APPLICATIONS OF GENERAL-PURPOSE RECONFIGURABLE LLRF PROCESSING ARCHITECTURES

Lawrence Doolittle (LBNL, Berkeley, California)*

Abstract

Traditional RF processing systems have involved heterodyned RF processing based on mixing a Local Oscillator to up and down convert RF signals through a baseband I/Q or Mag/Phase processing channel. These systems were traditionally custom engineered for each accelerator application. Recent technical developments in RF processing and the development of sufficiently fast reprogrammable digital processing functions lead to development of general-purpose RF processing functions which can incorporate a mix of heterodyned and direct digital down/up-converted processing ("software radio"). This general-purpose approach allows one design of hardware to be applicable to many RF processing tasks, where the firmware and software in the programmable functions define the application. An example design, with applications to linac LLRF control loops and electro-optic timing reference stabilization is presented.

INTRODUCTION

Industry trends, including Moore's law, have inexorably pushed signal processing from the analog to digital domains. The accelerator community's Low Level RF (LLRF) controls are representative of, but not a large component of, the military/industrial signal processing market. That market in turn is tangential to the telecommunications industry, and most of the key components and tools used are offshoots of those needed by telecom. The overall world telecom industry has sales on the order of US$\$10^{12}$/year, which helps to explain the remarkable performance-to-cost ratio of its products.

Like radar, radio astronomy, and wireless communications, accelerator RF controls measure and create RF signals. Above some fuzzy threshold around 200 MHz, it is still advantageous to use heterodyning techniques in the hardware between the RF and digital converters, leading to the familiar block diagram of figure 1.

The primary job of this system is to set the cavity's vector voltage using feedback. Cavity control's major difference from other RF applications is its sensitivity to latency, since like any feedback system, some of its performance scales as $1/T$.

To put this in a quantitative context, a typical modern LLRF system will use a pipelined 12 to 16 bit ADC running at 60 to 120 MS/s, digitizing a signal in the 1st, 2nd, or 3rd Nyquist zone. This chip itself introduces 5 to 15 cycles of latency. The DAC is often clocked faster than the ADC, since the output power of a DAC decreases with each advancing Nyquist zone. This paper draws a distinction between MS/s, that represents data rate, and MHz, that represents operating (carrier) frequencies.

The reconfigurable aspect to these designs is tied to the use of FPGA (Field Programmable Gate Array) chips to implement Digital Signal Processing (DSP). Most of this paper, therefore, covers techniques for fitting the relevant DSP into existing FPGA chips. There is also some configurability built into the design of that DSP logic, to give flexibility in testing and operation.

The reprogrammability of the FPGA allows the same hardware to be used for different purposes (e.g., pulsed *vs.* CW machine, or RF *vs.* optical signal handling). It also makes allowances for a design to improve with time, and have new features added.

BUILDING BLOCKS

Before embarking on any digital design, it is always important to have a solid grasp of the mathematical and signal processing steps required, and to have some analytic or numerical demonstration that the signal processing will meet the accelerator's needs. The key step is usually to evaluate if the feedback loops will remain stable and maintain the cavity field to within specified tolerances, in the face of expected perturbations.

The building blocks to construct the feedback and associated functions within an FPGA are, at the lowest level, mostly adders, multipliers, and registers. These get combined to make larger blocks. One of the best understood is a generalized digital filter, widely covered in signal processing textbooks. Later sections will discuss two filter architectures of special interest for LLRF, and a technique for transformations between rectangular and polar coordinates.

FEEDBACK

The simplest feedback to implement is a Proportional-Integral (P-I) loop. Basic feedback theory teaches us that the most latency-sensitive term is the P term. This data path can be constructed either with [1] or without [2] downconversion to baseband. Limitations of finite-precision arithmetic make constructing an accurate integral (I) term without downconversion to baseband difficult, although useful approximations can be created if a lower bound to frequency is specified.

With an I-Q sampled system, conversion between baseband and IF is trivial. Such systems suffer from odd signal

* This work was supported by the Office of Science, US Department of Energy

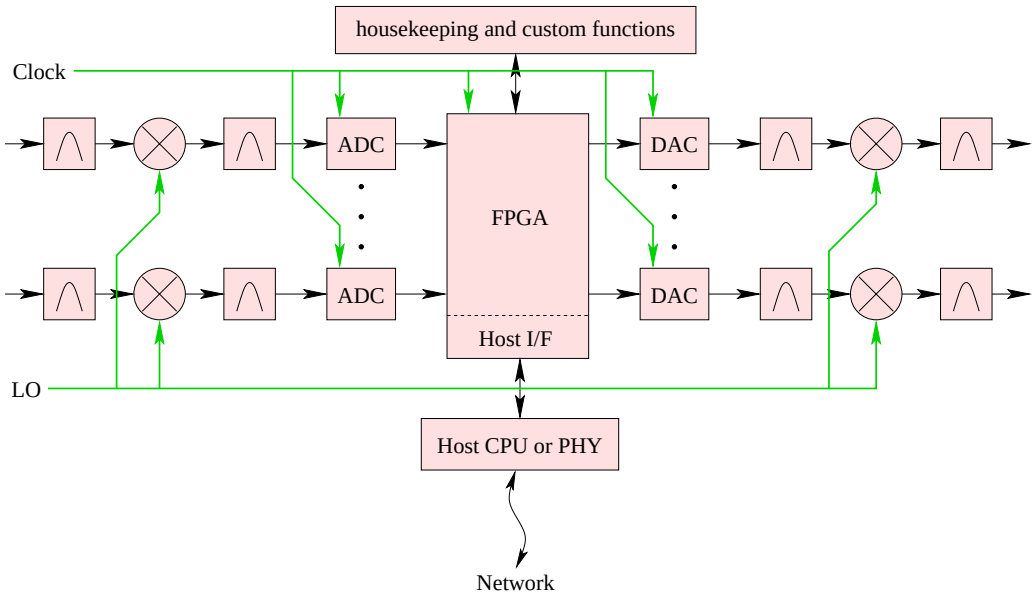

Figure 1: Familiar Block Diagram

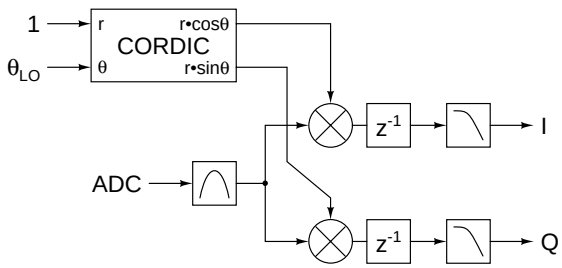

Figure 2: Simple conversion from non-IQ signal to I and Q

harmonics overlapping the signal itself, and ADC errors limit the ability to recover small signals by digital signal averaging. A non-IQ (but coherent) system can have higher useful dynamic range. Its downside is the more complex digital conversion between IF and baseband.

The standard approach for downconverting to baseband parallels traditional analog circuitry, shown in figure 2. This configuration is not ideal for feedback use, because of compromises required in the low-pass filter between spur suppression and latency. A slight rearrangement of these same components can reconstruct I and Q from pairs of input samples with no first-order spurs [3].

Pulsed machines (that have an RF pulse length that lasts a large number of ADC samples) can often benefit from pulse-to-pulse feedback, also known as adaptive feedforward. One component of the drive waveform is a table that is computed based on the previous cavity field error. Robustly convergent behavior is seen by combining a low pass filter in the plant with a time reversed low pass filter in the digital control loop [4]. This computation is not hard to implement in an FPGA fabric, as long as the gap between pulses is longer than the pulse length itself, and two copies of the pulse waveform fit in the FPGA's on-chip memory.

FILTERS

CIC (cascaded integrator comb) filters are a class of low-pass decimating filters that are easy to program and light on FPGA resources. They have the interesting feature that the bandwidth intrinsically scales with the decimation ratio m, and that decimation ratio does not change the data path, only how often the integrator clock is enabled. Consideration has to be made that an nth order CIC has a signal gain of m^n.

When creating operator comfort displays or recording waveforms for later analysis, it can be helpful to create decimating filters to limit the needs for data storage or communication bandwidth. While a CIC filter is normally used to start the process, such filters cannot give a passband that is a large fraction of the Nyquist band. A series of half-band filters can, and they still have a good balance of efficiency and resource usage. Their relatively large latency makes them a poor candidate for the actual feedback paths.

Both CIC and half-band filters can be built in an FPGA to handle a multiplexed stream of data. For instance, the I and Q data converted from a forward and reverse directional coupler can be multiplexed into a single data path containing four isolated data streams. When building filters of this type, it is useful to keep to a formalism of passing a strobe bit along with the data word. A 12-channel rendition of this concept lies at the heart of the waveform recording apparatus of LBNL's production phase stabilization code.

CORDIC

CORDIC [5] processing is a nonlinear transformation of great utility in LLRF signal handling, providing conversion between polar and rectangular representations of a complex number. The simplest and highest performance imple-

mentations in an FPGA are fully unrolled [6], which also makes them large consumers of FPGA resources. Nonlinear signal processing has to be treated with respect when designing a LLRF system. They work best when provided with a signal with excellent signal to noise ratio and limited dynamic range. The CORDIC logic itself generates data-dependent non-Gaussian errors due to arithmetic truncation, which is usually characterized as noise. That noise can be made arbitrarily small by choosing appropriate bit width and stage count in the CORDIC logic. As always, extensive simulations are used to verify adequate performance of the DSP design.

When a feedback loop is constructed in polar coordinates, the tendency is to use two CORDIC processes, one to move from ADC samples to magnitude and phase, the second to move back to DAC samples. The latency of such a system is dominated by the CORDIC pipelines, so the bandwidth does not need to be high. Jefferson lab pioneered using a single CORDIC block, with inputs multiplexed between the two functions, which they call a Tornado [7].

The phase distribution system described below measures a number of individual frequency components in the input waveform. After averaging their individual I and Q values, they are all multiplexed through a single CORDIC block to compute their individual angles.

When constructing a sine wave local oscillator, it is possible to use a CORDIC as shown in figure 2. If the LO has a rational relationship (with a denominator less than about 64) to the clock rate, and no additional phase manipulations are needed, a counter with lookup table can give a better tradeoff of accuracy *vs.* resources than a phase accumulator plus CORDIC. Both approaches can be considered a form of DDS. If used with a pulsed machine, a table-based design can have the table reloaded between pulses to set the phase and amplitude of the LO. This configuration could have some advantages at the front end of a many-cavity vector-sum controller.

OTHER TECHNIQUES

In some circumstances, it's possible to save resources by using bit-serial arithmetic [8] instead of full parallel arithmetic. For instance, a bit-serial CORDIC processor can be a small fraction the size of a fully unrolled CORDIC; of course, it's also dramatically slower. One version created by the author used $1/8$ the logic, but ran 500 times slower.

At the extreme of complex algorithms and relaxed latency requirements, it is possible to resort to traditional general purpose computer architecture, either hard cores or soft cores. Computers, in this context, can be thought of as an extreme case of the data multiplexing tricks mentioned above for filters and CORDIC blocks. In a Harvard architecture computer, the sequence of data multiplexed through multipliers and adders is chosen by means of a stored program. As long as the program stays simple and has limited conditional dependence on the data, the predictable delay intrinsic to digital synchronous logic can be maintained. Modern general-purpose CPUs, with their sophisticated caches, branch prediction units, and interrupts, have strayed far from that concept. They are therefore not suitable for the DSP component of LLRF control.

LLRF programming has many similarities to the currently burgeoning field of software radio. The up and down-conversion steps and the digital filter steps are common to the two.

FIBER OPTIC PHASE DISTRIBUTION

A fiber-optic-based S-band phase distribution system has been built (as shown in figure 3 using some of the techniques described above. The control hardware platform is LBNL's fourth-generation LLRF board [9]. The signal processing logic uses more than half of the resources of a Xilinx XC3S1000.

The ADC and digital processing runs at 87.5 MS/s, so the Nyquist span is 43.75 MHz. The DSP section has to measure six frequencies, and use feedback to generate two. The following table shows the actual analog frequencies measured, and their relationship to the sampling rate, both actual and as it appears aliased into the first Nyquist zone.

Interferometer beat	101.50 MHz	29/25	56/350
Subharmonic upper	60.25 MHz	241/350	109/350
Cal upper sideband	57.75 MHz	33/50	119/350
IF	56.00 MHz	16/25	126/350
Cal lower sideband	54.25 MHz	31/50	133/350
Subharmonic upper	51.75 MHz	201/350	149/350

All frequencies except the interferometer beat are down-converted from S-band with a 2800 MHz LO. The frequencies are all shifted slightly from the above values by locking the LO (from which the ADC clock is derived) to 2800.056 MHz instead of 2800 MHz. This allows averaging some systematic phase distortion that arises in the analog hardware.

The code base is written in a modular and testable form, that has permitted us to reconfigure channels between optical and RF, with one- and two-board experiments to study correlations between receivers.

Preliminary measurements show that the hardware and software is capable of transmitting S-band phase information over 2 km of optical fiber with 20 to 50 fs rms drift and noise during several days of operation [10].

CONCLUSIONS

General purpose hardware, fitting the mold of the familiar block diagram, is now capable of meeting most LLRF control requirements. The specific accelerator needs are then implemented in programmable digital logic, using techniques discussed above. Most of the intellectual investment in understanding how to operate the cavity, and by extension the beam, is therefore encapsulated in that programming. It is hoped that the understanding, and maybe even the programming, can be carried forward to future

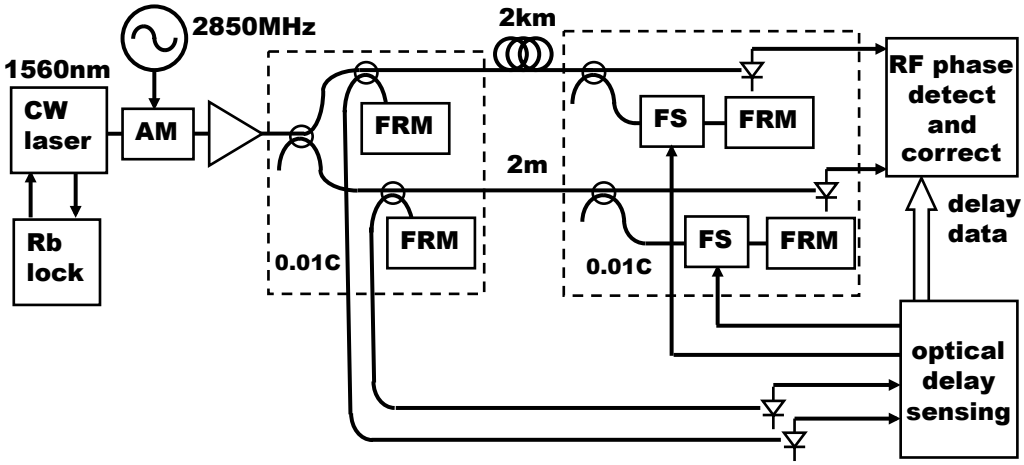

Figure 3: Schematic of phase distribution test equipment

hardware, as current hardware goes obsolete and becomes unmaintainable.

LLRF programming uses well understood building blocks, generally with parallels in previous generations of analog design. The reprogrammable nature of the hardware allows new requirements to be met in a scalable and cost-effective manner. This combination of simple hardware and complex programming will operate more reliably than an analog realization would, and the digital section operates without drift or $1/f$ noise. The invisibility and abstraction of the programming requires a serious commitment to design, simulation, and testing.

ACKNOWLEDGEMENTS

The work presented here has grown out of years of productive collaborations with laboratories around the world. In particular, SNS (ORNL), HINS and Project X (FNAL), FERMI@ELETTRA (Sincrotrone Trieste), LCLS (SLAC), NSLS-II (BNL), JLab, and SSRF have all contributed to the development or implementation of these technologies. We have also enjoyed the benefits of collaboration with FLASH (DESy) and the GNU Radio project.

The author wishes to thank Gang Huang of LBNL for his many contributions to the phase reference application.

REFERENCES

[1] L. Doolittle, “Low-Level RF Control System Design and Architecture,” APAC 2007.

[2] L. Doolittle, H. Ma, and M. Champion, “Digital Low-Level RF Control Using Non-IQ Sampling,” LINAC06.

[3] S. N. Simrock et al., “Considerations for the Choice of the Intermediate Frequency and Sampling Rate for Digital RF Control” EPAC 2006

[4] A. Brandt, “LLRF Automation and Adaptive Feedforward,” FLASH Seminar May 29th 2006.

[5] J. Volder, “The CORDIC Trigonometric Computing Technique,” IRE Trans. Electronic Computing, 1959

[6] R. Andraka, “A survey of CORDIC algorithms for FPGAs,” FPGA ’98

[7] J. Musson et al., “Digital Self-Excited Loop,” LLRF07

[8] R. Andraka, “Distributed Arithmetic,” http://www.andraka.com/distribu.htm

[9] L. Doolittle, “LLRF Evaluation Board,” http://recycle.lbl.gov/llrf4/

[10] J. W. Staples et al., “Demonstration of Femtosecond-Phase Stabilization in 2 km Optical Fiber,” PAC07