence on ADC of the signal hit, threshold turn-on for ADC about 1000

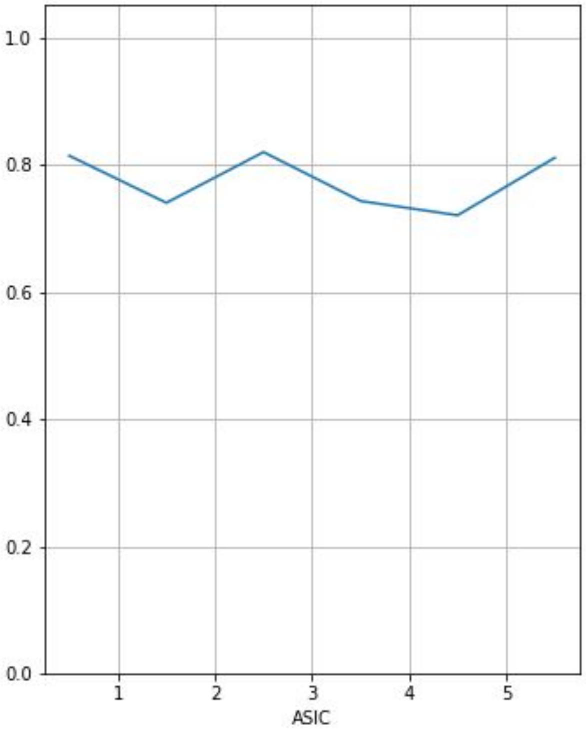
Cross talk probability for ADC(signal)>1000



Moderate dependence on board ID, for ID>25

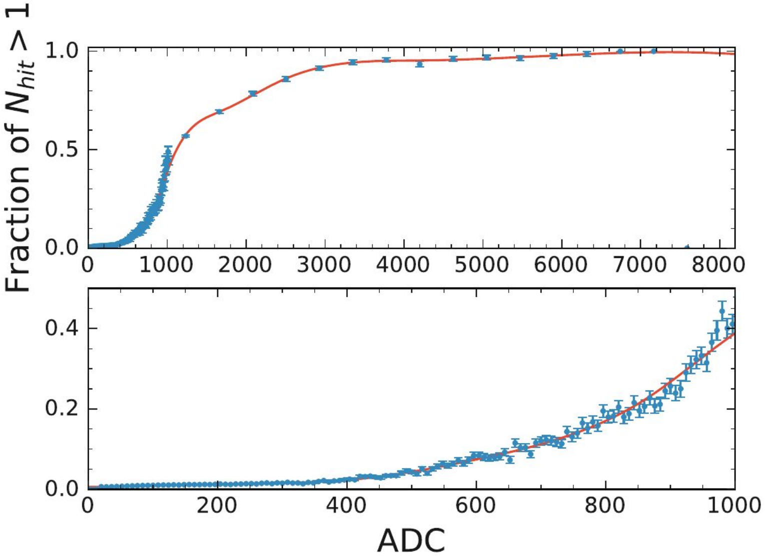
Cross talk probability for ADC(signal)>1000

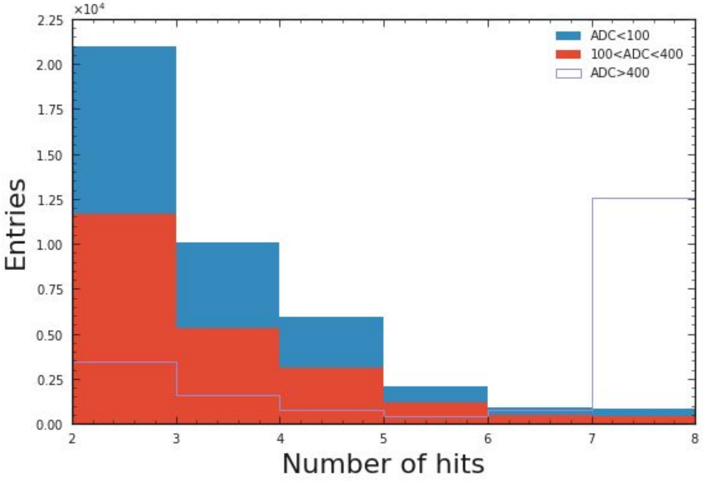




Moderate dependence on ASIC within board

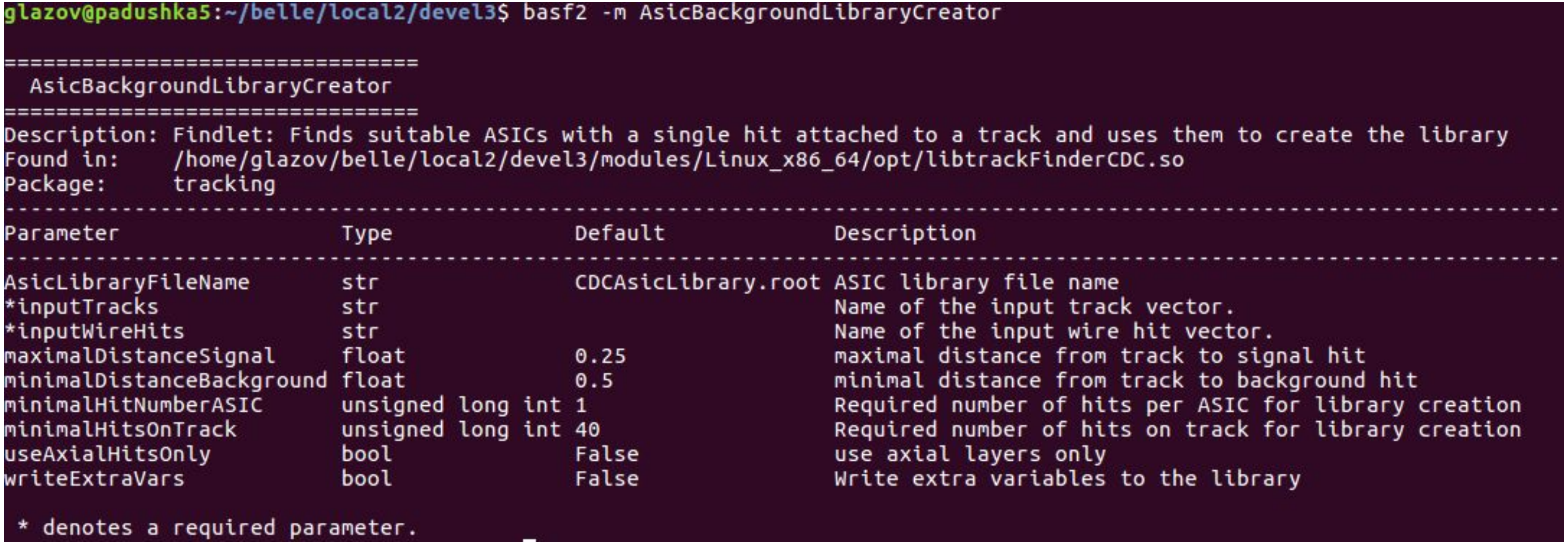
Library implementation





- Retain ADC-dependence only
- Store probability of cross talk vs ADC (1D histogram) plus library for ASICs with Nhit>1 (sorted in a sorted in ADC,channel std::vector, <10 MB)
- MC study shows that d-ray contamination can be neglected.

Software implementation: cosmic data analysis



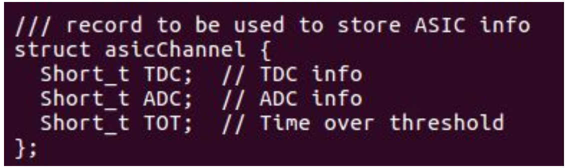
```
glazov@padushka5:~/belle/local2/devel3$ basf2 -m AsicBackgroundLibraryCreator

================================
  AsicBackgroundLibraryCreator
================================
Description: Findlet: Finds suitable ASICs with a single hit attached to a track and uses them to create the library
Found in:    /home/glazov/belle/local2/devel3/modules/Linux_x86_64/opt/libtrackFinderCDC.so
Package:     tracking
------------------------------------------------------------------------------------------------------------------------
Parameter                 Type              Default             Description
------------------------------------------------------------------------------------------------------------------------
AsicLibraryFileName       str               CDCAsicLibrary.root ASIC library file name
*inputTracks              str                                   Name of the input track vector.
*inputWireHits            str                                   Name of the input wire hit vector.
maximalDistanceSignal     float             0.25                maximal distance from track to signal hit
minimalDistanceBackground float             0.5                 minimal distance from track to background hit
minimalHitNumberASIC      unsigned long int 1                   Required number of hits per ASIC for library creation
minimalHitsOnTrack        unsigned long int 40                  Required number of hits on track for library creation
useAxialHitsOnly          bool              False               use axial layers only
writeExtraVars            bool              False               Write extra variables to the library

 * denotes a required parameter.
```

Module “AsicBackgroundLibraryCreator”, to be added to the path after CDC track reconstruction for cosmic events. Produces root file for selected hits with and without cross talk. Added to tracking/trackFindingCDC/ package.

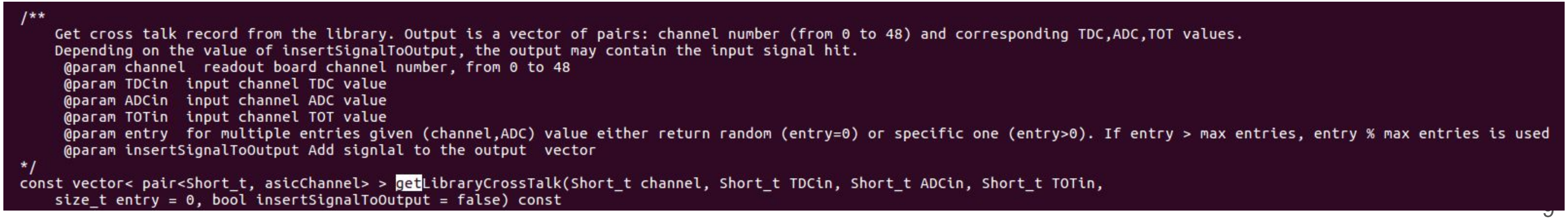
Branch feature/CDCAsicBackgroundLibrary

Software implementation: DB object

- New DB object   CDCCrossTalkLibrary.h in cdc/dbobjects/include
- Code to create payload from the analysis ntuple:
  cdc/examples/prepareAsicCrosstalkSimDB.py
- Payload creation and testing code is added to CDCDatabaseImporter class
- Interface to simulation provided by the getLibraryCrossTalk function



```
/// record to be used to store ASIC info
struct asicChannel {
  Short_t TDC;  // TDC info
  Short_t ADC;  // ADC info
  Short_t TOT;  // Time over threshold
};
```



```
/**
    Get cross talk record from the library. Output is a vector of pairs: channel number (from 0 to 48) and corresponding TDC,ADC,TOT values.
    Depending on the value of insertSignalToOutput, the output may contain the input signal hit.
     @param channel  readout board channel number, from 0 to 48
     @param TDCin  input channel TDC value
     @param ADCin  input channel ADC value
     @param TOTin  input channel TOT value
     @param entry  for multiple entries given (channel,ADC) value either return random (entry=0) or specific one (entry>0). If entry > max entries, entry % max entries is used
     @param insertSignalToOutput Add signlal to the output  vector
*/
const vector< pair<Short_t, asicChannel> > getLibraryCrossTalk(Short_t channel, Short_t TDCin, Short_t ADCin, Short_t TOTin,
    size_t entry = 0, bool insertSignalToOutput = false) const
```

Software implementation status

	Code	Debug/analysis	Convenience scripts
Library generation findlet module AsicBackgroundLibraryCreator	Done	Done	Started, need to add b2luigi
Database object CDCCrossTalkLibrary.h	Done	Done	Done
CDC Simulation (together with H. Ozaki)	Done (separate module)	---	Add tracking validation script

Further things to consider

- How many periods do we want to introduce for the cross talk library ? Probably at least two: before and after the threshold update (run 104 exp 8)
- For background obtained using (pseudo) random trigger events the cross talk is already present. What do we want to do with simulated background ?
  - Add cross talk hits to the overlay background files
  - Use the library from the latests period (?)
- There are cross talk hits which are outside single ASIC. Simulation of this contribution would be the next step.

Discussion

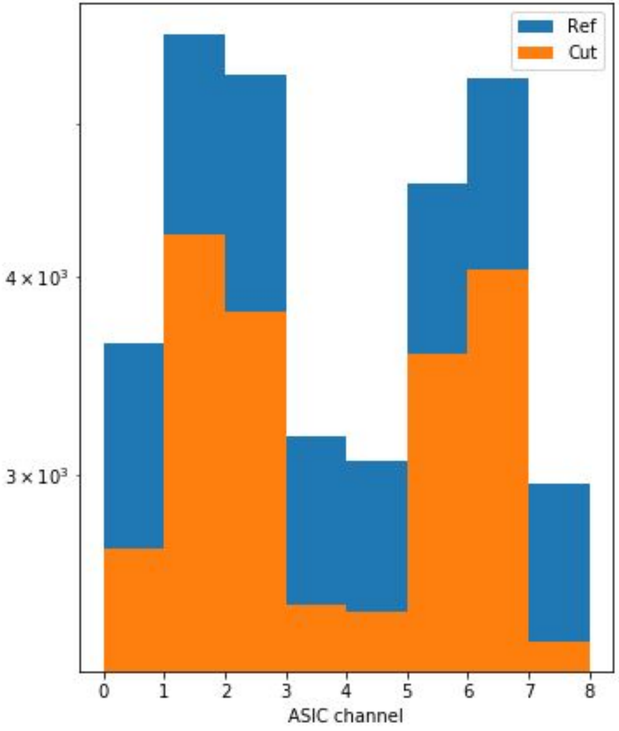
- What is the reason for sharp increase of cross talk probability for ADC>1000 ? Look at FADC raw data ?
- Instead of cosmics, would it be possible to use test pulse, injected to a single ASIC channel (no delta-ray problem) ?
- First look at the data, indicates that the library must be ADC and (for simplicity) ASIC_channel dependent. Given that ADC/TDC/TOT information can be compressed to a single 32 bit word, there should be no problem with the simulation job memory. Sufficient library can be collected in a 2 hour cosmic run.

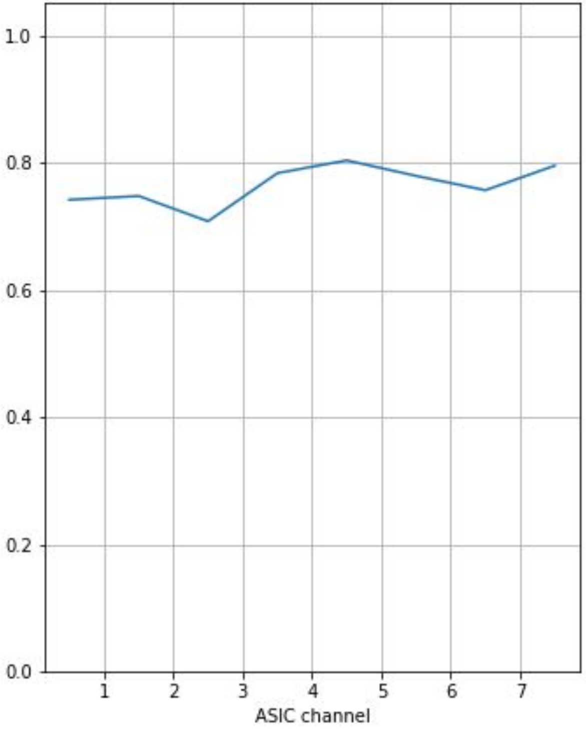
Possible interface for the simulation job:

```
struc ASIChits { int WireID; Int TDC; int ADC; int TOT}
```

```
vector<ASIChits> getAsicCrossTalk(int wireId, int ADC)
```
— initialization and translation to wireID to be done internally.

Cross talk probability for ADC(signal)>1000





Moderate dependence on channel within ASIC