nBLM Control System Design

Version-1.0 2017/12/01

	REDACTEUR ***Edited by***	**VERIFICATEURS** ***Reviewed by***	**APPROBATEUR** ***Approved by***
NOM **Prénom** ***Name***	Nadot Victor Mariette Yannick Bertrand Quentin	Gougnaud Françoise Segui Laura	ESS-ERIC
Date et signatures ***Date and visas***			

DISTRIBUTION LIST	
Internal : - Thomas Papaevangelou - Laura Segui - Françoise Gougnaud - Tom Joannem - Quentin Bertrand	*External :* - ESS
Copy to : - Florence Ardelier	

DOCUMENT REVISION HISTORY				
Version	Date	Who	Paragraph/page	Comments
Preliminary draft	2017/08/03	Mariette/Nadot	All/All	Creation of the document
V1.0	2017/12/01	Mariette/Nadot	All/All	Document sent to CDR1.1

TABLE OF CONTENTS

1. Introduction

1.1. Purpose

This document describes the specifications and design for the neutron sensitive Beam Loss Monitor (nBLM) control system. The control system interfaces and architecture of the 82 nBLM detectors are also presented.

1.2. Scope

CEA ESSI is in charge of supplying the control system of nBLMs. The FPGA firmware development will be done by ESS ERIC from the specifications described in the document.

1.3. Abbreviations, acronyms and definitions

Name or acronym	Definition
ADC	Analog to Digital Converter
ADU	Analog to Digital Unit
AMC	Advanced Mezzanine Card
BEE	Back-End Electronics
BIS	Beam Interlock System
CA	Channel Access
CEA	Commissariat à l'Energie Atomique
CU	Cooling Unit
EMMC	Enhanced Module Management Controller
ESS	European Spallation Source ERIC
EPICS	Experimental Physics and Industrial Control System
ESSI	ESS Irfu project
FEE	Front-End Electronics
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HEBT	High Energy Beam Transport
HWR	Half Wave Resonator
I/O	Input / Output
ICS	Integrated Control System
LCS	Local Control System
LINAC	Linear Accelerator
PM	Power Module
MCH	MTCA Carrier HUB
MCMC	MicroTCA Carrier Management Controller
MEBT	Medium Energy Beam Transport
MMC	Module Management Controller
MPS	Machine Protection System
MTCA	Or µTCA: Micro Telecomunications Computing Architecture
nBLM	neutron sensitive Beam Loss Monitor
PDR	Preliminary Design Review
PSA	Peak Shape Analysis
PV	Process Variable
RTM	Or µRTM: Rear Transmission Module
SAR	System Acceptance Review

SoW	Scope of Work
TBC	To Be Confirmed and/or yellow highlight
TBD	To Be Defined and/or yellow highlight
TBW	To be Written and/or yellow highlight
TOT	Time over threshold

1.4. Related Documents

	Name of reference document	ESS reference
[1]	nBLM Project PDR1.1	CHESS, ESS-0087794, 2016
[2]	nBLM System PRD1.2	July 2017
[3]	ESS CEA IKCA – Schedule AIK 7.9	CHESS, ESS-0052571, 2016
[4]	ESS_IKC_BLM_Control_2017-10-04	
[5]	nBLM system risk analysis	
[6]	Modes covered by nBLM system	

1.5. Context

The purpose of an nBLM detector is to detect fast neutrons in order to measure the beam loss. The beam loss measurement is mandatory in order to:

- Ensure the accelerator security with a fast beam stop in case of a beam loss accident
- Contribute to the human safety with a low level line activation (<1 W/m)

Therefore the nBLM detector is sensitive to **fast neutrons,** but insensitive to unnecessary X and γ rays and thermal neutrons.

A nBLM detector could be a “fast” detector or a slow “detector”, few differences are set in the FEE and the packaging (polyethylene for the slow detector). The characteristics of each detector are described in the chapter 4.2).

Originally one nBLM module was a slow and a fast detector attached together. Now we do not talk about “module” but “detector” as it could be or not separated.

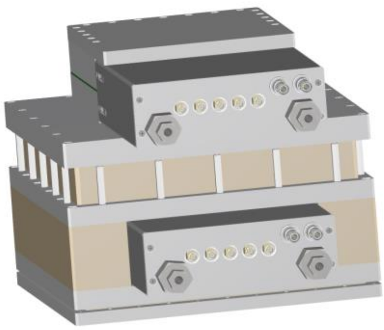


Figure 1: *Picture of the nBLM prototype. There are 2 detectors attached together. The “fast” detector is seen in the front-top side, while the “slow” detector is attached behind it, surrounded by the polyethylene.*

The nBLM detector is a Micromegas detector as defined in documents [1] and [2]. It is a gaseous detector with a gas chamber. There are one input and one output for the gas (He+10%CO2) for each detector.

Each Micromegas detector also needs to be powered with high voltage, thus 2 high voltage channels are needed per nBLM detector.

Moreover a preamplifier of the FEE is used for each nBLM detector, thus we need to power it with low voltage.

Along the Linac, the total number of nBLM detectors is 82 (see chapter 3).

2. Hardware description

2.1. Interface

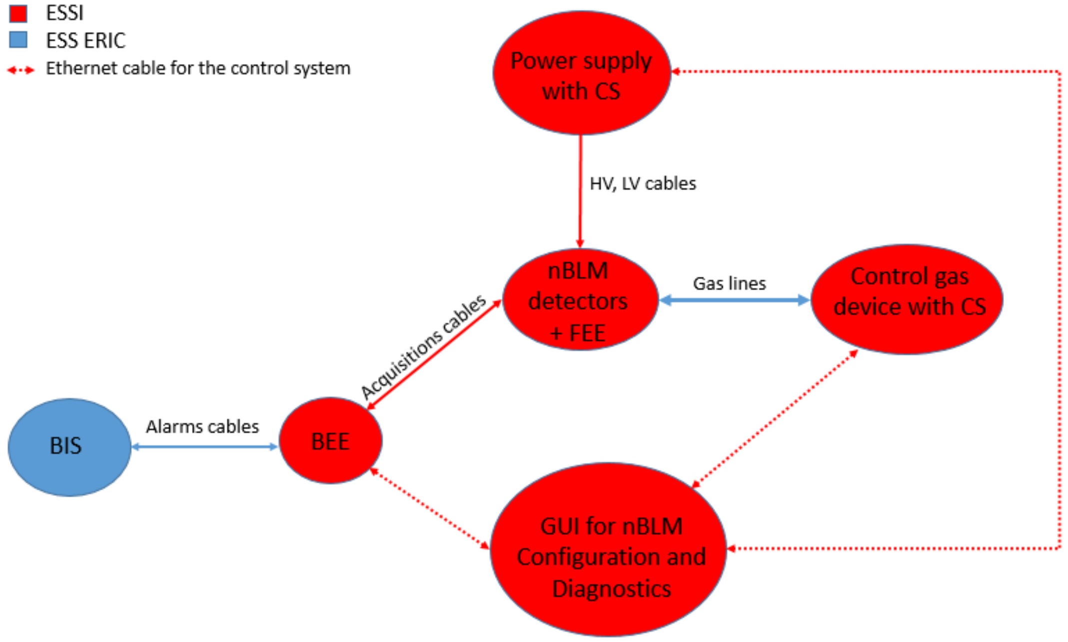


Figure 2: *nBLM interface overview*

Remark: The FPGA firmware development inside the BEE is not in the scope of ESSI

2.2. Descriptions of hardware devices involved in the nBLM control System

The Control system must manage the slow control of the HV, LV, and Gas devices. It also manages the detection settings, acquisitions from BEE and archiving.

2.2.1. Gas management device

Monitoring/PLCs/instrumentation architecture for the gas is divided into 3 subsets: Gas Storage (GS), Gas Distribution (GD) and Gas DTL Line (GDTL).

2.2.1.1. Hardware architecture for the gas control

The hardware architecture complies with ESS standards:

- Kontron Industrial PC
- Siemens S7-1500 CPU
- Siemens Input/output cards

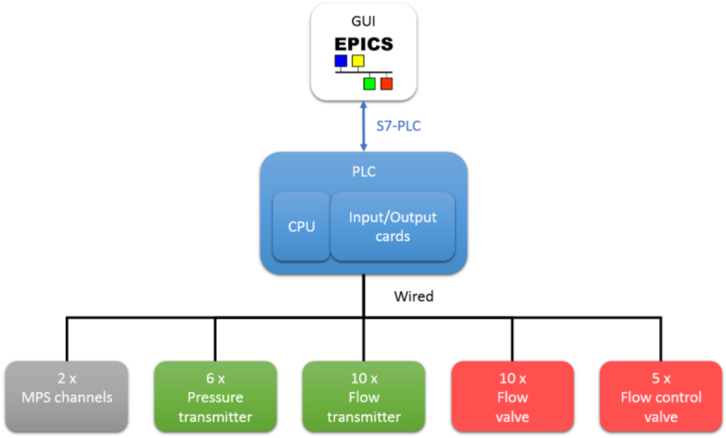


Figure 3: *Gas control overview*

An EPICS server running on PLCs was developed by ESSI and communicates with the s7plc EPICS driver running on the Kontron industrial PC.

Input cards acquire the sensor pressure values. Flow valves are controlled by digital output cards. Flow measurements and flow controls valves would be performed by Profibus Network.
The PLC will manage the gas interlock.

In Figure 5 we can see an example of the power supply, the CPU and some of the cards, and how they will fit into a DIN-rail. We can also propose the option without profibus, but we prefer its use as it would reduce the number of cards.

2.2.1.2. Presentation of the hardware for the gas control

The main goal of the PLC is to ensure independently the integrity of the 10 detector sets. The PLC also manages the gas flow regulation.

The PLC performs flow control (PID type) for each set of detectors. It will ensure the security of the detectors by the control of the electrovalve. All the controls settings and warning/fault thresholds can be fully adjusted by the user.

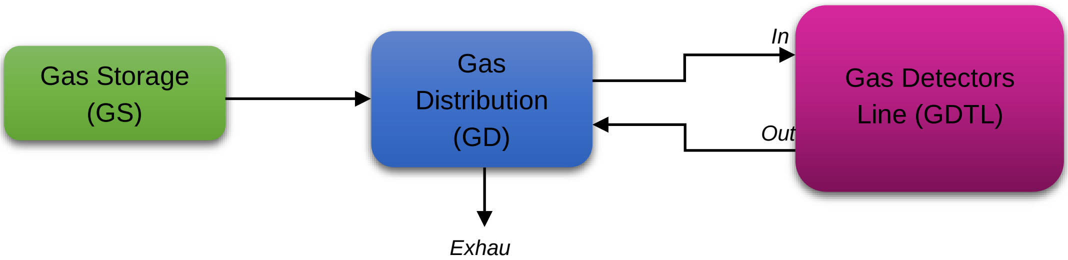


Figure 4: *Gas system architecture*

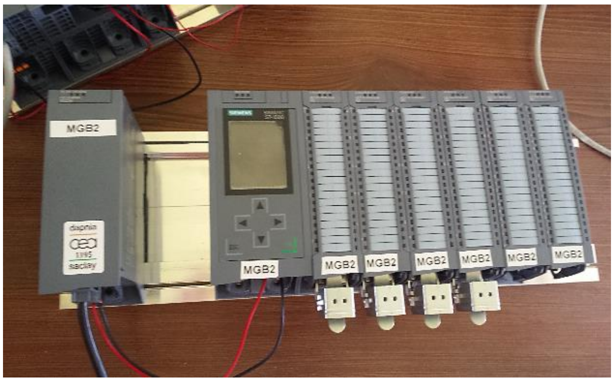


Figure 5: *PLC system for the gas control, with the CPU and some analogic and digital Siemens cards used.*

2.2.2. Hardware for High Voltage and Low Voltage

To control high and low voltages, the CAEN SY4527 crate has been chosen:

- To supply Mesh and Drift of each detector, A7030 modules (3kV/1mA) will be plugged inside the crate. Each A7030 module can control 48 channels.
- To supply low voltages of the pre-amplifiers, A2519 modules (15V/5A) will be plugged inside the crate. Each A2519 module can control 8 channels.

The SY4527 have 16 slots.

The CAEN SY4527 provides an EPICS service to access to Process Variables using the CA protocol. The EPICS IOC is integrated in the CPU of the crate.

The CAEN SY4527 takes in charge its interlocks.

2.2.3. Hardware for the Back-End Electronics (BEE)

For the fast acquisitions (BEE), the architecture is ICS compliant. We use a MTCA crate with a power module, a MCH board and one or several AMC board recommended by ICS: the IOxOS IFC1410 board.

The IFC1410 board has 2 FMC ports where we can connect ADC3111 acquisition FMC board for example. The FMC board must be compatible with the IFC1410 carrier board and also compliant with the ESS recommendation.

2.2.3.1. MTCA crate

MTCA = µTCA: Micro Telecomunications Computing Architecture

A cooling unit, a power module, a backplane and a MCH board are the basic minimum configuration.

Communications between AMC and MCH are done through the backplane.

The adjacent figure shows the board architecture.

MCMC : MicroTCA Carrier Management Controller
MCH : MTCA Carrier HUB
MMC : Module Management Controller
AdvancedMC : AMC : Advanced Mezzanine Card
EMMC : Enhanced Module Management Controller
PM : Power Module
CU : Cooling Unit
µRTM : MicroTCA.4 Rear Transition Module

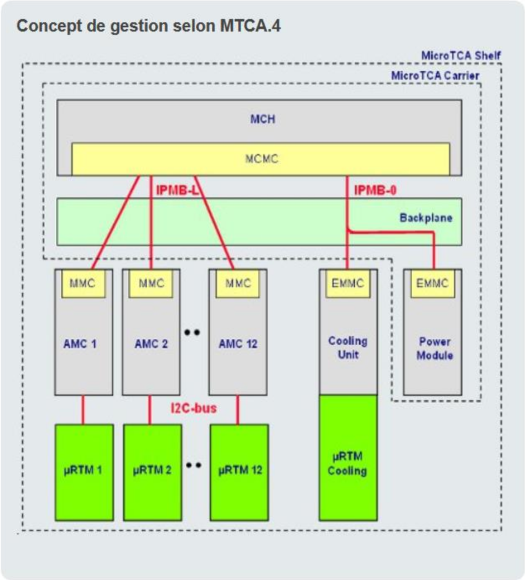


Figure 6: *MTCA architecture*

2.2.3.2. MCH board description

These boards configure the communications in the backplane.

2.2.3.3. AMC boards IOxOS IFC1410

IOxOS Technologies provides the IFC1410, an MTCA.4 Intelligent FMC Carrier in AMC form factor featuring NXP QorIQ T Series and Xilinx Artix-7 and Kintex UltraScale devices.
The IFC1410 can carry 2 FMC boards and be connected to one µRTM.

2.2.3.4. IOxOS ADC3111 FMC board

Each FMC ADC3111 can read 8 analog signals.
The input voltage range is -0.5V to 0.5V.

As the timing response of the detector is very fast (at the precision of ns) and the total duration of an event is of the order of 100-200 ns, the acquisition speed of 250 Msps of the FMC board is satisfying.

2.2.3.5. Timing receiver board

The timing receiver board will be ESS choice compliant.

2.2.3.6. FMC digital Input/output

The digital input/output board will be ESS choice compliant.

This FMC Digital Input/Output board (DIO) will be configured to use up to 8 outputs (one fast alarm output per detector).

This DIO FMC is directly connected to the Kintex UltraScale FPGA of the IFC1410 board. Thus the FPGA firmware could send alarm without intermediate software.

2.2.3.7. Overview of a possible BEE/MTCA crate configuration for the nBLM

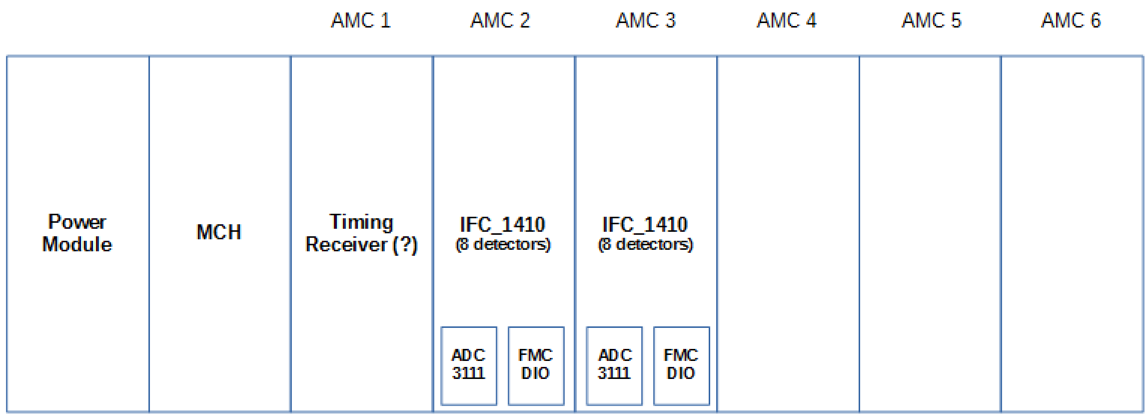


Figure 7: *Possible BEE or MTCA crate for nBLM acquisitions*

3. Distribution

3.1. Detector location along the Linac

The total number of nBLM detectors is 82. But in the initial agreement described in document [3], it was specified that 42 nBLM modules (thus 84 detectors) should be provided. Thus 2 detectors will be spares.

The number of detectors per section is the following:

	Detectors
MEBT	4
DTL	42
SPK	26
MB	2
HB	2
PBW	6
Total	**82**

We see the *nBLM detector location along the Linac* in *Figure 8*, as said in the chapter 1.5, that a pair of fast and slow detectors could be placed together.

Legend of the *nBLM detector location along the LinacFigure 8*:

S: slow nBLM detector

F: fast nBLM detector

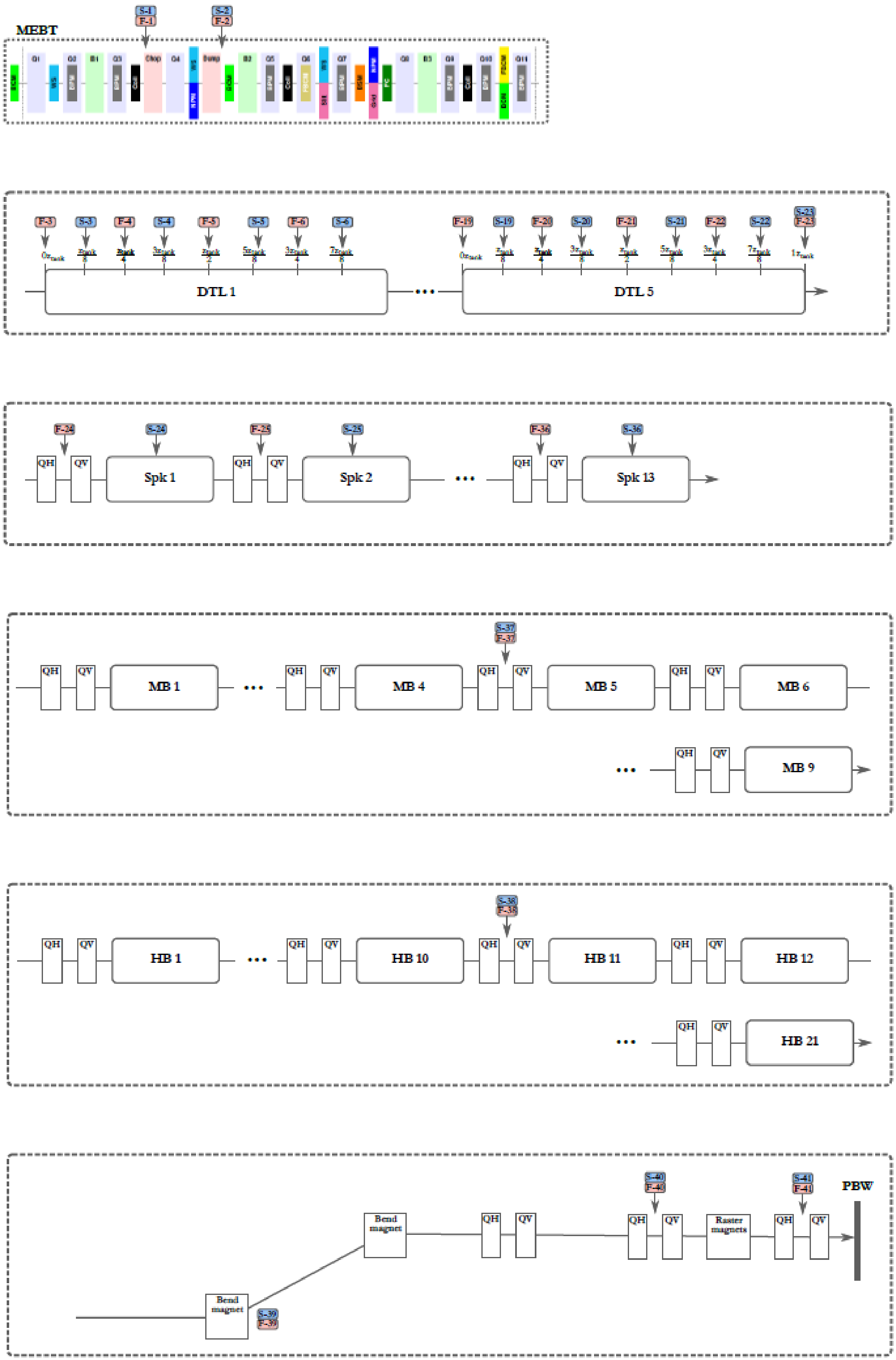


Figure 8: *nBLM detector location along the Linac*

3.2. LV and HV distribution

- One LV channel can supply several pre-amplifiers. There are 8 channels in one A2519 module, it is enough for all pre-amplifiers.

- As 2 channels (mesh and drift) are required per detector, 82 x 2 = 164 HV channels are needed. There are 48 channels in one A7030 module. Thus we need four A7030 modules.

Thus LV and HV boards will use 5 slots in the SY4527 crate and we will only need one SY4527 crate (it has been decided to use 2 SY4527 due to the long HV cable) for the whole HV and LV modules. The Figure 9 is an overview of the LV and HV distribution.

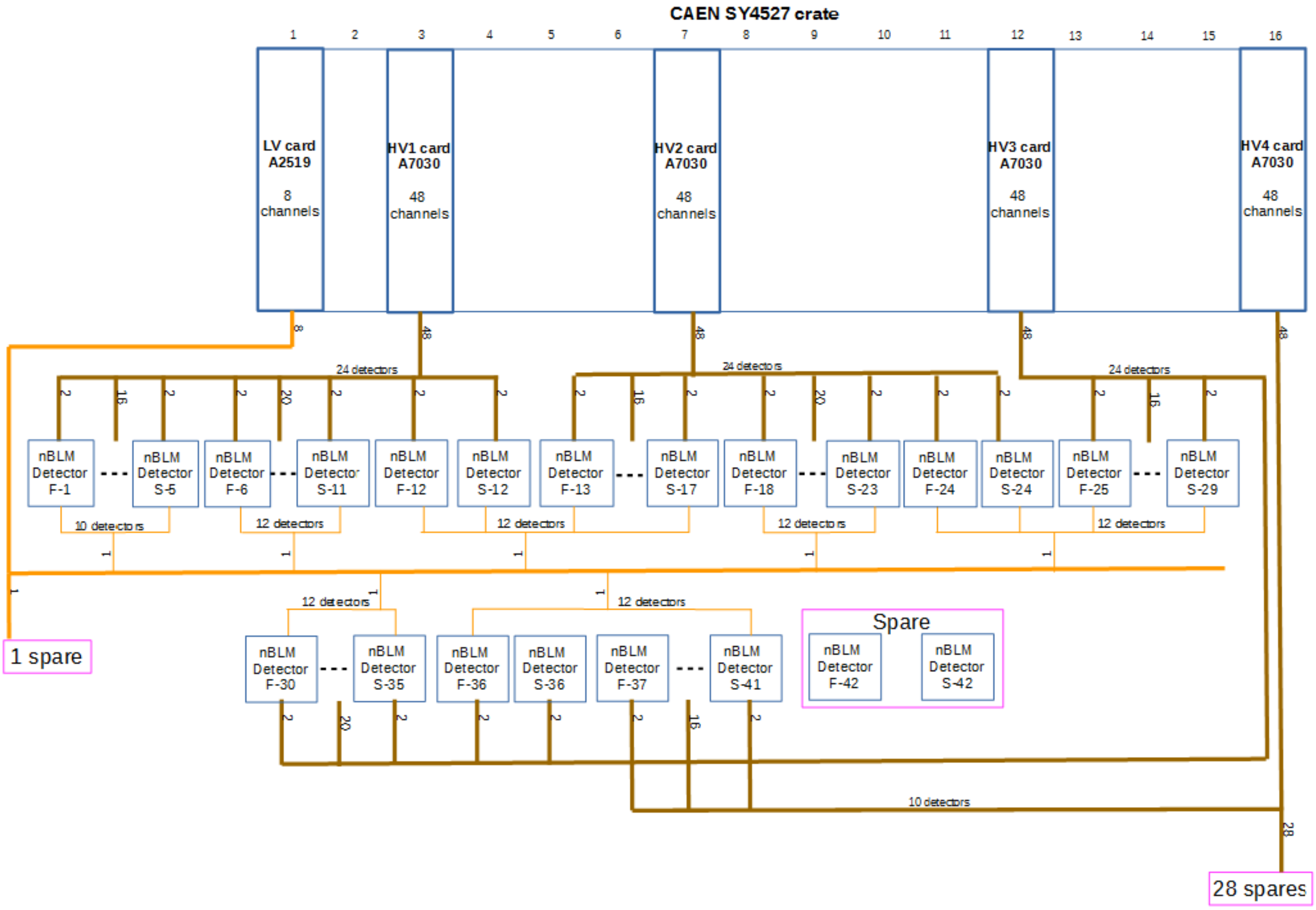


Figure 9: *Example of a hardware distribution for LV and HV*

A second crate for cable length convenience will be added.

3.3. Gas distribution

10 gas lines will provide gas for the whole 82 detectors. The following table gives gas line per section:

Detectors in	Gas line	Number of detectors
MEBT-DTL1	Line 1	12
DTL2	Line2	8
DTL3	Line 3	8
DTL4	Line4	8
DTL5	Line 5	10
SPK1-4	Line6	8
SPK5-8	Line 7	8
SPK9-13	Line8	10
MB-HB	Line 9	4
Bend Magnet	Line 10	6
Total	**10**	**82**

3.4. Acquisition distribution

The acquisitions in the BEE are done with the ADC3111 FMC module. As seen in chapter 2.2.3 there will be only one ADC3111 FMC board per IFC1410 board, hence 8 acquisition channels maximum.

We decided to make neighbour pairs of fast and slow acquisitions for software architecture convenience. Channels repartition in ADC3111 inputs are one fast detector and the slow neighbour detector set respectively on 2 consecutive channels and so on.

To avoid blind section in case of a hardware failure (IFC1410 or ADC3111), it has been decided to cross detector pairs on several ADC3111 modules. Here is the acquisition input distribution for the first and second ADC3111 modules:

- ADC n°1 on the IFC1410 n°1:
  - channel1: fast nBLM1 detector (F-1)
  - channel2: slow nBLM1 detector (S-1)
  - channel3: fast nBLM3 detector (F-3)
  - channel4: slow nBLM3 detector (S-3)
  - channel5: fast nBLM5 detector (F-5)
  - channel6: slow nBLM5 detector (S-5)
  - channel7: fast nBLM7 detector (F-7)
  - channel8: slow nBLM7 detector (S-7)

- ADC n°2 on the IFC1410 n°2:
  - channel1: fast nBLM2 detector (F-2)
  - channel2: slow nBLM2 detector (S-2)
  - channel3: fast nBLM4 detector (F-4)
  - channel4: slow nBLM4 detector (S-4)
  - channel5: fast nBLM6 detector (F-6)
  - channel6: slow nBLM6 detector (S-6)
  - channel7: fast nBLM8 detector (F-8)
  - channel8: slow nBLM8 detector (S-8)

Regarding the acquisition we need a minimum of 11 FMC modules and 11 IFC1410 boards to cover the 82 nBLM detectors (In case where all the 8 channels of each FMC module are used).

3.5. Distribution/configuration table

The next configuration table gives locations, HV, LV, and gas (acquisition TBD) channels for each detector.

	ADC3111 FMC		Detector		To Be Defined				HV channel	
IFC1410	Number	channel	Type	Number	Rack name	Location	Segment	Gas line	Mesh	Cathode
1	1	1	F	1	FEB-050	MEBT	-	1	1	2
		2	S	1	FEB-050	MEBT	-	1	3	4
		3	F	3	FEB-050	DTL	1	1	9	10
		4	S	3	FEB-050	DTL	1	1	11	12
		5	F	5	FEB-050	DTL	1	1	17	18
		6	S	5	FEB-050	DTL	1	1	19	20
		7	F	7	FEB-050	DTL	2	2	25	26
		8	S	7	FEB-050	DTL	2	2	27	28
2	2	1	F	2	SPK-010	MEBT	-	1	5	6
		2	S	2	SPK-010	MEBT	-	1	7	8
		3	F	4	SPK-010	DTL	1	1	13	14
		4	S	4	SPK-010	DTL	1	1	15	16
		5	F	6	SPK-010	DTL	1	1	21	22
		6	S	6	SPK-010	DTL	1	1	23	24
		7	F	8	SPK-010	DTL	2	2	29	30
		8	S	8	SPK-010	DTL	2	2	31	32
3	3	1	F	9	FEB-050	DTL	2	2	33	34
		2	S	9	FEB-050	DTL	2	2	35	36
		3	F	11	FEB-050	DTL	3	3	41	42
		4	S	11	FEB-050	DTL	3	3	43	44
		5	F	13	FEB-050	DTL	3	3	49	50
		6	S	13	FEB-050	DTL	3	3	51	52
		7	F	15	FEB-050	DTL	4	4	57	58
		8	S	15	FEB-050	DTL	4	4	59	60
4	4	1	F	10	SPK-010	DTL	2	2	37	38
		2	S	10	SPK-010	DTL	2	2	39	40
		3	F	12	SPK-010	DTL	3	3	45	46
		4	S	12	SPK-010	DTL	3	3	47	48
		5	F	14	SPK-010	DTL	3	3	53	54
		6	S	14	SPK-010	DTL	3	3	55	56
		7	F	16	SPK-010	DTL	4	4	61	62
		8	S	16	SPK-010	DTL	4	4	63	64
5	5	1	F	17	FEB-050	DTL	4	4	65	66
		2	S	17	FEB-050	DTL	4	4	67	68
		3	F	19	FEB-050	DTL	5	5	73	74
		4	S	19	FEB-050	DTL	5	5	75	76
		5	F	21	FEB-050	DTL	5	5	81	82
		6	S	21	FEB-050	DTL	5	5	83	84
		7	F	23	FEB-050	DTL	5	5	89	90
		8	S	23	FEB-050	DTL	5	5	91	92
6	6	1	F	18	SPK-010	DTL	4	2	69	70
		2	S	18	SPK-010	DTL	4	2	71	72
		3	F	20	SPK-010	DTL	5	5	77	78
		4	S	20	SPK-010	DTL	5	5	79	80
		5	F	22	SPK-010	DTL	5	5	85	86
		6	S	22	SPK-010	DTL	5	5	87	88
		7	F	24	SPK-010	SPK	1	6	93	94
		8	S	24	SPK-010	SPK	1	6	95	96
7	7	1	F	25	SPK-050	SPK	2	6	97	98
		2	S	25	SPK-050	SPK	2	6	99	100
		3	F	27	SPK-050	SPK	4	6	105	106
		4	S	27	SPK-050	SPK	4	6	107	108
		5	F	29	SPK-050	SPK	6	7	113	114
		6	S	29	SPK-050	SPK	6	7	115	116
		7	F	31	SPK-050	SPK	8	7	121	122
		8	S	31	SPK-050	SPK	8	7	123	124
8	8	1	F	26	MBL-090	SPK	3	6	101	102
		2	S	26	MBL-090	SPK	3	6	103	104
		3	F	28	MBL-090	SPK	5	7	109	110
		4	S	28	MBL-090	SPK	5	7	111	112
		5	F	30	MBL-090	SPK	7	7	117	118
		6	S	30	MBL-090	SPK	7	7	119	120
		7	F	32	MBL-090	SPK	9	8	125	126
		8	S	32	MBL-090	SPK	9	8	127	128
9	9	1	F	33	SPK-050	SPK	10	8	129	130
		2	S	33	SPK-050	SPK	10	8	131	132
		3	F	35	SPK-050	SPK	12	8	137	138
		4	S	35	SPK-050	SPK	12	8	139	140
		5	F	37	SPK-050	MB	-	9	145	146
		6	S	37	SPK-050	MB	-	9	147	148
		7	F	39	SPK-050	PBW	-	10	153	154
		8	S	39	SPK-050	PBW	-	10	155	156
10	10	1	F	34	MBL-090	SPK	11	8	133	134
		2	S	34	MBL-090	SPK	11	8	135	136
		3	F	36	MBL-090	SPK	13	8	141	142
		4	S	36	MBL-090	SPK	13	8	143	144
		5	F	38	MBL-090	HB	-	9	149	150
		6	S	38	MBL-090	HB	-	9	151	152
		7	F	40	MBL-090	PBW	-	10	157	158
		8	S	40	MBL-090	PBW	-	10	159	160
11	11	1	F	41	SPK-050	PBW	-	10	161	162
		2	S	41	SPK-050	PBW	-	10	163	164
		3			SPK-050					
		4			SPK-050					
		5			SPK-050					
		6			SPK-050					
		7			SPK-050					
		8			SPK-050					

4. FPGA firmware specification

4.1. Main characteristics for fast and slow neutron detectors

For slow or fast detector, the deposited energy by a neutron will give a specific shape. This shape can be fitted using a Landau distribution or a polynomial fit, it gives a big peak under the noise. The pulse duration (see the definition below) is about 100-200 ns.

Therefore a fast ADC is major, the chosen ADC3111 (see chapter 2.2.3.4) gives acquisition with 250 MSamples/s (every 4ns) which is enough.

Below is an ideal shape (Landau distribution) of one neutron detection (the neutron peak is negative). Useful characteristics are also displayed:

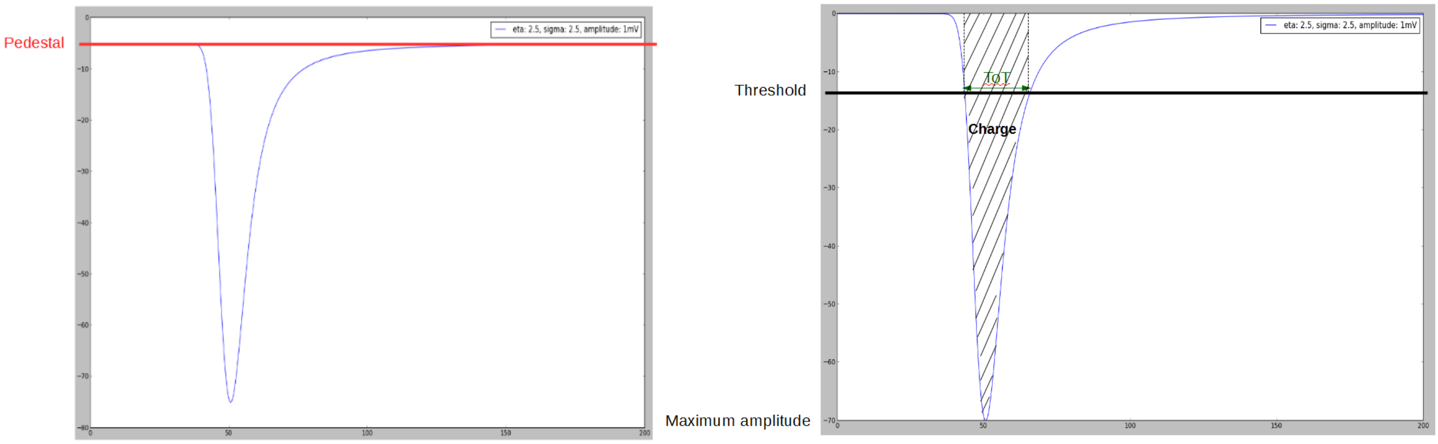


Figure 10: *Ideal shape or peak for one neutron (In the right figure the pedestal has been subtracted)*

Definition of characteristics useful for the analysis:

- **Pedestal:** it is the noise level (which could be different from 0)
- **Maximum amplitude (A):** it is the maximum amplitude of the peak after having subtracted the pedestal level
- **Charge (Q):** it is the integral of the peak signal from the bin below the threshold to the bin above the threshold (see ***Figure 10***).
- **Time over Threshold (TOT):** It is the duration time of the peak below the threshold.
- **Pile-up:** when many neutrons are acquired at the same time or continuously, the shape is different from the ideal shape of one neutron.

Other variables may be useful:

- **Charge over amplitude** (Q/A): ratio between the charge and maximum amplitude for one peak. For the neutrons it should be constant (only in the slow detector) as we have a quite stable energy. For a noisy event or a spark the value will be different.
- **TOT min:** defines a minimum duration of a peak to consider that it is a neutron.

- **TOT max:** defines a duration of a peak from where we consider that there is not only one peak (2 or more neutrons detection could be acquired at the same time)
- **TOT alarm:** defines a maximum duration where the signal is below the threshold. We consider that there is not only one peak, many neutrons are acquired continuously during this maximum duration (pile-up).
- **Mean amplitude:** define the mean amplitude for the slow detector (see the following chapter).

4.2. Differences between fast and slow detectors

This chapter summaries the difference between fast and slow detectors.
For more details, see the document [6].

Slow detector:

- For the slow detection, the neutron moderation process will delay a big part of the events for times from some 10 ns to 200 µs. (17% neutrons in 10 µs, 100% in less than 300µs)
- The mean amplitude (or charge) neutron value is constant for a slow detector at a specific location.
- The detector sensitivity is good for neutrons inside a big energy range (10^{-6} to 100 MeV)
- Angle detection : 4π

This detector is very efficient for neutron counting. It will be monitored to detect small beam loss during long time, commissioning, etc.

Fast detector:

- The fast detector will acquire 100% neutrons in 0.01 µs
- The amplitude and the charge are NOT always the same for all neutrons
- The detector sensitivity is good for neutrons with energy from 1 MeV
- Angle detection : 2π

This detector is very efficient for fast detection of a big number of neutrons. A fast alarm will be generated in case of a consequent beam loss.

The following basic figures outline the different sensitivity and distribution for fast and slow detector:

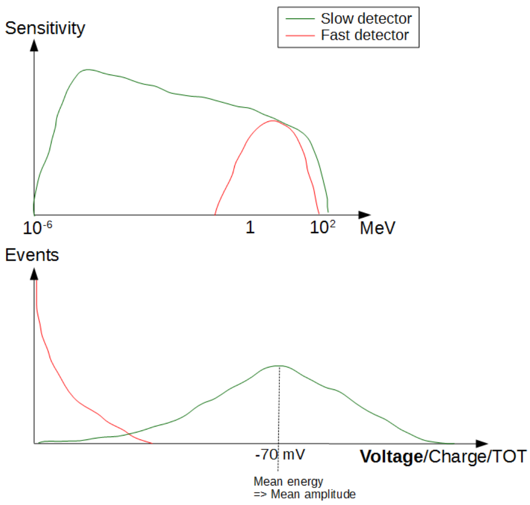


Figure 11: Sensitivity and distribution for fast and slow detectors

Due to the spread out amplitude in the fast detector we talk about **events** (and not neutrons like in a slow detector) for a typical acquired landau shape. We couldn't know with a shape if it is one or many neutrons. For this detector we will then use the TOT variable:

$$Number\ of\ events = \frac{TOT\_mesured}{TOT\_1event}$$

The amplitude distribution in the slow detector is like a Gaussian distribution. Thus if we know the **mean amplitude (or mean Q)** for one neutron it will be easy to calculate how many neutrons will be:

$$Number\ of\ neutrons = \frac{Q\_mesured}{Q\_1neutron}$$

Settings/calibration:

Settings/calibration will be necessary.

We will need to fit each event to a correct distribution determined experimentaly to get the values of the peak.

For each detector, some parameters (analyse threshold, pedestal, etc.) settings could be different. It will depends on the nBLM location (high or low energy section), preamplifier settings, detector type etc.

Detector stability (slow detector):

We want to qualify each detector frequently to check if the neutron detection is still performant.

For the slow detector, the purpose is to determine the mean amplitude and the **mean TOT** and to check the stability of these values over the time (see chapter 7.1)

For the fast detector it has not been defined yet. (Noise rate monitoring ? comparison of the integral in each run... TBD)

4.3. FPGA firmware goal

The Kintex Ultrascale FPGA is integrated into the IOxOS IFC1410 card. This FPGA already makes basic acquisitions up to 16 detectors via 2 ADC3111 FMC cards. Currently, an existing EPICS IOC (ifcdaq) runs in order to view the acquired data.

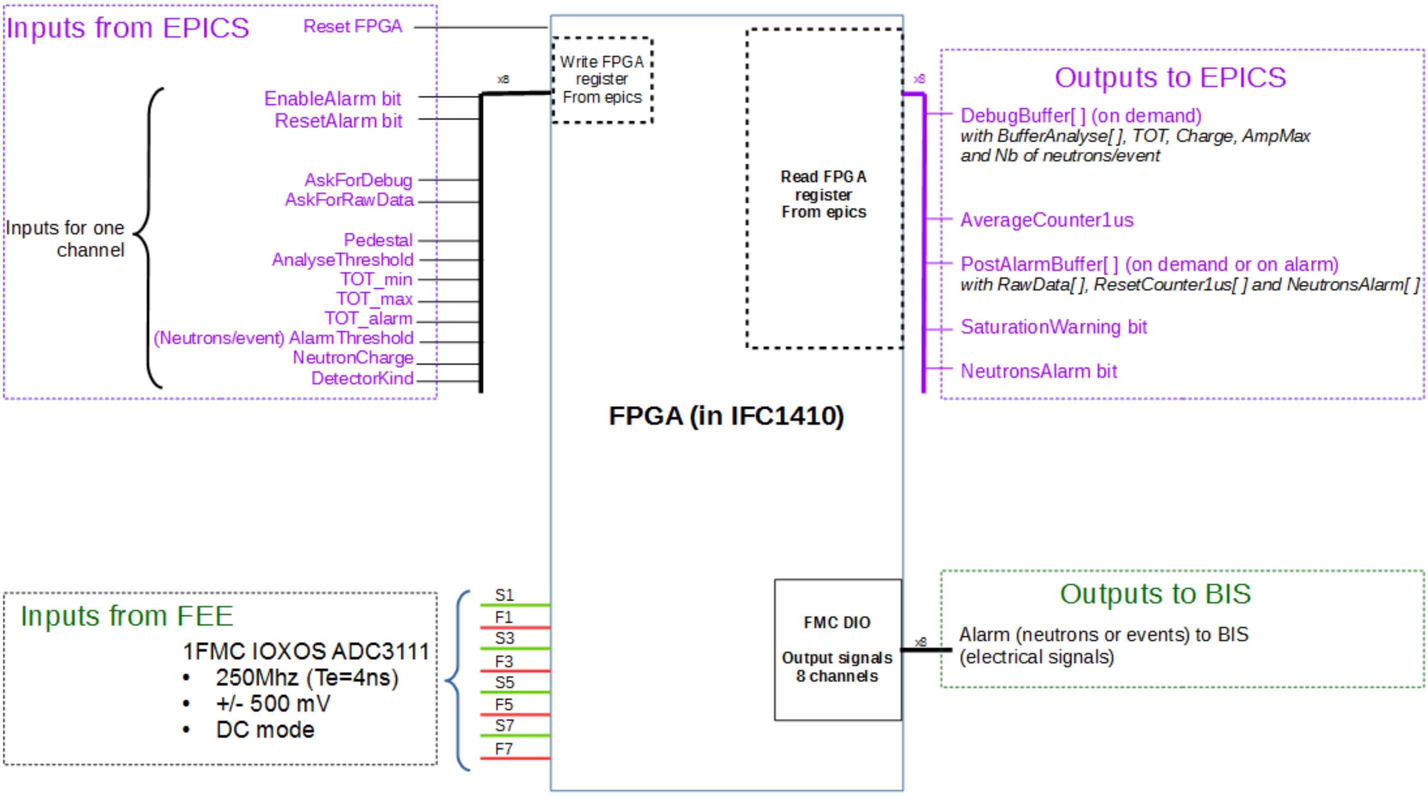
Thus from the actual VHDL source code, the next requirements should be added.

The main goal is to analyse the signal, detect neutrons/events, and count them. If a threshold counting, given an input parameter, is reached then the FPGA will send an electrical signal alarm to the BIS/MPS. The MPS will then decide to stop or not the machine.

In addition of this security feature, the FPGA will provide EPICS with data for monitoring.

Remark: The FPGA firmware is also called bitstream.

4.4. FPGA interfaces



From the FPGA point of view there are 4 kinds of I/Os: slow and fast acquisition signals from detectors, input parameters for the analysis, alarm output signals and output data for post-analysis and monitoring

Figure 12: *FPGA inputs and outputs*

4.4.1. Acquisition signals

The detector acquisition is provided by the ADC3111 FMC cards. The range is -500 to 500 mV. The sampling frequency is 250MHz (T=4ns). The acquisition is permanent (every 4 ns) for all connected detectors. All acquisitions are done in parallel.

Remark: All timing in FPGA data are converted into number of bins. For example 1µs is converted into 250 as the sampling period is 4ns.

4.4.2. Input parameters for the analysis.

The FPGA will manage up to 8 channels. An input Reset FPGA bit will reset the FPGA IPs.

As said in chapter 4.2, each channel settings could be different, thus there are setting data for each one.

All these input parameters could be set in EPICS, (see the PV list in chapter 6.3.1). The following table describes the input parameters to implement:

Inputs	Description	Type	Range
Reset FPGA	Reset the FPGA	boolean	0 or 1
EnableAlarm bit	1 : can't trigger the alarm, 0: can trigger the alarm This input could be useful for tests for example. In normal use the AlarmMPS shouldn't be disabled.	boolean	0 or 1
ResetAlarm bit	As AlarmMPS is latched. It resets it	boolean	0 or 1
AskForDebug	Ask for keeping the last analysis buffer done after a peak detection, and transfer it to EPICS	boolean	0 or 1
AskForRawData	Ask for keeping the last raw data acquisition, and transfer it to EPICS	boolean	0 or 1
Pedestal	Pedestal value to subtract from the input signal to remove the offset	integer	TBD
AnalyseThreshold	Below this threshold, the data are buffered in the BufferAnalyse	integer	TBD
TOT_min	Calculated from the mean Time over Threshold of a neutron (mean TOT) (see Figure 10). In a slow detector, over TOT_min the BufferAnalyse will be analysed by the Analysis algorithm	integer	TBD
TOT_max	Calculated from the mean Time over Threshold of a neutron (mean TOT) (see Figure 10). In a fast detector, when the BufferAnalyse is still running over TOT_max, an event will be added each TOT_max	integer	TBD
TOT_alarm	When the BufferAnalyse is still running over TOT_alarm, it means too many events/neutrons are detected. The AlarmMPS is then trigged (see next paragraph)	integer	TBD
AlarmThreshold	When the neutron/event counting rate is over this threshold, the AlarmMPS is generated (only if EnableAlarm is 0)	integer	TBD
NeutronCharge	It is the neutron charge value. It is the result of the integral of an ideal neutron peak signal from the bin below the threshold to the bin above the threshold (see Figure 10).	integer	TBD
DetectorKind	It indicates to the FPGA if the channel is associated to a slow or a fast detector	integer	TBD

4.4.3.Alarm output signals

When abnormal number of neutrons is counted, the FPGA must send a fast electrical signal to the BIS/MPS.

As described in chapter 2.2.3.6, a DIO-FMC module is plugged in the IFC1410 directly connected to the Kintex Ultrascale FPGA. There are up to 8 detectors per FPGA, so up to 8 digital alarm signals to the BIS/MPS.

For each channel, when this alarm (AlarmMPS) occurs it will be persistent (latched) until the ResetAlarm is required from the operator (EPICS).

4.4.4.Output data for post-analysis and monitoring

In the FPGA design, there are 2 acquisition buffers for post-analysis per channel, one is a small buffer (BufferAnalyse) and the second one is a big buffer (RawData).

Both buffers could be requested by the operator respectively with AskForDebug and AskForRawData inputs data.

The BufferAnalyse buffer will only be filled up when the acquired data will be below the AnalyseThreshold. It has a maximum size of TOT_alarm samples (2 bytes). As TOT_alarm is 250 bins (250 x 4ns = 1µs), it means 500 bytes per channel.

The circular buffer RawData will always be filled up with raw data from ADC. As said before, this buffer could be uploaded after an operator request. But it will also be automatically uploaded after an AlarmMPS. As we want to save pre-trig data during 10 µs and post-trig data during 3 ms (thus a total of 3010 µs), it means 752500 samples (2 bytes), thus about 1.5 Mbytes per channel.

To these 2 post-analysis buffers, we add other useful data (see Figure 13), thus the BufferAnalyse becomes the DebugBuffer and the RawData becomes the PostAlarmBuffer. The maximum size of the buffer DebugBuffer is 250 x 2 words + 4 words = 508 bytes per channel. The size of the buffer PostAlarmBuffer is 752500 x 2 words = less than 3 Mbytes per channel.

For monitoring we will also need to display analysis results. For example we want to know the average of the number of neutrons detected per microsecond. These data will be archived.

The following table describes the post-analysis buffers and monitoring data to implement:

Outputs	Description	Type	Range
DebugBuffer	Composed with BufferAnalyse buffer (TOT_alarm maximum size), a TOT value (in number of bins), a Charge value (see charge in Figure 10), an AmpMax value (maximum amplitude of the peak), and the number of neutrons/events detected inside the BufferAnalyse	Array of integer	TBD
AverageCounter1us	Monitoring data indicating an average of the number of neutrons detected per microsecond. The average is done every 500 ms.	integer	TBD
PostAlarmBuffer	Composed with RawData buffer associated with ResetCounter1us and NeutronsAlarm informations. The ResetCounter1us and NeutronsAlarm give informations when the counter of neutron on 1µs is reset and when neutrons alarms occur if there are.	integer	TBD
SaturationWarning	Monitoring data to indicate when saturations occur	Boolean vector	TBD
NeutronsAlarm	Monitoring data to indicate when neutrons alarms occur	Boolean vector	TBD

4.5. FPGA algorithm for neutron detection for one channel

The algorithm specification is the same for fast or slow detectors and the same for the 8 channels.

The FPGA design is described in the *Figure 13*. Each block could be an FPGA IP.

This algorithm has been coded in python program. Thus to check the final FPGA implementation, we could compare results from the FPGA to the python script.

The python script “simu_fpga_algo.py” is delivered by ESSI.

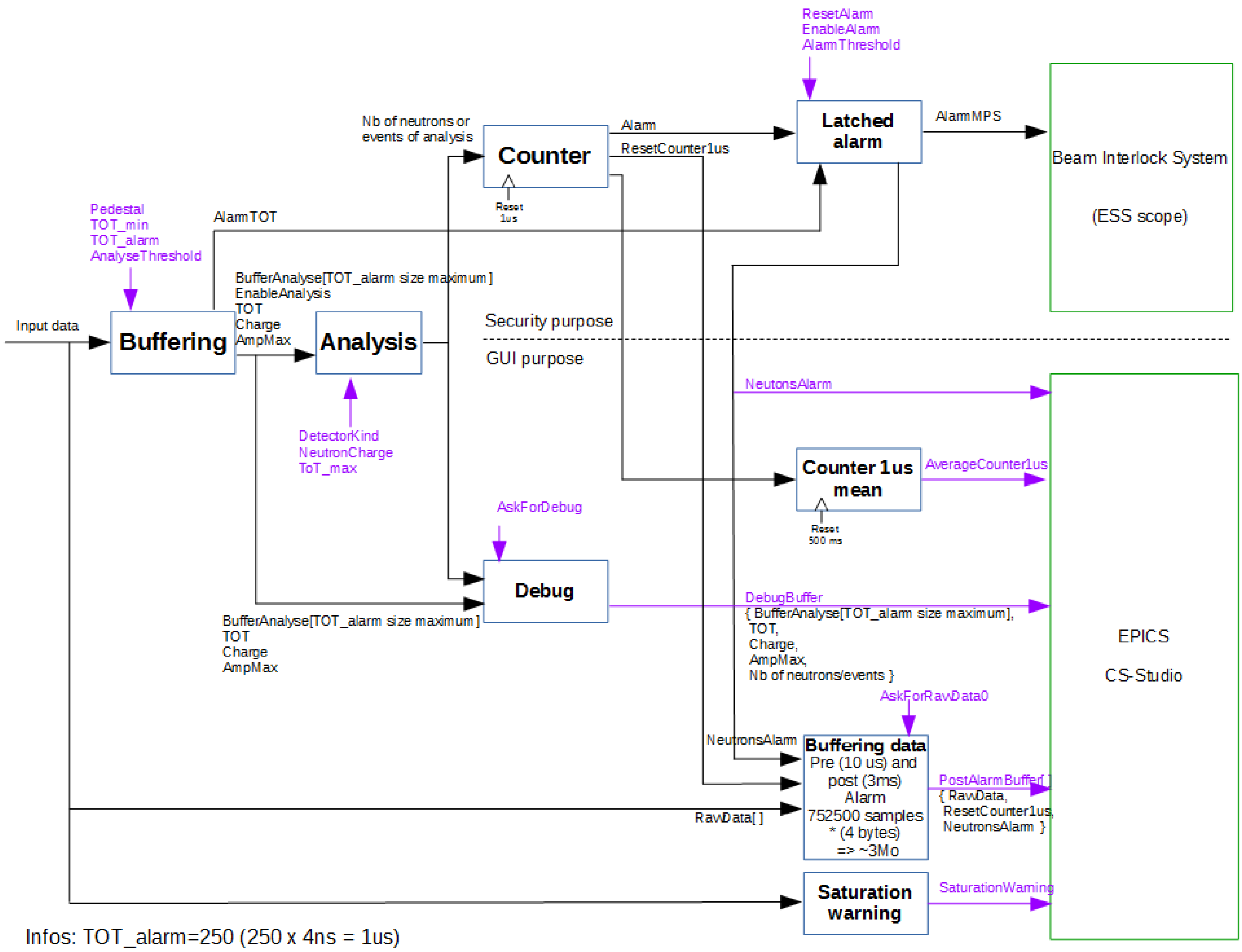


***Figure 13:** Functional FPGA diagram (slow or fast detector) for one channel (1 detector)*

The next table describes the logic of each block in the Figure 13. These blocks are also described in flowcharts in Appendix A.

Remark : input = Input data – Pedestal

	Buffering
when	input < thresholdAnalyse
until	input >= thresholdAnalyse or (input < thresholdAnalyse & TOT >= TOT_Alarm)
computing	Charge, AmpMax, TOT, BufferAnalyse, EnableAnalysis (end of bufferization)

	(Buffer) Analysis
when	Analysis trigged by Buffering (EnableAnalysis)
until	end of computing
computing	when TOT < TOT_min: AnalyseForNothing++
	when TOT >= TOT_min: UsefullAnalyse++, NbOfNeutrons+= Charge/NeutronCharge OR NbOfEvents+= roundingUp(TOT/TOT_max)

	(neutrons/events) Counter
when	reset every 1µs
until	always
computing	NbOfNeutronsPer1us += NbOfNeutrons (EPICS monitoring when detectorKind is SLOW)
	NbOfEventsPer1us += NbOfEvents (EPICS monitoring when detectorKind is FAST)

	Latched alarm
when	NbOfNeutronsPer1us > AlarmThreshold when detectorKind is SLOW
	NbOfEventsPer1us > AlarmThreshold when detectorKind is FAST
	or TOT >= TOT_alarm
until	User reset (EPICS)

	Saturation warning
when	Input >= 500mV or Input <= -500mV
until	Always
computing	Saturation = 1 Saturation = 0 after 1s if no saturation

	(EPICS) Counter 1us mean
when	Reset every 500 ms
computing	sum(NeutronsOrEventsPer1us)/500000

	Debug
when	AskForDebug (from EPICS) and analysis completed (EnableAnalysis)
until	Transfer ended
computing	Send the DebugBuffer (with BufferAnalyse[ ], TOT, Charge, ampMax, and NbOfNeutron or NbOfEvents)

	Buffering data Pre(10 us) and post (3 ms) Alarm
when	AskForRawData (from EPICS) or alarmMPS edge
until	Transfer ended
computing	Buffering 3,01 ms of rawData, resetCounter1us and NeutronsAlarm

	Warning
when	Sparks, calibration… TBD

Results of neutrons detection with real data set in input are displayed in Appendix B.

5. Control system

5.1. Control system test stand in Saclay

The upper layer will consist of one PC dedicated to development and GUI displays, the EEE server in ESS and another PC dedicated for the archiving Archiver Appliance.

The middle layer is dedicated to the EPICS servers or IOCs:

- An EPICS IOC is used for the fast acquisition (IFC1410)
- HV and LV are controlled by an SY-4527 B crate IOC (manufacturer IOC)
- An EPICS IOC is used for a Kontron Industrial PC linked to the PLC in charge of the gas management. The Protocol between the PLC and Kontron Industrial PC is S7.

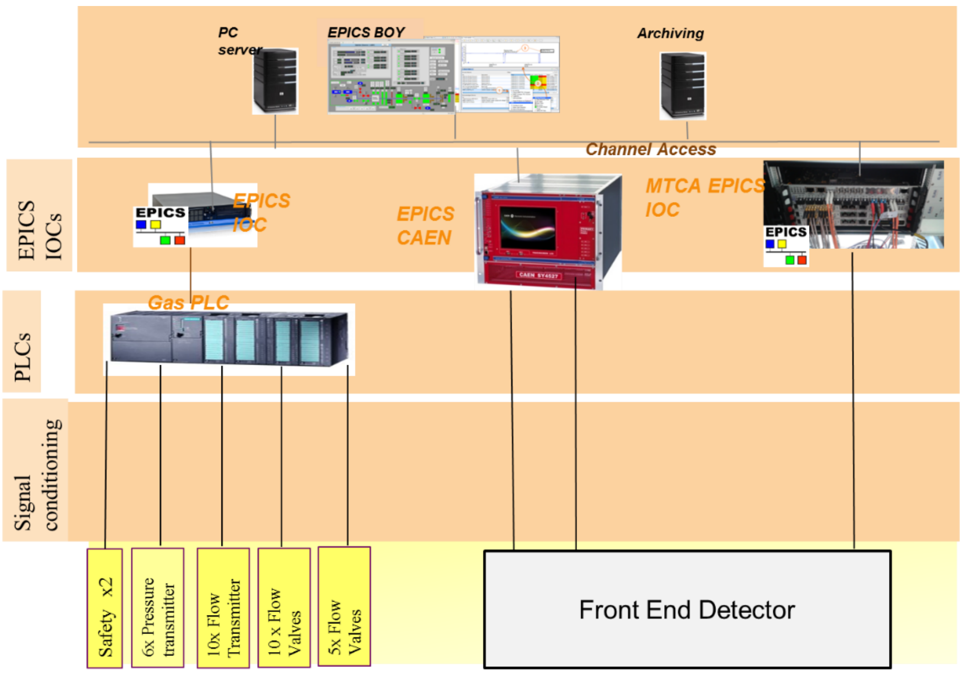


Figure 14: *nBLM control system overview*

5.2. Gas Control System

The gas system is controlled bya Siemens PLC, although we need to pass control parameters to the Control System via Channel Access. There will be **10 gas lines** in operation going from the gallery to the tunnel. They will be installed in parallel and each one will give gas to a group of detectors placed on different locations. Therefore, we are going to monitor the flow per line, and not per detector. The different functionalities are listed in the following table.

Analog input	
Per line • Pressure IN • Flow IN • Flow OUT System • Pressure IN • Flow IN • Flow OUT	Pressure reading : 4-20mA Flow reading : Profibus
Analog output	
Per line • Flow control valve	Profibus
Digital output	
Per line • Flow valve IN • Flow valve OUT System • Flow valve IN • Flow valve OUT	
Settings	
Per line • Flow setpoint • Flow min (warning) • Flow min (fault) • Flow max (warning) • Flow max (fault) • ∆Flow (warning) • ∆Flow (fault) • Pressure limit (warning) • Pressure limit (fault)	Flow min (warning)>Flow min (fault) Flow max (warning)<Flow maw (fault) Pressure limit (warning)<Pressure limit (fault)
System • Initialization time – First boot • Initialization time – Setpoint changed	Without warning and fault
Controllers	
ON/OFF • Per line • All the system Change bottles stack	
Status	
Per line • Ok status • Warning status • Fault status System • Ok status (if all lines = Ok) • Warning status (if at least 1 line = Warning) • Fault status (if at least 1 line = fault)	
Fault and warning	
Per line • Warning	Warning = no action on process Fault = close flow valve IN/OUT of the line

○ Flow IN > Flow max warning ○ Flow IN < Flow min warning ○ Pressure > Pressure warning ○ ΔFlow > ΔFlow warning • Fault ○ Flow IN > Flow max fault ○ Flow IN < Flow min fault ○ Pressure > Pressure fault ○ ΔFlow > ΔFlow fault System • Warning ○ Flow IN > Flow max warning ○ Flow IN < Flow min warning • Fault ○ Flow IN > Flow max fault ○ Flow IN < Flow min fault	

5.3. HV Control System

Each detector has two high voltages inputs: one for the mesh and one for the cathode. The controls for both are identical. All the channels are driven individually. With the IOC running on the CAEN crate, the user can:

- Enable the channel
- Get channel status
- Get channel voltage/current
- Get card and channel number

As an admin, you can also:

- Set the amplitude/current
- Set maximum voltage/current
- Set offset current
- Set the ramp-up and down voltage
- Set trip parameters
- Set network configuration

5.4. LV Control System

The FEE of several detectors are fed by the same low voltage module. In principle, we will group them by location (for example, all the detectors around DTL-1 with same low voltage line). Therefore, the controls will be per line and not per detector. The functionalities are the same as the HV.

Notice that for the low voltage, because of the length of the cables of ~50m, we need to give a higher value on the module to have ±5V on the detector (measurements on-going).

5.5. Acquisition Control System

A signal output from a detector (fast or slow) will be acquired, and the neutron counting will be done by the FPGA. The counting rate, in comparison with threshold values, will be used to reveal accidental (instantaneous increase) or continuous beam losses (long term increase), contributing to the safe operation of the accelerator.

In both cases, the FPGA must quickly send an alarm without waiting any order from the control system. In the continuous beam losses, the control system could send a warning before as it is a slow process.

In fact, the BEE control is used as a setting and diagnostic tool (with archiving) for the accelerator commissioning, the beam tuning and the long term monitoring of the beam.

To accomplish this work each nBLM group (see ***Figure 15***) has a BEE for the Control System with its IOC.

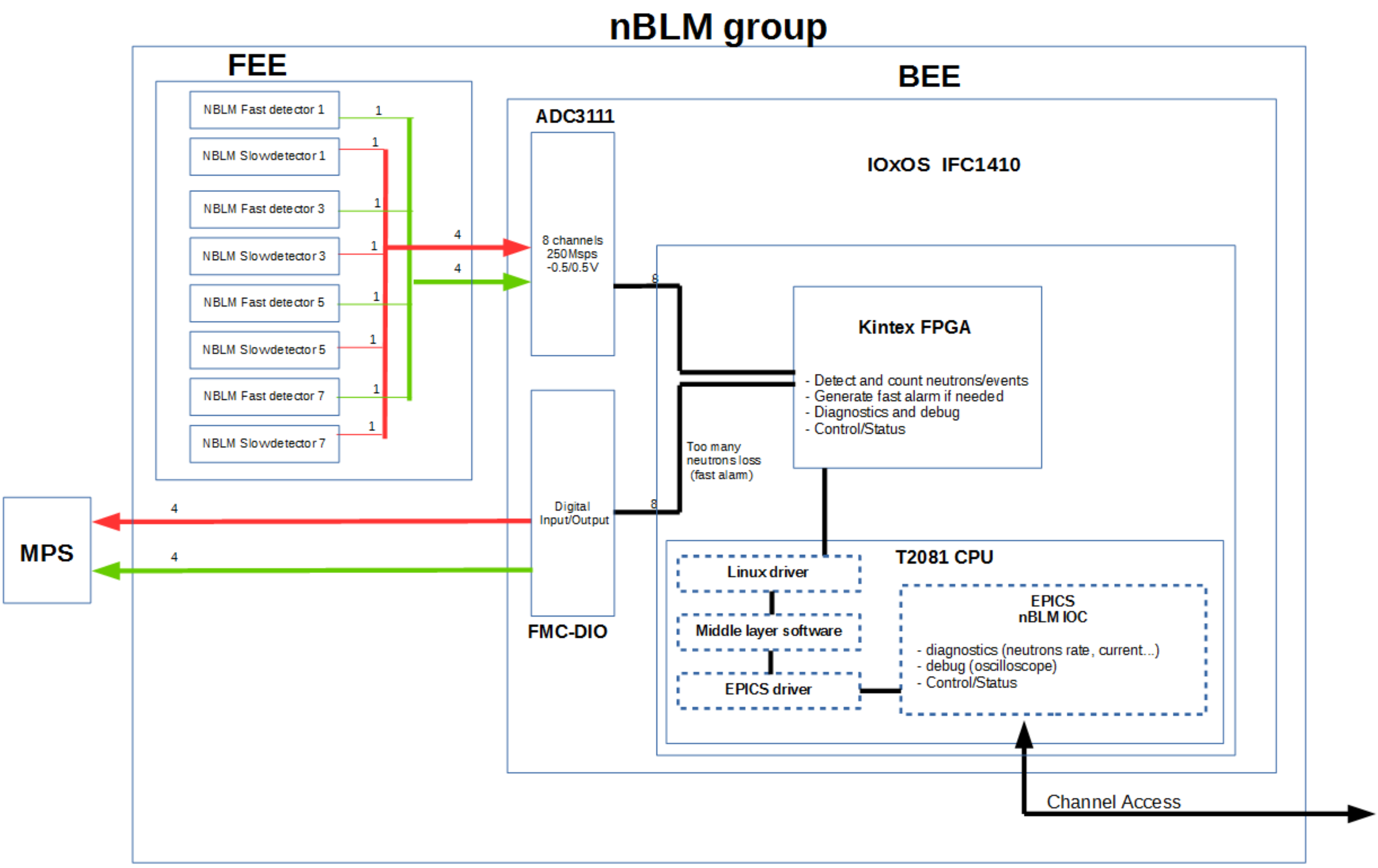


Figure 15: *nBLM group control system*

The IOC run on a Linux operating system inside the CPU T2081. The IFC1410 framework provided by ESS/ICS for the communication between the FPGA and the IOC is described in the ***Figure 16***.

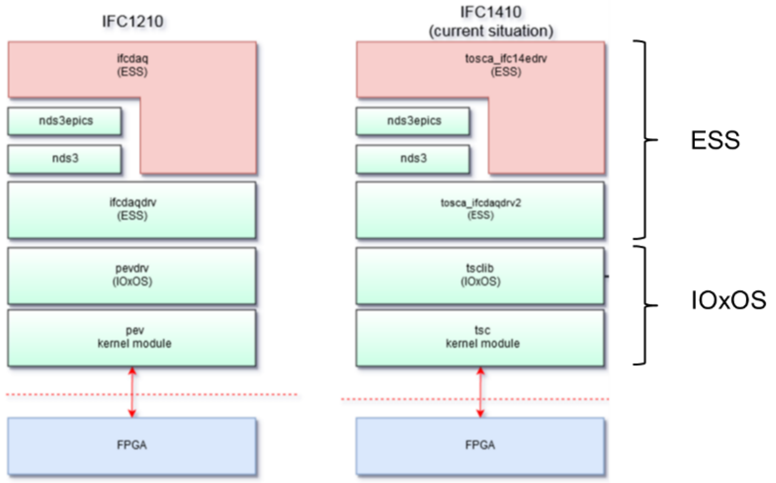


Figure 16: *IFC1410 Framework (similar to the VME IFC1210 framework)*

- Tsc kernel module and tsclib are Linux drivers: FPGA registers access via PCIe
- Tosca_ifcdaqdrv2 is a middle layer software: It purchases high level process function (set decimation, gain, etc.) for several kinds of FMC board
- Tosca_if14edrv is an EPICS driver. The nBLM IOC will use this driver

When the firmware FPGA will be modified with new registers, modifications must be applied in the middle layer software and in the EPICS driver to add new process function and new process variable. These modifications will be done by ESS-ERIC as defined in the agreement [4].

5.6. Beam Timing

The proton beam is pulsed. The protons pulse frequency is 14 Hz. Thus one proton pulse each 71.4ms. And the pulse duration is about 3.4 ms.

It means that in case of a beam loss it is probably enough to analyse about 3 ms post-alarm data.

We need correlation between post-mortem data and timing system for beam loss analysis.

6. EPICS specification

6.1. Channel access architecture

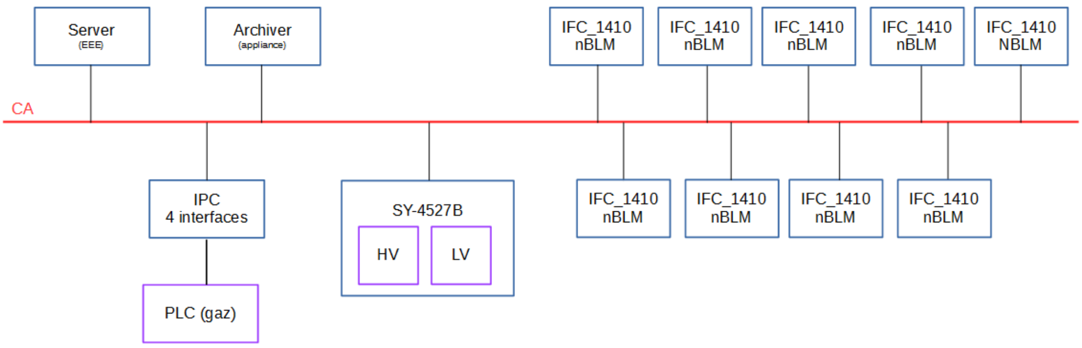


Figure 17: *CA Network architecture*

6.2. Functional EPICS database communication

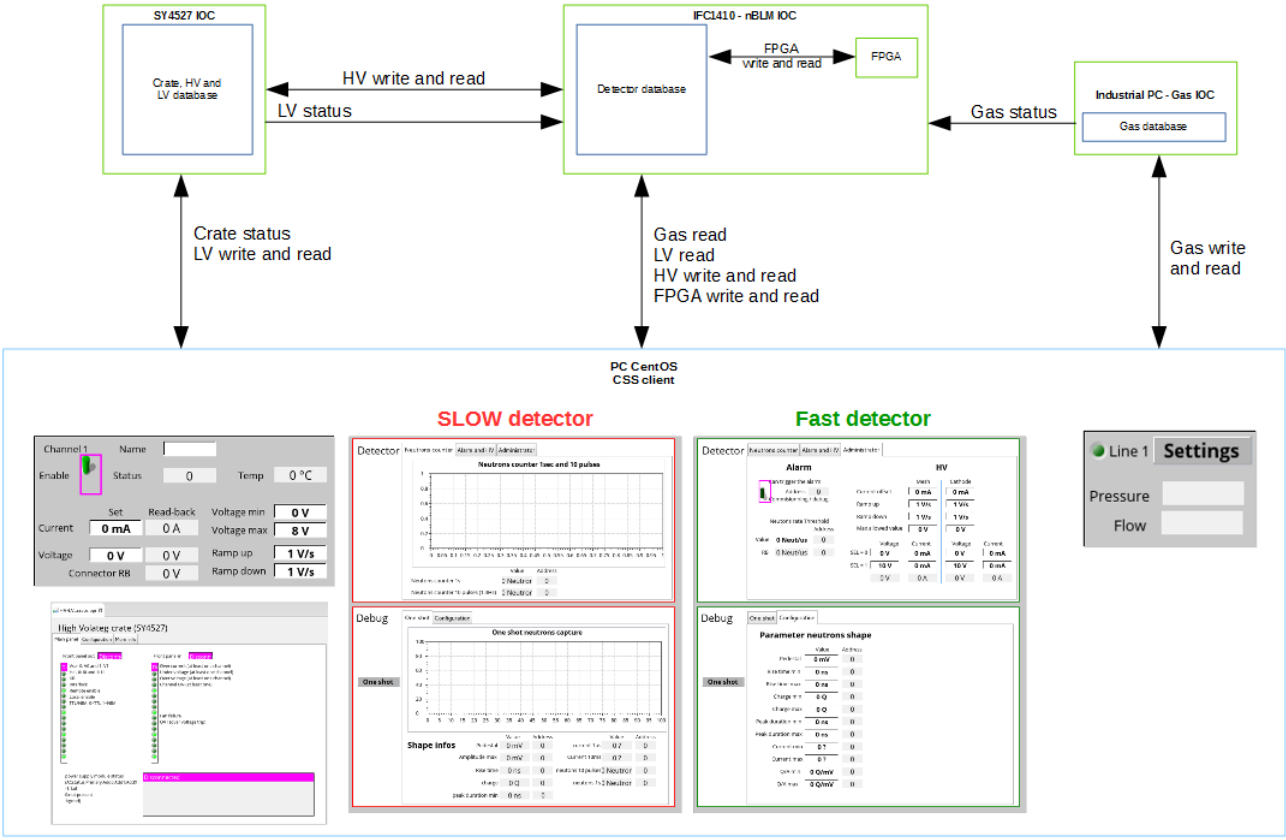


Figure 18: *Functional EPICS database communication for 2 detectors (fast + slow)*

Interlock status (HV, LV and gas status) go to the nBLM IOC. That means each nBLM IOC is aware of the state of its associated devices (gas, HV and LV) and decides to generate or not a global warning or an alarm status.

6.3. EPICS PVs lists (TBD)

6.3.1. PVs list for a nBLM detector (slow or fast)

Add archiving frequency

Name	EPICS input or output	Scalar or waveform	Number of elements	Update rate	Comment
DebugBuffer	input	waveform	508 bytes	on demand (with AskForDebug)	4ns / sample
AverageCounter1us	input	scalar	-	2 Hz (500ms)	Neutron counting for GUI information
PostAlarmBuffer	input	waveform	3 MBytes	on demand (with AskForRawData)/on alarm	4ns / sample

NeutronsAlarm	input	scalar	-	2 Hz (500 ms)	Alarm state for GUI information Vectors of 16 bits (1bit/channel)
SaturationWarning	input	scalar	-	2 Hz (500 ms)	Warning state for GUI information Vectors of 16 bits (1bit/channel)
MeanAmplitude	input	scalar	-	Once a month	Result of a script (see chapter 0)
MeanTOT	input	scalar	-	Once a month	Result of a script (see chapter 0)
EnableAlarm	output	scalar	-	on change	Vectors of 16 bits (1bit/channel) 1 : can't trigger the alarm, 0: can trigger the alarm This input, coming from GUI, can disable the alarm. In normal use alarms should be enabled.
ResetAlarm	output	scalar	-	on change	Vectors of 16 bits (1bit/channel) As each alarm is latched. It resets the alarm
AskForDebug	output	scalar	-	on change	Vectors of 16 bits (1bit/channel) It will keep the last BufferAnalyse and associated informations after the end of an FPGA buffering
AskForRawData	output	scalar	-	on change	Vectors of 16 bits (1bit/channel) It will keep the last RawData and associated informations
Pedestal	output	scalar	-	on change	Pedestal value to subtract the offset from the input signal
AnalyseThreshold	output	scalar	-	on change	For neutron detection algorithm. See chapter 4.4
TOT_min	output	scalar	-	on change	For neutron detection algorithm. See chapter 4.4
TOT_max	output	scalar	-	on change	For neutron detection algorithm. See chapter 4.4
TOT_alarm	output	scalar	-	on change	For neutron detection algorithm. See chapter 4.4
AlarmThreshold	output	scalar	-	on change	Activation of 1 W/m must be converted into neutron count rate When the neutron count rate (evaluates every µs) is over this threshold, an alarm is generated (if Enable alarm is 0)
NeutronCharge	output	scalar	-	on change	For neutron detection algorithm. See chapter 4.4
DetectorKind	output	scalar	-	on change	Vectors of 16 bits (1bit/channel) 0 : Fast detector 1 : Slow detector
Instrument enable control	input	scalar	-	on demand	enabled, disabled
Instrument enable status	output	scalar	-	on demand	enabled, disabled

Temperature	output	scalar	-	10 Hz	temperature of the instrument?
Gain	input	scalar	-	on demand	
Filter parameters	input	?	?	on demand	?
Self test	input	scalar	-	on demand	?
Self test status	output	scalar	-	on event	was self test successful?
Mode	input	scalar	-	on demand	normal, calibration
Heartbeat	output	scalar	-	on event	?
~~Threshold~~	~~input~~	~~scalar~~	~~-~~	~~on demand~~	~~gray/pulse~~
~~Activation limit~~	~~input~~	~~scalar~~	~~-~~	~~on demand~~	~~1 W/m (see ACC.SYR-20 Beam Loss Limit)~~

6.3.2. PVs list for the gas TBD

Name	Input or output	Scalar or waveform	Number of elements	Update rate	Comment
Pressure from bottles threshold: Pbottle th					
Flow set for the lines	input				
Flow threshold for the lines	input				
Flow set for the whole system	input				
Flow threshold for the whole system	input				
ΔF limit for leaks (absolute value)	input		1		
Pressure set for the lines					
Pressure threshold for the lines					
Pressure set for the whole system					
Pressure threshold for the whole system					

6.3.3. PVs list for HV and LV TBD

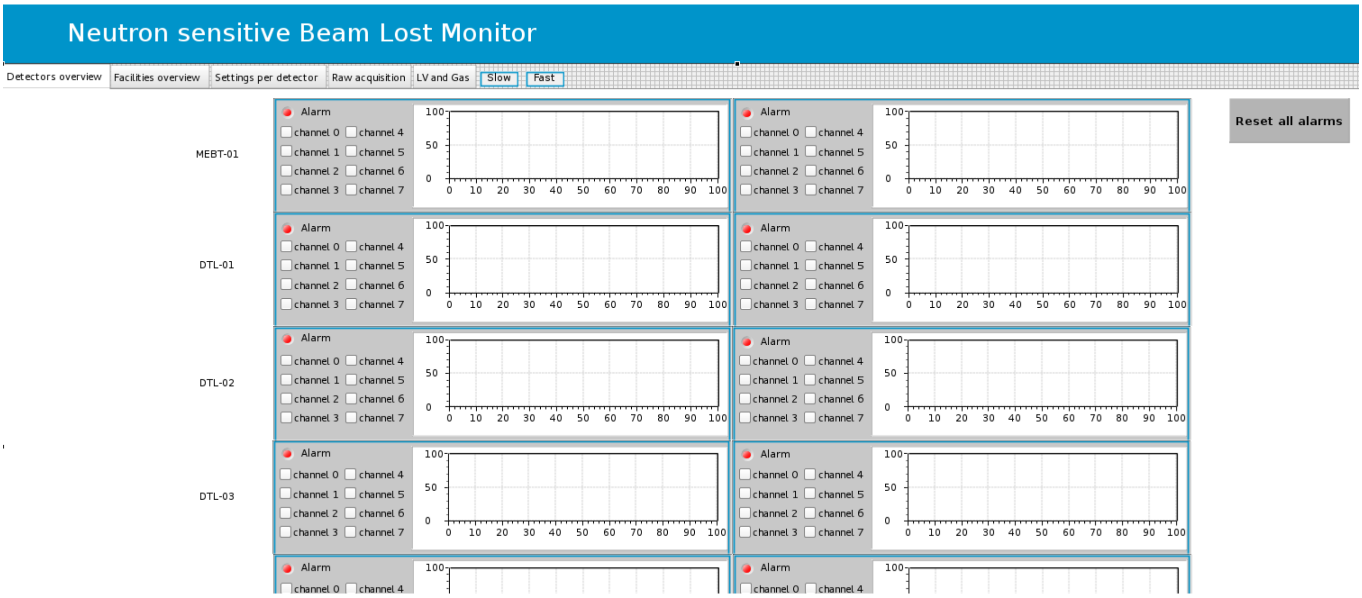
Name	Input or output	Scalar or waveform	Number of elements	Update Rate	Comment
High voltage PS control	input	scalar	-	on demand	on, off, standby
High voltage PS status	output	scalar	-	on event	on, off, standby, fault, over voltage, over current, ..
High voltage PS voltage	input	scalar	-	on demand	set voltage
High voltage PS voltage	output	scalar	-	1 Hz	actual voltage
High voltage PS current	output	scalar	-	1 Hz	actual current
High voltage PS temperature	output	scalar	-	1 Hz	temperature readout
Low voltage PS control	input	scalar	1		For 5 lines
Low voltage PS status	output	scalar	1		For 5 lines
Low voltage PS voltage	input	scalar	-	on demand	set voltage
Low voltage PS threshold	Input				For 5 lines
Low current control ?					For 5 lines
Low current threshold ?					For 5 lines

6.4. GUI for the CS of all nBLMs

4 main tabs:

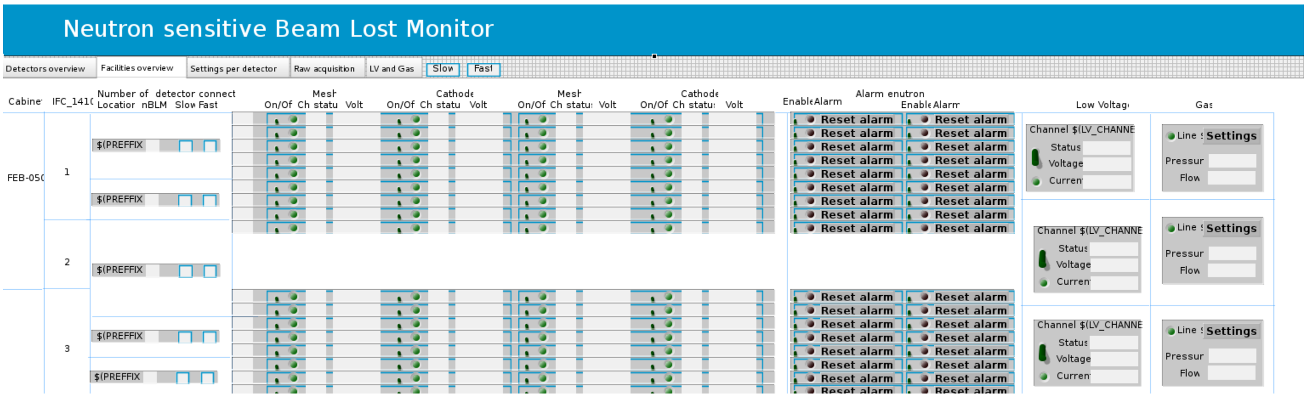
- Detector overview : global view of neutron counting and Alarms for each detector
- Device overview : global view of the 10 lines gas status, 7 LV channels and 164 HV channels
- LV and Gas settings
- Settings for a pair of fast and slow detector

Detector overview



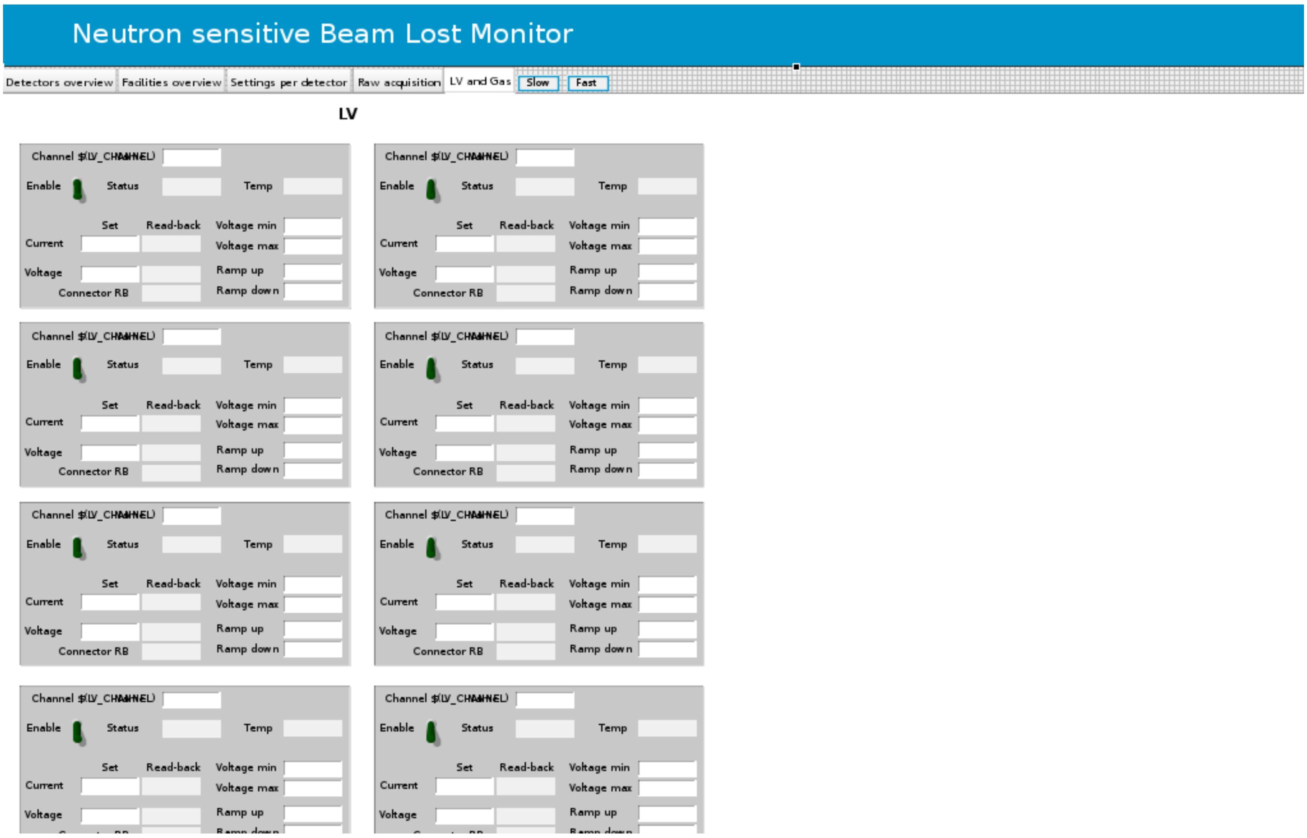
…

Device overview



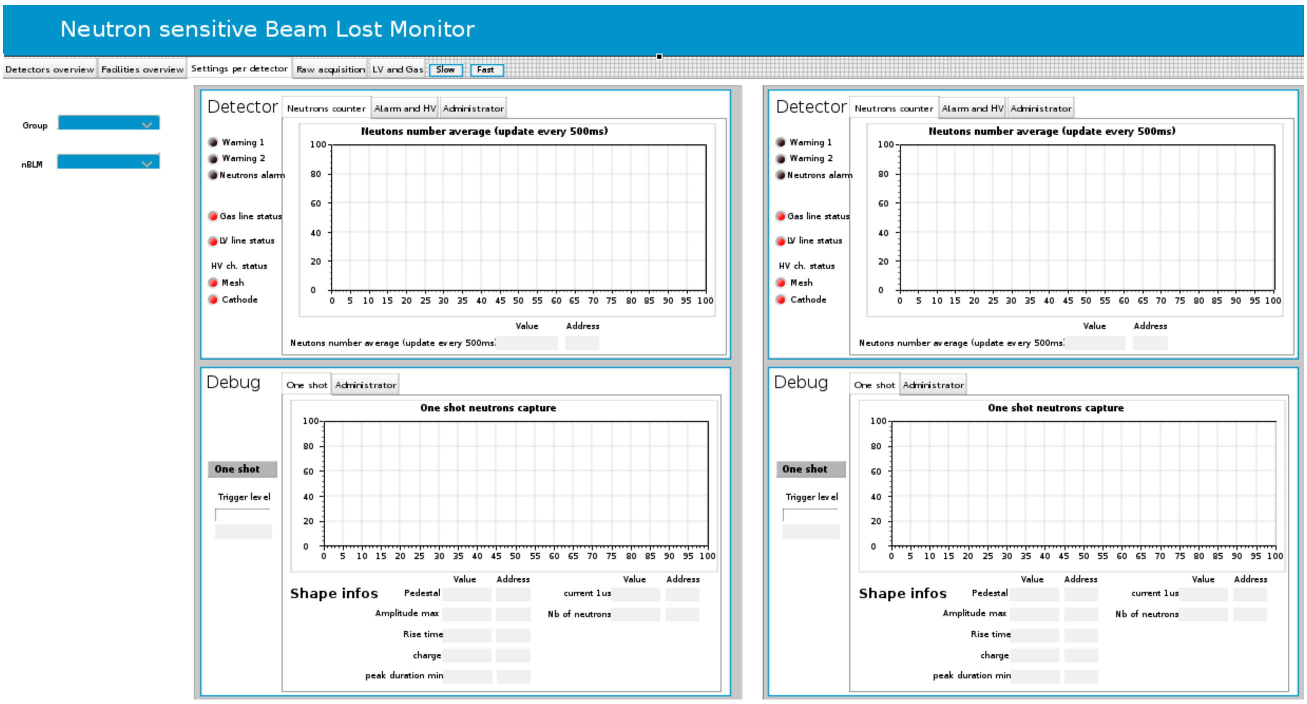
…

LV and Gas settings

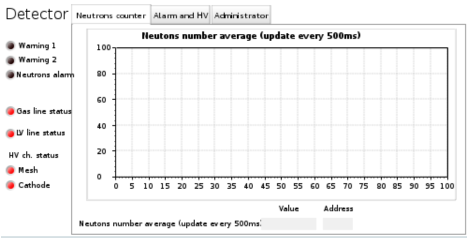


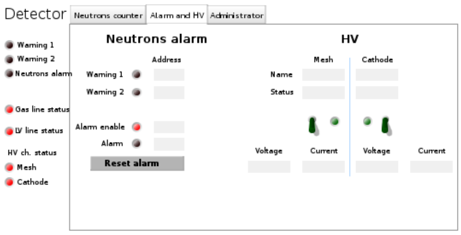
…

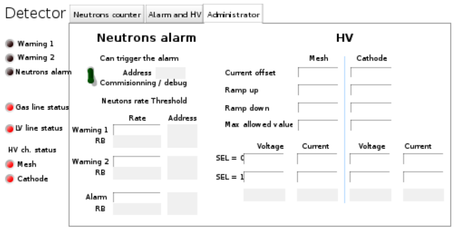
Settings for a pair of fast and slow detector

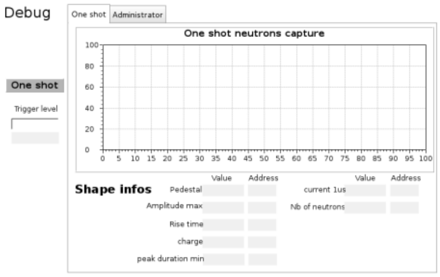


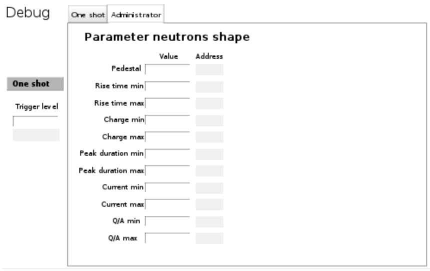
Inside the “Settings per detector” tab there are sub-tabs. The following figures are the view of these sub-tabs:











Expert mode: TBD

7. Malfunction/Stability

7.1. Detector stability

As said in chapter 4.2, we want to qualify each detector frequently to check if the neutron detection is still performant.

Thus adding an automatic script is advised, it will be frequently executed to check the detector stability over the time.

For the slow detectors, a script similar to the “calibration.py” (see chapter 9) could be executed. It will give distribution TOT histogram but also Amplitude and Charge histograms from the PostAlarmBuffer (saved thanks to the AskForRawData command). It will give the mean Amplitude, mean Q and mean TOT (for residual neutron inside the linac).

The CSS EPICS will trace the mean Amplitude and the mean TOT and will check the stability of these values over the time and the width of the distribution.

For the fast detectors, this checking is not applicable, thus we will count the events per second to verify the stability of the residual neutron rate inside the linac.

7.2. Malfunctions

See also document ref [5].

Possible malfunctions:

1. Gas flow problem (drop in, too much, slow leak)
2. No Gas
3. No pressure meter or flowmeter measure
4. Gas pollution
5. Front End Electronic failure
6. Back End Electronic failure: MTCA, IFC1410, ADC3111, FMC-DIO
7. HV/LV problem (crate or board failure, HV/LV down, HV/LV instability)
8. Detector problem (bad signal due to noise, sparks, ageing etc.)
9. Software/firmware crash
10. Bad/No detection settings

Continuous process to detect problems:

a. Gas flow and pressure monitoring
b. Remaining gas measure from pressure and flow
c. Mean Amplitude, mean TOT and width of the distribution stability over the time (see previous chapter)
d. Events/neutrons counter per second to verify the stability of the residual neutron rate (see previous chapter)
e. HV, LV Monitoring
f. IFC1410 alive counter

Possible actions:

A. EPICS or electrical alarm
B. EPICS warning
C. Watchdog

Moreover, the cross acquisition distribution (see chapter 3.4) limits the BEE failure impact.

8. Interface with MPS

There is one digital output alarm per detector. An alarm is set when the event/neutron number is over a rate for any detector (see chapters 4.4.3 and 4.5).

When a beam accident occurs, the MPS must receive a signal 5µs maximum after the accident.

Here is the timing budget of FPGA IPs concerned by the alarm production:

	Slow detector			Fast detector		
Steps	Normal case	Worst case	Comments	Normal case	Worst case	Comments
Response time detector	10000	300000	17 % neutons in 10µs	10	10	
Wire from detector to ADC	150	300	50 m cable = 150 ns delay	150	300	50 m cable = 150 ns delay
ADC delay	56	56	ADS42LB69 latency : 14cycles at 250 MHz	56	56	ADS42LB69 latency : 14cycles at 250 MHz
Buffering	200	1000	TOT_alarm = 1us	200	1000	TOT_alarm = 1us
Analysis/Counter	4	4		4	4	
Latched alarm (AlarmMPS)	4	4		4	4	
Total time (ns)	10414	301364		424	1374	
Total time (us)	10,414	301,364	0	**0,424**	**1,374**	

Thus the fast detector is compliant to answer to the requirement as the worst case is 1.374 µs.

The final decision to stop or not the beam will be taken by the BIS/MPS. The MPS can see all installed detectors as a system rather than individual devices. With the process variable, the MPS can locate the beam loss.

Moreover the use of coincidence between neighbouring detectors will increase the reliability of the system, by suppressing false triggers due to detector malfunction (discharge etc.).

9. Detector simulation/setting (TBD)

The methodology to configure a detector (fast or slow) will be the following:

1. Make raw acquisition with the neutron source. It could be done with an oscilloscope or on demand with the AskForRawData parameter in the EPICS CSS.

2. Give the raw data to the python script named "calibration.py". The script will display the whole acquisition. It will also detect all maximum amplitude of peak below a predefine level and it will give a distribution histogram of TOT associated with amplitude mean.

   When the script has been run, with the global view we can evaluate the Pedestal.

Once the pedestal is set, we set the analyze threshold below the noise. Several windows of 2000 samples are displayed. These windows will be displayed with at least one peak below the predefine level. In these windows, for each peak an ideal Landau shape is also displayed. Thus parameters eta and sigma of the Laudau function could be adjusted to fit the peak.

The TOT histogram shows the distribution of the TOT, thus we could find the TOT for a neutron and the associated mean amplitude.

3. Enter mean amplitude, eta, sigma and the Pedestal parameters found with the "calibration.py" script into the script named "script simu_fpga_algo.py". Set the AnalyseThreshold, the TOT percentage, it will calculate the charge Q and the TOT for the corresponding AnalyseThreshold.
   TOT_alarm can be also set.

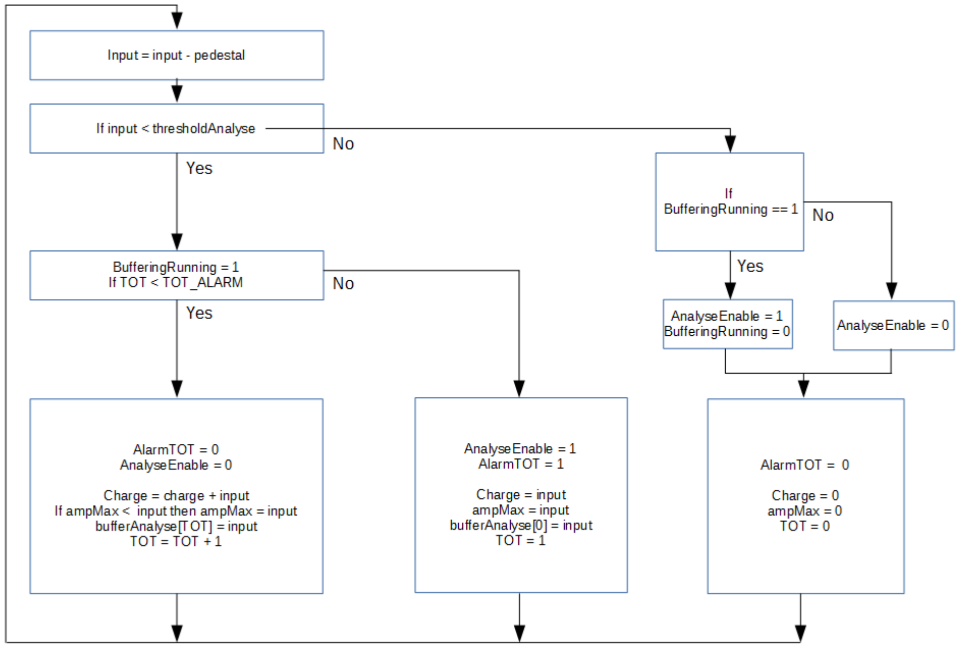
   With the same input raw data, we could see results of these setting.

4. If results are good then we set the same parameters in the EPICS CSS for the appropriate detector else we restart the procedure from 2.
   At least set the DetectorKind.

Appendix A

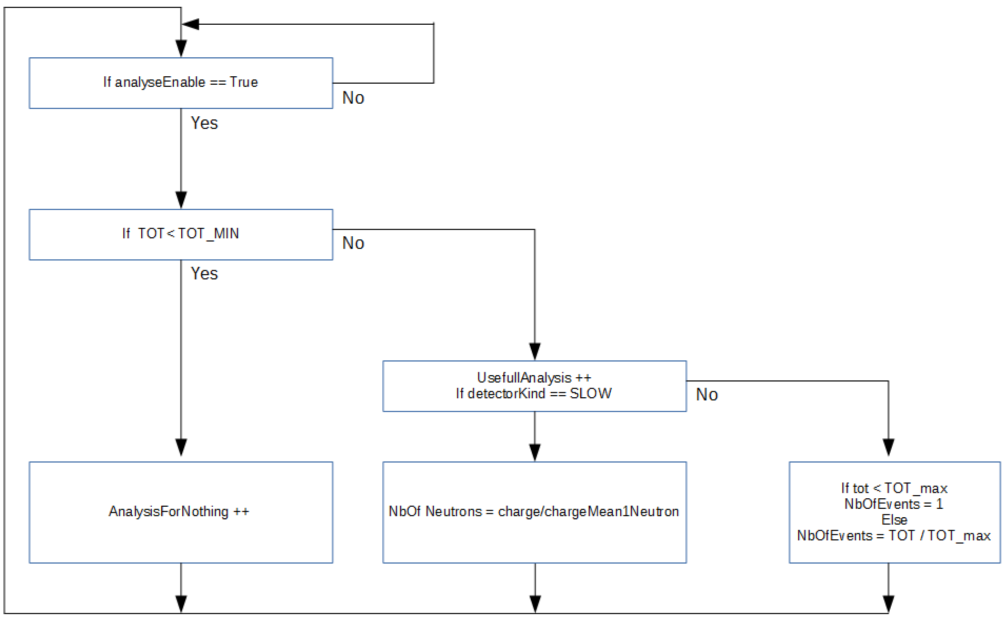
Flowcharts for each IP block:

FPGA algo bufferisation for analysis



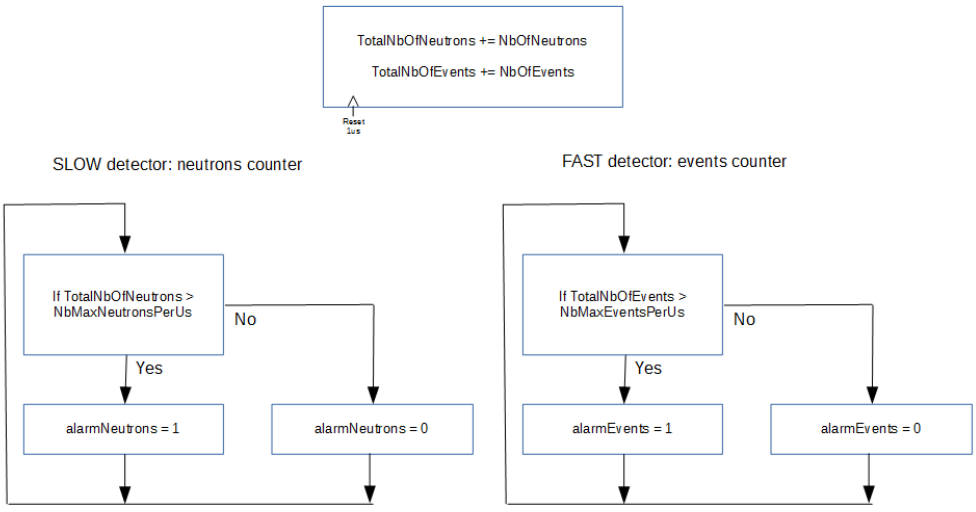
In this algo, all the "time" values are not in time scale but in samples, ex: TOT_ALARM = 100 (=Te * 100 ns)

FPGA algo analyse



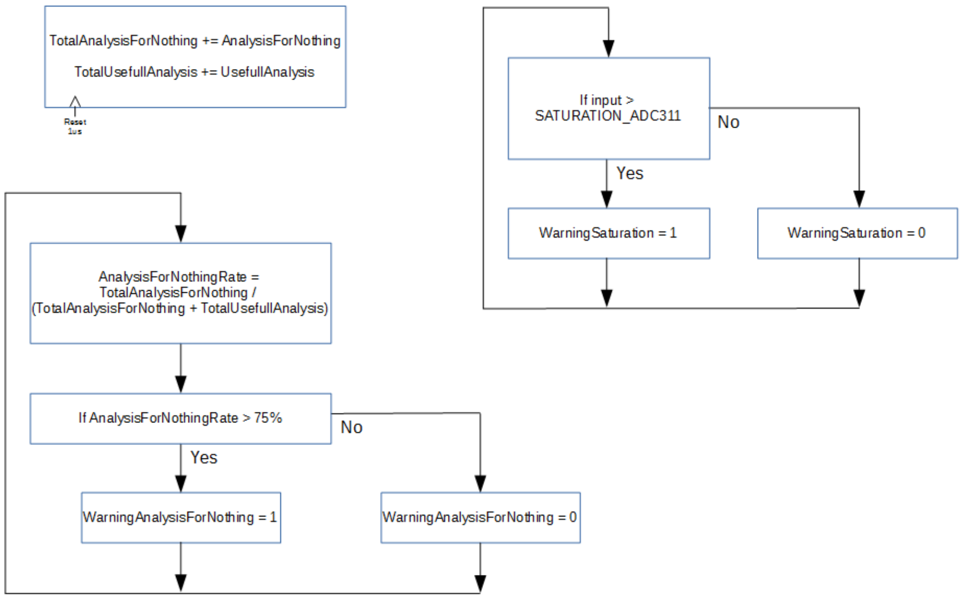
In this algo, all the "time" values are not in time scale but in samples, ex: TOT_ALARM = 100 (=Te * 100 ns)

FPGA algo alarms: neutrons and events counter



In this algo, all the “time” values are not in time scale but in samples, ex: TOT_ALARM = 100 (=Te * 100 ns)

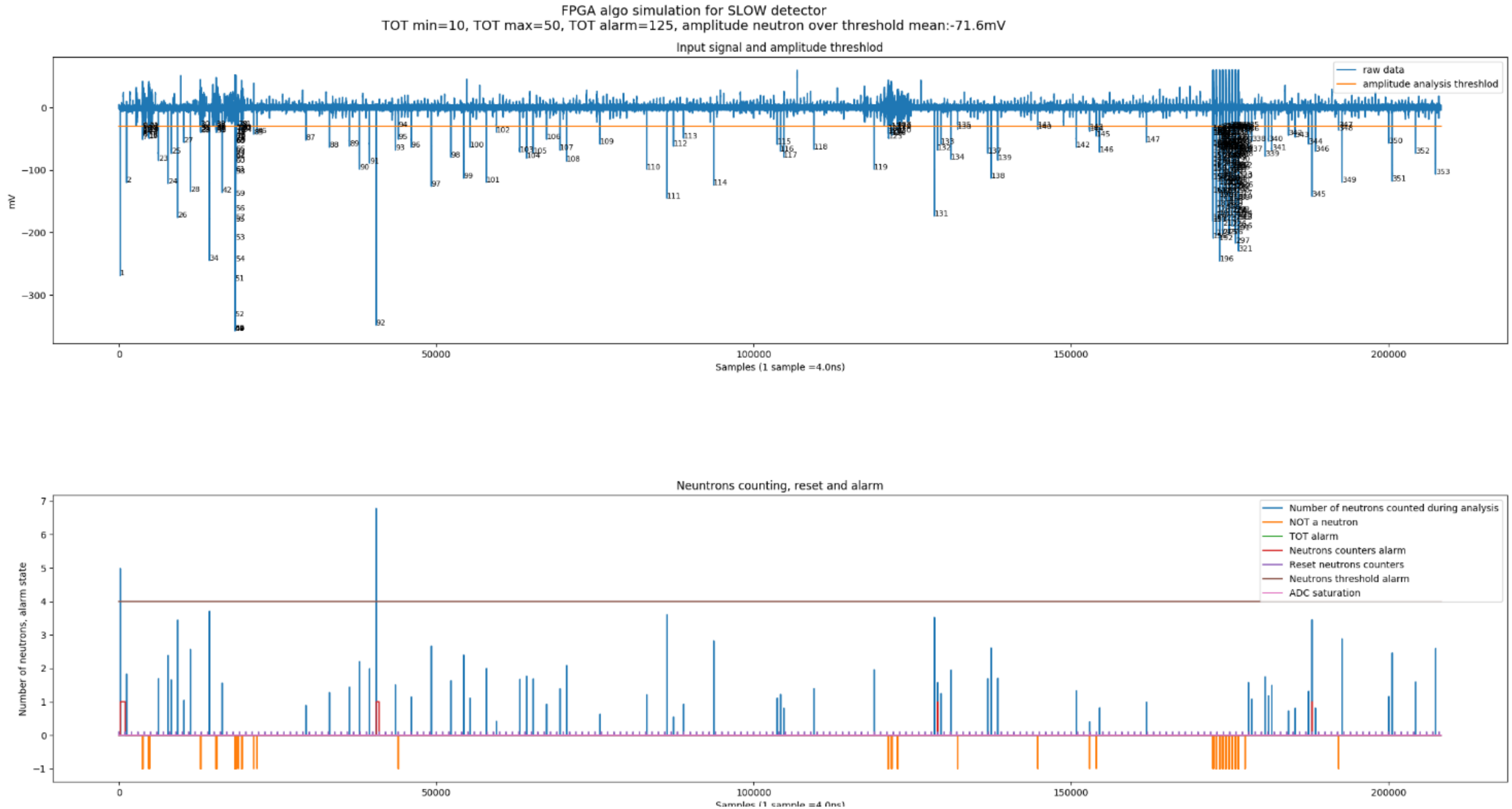
FPGA algo warning: rate analysis for nothing and channel saturation



In this algo, all the “time” values are not in time scale but in samples, ex: TOT_ALARM = 100 (=Te * 100 ns)

Appendix B

View of data acquired by a slow detector and associated results of the python algorithm:



Zoom around analysis number 42 and associated results of the python algorithm (we can see sparks around analyses number 43 to 78 and noise at the beginning and the end of the graph):