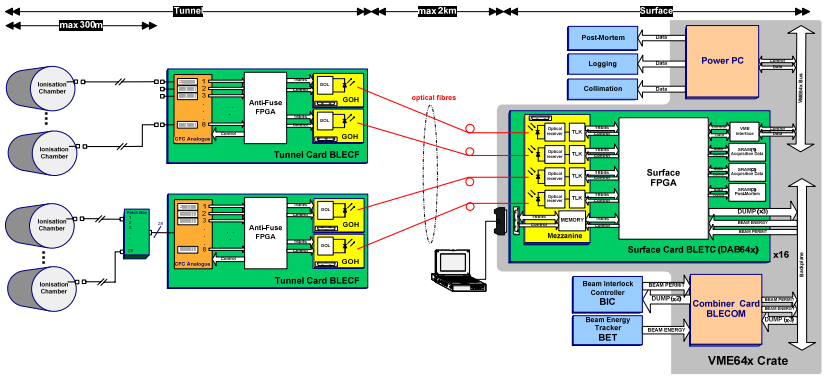
An FPGA Based Implementation for Real-Time Processing of the LHC Beam Loss Monitoring System's Data.

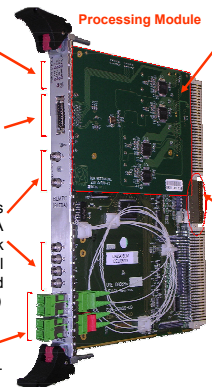
C. Zamantzas, B. Dehning, E. Effinger, J. Emery, G. Ferioli

CERN, 1211 Geneva 23, Switzerland

Abstract: The strategy for machine protection and quench prevention of the *Large Hadron Collider* (LHC) at the *European Organisation for Nuclear Research* (CERN) is mainly based on the *Beam Loss Monitoring* (BLM) system. At each turn, there will be several thousands of data to record and process in order to decide if the beams should be permitted to continue circulating or their safe extraction is necessary to be triggered. The processing involves a proper analysis of the loss pattern in time and for the decision the energy of the beam needs to be accounted. This complexity needs to be minimized by all means to maximize the reliability of the BLM system and allow a feasible implementation. Processing data in real-time requires dedicated hardware to meet demanding time or space requirements where performance is often limited by the processing capability of the chosen technology. Modern field programmable gate arrays (FPGAs) include the resources needed to design complex processing and can be reprogrammed making them ideal for future upgrades or system specification changes. In this paper, an FPGA based implementation is explored for the real-time processing of the LHC BLM data. It gives emphasis on the highly efficient *Successive Running Sums* (SRS) technique used that allows many and long integration periods to be maintained for each detector's data with relatively small length shift registers that can be built around the embedded memory blocks.

BLM System Overview





LEDs showing system and communication status

Connector to program the FPGA & Configuration devices. It gives additional access to 9 FPGA's I/Os.

LEMO connectors for accessing FPGA I/Os. (Two clock inputs, two general purpose inputs and two similar outputs.)

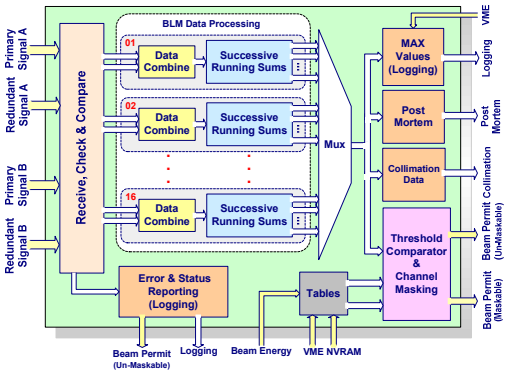
E2000-APC input connections to four optical fibres.

The **BLM Mezzanine** is hosting the receiver parts for four optical links. It handles the de-serialization and decoding of the four optical gigabit data transmission lines in parallel and provides the received data to the DAB64x card's FPGA device for processing.

The **P0** connector is using a custom-made backplane on the VME64x crate to daisy-chain, through each of the processing modules, the two beam permit lines and provide the beam energy data input.

Around 4000 *Ionization Chambers* are the detectors of the system. Tunnel cards, called *BLECFs*, acquire and digitise the data from the detectors and transmit those at the surface using *Gigabit Optical Links*. There, the data processing cards, named *BLETCs*, receive those data and decides whether or not the beam should be permitted to be injected or continue circulating. Each surface card receives data from 2 tunnel cards, which means that it can treat up to 16 channels simultaneously, and provides data for on-line display and post-mortem analysis.

Surface FPGA's Processes



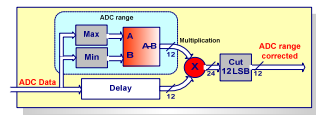
Between the blocks responsible for the correct reception and the comparison, with the relevant for the channel and the beam energy threshold values, is the BLM data real-time processing.

In the configuration chosen, **the system is able to treat 16 channels in parallel and maintain 12 integration periods for each of them spanning in lengths from 40µs up to 84s.**

In order to achieve nine orders of dynamic range (the acquisition) is making use of both Current-to-Frequency Converter (CFC) and ADC circuitries, and merges those data subsequently.

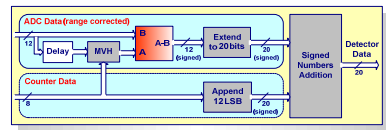
For supervision and logging purposes, data will be sent over the VME-bus for on-line viewing and storage by the Logging and Post-Mortem systems. **The BLM system will drive an online event display and write extensive online logging** (at a rate of 1 Hz) **and post-mortem data** (the acquired data from the last 20,000 turns plus averages of the same data of up to the last 40 minutes) **for offline analysis**. Finally, for the Collimation system and the correct **alignment and setup of the collimators** one more set of data is available. Those data contain whenever requested the last 20.48 ms of acquired data in the form of 32 x 640 µs sums for each detector.

CFC & ADC Data Combine



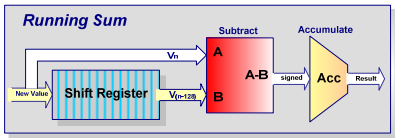
On the beginning stage, the **ADC value is normalised** by its effective range. The min and max of the ADC values received are continuously calculated. Their difference signifies the effective range of the ADC circuitry and is used to normalise each received value.

The two types of data acquired for each detector are of different type and a pre-processing is needed in order those to be combined seamlessly. The measurement of the frequency produced by the Current-to-Frequency Converter with a counter relates to the current accumulated between the last acquisitions. On the other hand, the voltage measured by the ADC is the fraction remained between the last count and the first from the next acquisition.

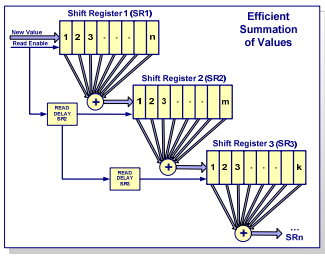


In order to combine those data the difference of the last two ADC measurements is calculated. It corresponds to the counter fraction of the last 40 µs and thus could be **added to the counter value**. Of course, since the difference could be a negative number, signed number arithmetic is used for the addition.

Basic Principles Used



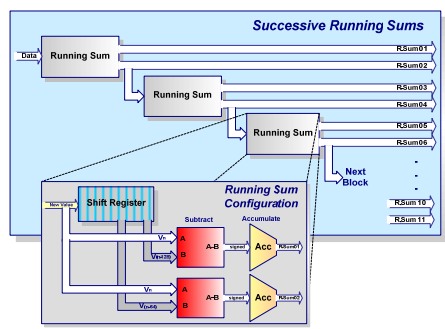
The **Running Sums** can be produced simply by adding the newly acquired value to a register and subsequently subtracting from it a value coming delayed by a number of cycles in a shift register. A similar configuration, but more efficient, would be to **add the difference of those two values**, the new and the delayed, **to an Accumulator** using signed numbers notation.



Additionally, **long histories** of the acquired data are needed for the construction of long moving windows. The technique employed to reach long integration periods with relatively small in length shift registers, that overcomes the storage problem of long histories of the acquired data, is tackled by the **consecutive storage of sums of the received values**.

Successive Running Sums

In this technique, the storage problem of long histories of the acquired data, which are needed for the construction of long moving windows, is tackled by the cascading of multiple moving windows. Thus, **the already calculated running sums are used** in order **to calculate bigger in length running sums** which translates to a huge data reduction and resource sharing.



Furthermore, in the design proposed, the sum of the Shift Register contents is always kept and updated in the Running Sums. Thus, **the RS output is also used directly to feed the following stage's input**.

Finally, **multipoint shift registers** are being employed whose feature can be effectively **used to combine overlapping memory contents**, therefore minimising even more the resource utilisation.