# A COMPACTRIO-BASED BEAM LOSS MONITOR FOR THE SNS RF TEST CAVE*

W. Blokland, ORNL, Oak Ridge, TN, U.S.A.
G. Armstrong, Maverick Systems, Kingston, TN, U.S.A.

## *Abstract*

An RF Test Cave has been built at the Spallation Neutron Source (SNS) to be able to test RF cavities without interfering with the SNS accelerator operations. In addition to using thick concrete wall to minimize radiation exposure, a Beam Loss Monitor (BLM) must abort the operation within 100 usec when the integrated radiation within the cave exceeds a threshold. We chose the CompactRIO platform to implement the BLM, based on its performance, cost-effectiveness, and rapid development. Each in/output module is connected through an FPGA to provide point-by-point processing. Every 10 usecs, the data is acquired, analyzed, and compared to the threshold. Data from the FPGA is transferred using DMA to the real-time controller, which communicates to a gateway PC to talk to the SNS control system. The system includes diagnostics to test the hardware and integrates the losses in real-time. In this paper, we describe our design, implementation, and results.

## INTRODUCTION

In particle accelerators, Beam Loss Monitors (BLMs) are critical devices used to protect the machine and personnel. The BLM must operate reliably and in real-time. BLM mostly use ion chambers or scintillators to detect the radiation due to beam losses or high power RF. The signal from these detectors must be acquired and processed in real-time to calculate the losses during the beam pulse and decide whether to abort the beam.

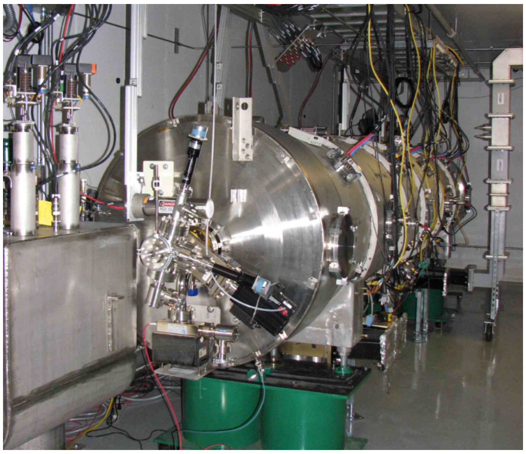


Figure 1. The SNS RF Test cave.

The goal of the RF Test Cave, see figure 1, is to test the SNS Superconducting Cavities, [1]. Radiation from the RF must be measured to ensure proper operation of the RF system and protect the machine and people.

This paper focuses on that part of the BLM that implements the data-acquisition and the signal processing.

A typical implementation is VME-based with a real-time operating system such as VxWorks as is the current SNS BLM system, [2]. As technology advances, other techniques have become available. In particular, National Instruments now provides FPGA processing in combination with a real-time controller, PXI or CompactRIO based. The cost of the CompactRIO system, including the Real-Time controller and IO modules is about $10k for 12 channels (this can be extended to 16 channels for $500 dollars), or about $850-$700 per detector. A rackmounted PC of about $2k is needed to function as a gateway to EPICS. The CompactRIO system has an FPGA in its backplane to talk to the IO modules, process the data, and communicate the results to the cRIO Real-Time controller.

The RF Test Cave has 10 detectors and measures only cavity radiation due to the RF, since no beam is being accelerated.

## IMPLEMENTATION

The signals from the detectors are amplified and wired to 3 simultaneous sampling modules with each 4 channels of 16-bit, 100kHz digitizers. The HV power supply can be controlled through the Analog Out (AO) module. The HV readbacks are acquired through Analog In (AI) module. A Digital Input Output (DIO) module with bidirectional IO lines is used to detect the beam trigger to start taking data, and also to output a beam abort signal. A second DIO module is used to turn LEDs on or off, to test the PMTs by producing light in the scintillators. The system with EPICS Gateway is shown in figure 2.

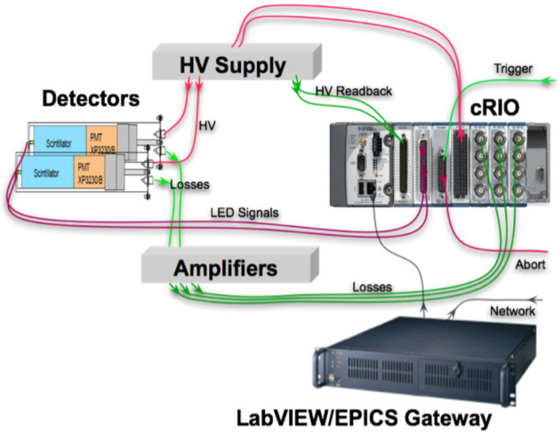
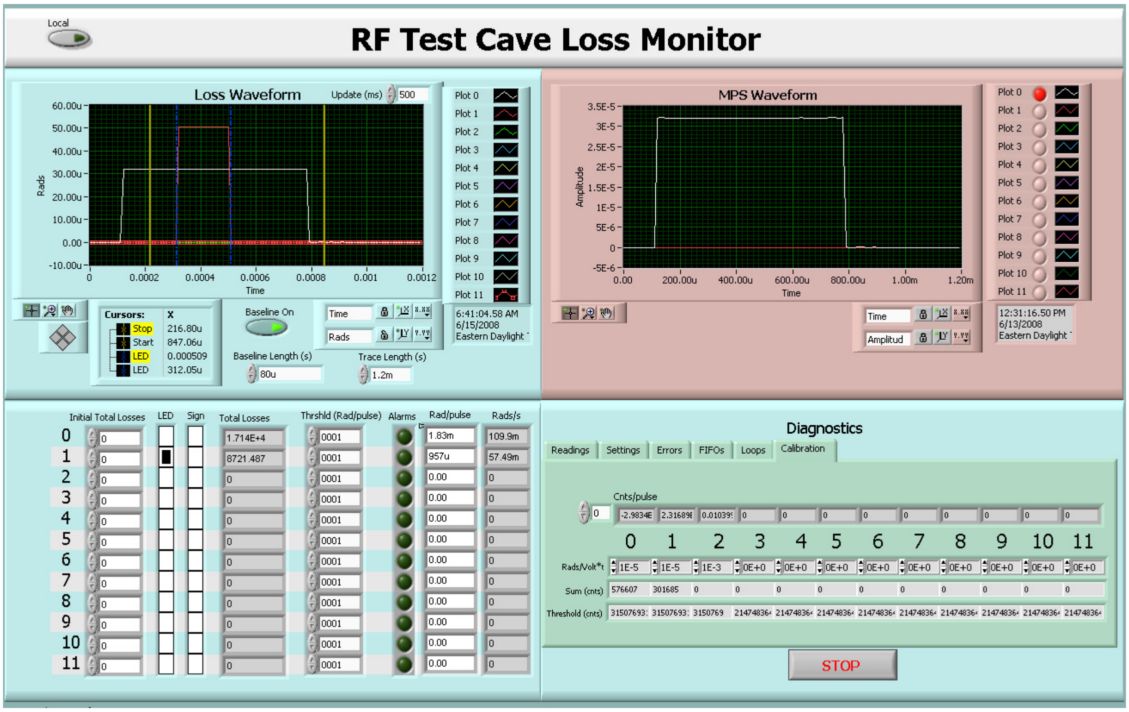


Figure 2 The data-acquisition system.

* ORNL/SNS is managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725



RF Test Cave Loss Monitor

Update (ms) 500

| Cursors: | X |
|---|---|
| Stop | 216.80u |
| Start | 847.06u |
| LED | 0.000509 |
| LED | 312.05u |

Baseline On

Baseline Length (s) 80u

Trace Length (s) 1.2m

6:41:04.58 AM
6/15/2008
Eastern Daylight

12:31:16.50 PM
6/13/2008
Eastern Daylight

|  | Initial Total Losses | LED | Sign | Total Losses | Thrshld (Rad/pulse) | Alarms | Rad/pulse | Rads/s |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | □ | □ | 1.714E+4 | 0001 | ● | 1.83m | 109.9m |
| 1 | 0 | ■ | □ | 8721.487 | 0001 | ● | 957u | 57.49m |
| 2 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 3 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 4 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 5 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 6 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 7 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 8 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 9 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 10 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |
| 11 | 0 | □ | □ | 0 | 0001 | ● | 0.00 | 0 |

Diagnostics

Calibration

Cnts/pulse 0

|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cnts/pulse | -2.9834E | 2.31689E | 0.01039S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Rads/Volt*t | 1E-5 | 1E-5 | 1E-3 | 0E+0 | 0E+0 | 0E+0 | 0E+0 | 0E+0 | 0E+0 | 0E+0 | 0E+0 | 0E+0 |
| Sum (cnts) | 576607 | 301685 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Threshold (cnts) | 31507693: | 31507693: | 3150769 | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ | 21474836‹ |

Figure 3. The front-panel of the program on the Real-Time controller.

.

## *Software Overview*

To interface the system to EPICS, a rackmount computer functions as a bridge between the real-time controller and the rest of the control system. Thus, in effect, there are three environments to program:

- LabVIEW FPGA: implements the low-level point-by-point processing of the data,
- LabVIEW Real-time: implements the real-time calculations such as Rads/sec, and
- LabVIEW for Windows: implements the gateway to the EPICS-based control system.

The LabVIEW project manager manages and integrates each of the environments within one project. All drivers for the hardware are included and integrated within the programming language. LabVIEW provides the same data-flow paradigm whether you program for windows, Real-time or FPGA. Programming LabVIEW within the FPGA environment takes some adjustment to account for the different style of programming needed for the FPGA to optimize for program speed and size.

## *FPGA Software*

The FPGA code implements all the point-by-point processing and the main program iterates at 100kHz, the maximum for the cRIO digitizers. In the current BLM system for the SNS accelerator, a 100kHz VME digitizer first takes a complete trace of the beam loss signal. Then the VME controller does a DMA transfer of the traces and after that has to process the data. This means that the controller can only trigger the beam abort after the beam pulse has passed. This is why an analog circuitry handles the fast abort function.

However, with the point-by-point processing of the FPGA, the beam abort is easily implemented by comparing the sum of the acquired data so far to a threshold and, if exceeded, setting the abort line. Thus, the program can compare the losses to the thresholds and possibly generate a beam abort on each 10 usec iteration. Similarly, the generation of a pulse to drive the LED during the acquisition is implemented simply by turning a Digital Out on or off based on the acquired sample number as shown in Figure 4.

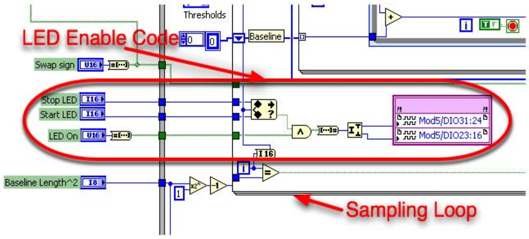


Figure 4. The oval loop shows all the code to implement the Test Pulse for the LED for all channels.

The point-by-point processing also implements the baseline correction, sign inversion, and integration of the total losses for each channel.

The total losses are represented in a 64-bit fixed-point format to deal with the large range of the adjustable gain, about 10^6, depending on the HV settings for the PMTs.

The acquired data, baselines, sums, and abort status are all sent to the Real-time controller using DMA FIFOs. As all calculations that must be done within the beam repetition rate, 60Hz, are done in the FPGA, the real-time controller is only passed data at a rate, default of 2Hz, required for display purposes. If a beam abort is set then the loss waveforms are tagged and DMA'd to the real-time controller for display in a separate window.

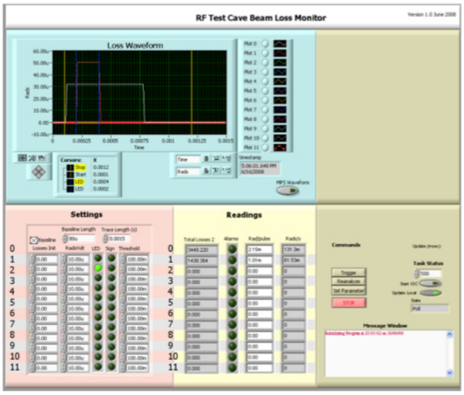
The settings and readbacks such as thresholds, gains, LED enable, and HV readbacks are transferred through the Interactive Front Panel Communication.

### *Real-time Software*

The Real-Time controller interacts with the FPGA code to get the data in and out. It scales the acquired loss traces from raw binary to Rads/s and calculates the losses per second using the change in total losses. The controller communicates with the gateway using LabVIEW Shared Variables. Its front panel is shown in Figure 3. While the real-time controller is embedded and doesn't have a display, LabVIEW can show the front-panel on the host computer while the code executes on the RT controller. During normal operations only the front-panel of the host computer's program, the gateway software, will be shown.

### *Gateway Software*

The gateway PC integrates the CompactRIO system to the EPICS-based control system. The PC runs a program based on the standard SNS template, [3], including the EPICS IOC. This program links the EPICS Process Variables of the IOC to the Shared Variables of the Real-time controller, see Figure 5.



Cursors: X
Stop 0.0012
Start 0.0001
LED 0.0004
LED 0.0002

Time
Rads

Timestamp
5:06:01.648 PM
6/16/2008

Settings

☒ Baseline
Baseline Length 80u
Trace Length (s) 0.0015

|  | Losses Init | Rads/Volt | LED | Sign | Threshold |
|---|---|---|---|---|---|
| 0 |  |  |  |  |  |
| 1 | 0.00 | 10.00u | ● | ● | 100.00m |
| 2 | 0.00 | 10.00u | ● | ● | 100.00m |
| 3 | 0.00 | 10.00u | ● | ● | 100.00m |
| 4 | 0.00 | 10.00u | ● | ● | 100.00m |
| 5 | 0.00 | 10.00u | ● | ● | 100.00m |
| 6 | 0.00 | 10.00u | ● | ● | 100.00m |
| 7 | 0.00 | 10.00u | ● | ● | 100.00m |
| 8 | 0.00 | 10.00u | ● | ● | 100.00m |
| 9 | 0.00 | 10.00u | ● | ● | 100.00m |
| 10 | 0.00 | 10.00u | ● | ● | 100.00m |
| 11 | 0.00 | 10.00u | ● | ● | 100.00m |

Readings

|  | Total Losses 2 | Alarms | Rad/pulse | Rads/s |
|---|---|---|---|---|
| 0 | 3449.220 | ● | 2.15m | 131.3m |
| 1 | 1438.364 | ● | 1.01m | 61.53m |
| 2 | 0.000 | ● | 0.00 | 0 |
| 3 | 0.000 | ● | 0.00 | 0 |
| 4 | 0.000 | ● | 0.00 | 0 |
| 5 | 0.000 | ● | 0.00 | 0 |
| 6 | 0.000 | ● | 0.00 | 0 |
| 7 | 0.000 | ● | 0.00 | 0 |
| 8 | 0.000 | ● | 0.00 | 0 |
| 9 | 0.000 | ● | 0.00 | 0 |
| 10 | 0.000 | ● | 0.00 | 0 |
| 11 | 0.000 | ● | 0.00 | 0 |

Update (msec)
Task Status 500
Start IOC
Update Local
State Poll

Initializing Program at [illegible]

Figure 5. The Front-panel of the Gateway software.

## RESULTS

The system has been tested in the lab with pulse generators to trigger the system and provide a pulse as a signal to the digitizers. To test the DMA FIFOs under duress, the fill rate by the FPGA code was set to exceed the read out rate by the Real-Time controller. During such a test, data is lost but the number of lost samples is logged so that the FIFOs can resynch once the fill rate was once again below the read out rate. Normally the fill-out rate is set a 2Hz, much lower than the maximum read-out rate. The max rate is about 50Hz with all twelve 150 sample-long traces displayed and transferred.

An interesting aspect of the FPGA point-by-point processing is that it doesn't matter what the retrigger rate is. As long the next trigger occurs after the trace has been completed, the system is ready for the next trigger. Thus a 1msec acquisition can retrigger at approx. 1 kHz. As all the data is processed within the FPGA this does not lead to an additional load on the Real-Time controller. This is unlike the current SNS BLM where longer traces mean that it takes longer to finish the acquisition, longer to transfer the data from the digitizer to the real-time controller, and longer to process the data on the real-time controller.

The length of the trace can also be varied without affecting the real-time performance. Currently the DMA FIFOs are limited to hold up 15msec, the maximum for a 60Hz rep rate, worth of data for the 12 channels.

## CONCLUSIONS

The CompactRIO is a viable alternative to the VME-based systems. The point-by-point processing of the FPGA makes the implementation of many functions of a BLM system very easy. While there is a learning curve for writing LabVIEW code for FPGA, it does provide a well-integrated environment to implement FPGA and Real-Time functions. In most cases, one will have to deal with multiple vendors and multiple tools to implement such a point-by-point processing application.

## FUTURE

We hope that soon we will have actual loss signals from when the Test Cave is running so that we can show this data on the local consoles using the EDM screens. We expect that the EPICS IOC will become available for the CompactRIO Real-Time controller so that the Gateway PC is no longer needed. This implementation and others will be evaluated as replacements for the current VME-based SNS BLM system

## REFERENCES

[1] D. Stout, et al, "Status of the Spallation Neutron Source Superconducting RF Facilities",ORNL, Oak Ridge, Tennessee

[2] Y. Leng,et al "Data Acquisition for SNS Beam Loss Monitor System", ICALEPCS 2003, Gyeogiu, Korea, October 13-17, 2003, pp 525-26.

[3] W. Blokland et al, "Network Attached Devices at SNS", DIPAC'03, Mainz, Germany May-5-7, 2003, pp 146-8.