THE SNS FRONT END LLRF SYSTEM *

L. Doolittle, C. C. Lo, M. Monroy, and A. Ratti
E. O. Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract

LBNL has built the Front End for the Spallation Neutron Source (SNS) project,** an H$^-$ injector that is designed to deliver 52 mA of beam current at 2.5-MeV energy in 1ms batches of 650-ns pulses at 60 Hz, for a macro beam duty factor of 6%. The Front End is comprised of an ion source, an electrostatic Low Energy Beam Transport line (LEBT), an RFQ, and a Medium Energy Beam Transport line (MEBT). The RFQ accelerates the beam from 65 keV to 2.5 MeV. The MEBT includes four bunching cavities that are designed to preserve the longitudinal bunch length during transport down the 3.7 m long beam line. An FPGA-based Low Level RF (LLRF) control system has been built and is now used to control cavity amplitudes to better than 1% and relative phases within 1 degree. The signal-processing chain that includes ADC, DSP in FPGA, and DAC, adds an analog delay of about 250 ns. The fast digital processing is networked to the EPICS global controls system with a commercial miniature embedded Linux computer (nanoEngine).

1 INTRODUCTION

The LLRF system controls four 402.5 MHz rebuncher cavities of the SNS MEBT. Phase-coherent operation of these cavities was needed during early commissioning activities at Berkeley.

To merge successfully with the SNS RF reference infrastructure, the system was designed to make use of a 352.5 MHz LO, 50 MHz IF source, 40 MHz clock, and 10 MHz synchronisation. Figure 1 shows the basic architecture selected. The 50 MHz input signals (mixed down from 402.5 MHz) from the cavity field probe, forward power, and reflected power, are sampled at 40 MS/s by individual 12-bit ADCs. Digital signal processing in an FPGA controls two 12-bit analog output channels, each updated at 20 MS/s, control a 50 MHz vector modulator. A networked host computer connects the fast signal-processing chip to the outside world via Ethernet. Timing and interlocks are routed through the same FPGA.

A system simulator was written to verify that such a system could provide enough appropriate feedback to keep the cavity fields and phases nearly constant during beam loading transients. Out of a total feedback loop latency of 1100 ns, 680 ns are simply due to cables and the 20 kW amplifier. Of the remaining 420 ns that are associated with the LLRF chassis, 170 ns come from analog filters, and 250 ns from the ADC/FPGA/DAC processing step.

Note that, as usual for cavity control, there are two feedback loops. A fast one that operates within the 1-ms macro pulse, to keep the cavity field vector on its set point, and a slow one that keeps the cavity on tune mechanically.

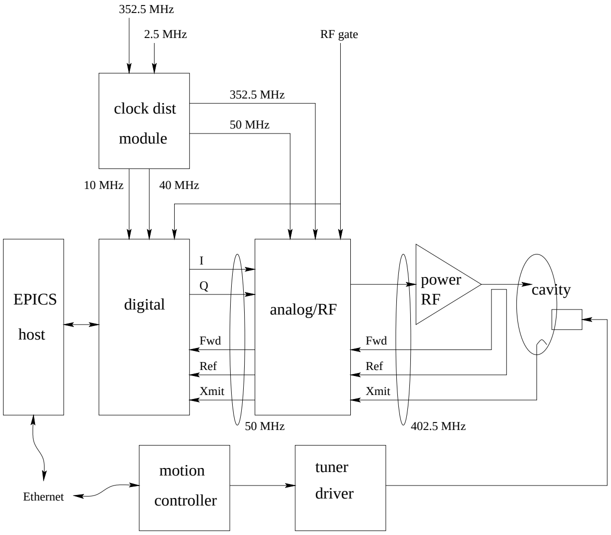

Figure 1. Initial system block-diagram

2 HARDWARE

Previous experience with Xilinx 4000 series FPGA chips convinced us that their next-generation, Virtex-derived, chips could easily handle the data path at 40 MHz with a moderate amount of pipelining. A pure digital signal processing flow was blocked out (see Figure 2), implementing auto-offset subtraction, set point, complex proportional gain, scalar integral gain, and a feed-forward table. I and Q channels are never separated; only a 10-MHz sign flip is required to change the input I/Q/-I/-Q sequence to the output I/Q/I/Q sequence. Xilinx's patented writable logic-cell feature is used to advantage in the three data-path multipliers (two to set the complex proportional gain and one to set the scalar integral gain). Such writable cells convert the conventional ``KCM" (constant coefficient multiply) used in DSP on FPGA (to reduce logic element usage and increase speed) into a DKCM (dynamic constant coefficient multiply), where the coefficient can be reloaded, in our case, between beam pulses.

The hardware was kept as simple as possible. Very little is included on the digital board besides the 4 input

* This work is supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC03-76SF-00098.

**The SNS project is being carried out as a collaboration of six US Lab-oratories: Argonne National Laboratory (ANL), Brookhaven National Laboratory (BNL), Thomas Jefferson National Accelerator Facility (TJNAF), Los Alamos National Laboratory (LANL), E. O. Lawrence Berkeley National Laboratory (LBNL), and Oak Ridge National Laboratory (ORNL). SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Dept. of Energy.

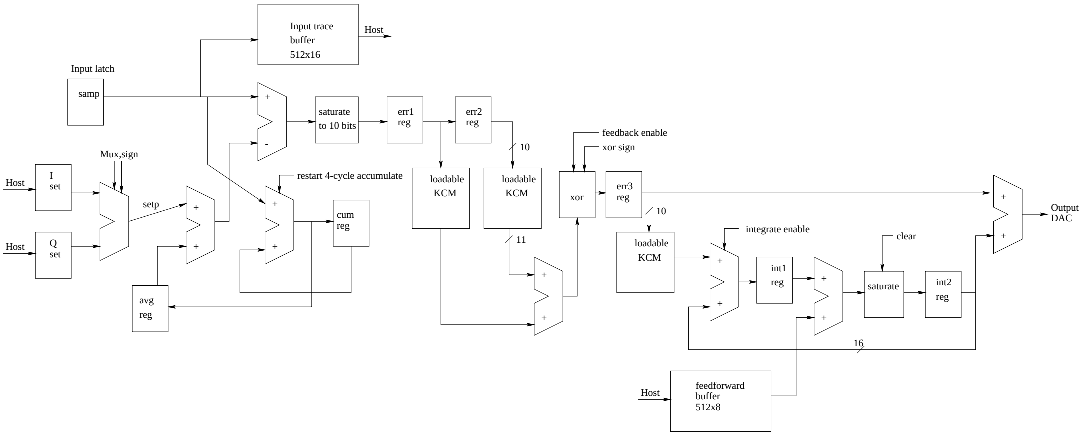

Figure 2. Block diagram of the data path programmed into FPGA

ADC's (TI/Burr-Brown ADS808), the FPGA (Xilinx XC2S150), and the output dual DAC (Analog Devices AD7965). See Figure 3. The embedded computer's expansion bus is tied directly to the FPGA, so a virtual register map (64K x 16) can be implemented under complete control of the FPGA firmware. The four FPGA JTAG TAP pins are wired directly to the embedded computer's digital I/O lines, allowing the FPGA to be programmed without recourse to the usual separate chip of ROM. Of course, the embedded computer is programmed to get that FPGA bit file from the network, so firmware updates are trivial to propagate.

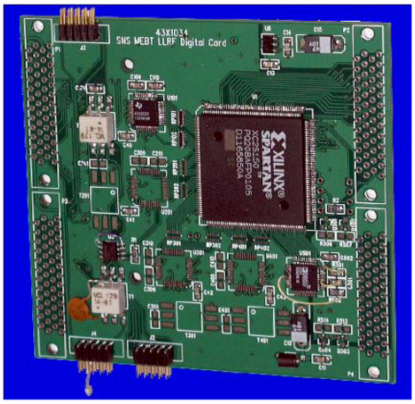

Figure 3. Digital board

Limited FPGA pin count and engineering time kept us from adding external memory to the FPGA. As a result, only 6 kBytes of data from within a 1-ms pulse can be stored. In contrast, during the same pulse 200 kBytes stream in through the high-speed ADCs. This limited storage isn't all bad; such large quantities of data could easily overwhelm the host computer and/or the network. Our experience showed this memory to be adequate for both operator-comfort displays (with about the information content of a digital storage scope) and acquisition of leading-edge waveforms (to compute detuning angles).

A single-board computer called a nanoEngine, from Bright Star Engineering, was chosen for this project, as a known good base on which to run EPICS on Linux. Relevant features include:

200 MHz StrongARM CPU
32 Mbytes SDRAM
4 Mbytes Flash
Bus expansion

100 base T Ethernet
Serial (RS-232) console
10 digital I/O

The footprint is a single high-density 160-pin connector over which all power and signal lines are routed, including Ethernet. The 200-MHz processor is capable of nearly 200 MIPS operation, as long as it runs mostly out of the onboard 8K data/8K program cache. The curve-fitting routines in particular had to be carefully constructed to avoid blowing this cache, as well as to avoid heavy floating-point work—there is no floating-point hardware on a StrongARM chip; those instructions are emulated (slowly) in software.

The only non-programmable logic in the system controls a PIN diode shutoff for the output 402.5 MHz signal. When the externally provided RF gate drops, no RF output is possible. This signal is also used as the trigger for a timing-state machine (programmed in the FPGA). The feed-forward table was used extensively to generate test patterns. When first hooked up to a cavity, with feedback disabled, this let us use the system as a pulse generator and vector oscilloscope.

The RF chassis was constructed with connectorized (SMA) components, to permit rearrangement of components and adjustment of signal levels without incurring the schedule delay of revising circuit boards (see Figure 4). Because of the tight schedule and in an attempt to eliminate risks, enough phase shifters and additional control and monitoring points were added that an analog control loop could have operated the cavities if that had been found necessary. Much of that circuitry sat idle during the commissioning process, partly because the digital section worked, and partly because the accelerator beam-timing signals were frequently being changed. This ‘moving target’ could be hit far more easily with adjustments to the FPGA configuration than adjusting four copies of analog hardware.

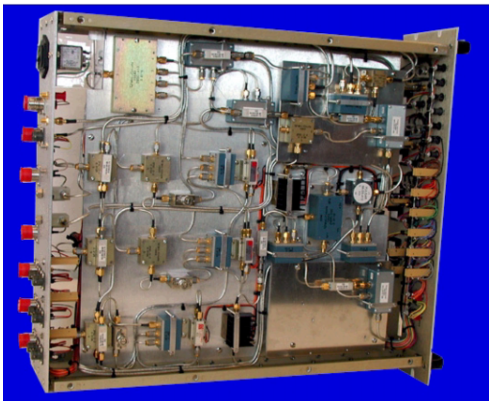

Figure 4. RF Chassis

The chassis is convection cooled (no fans). Total system dissipation is about 35 W; the largest contributor (15 W) is the 352.5-MHz amplifier that drives the LO port of four mixers with +20 dBm each. The ADS808's are powered off when not in use; this avoids 2 W of dissipation on the digital board. These chips make a complete transition out of sleep mode to providing accurate conversions in about 40 microseconds.

3 PROGRAMMING

The cavity detune angle must be computed, based on RF waveforms, in order to close the tuner feed-back loop. Initial attempts to use the trailing edge of the field probe were stymied by the coupling between the cavity and the output cavity of the 20-kW driver amplifier, and due to the fact that the tune of this coupled system changed as a function of power level. We therefore switched to a method based on the leading edge of the reflected wave. The host computer analyses 500 points of raw ADC data, and generates a six-parameter fit to the exponential decay waveform, in less than 1 ms. The result is an unambiguous measure of the cavity frequency-error. A simple set of averaging, glitch rejection, and deadbanding ran a bang-bang control of an external (network connected) stepper motor controller. Once realistic bounds were set on the fit parameters and quality and the sign of the feedback was selected correctly, this system ran flawlessly.

The EPICS interface was based on xcas. The FPGA delivered an interrupt at the end of an RF pulse, triggering software to exchange data between the EPICS data structures and the FPGA before the next pulse started, 15 ms later. The firmware and software used a handshake flag to monitor this process, as a check that the software was adequately responsive. No such errors were logged, despite the use of a stock, non-real-time Linux kernel.

The development of the FPGA firmware and the support software was done in a co-ordinated manner, so that each task could use an appropriate mix of hardware and software. Typical race conditions were eliminated at the source. For example, latched-error conditions were picked up by the software with a read-and-clear function; the software, working in a single thread-process model, could then increment error counters for its internal use. The FPGA logic was coded entirely in synthesizable VHDL, using Virtex primitives only for RAM, multipliers, and clock distribution. No special constraints were required to meet the 40 MHz clock rate.

All of the signal-processing logic in the FPGA was configured in a single 40 MHz clock domain. A small amount of logic was placed in the clock domain of the host bus, 25 MHz. The usual techniques for crossing the two domains were mostly unnecessary, as the fast signal processing takes place within the 1-ms beam pulse, and the host activity is limited to an interrupt routine following the pulse. The handshake discussed above was used to verify this assumption. Self-check logic was added to detect missing or erratic clocks, triggers, or synchronization pulses.

In the available commissioning time, we learned how to systematically turn on a cavity, adjusting the various phases and gains. That did not leave any time to codify these techniques into automatic start-up and self-tune software.

Figure 5 shows an example of a short RF pulse, with a beam-loading transient that starts at the 120-microsecond co-ordinate. These data were saved from a running LLRF control system and converted here from Real/Imaginary to display their magnitude only.

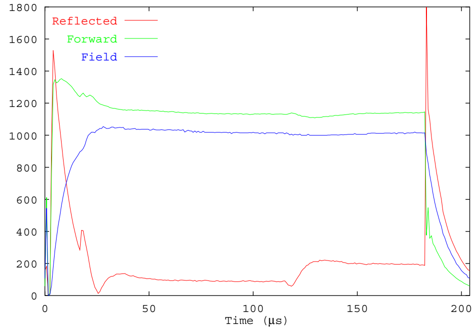

Figure 5. RF pulse example

4 ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of and equipment from the LANL SNS LLRF team as well as the support of the SNS project office at ORNL.

5 REFERENCE

[1] SNS Beam Instrumentation Workshop, Berkeley (1999).