DEVELOPMENT OF THE THERMAL BEAM LOSS MONITORS OF THE SPIRAL2 CONTROL SYSTEM

C Haquin, GANIL, Caen, France
F. Negoita, IFIN-HH, Bucharest, Romania

Abstract

The Spiral2 linear accelerator will drive high intensity beams, up to 5 mA, to up to 200 kW at linac exit. Such beams can seriously damage and activate the machine! To prevent from such situation, the Machine Protection System (MPS) has been designed. This system is connected to diagnostics indicating if the beam remains under specific limits. As soon as a diagnostic detects its limit is crossed, it informs the MPS which will in turn take actions that can lead to a beam cut-off in appropriated timing requirements. In this process, the Beam Loss Monitors (BLM) are involved in monitoring prompt radiation generated by beam particles interactions with beam line components and responsible for activation, on one side, and thermal effects, on the other side. BLM system relies mainly on scintillator detectors, NIM electronics and a VME subsystem monitoring the heating of the machine. This subsystem, also called «Thermal BLM», will be integrated in the Spiral2 EPICS environment. For its development, a specific project organization has been setup since the development is subcontracted to Cosylab. This paper focuses on the Thermal BLM aspects and describes this development process.

BLM SYSTEM OVERVIEW

Requirements

From functional point of view the main requirement of the BLM system is to deliver a “BEAM STOP” signal to the machine protection system [1] in case the beam losses exceed the established thresholds (activation limits calculated correspond to 1 W/m losses for the 40 MeV deuteron beam).

But, from safety point of view, there are tough reliability requirements since the part of the BLM system responsible for generating the “BEAM STOP (Activation)” signal is classified as ”equipment contributing to personnel safety” (EPS). Consequently, BLM system architecture relies on robust devices, and well-tried techniques.

The Detectors

Detector design relies on BC430 plastic scintillator from Saint-Gobain, which has a good response to all kind of radiations and particles like protons, deuterons, heavy ions, neutrons, gamma and X radiations emitted in nuclear reactions induced by beam losses in structure materials of the accelerator. Then, scintillator is viewed by a Hamamatsu photomultiplier R329-02 (12 dynodes) polarized by a high-voltage power supply. A mu-metal shield is used to screen the photomultiplier against static magnetic fields present around the dipoles and quadrupoles used for beam steering. Finally, the discriminator converts the charge collected by the photomultiplier anode into a voltage pulse. For picking up the signal from noise a voltage comparator with variable threshold is used. The circuit delivers TTL logic pulses with variable width, depending on the time over threshold. The pulses are then converted into NIM pulses in order to be sent along a 30-50 m cable with 50 ohm characteristic impedance.

The pulsed logical NIM signals delivered by BLM detectors are used, for the activation monitoring, by a NIM alarm module specifically developed by IFIN-HH (see Fig. 1). This module consists in counters associated to each detector and periodically preloaded with a threshold. The “BEAM STOP” signal is generated as soon as one of the counters reaches zero.

A total number of 32 detectors will be installed in accelerator hall: one detector for each cryomodule (20 pieces) and 12 distributed along high-energy beam lines.

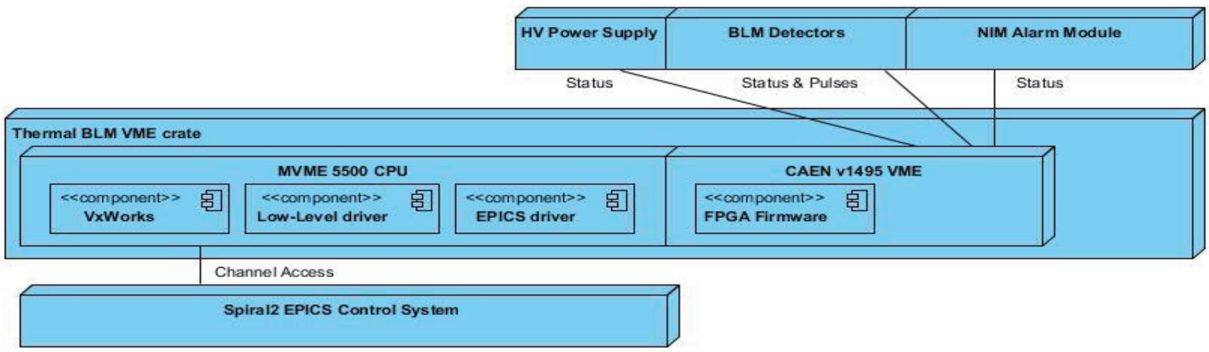

Figure 1: Deployment diagram of the BLM monitor system.

ISBN 978-3-95450-139-7

Copyright © 2014 CC-BY-3.0 and by the respective authors

THERMAL BLM ARCHITECTURE

Requirements

Whenever thermal effects exceed thresholds, a "BEAM STOP Thermal" signal must be raised for the MPS.

The logical NIM signals delivered by BLM detectors are also used for thermal effects monitoring but, in this case, the same response of BLM detectors corresponds to very different thermal effects as function of ion beam type and energy. Though this architecture also relies on counters periodically preloaded, thresholds values are different and must be changed for each beam (and each detector depending on its position along beam line). Furthermore, in order to convert BLM detectors counting rates into intensity of beam loss (or power of beam loss) a bunch of coefficients have to be computed also for each beam. All those thresholds and coefficients will be computed in advance, by a machine dedicated to parameters preparation, and will be loaded at run time.

It is clear that thermal effects monitoring requires a more flexible and powerful architecture than activation monitoring, luckily it is not EPS classified.

Finally, BLM Thermal monitoring must be fully integrated in the EPICS Spiral2 Control System.

Architecture

The VME solution chosen for the Spiral2 Control System is basically compatible with the needs of the Thermal BLM. So, a Motorola MVME 5500 CPU board is used but, for the first time in the Spiral2 project, we coupled it with a CAEN V1495 General Purpose VME board hosting a FPGA which also have enough input/output capabilities. Pulsed and status signals from detectors, NIM alarm modules status signals and High Voltage power supply status are connected to FPGA board input. The VME CPU interfaces the CAEN V1495 board on one side and communicates with the Spiral2 EPICS Control System via, the Channel Access protocol, on the other side (see Fig. 1). The functionality dispatch between the two boards is described below.

FPGA Functionalities

The first task of the firmware developed for the FPGA is to count the incoming pulses from the 32 detectors, and to raise the "BEAM STOP Thermal" signal as soon as one of the detector reaches one of the 2 level alarm, each alarm level having its own integration time.

The second task is to notify the VME CPU with a VME interrupt whenever a "BEAM STOP Thermal" signal is raised or periodically for the VME to retrieve counting rates and status bit registers.

Additionally, all the preload values for the counters, the alarm periods, monitoring period, expected state of each status signals must be made programmable at run time.

Finally, there is a Ready signal for the VME CPU indicating that the firmware is initialized and ready to be configured (see Fig. 2).

The development for the FPGA is the first part of the development subcontracted to Cosylab.

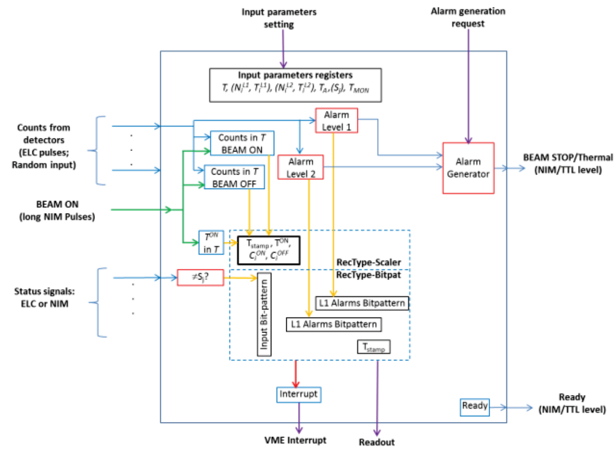

Figure 2: Block Diagram of FPGA processes.

CPU Functionalities

For the interface between VME CPU and the CAEN V1495 board, a low level driver, relying on the VxWorks operating system of the VME CPU, controlling the VME bus and allowing communication with upper software layers, was developed. It mainly aims at handling the Ready signal, transferring the programmable parameters and serving the interrupt raised by the CAEN V1495 board. This driver is the second part of the development subcontracted to Cosylab.

Next, VME CPU and the Spiral2 EPICS Control System interface, of course relies on an EPICS IOC in which a BLM application module was developed. The module consists in an EPICS Record database and a driver derived from asynPort driver [2].

This module is the last part of the development subcontracted to Cosylab and since it is the final link between "Thermal BLM" and the Spiral2 Control System we had to specify and verify it very precisely in order to make it compliant with our exploitation needs and Control System rules. In other words, we had to stick closer than usually to the V development cycle!

We adopted a very categorical and verifiable requirements specification approach, inspired from the classified EPS, coupled with UML modelling diagrams to describe the expected dynamic behaviour of the module and took advantage of the EPICS record formalism to specify the interface. This process turns out to be very efficient and useful very early in the project not only for us but also for our subcontractor in order to estimate the cost of the development.

Requirements are dispatched in Development, Environment, Deliverables, Interface, Functional, Implementation, Performance, Test categories (Tab. 1).

Table 1: Requirement Specifications Example

Identifier	Functional Requirement
BLM_SP2-SRS-F-46	$(EQPT):$(DID)_CoeffCons shall be used for CA clients to set each detector Transformation Coefficient

Copyright © 2014 CC-BY-3.0 and by the respective authors

The dynamic behaviour is specified mainly with state and activity UML diagrams (see Fig. 3).

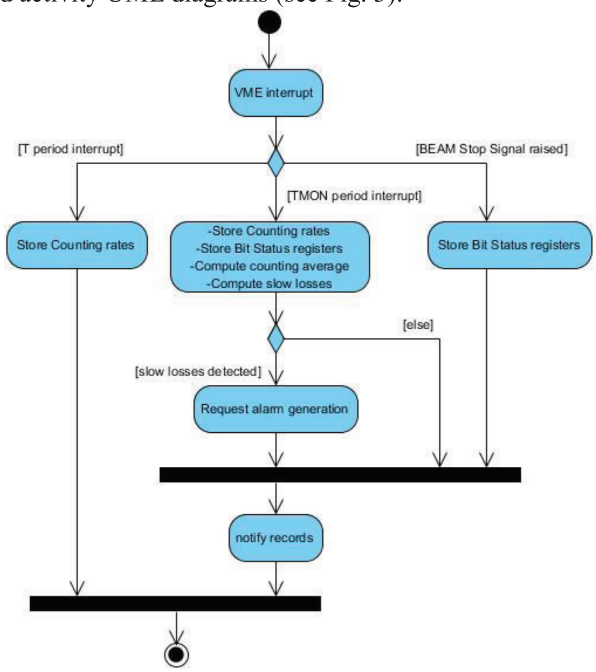

Figure 3: Activity diagram of CPU processes.

Thermal BLM System Functionalities

From end user point of view, the following functionalities are required:

- Setup functionality, to set the alarm thresholds, integration time for each alarm level, the monitoring period (T_{MON}) at which data are transferred for visualization, and also the expected state of the binary status of the detectors, NIM alarm modules and HV power supply.
- Readout functionality retrieves, for visualization, in real time at a T_{MON} rate, the counting rates, Alarms status and Status signals.
- Back Ground estimation/subtraction for providing the possibility to store the background counting rates when there is no Beam and to subtract it to the counting rates when there is actually the Beam.
- The Slow Losses functionality applies a specific algorithm to the counting rates so as to detect slow losses that may damage the machine by thermal effect. Additionally to the individual counting rate of each detector, which is efficient in detecting a localized loss causing overheating above the threshold, the slow losses algorithm is able to take into account the counting rates of several detectors and though each detector taken individually is below its threshold the whole loss may be too important. For this functionality, a matrix of coefficients and a bunch of thresholds must be set. At runtime, the calculated slow losses can be monitored at a T_{MON} rate.
- The Post Mortem functionality allows deep analysis of what just occurred when the BLM system raises one of the “BEAM STOP activation” or “BEAM STOP thermal” signals. In such a case, the latest counting rates that are actually stored at a micro second rate between each T_{MON}, which is at a second rate, can be uploaded at application level.

The interface between all those functions and the user applications rely on standard EPICS records and the Channel Access protocol and follow the rules defined by the Spiral2 Standard Interface [3].

PROJECT ORGANISATION

The whole BLM system of the Spiral2 project is a deliverable under the responsibility of IFIN-HH, Bulgaria, laboratory. Only the specification of the software running on the VME CPU is handled by the Spiral2 Control Command group, from GANIL, to ensure the compliance with the Spiral2 control system. The software development on both FPGA side and VME side is subcontracted to Cosylab. For this purpose, a specific development platform was setup including VME crate and boards but also a development station installed with the EPICS distribution in its Spiral2 version [4] and the so called “topSP2” Spiral2 development environment. Additionally, a SVN repository was setup with access for Cosylab to backup and deliver the developments.

The project has been splitted in two phases, the first phase include FPGA development, VxWorks low level driver and a first shot of the EPICS driver based on the asynPort driver including only Setup and Readout functionalities. First phase development is finished and tests are ongoing with promising results. For the second phase, EPICS driver specification must be finalised then FPGA notification mechanism will be optimised and the remaining Back Ground, Slow Losses, Post Mortem functionalities will be implemented.

The deadline for this project is mid 2014, it’s a tight schedule [5] like in every project but we’ll make it!

REFERENCES

[1] C. Berthe et al, “Machine Protection System for the SPIRAL2 facility”, MOPPC042, ICALEPCS 2013, to be published.

[2] http://www.aps.anl.gov/epics/modules/soft/asyn/R4-12/asynPortDriver.html

[3] C. Haquin et al. “Spiral2 control command: a standardized interface between high level applications and Epics IOCs”, Icalepcs 2011, Grenoble, France, October 2011.

[4] D. Touchard et al, “The Spiral2 command: control software organization and management“, Icalepcs 2009, Kyoto, Japan, October 2009.

[5] C. Haquin et al, “The Spiral2 control system progress towards the commissioning phase”, MOCOAAB03, ICALEPCS 2013, to be published.

Copyright © 2014 CC-BY-3.0 and by the respective authors

ISBN 978-3-95450-139-7