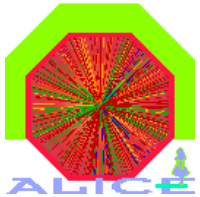
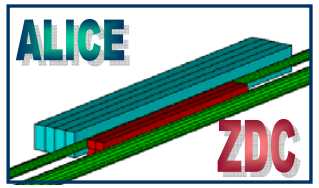


ZDC: online calibration & Shuttle preprocessor



C. Oppedisano and E. Scomparin

ONLINE CALIBRATION STRATEGY
- ➡ DATA SOURCES
- ➡ CALIBRATION STRATEGY
- ➡ USER REQUIREMENTS

STATUS OF THE SHUTTLE PREPROCESSOR
- ➡ IMPLEMENTATION AND TEST

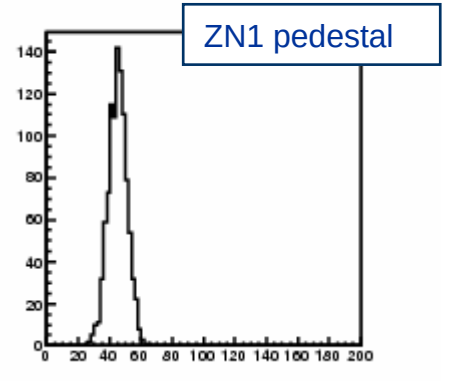
ONLINE CALIBRATION

DATA SOURCES

DCS ➡ HV of the photomultipliers ➡ 22 floats
ZDC table positions ➡ 4 floats (once per RUN)

DAQ ➡ PEDESTAL RUNS ➡ 91 ADC values per event
(47 in-time values + 44 out-of-time values)

Trigger ➡ dedicated pedestal runs
AND generator trigger during physics runs



EM dissociation TRIGGERS ➡ energy calibration of the ZDC response
4 floats (one for each hadronic ZDC)

Trigger ➡ EM dissociation events during physics data taking
(triggered by the ZDC itself)

CALIBRATION STRATEGY

Monitoring machine with root ➡ at the end of the RUN a macro processes the produced histograms to extract the data needed for calibration

AliZDCCalibData inherits from TObject ➡ the ZDC calibration object can be transferred by the Shuttle preprocessor and written in the OCDB

AliZDCCalibData data members:

DAQ

```
// --- Pedestals
Float_t  fMeanPedestal[47];     // Mean pedestal values
Float_t  fMeanPedWidth[47];     // Mean pedestal values
Float_t  fOOTPedestal[44];      // "Out of Time" pedestal values
Float_t  fOOTPedWidth[44];      // "Out of Time" pedestal values
Float_t  fPedCorrCoeff[2][44]; // Fit of correlation in-time vs. out-of-time
// --- E calibration
Float_t  fEnCalibration[6];     // Coeff. for energy calibration
```

DCS

```
// --- PMTs HV values
Float_t  fPMTHVVal[22];         // PMTs HV values
// --- Values for alignement
Float_t  fZDCTablePos[4];       // Vertical value for ZDC tables
```

CALIBRATION ALGORITHM

Working prototype ROOT macro to create calibration data from RAW data in the MONITORING machine

- 2 main functions:
  - AnalyzePed

```
AnalyzePed(const char *fDirPedRawFile, Float_t *MeanPed,
  Float_t *MeanPedWidth)
```

creates pedestal histograms and extracts the parameters needed for the calibration (mean values and widhts)

  - ZDCCalibEn

```
ZDCCalibEn(const char *fDirEMDRawFile, Float_t *CalibCoeff)
```

fills the needed histograms and analyze them to obtain the energy calibration coefficients needed for the calibration

- histograms saved as reference:
  - 47 1-dim. pedestal histos + 44 2-dim. Correlations (in-time vs. out-of-time)
  - 4 1-dim. EM dissociation spectrum histos

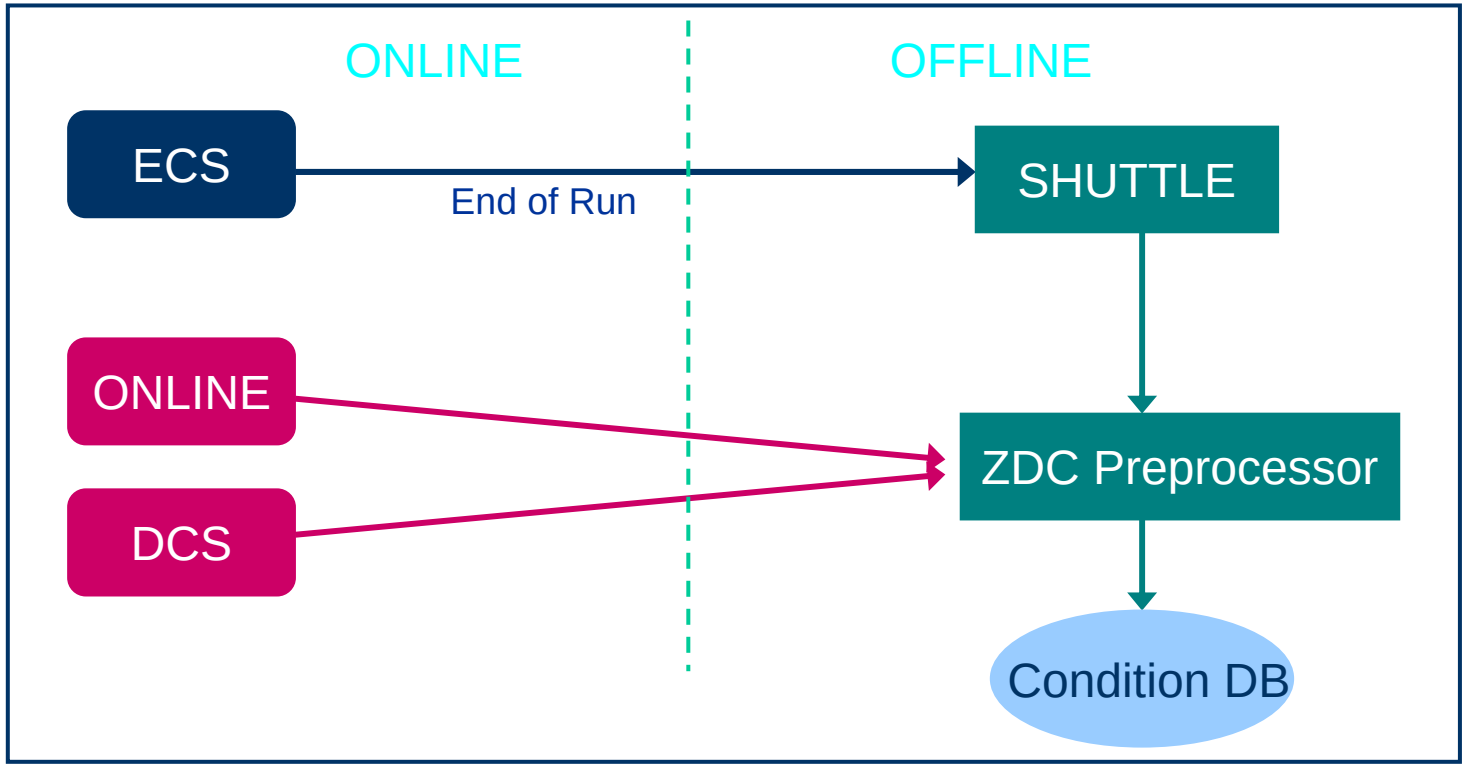
USER REQUIREMENTS

Parameter	Data format/size per channel	Total data size		Source	# of required events/sampling rate	Processing level
		OCDB	reference			
Pedestals	1 float	600 Bytes	5 Mbytes	DAQ	1.00E+05	sub-event
Energy calib. factors	1 float	100 Bytes	50 Kbytes	DAQ	1.00E+05	sub-event
HV of all towers	1 integer	100 Bytes	-	DCS	-	-
Vertical position	1 integer	16 Bytes	-	DCS	-	-

ZDC PREPROCESSOR (I)

ZDC preprocessor implemented and tested
in the framework of the test suite provided by Alberto and JanFiete

DATA SOURCES FOR ZDC PREPROCESSOR



Output of ZDC online calibration routines ➡ input data for Shuttle

ZDC PREPROCESSOR - TEST

TestZDCPreprocessor.C macro to test the ZDC Preprocessor:

- create (or read) a DCS parameter map
- process files that simulate online (DAQ, DCS) files
- instantiate AliZDCPreprocessor

Classes already committed in the ZDC directory:

- AliZDCCalibData
- AliZDCPreprocessor
- AliZDCDataDCS

Calibration deeply tested

DCS data processed succesfully

Algorithm for DAQ procesing still in development

ZDC PREPROCESSOR (II)

AliZDCPreprocessor class implemented

➡ derives from AliPreprocessor and includes an AliShuttleInterface instance

```
AliZDCPreprocessor(const char* detector, AliShuttleInterface* shuttle);
```

➡ Initialize and Process methods implemented

```
virtual void Initialize(Int_t run, UInt_t startTime, UInt_t endTime);
virtual UInt_t Process(TMap* dcsAliasMap);
```

Initialize ➡ initialization of AliZDCDataDCS object

Process ➡ process DCS input data forwarded to AliZDCDataDCS class
➡ get files with calibration parameters from monitoring machine
➡ stores the final AliZDCCalibData object in a CDB entry

AliZDCDataDCS class

➡ derives from TObject

➡ ProcessData method to process DCS data points

```
void ProcessData(TMap& aliasMap, Float_t *CalibData);
```