

A low-cost, FPGA-based servo controller with lock-in amplifier

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2012 JINST 7 P10026

(<http://iopscience.iop.org/1748-0221/7/10/P10026>)

View the [table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 128.143.23.241

This content was downloaded on 05/10/2013 at 16:52

Please note that [terms and conditions apply](#).

RECEIVED: June 28, 2012

REVISED: August 29, 2012

ACCEPTED: October 2, 2012

PUBLISHED: October 29, 2012

A low-cost, FPGA-based servo controller with lock-in amplifier

G. Yang,^{a,b,1} J.F. Barry,^a E.S. Shuman,^a M.H. Steinecker^a and D. DeMille^a

^a*Department of Physics, Yale University,
P.O. Box 208120, New Haven, CT 06520, U.S.A.*

^b*Department of Physics and James Franck Institute, University of Chicago,
5720 S. Ellis Ave, Chicago, Illinois 60637, U.S.A.*

E-mail: yangge@uchicago.edu

ABSTRACT: We describe the design and implementation of a low-cost, FPGA-based servo controller with an integrated waveform synthesizer and lock-in amplifier. This system has been designed with the specific application of laser frequency locking in mind but should be adaptable to a variety of other purposes as well. The system incorporates an onboard waveform synthesizer, a lock-in amplifier, two channels of proportional-integral (PI) servo control, and a ramp generator on a single FPGA chip. The system is based on an inexpensive, off-the-shelf FPGA evaluation board with a wide variety of available accessories, allowing the system to interface with standard laser controllers and detectors while minimizing the use of custom hardware and electronics. Gains, filter constants, and other relevant parameters are adjustable via onboard knobs and switches. These parameters and other information are displayed to the user via an integrated LCD, allowing full operation of the device without an accompanying computer. We demonstrate the performance of the system in a test setup, in which the frequency of a tunable external-cavity diode laser (ECDL) is locked to a resonant optical transmission peak of a Fabry-Perot cavity. In this setup, we achieve a total servo-loop bandwidth of ~ 7 kHz and achieve locking of the ECDL to the cavity with a full-width-at-half-maximum (FWHM) linewidth of ~ 200 kHz.

KEYWORDS: Lasers; Digital signal processing (DSP); Digital electronic circuits; Analogue electronic circuits

¹Corresponding author.

Contents

1	Introduction	1
2	System design	3
3	Principles of operation	3
3.1	Signal processing	4
3.1.1	Modulation generation and lock-in amplifier	4
3.1.2	Servo control	6
3.1.3	Ramp generator	7
3.1.4	Monitor outputs	7
3.2	User interface	7
4	Implementation	8
4.1	Modulation waveform synthesizer	9
4.2	Filter unit with gain	10
4.3	Dynamic output bandwidth allocation	11
4.4	FPGA usage	12
5	Performance	13
5.1	Test setup	13
6	Conclusions and possible future extensions	15
A	Assembly, mounting, and connectorization	15
B	Electronic characteristics	16
B.1	Slow DAC	16
B.2	Fast DAC	16
B.3	Analog capture circuit	17
C	RC filter equations in discrete time	17

1 Introduction

The use of active feedback control circuits to stabilize the frequency of tunable lasers is widespread in atomic physics and other fields [1–3]. Traditionally, analog phase-sensitive detector and proportional-integral (PI) servo circuits are used for generating frequency-sensitive signals and providing feedback to the laser controller electronics. Tuning of parameters (such as gains and

filter time constants, signal weights in summing amplifiers, etc.) for such detector and servo circuits is typically accomplished either with potentiometers (with a limited dynamic range) or by physical replacement of resistors, capacitors, etc. This makes large parameter changes onerous in such systems.

A variety of digital implementations of phase-sensitive detectors (or lock-in amplifiers) using microprocessors and digital signal processors have also been realized [4–10]. These can enable flexible implementations of processing and feedback control. Recently, field programmable gate arrays (FPGAs) have received substantial attention for use as phase-sensitive detectors [11, 12]. In these devices, the programmer has direct access to the gate-level design of the system. Truly parallel signal-processing algorithms can be implemented as dedicated hardware modules, like those in an analog circuit, and run at sampling rates fast enough to emulate the analog circuit functionality (> 1 MHz). In addition, it is possible to incorporate peripherals such as liquid-crystal displays (LCDs) and keyboards, enabling convenient operation as stand-alone systems. These features make FPGA-based phase-sensitive detectors ideally suited for incorporation in a digital replacement for traditional analog laser-locking circuits. Indeed, the use of FPGAs has been described for a specialized laser-locking application, but implementation details are not given [13].

Nevertheless, the design of FPGA logic in hardware description language (HDL) requires knowledge of the specific FPGA’s integrated circuitry and logic modules, resulting in a steep learning curve. Building digital signal processing cores from the ground up and connecting them with the necessary peripheral interfaces may present a considerable challenge to researchers without previous experience in digital logic. As a digital system based on off-the-shelf hardware, however, once the desired functionality has been implemented on one FPGA board, the system may be easily reproduced by researchers without requiring detailed understanding of the underlying hardware.

In this work, we describe the design and implementation of an FPGA-based system that incorporates a built-in waveform synthesizer, a lock-in amplifier, two channels of PI servo control, and a ramp generator on a single FPGA chip. Although adaptable to a variety of applications, this system has been designed to facilitate frequency locking of a diode laser, as a stand-alone, drop-in replacement for traditional analog locking circuits.¹ The goal of this work is to provide a system that is cheap and easy to assemble in quantity, as a single experiment may call for locking ten or more lasers. The total cost of the FPGA board and its peripherals is $< \$400$. Unlike traditional analog circuits, which usually require custom PCB fabrication and soldering of numerous components, this system consists of only a few off-the-shelf modules that are easily assembled. The only soldering required is for panel-mounted cable connectors to route signals to and from the appropriate input and output pins on the FPGA board. The system provides performance comparable to that of standard peak-dither analog locking circuits; as an example, we demonstrate using this system to lock an external-cavity diode laser (ECDL) to a Fabry-Perot cavity, with linewidth < 1 MHz. Moreover, in contrast to traditional analog locking circuits, this system is designed to allow easy

¹During preparation of this manuscript, we became aware of recent work along similar lines, describing the use of an FPGA board for laser frequency locking [14]. Major differences exist, however. While the system described by Schwettmann, et al., is designed to accept an error signal input proportional to the frequency deviation of the laser, our system also incorporates a lock-in amplifier and an integrated modulation waveform generator to allow for peak-dither locking. Also, in their system, parameters are read out and modified via an attached computer, whereas our system is designed to stand alone, with parameter read-out and modification via an integrated display and onboard controls.

reconfiguration over a broad range of locking parameter values (e.g., gains, filter time constants, etc.); optimization of these parameters can be accomplished without the need for soldering or physical changes to the circuit. This system focuses on achieving moderate performance at a low cost, but the design may be adapted for use with higher performance FPGA boards, ADCs, and DACs.

The user manual for this system, the source code (written in VHDL), and the FPGA configuration file (the compiled source code which programs the FPGA) are available on our group website (www.yale.edu/demillegroup/fpga.html).

2 System design

The system-level schematic is shown in figure 1. The system consists of the following hardware components: DACs and an ADC, the FPGA, an LCD, control switches, and a rotary knob. These components constitute the physical inputs and outputs of the system and require a layer of interface to convert signals back and forth to register values.

In general, the price of FPGA platforms is determined by the FPGA size and speed and the level of sophistication of onboard peripherals. We used the Xilinx Spartan-3AN starter kit (Digi-Key, 122-1528-ND). This moderately-priced platform is adequate for emulating the functionality of standard analog laser servo electronics. The Spartan-3AN starter kit incorporates a dual-channel, simultaneous-sampling ADC with 1.5 MS/s sampling rate per channel and a quad-channel DAC with 1.5 MS/s bandwidth shared across all four channels (referred to as the slow DAC). Additional commercially-available peripheral modules (Digilent, Pmod R2R) provide four 8-bit resistor-ladder DACs with 25 MS/s bandwidth each (referred to as the fast DACs); these DACs offer faster analog outputs but with reduced precision and high output impedance. The fast DACs connect with the FPGA evaluation board via an interface board (Digilent, FX2 Module Interface Board). Further details regarding the assembly and connectorizing of the FPGA board, the ADC, and both types of DACs are given in the appendix. We note that for laser frequency locking the total servo loop bandwidth is often a key parameter defining system performance. In our system, the slow DAC channels are used to directly control the laser (due their higher precision and lower output impedance relative to the fast DAC), while the fast DAC channels are used as monitors. Additionally, our design includes a dynamic bandwidth allocation unit which allows arbitrary distribution of the total slow DAC bandwidth among its four channels.

3 Principles of operation

The basic principles of operation of the system are shown in figure 2. The signal processing unit and the user interface unit interact via a set of control parameters, such as the user-defined modulation frequency and various gain settings. These control parameters and functionalities are described below. To aid in tuning the control parameters, each physical output may be easily switched among several monitor signals depending on the user-selected mode (see below). Our implementation allows on-the-fly system tuning and switching between modes. Changes to the control parameters do not disrupt the operation of the signal processing unit.

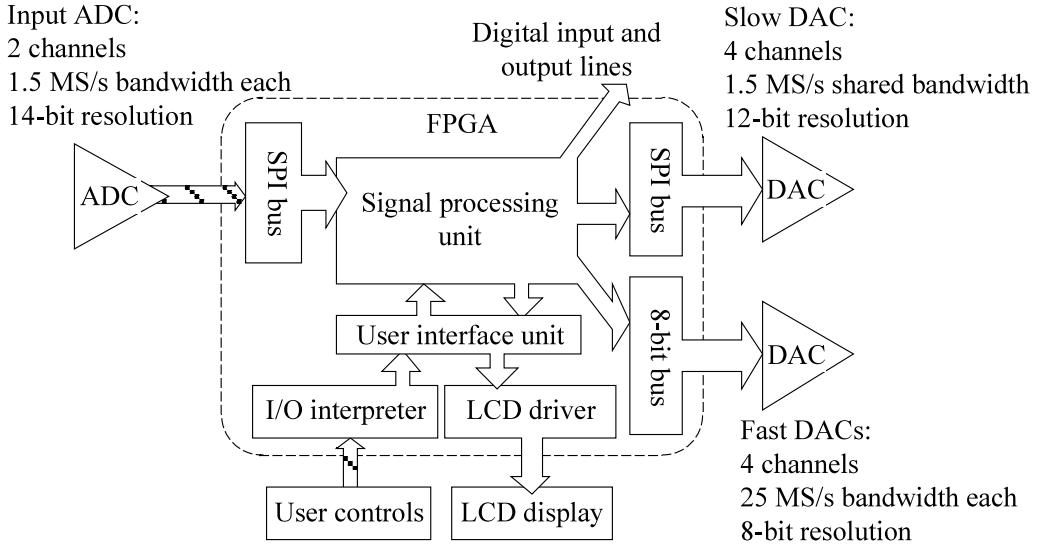


Figure 1. System design schematic. The system consists of the ADC, DACs, user controls, LCD display, and FPGA. The design is a system-on-a-chip, where the signal processing and user interface units occupy the same FPGA element as the Serial Peripheral Interface (SPI) bus, 8-bit parallel bus, LCD driver, and I/O interpreter for the controls. Individual units run independently and in parallel. Note that two kinds of DACs are used; a 12-bit, serial-controlled, four-channel DAC outputs at 1.5 MS/s (and this bandwidth is shared among all four channels), while four faster 8-bit resistor-ladder DACs each output at up to 25 MS/s. Multipurpose digital outputs are shown as well.

3.1 Signal processing

The signal processing unit includes a lock-in amplifier, a built-in waveform synthesizer with adjustable phase (for generating the lock-in modulation signal), two channels of PI servo control (one fast and one slow), and a ramp generator that can replace the servo outputs (to aid in finding a spectral line, as is typically needed for laser locking). During normal operation for laser frequency locking, the lock-in amplifier generates an error signal (proportional to the deviation between the laser frequency and the frequency corresponding to a resonant spectral peak), while the fast and slow servo controls provide feedback over different frequency ranges determined by the various filters (figure 2).

3.1.1 Modulation generation and lock-in amplifier

During operation, the modulation waveform synthesizer generates a periodic modulation drive signal that can be used to implement phase-sensitive (lock-in) detection. For laser frequency locking, this signal may drive an acousto-optic modulator (AOM) which dithers the laser frequency in the vicinity of a resonant optical transmission peak of a Fabry-Perot cavity.

The input signal from the experiment is demodulated by multiplication with a phase-shifted version of this modulation signal and then low-pass filtered (see section 4.2). We refer to the demodulated signal as the error signal (since in a servo-locking application, this is proportional to the deviation between the measured process variable and its setpoint). The modulation waveform's amplitude and phase shift can be tuned with 12 bits of depth. The waveform can be a sinusoid

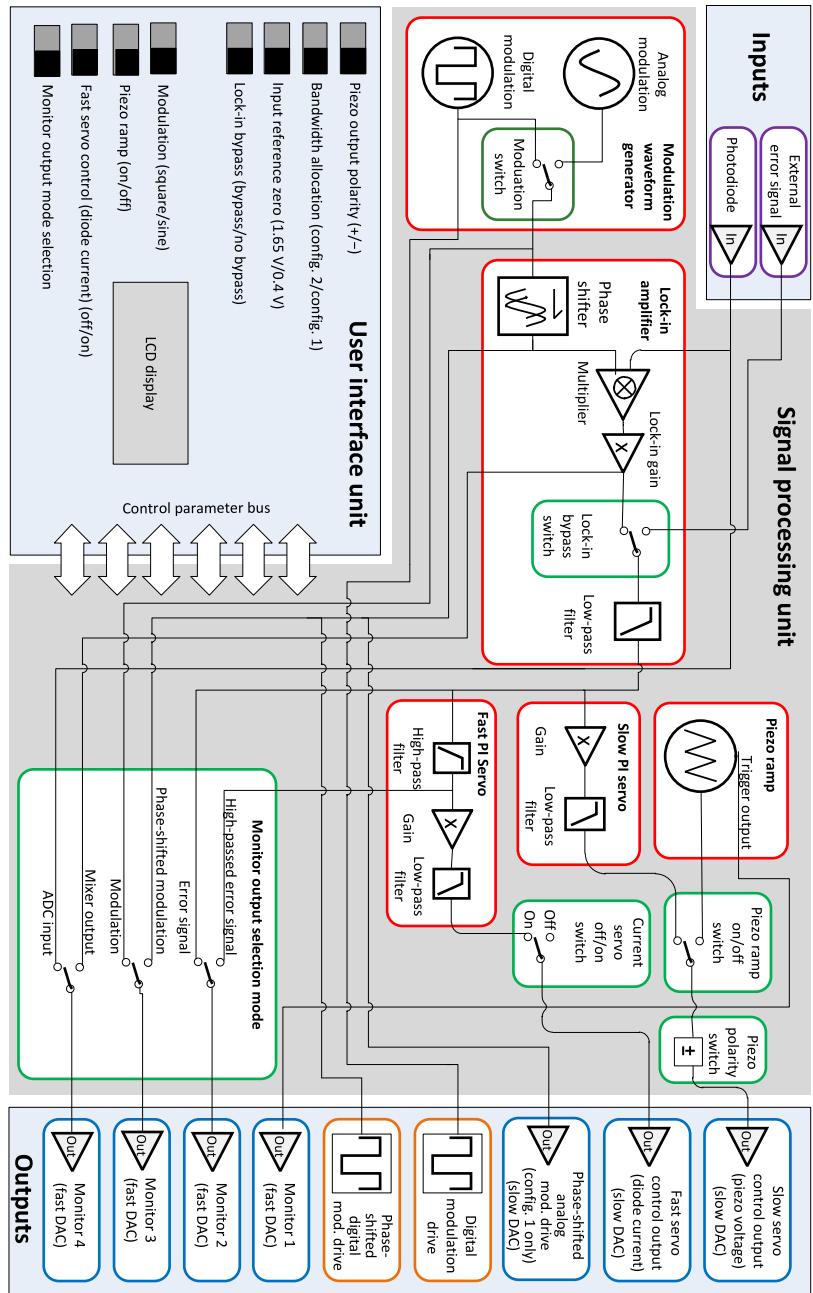


Figure 2. Signal processing conceptual schematic diagram. The main outputs of this unit include the modulation output, the fast servo control (diode current) output and the slow servo control (piezo voltage) output. The main input accepts either a modulated signal (sent to the lock-in for demodulation) or an error signal that can be used directly by the servo control (in which case the lock-in is bypassed). The physical output channels for the monitors each support several different signals, of which one is chosen given the user-selected mode. This mode is selected using a set of physical switches. Four monitor outputs using fast resistor-ladder DACs are available to display internal signals from various critical points in the processing stream, e.g., the mixer signal, the error signal, the signal from the high-pass filter, and so on.

or a square wave; the sine wave is produced on an analog output port, while the square wave is produced on a separate digital output port that is fixed at 3.3 V peak-to-peak amplitude.

There are two ways of outputting the sinusoidal modulation signal. In configuration 1, the sinusoidal modulation is output on channel 1 of the 12-bit quad-channel DAC. In this mode, 68% of the 1.5 MS/s bandwidth is allocated to the modulation output, leaving 30% for the fast servo control (diode current) output and 2% for the slow servo control (piezo voltage) output. In configuration 2, the sinusoidal modulation signal is instead output on one of the 25 MS/s 8-bit DAC channels shown in figure 1. Because this DAC has high output impedance (see below), an additional analog buffer circuit is needed to drive most actuators when using this output. In configuration 2, approximately 98% of the 1.5 MS/s, 12-bit quad DAC bandwidth can be appropriated for the fast servo control (diode current) output, with the balance used for the slow servo control (piezo voltage) output. This effectively increases the maximum loop bandwidth of the system. A physical slide switch allows the user to change between configuration 1 and 2.

The analog input capture circuit accepts signals in the range 0.4–2.9 V. The mathematical reference zero can either be placed at 0.4 V or 1.65 V. A physical slide switch allows the user to switch the input reference level between these two values. This allows the system to be used either with unipolar input signals (e.g., the signal from a photodiode following a Fabry-Perot cavity) or with bipolar error signals (if they have been offset by 1.65 V). The latter mode is useful when bypassing the internal lock-in amplifier and providing an external error signal.

The lock-in gain and low-pass filter are implemented as a single unit. The gain has a 16-bit dynamic range and is placed before the low-pass to prevent sudden jumps of the error signal that can disrupt a locked servo loop. The filter's cutoff frequency can be tuned from 0.22 mHz to 238 kHz. The floating point design uses 12 significant bits and a 2-bit exponent; the base for the exponent is 1/64. Floating-point operations are implemented as a combination of fixed-point operations. This design results in good tuning range while taking full advantage of the high efficiency of fixed-point multiplication and addition inside an FPGA.

Since the analog inputs run at a much slower speed (1.5 MHz per channel) than the main clock of the FPGA (50 MHz), we minimized resource consumption by implementing a pipelined gain-filter unit, using a single process to apply three low-pass filters and one high-pass filter. The initial latency of the pipeline is four clock cycles, and each additional sample takes two clock cycles. Details of this implementation can be found in section 4.

To connect the system to a typical laser system, a few additional electronics are needed. First, because the analog capture circuit of the FPGA board only takes in signals from 0.4 V to 2.9 V, one typically needs an offset amplifier for the input signal (which might extend to ground or be bipolar). In addition, a piezo controller with a variable offset (as typically provided with commercial tunable lasers) is needed to tune the laser to a specific resonant optical transmission line used as the reference for locking. Finally, in our implementation the modulation output is summed with a DC offset signal before being used to drive an AOM that is external to the laser.

3.1.2 Servo control

The two-channel servo control provides both a slow and a fast signal for feedback in a servo loop. The slow servo control channel takes in the error signal, multiplies it with a user-defined gain, and applies a low-pass filter. The fast servo control channel operates similarly but also includes a high-

pass filter before the low-pass filter. This allows any low-frequency feedback to be provided solely by the slow servo control channel, which is expected to have a larger range than the fast servo control channel. In our test setup, the fast servo control is used to drive the injection current of the laser diode (which provides a small dynamic range in the laser frequency), while the slow servo control is used to drive the piezoelectric actuator of the diffraction grating in the laser's external cavity (with a large dynamic range). However, the same two-channel servo loop can potentially be used in a variety of other servo-locking schemes.

To use the system as a dedicated two-channel PI servo controller (with an error signal generated externally rather than via the internal lock-in), the lock-in amplifier can be bypassed by flipping a physical slide switch. In this mode of operation, the output bandwidth allocation ratios can be changed via another physical slide switch to provide $\sim 98\%$ of the bandwidth (1.46 MS/s) to the fast PI channel and $\sim 2\%$ (35 kS/s) to the slow PI channel (shown in figure 3).

3.1.3 Ramp generator

A ramp mode is available in the system for tuning and locking to a specific resonant optical transmission peak. In the ramp mode, the two servos are disabled. Instead, a triangle wave is sent to the slow servo control (piezo voltage) output. This mode can be toggled on and off with an onboard switch. The amplitude and the frequency of the ramp can be adjusted using the LCD and onboard knobs, similar to the other control parameters. A 3.3 V digital square wave synchronized with the ramp is available via monitor output 1 for use as a trigger signal.

3.1.4 Monitor outputs

For help in debugging the system operation, we implemented monitor check points throughout the signal processing sequence. Using the fast DAC outputs (figure 1), the user can access seven optional checkpoints for display on an oscilloscope. A physical slide switch toggles the outputs between two sets of monitor signals (see figure 3).

3.2 User interface

The operation modes and control parameters in the signal processing unit are modified via an intuitive user interface. The arrangement of the physical controls and their corresponding functions are shown in figure 3. Slide switches allow the user to toggle between different operation modes. The control parameters, shown in table 1, are accessed in a circular fashion (figure 4) using two pushbuttons to select either the previous or next control parameter. The LCD displays the current value of the selected control parameter. The selected control parameter may be adjusted in both coarse and fine increments using the rotary knob.

The schematic of the user interface is shown in figure 5. The control parameters are stored in a parameter vector, which is used by the signal processing unit for its calculations. A list of all control parameters, along with their available ranges, bit depths, and scaling of tuning, is presented in table 1.

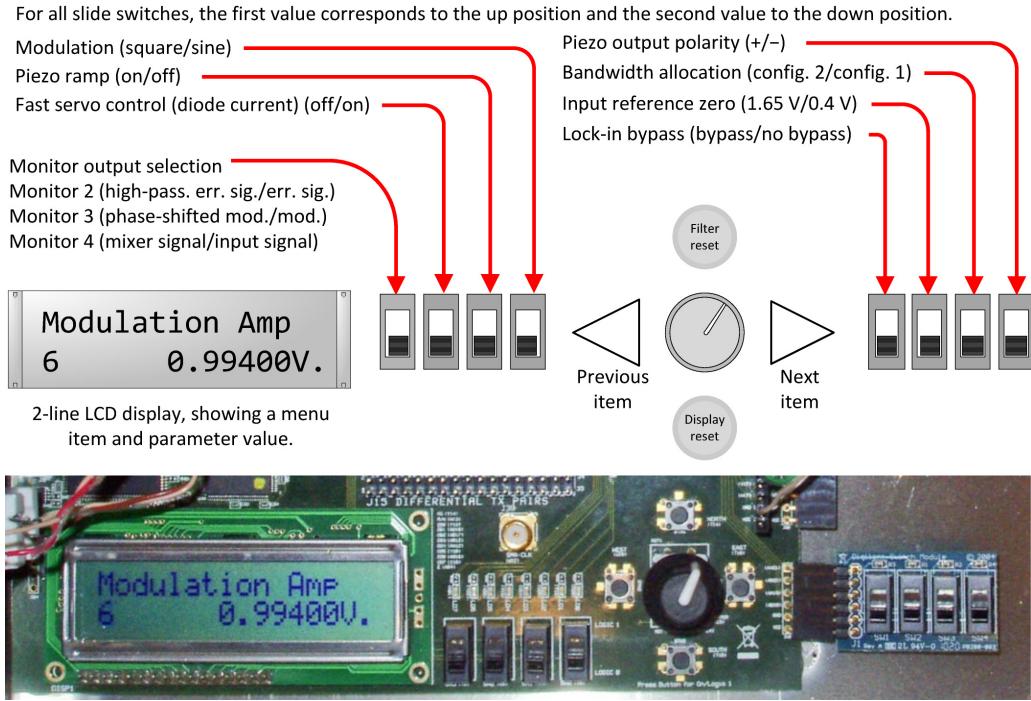


Figure 3. Controls and display (from left to right): A two-line LCD allows for parameter display. The left four mode switches control the monitor outputs, the fast servo (diode current), the piezo ramp, and the modulation waveform. Next to the rotary knob are the previous and next menu selection buttons and, above and below respectively, the global filter reset and display reset buttons. The right four mode switches are lock-in bypass, input reference zero (1.65 V/0.4 V), bandwidth allocation configuration, and piezo output polarity. Details of the functions of these controls are available in the user manual.

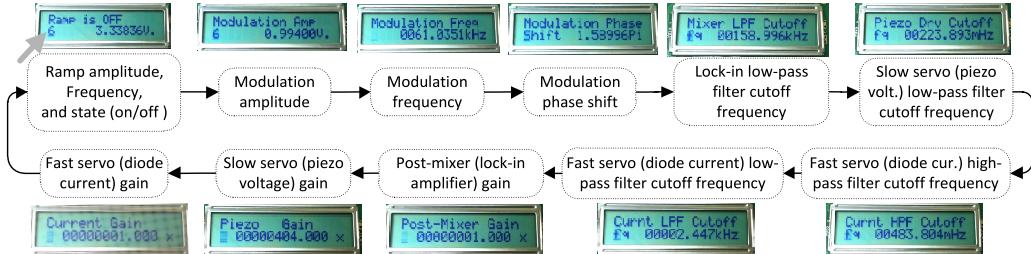


Figure 4. Menu items. The diagram shows the control parameters arranged in a loop, with a screenshot of the LCD display for each parameter. The gray arrow indicates the ramp frequency exponent, which is displayed for the both the ramp and modulation amplitude screens. The ramp frequency in Hz is $\approx 1500/2^x$, where x is the ramp frequency exponent. The default ramp frequency exponent of 6 corresponds to a ramp frequency of ≈ 24 Hz.

4 Implementation

In this section we provide details of how some specific elements in the signal processing unit are implemented.

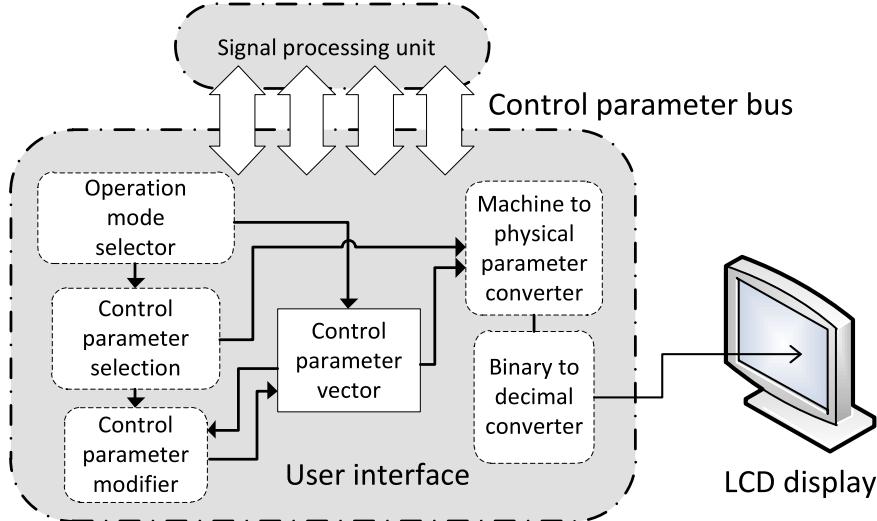


Figure 5. User interface schematic. The diagram shows the data flow for selection, presentation, and modification of control parameters and operation modes. Mode-dependent monitor outputs are shown in figure 2 as part of the signal processing unit.

Table 1. Control parameters.

Parameter	Range	Bit depth	Tuning scale	Notes
Modulation amplitude	0–3.3 V	12-bit	Linear	Fixed point
Modulation frequency	3 kHz – 12.5 MHz	12-bit	Linear	Fixed point
Modulation phase offset	0– 2π	14-bit	Linear	Fixed point
Lock-in low-pass filter	0.22 mHz – 238 kHz	12-bit*	Exponential	Floating point
Slow servo (piezo voltage) low-pass filter	0.22 mHz – 238 kHz	12-bit*	Exponential	Floating point
Fast servo (diode current) high-pass filter	0.22 mHz – 238 kHz	12-bit*	Exponential	Floating point
Fast servo (diode current) low-pass filter	0.22 mHz – 238 kHz	12-bit*	Exponential	Floating point
Post-mixer (lock-in amplifier) gain	0.015–1024	16-bit	Exponential	Fixed point
Slow servo (piezo voltage) gain	1–65536	16-bit	Exponential	Fixed point
Fast servo (diode current) gain	0.015–1024	16-bit	Exponential	Fixed point
Ramp amplitude	0–3.3 V	11-bit	Linear	Fixed point
Ramp frequency	0.046 Hz – 1.5 kHz	4-bit	Exponential	Increments by 2 \times

*Parameter has 12-bit resolution but is tunable over a 30-bit dynamic range.

4.1 Modulation waveform synthesizer

A schematic of the modulation waveform synthesizer is shown in figure 6. A 14-bit register records the accumulated phase. During each clock cycle, the accumulated phase register increases by a step size equal to 2^{14} times the user-defined modulation frequency divided by the FPGA clock frequency. The accumulated phase register value is then used to look up the corresponding waveform value from a 12-bit-precision sinusoidal lookup table. This waveform is multiplied by a constant to achieve the user-defined peak-to-peak modulation amplitude, and the result is output as the modulation signal. If a square modulation waveform is chosen, the output is determined by whether the phase counter is greater than or less than π . Values greater than π result in a Boolean high output, and values less than π result in a Boolean low output.

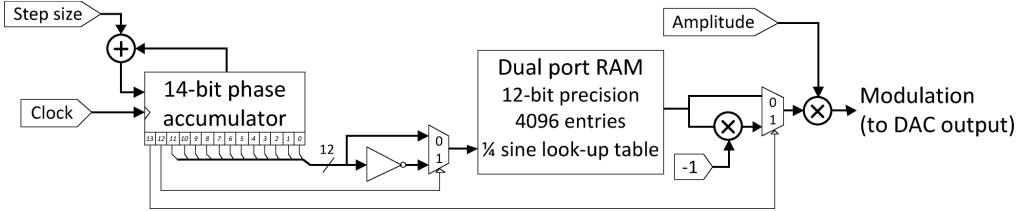


Figure 6. Modulation waveform synthesizer. The diagram shows a schematic of the synthesizer implementation. Dual-port RAM is used to implement the quarter-sine-wave look-up table. The 14-bit phase is translated to address the 12-bit quarter-wave look-up table in two steps. First, depending on the value of the second-highest-order bit, the value of the phase may be subtracted from π . Second, depending on the value of the highest-order-bit, the output value of the look-up table may be inverted. The phase-shifted modulation signal (not shown) is achieved using the same implementation as depicted above except that a user-defined phase is added to the accumulator, and this addresses the other port of the dual-port RAM. Implementation of the square wave modulation is not shown here.

The lookup table has 4096 (or 2^{12}) entries. Because only a quarter period of the sine wave is stored, the number of entries is appropriate to match the 14-bit phase accumulator register. In our implementation, an embedded dual-port block RAM is used to provide this lookup table, allowing two read-outs per clock cycle. Hence, in addition to the non-offset waveform, a phase-shifted waveform is produced by simply adding a user-defined phase offset to the pointer which retrieves the value from the lookup table.

To eliminate glitches in the square wave modulation mode that can otherwise occur when the user changes the modulation frequency, the phase accumulator is reset to zero every time this frequency parameter is changed.

4.2 Filter unit with gain

As shown in figure 2, four filters total are required among the output of the lock-in multiplier and the slow servo control (piezo voltage) and fast servo control (diode current) outputs. A multiplexed architecture is therefore built to use a single piece of hardware for the four filters sequentially as shown in figure 7a–b. A four-stage pipeline increases the processing efficiency from 16 clock cycles for four filters to 10 clock cycles, less than one third the time interval between consecutive samples from the ADC. As a result, the signal processing occurs at the maximum rate determined by the analog input devices.

Each filter has a cutoff frequency that can be set from 0.22 mHz to 238 kHz. Inside the filter module, a constant proportional to the cutoff frequency is stored as a 14-bit floating point number, with 12 bits devoted to the value and 2 bits devoted to the exponent. Floating point calculations are done via a multiplier which can have a gain of either 1, $1/64$, $1/64^2$, or $1/64^3$, depending on the value of the 2-bit exponent. This allows for a total dynamic frequency range of 2^{30} . Each filter module also has a pre-filter gain that can be changed within a 16-bit dynamic range, and each module can be set to function as either a first-order high-pass filter or a first-order low-pass filter.

Our design uses Infinite Impulse Response (IIR) filters [15]. The equations for IIR low-pass and high-pass filters mimicking single-stage low-pass and high-pass RC filters (with the same RC

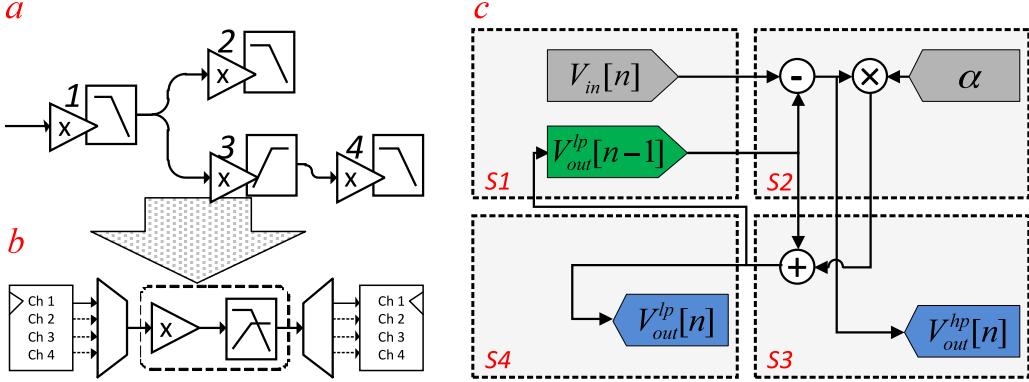


Figure 7. Pipelined IIR low-pass and high-pass filter module with pre-filter gain. a. The chronological sequencing of the four filter modules in the signal processing unit. b. These four units share the same piece of hardware and are multiplexed in time into four distinct filter modules. c. Low-level filter design. The diagram shows schematics of the filter processing flow for a single filter. S1–4 in red mark the four distinct stages. Inputs are shown in gray, while outputs are shown in blue. The feedback from the low-pass output is stored in the output buffer (green) before the next sample is acquired. Here the module is shown when used as a single filter. For clarity, neither the pre-filter gain nor any complexity stemming from α 's representation as a floating point number is shown.

time constant) may be written as (see appendix)

$$V_{out}^{lp}[n] = \alpha V_{in}[n] + (1 - \alpha) V_{out}^{lp}[n - 1] \quad (4.1)$$

and

$$V_{out}^{hp}[n] = (1 - \alpha) (V_{in}[n] - V_{out}^{lp}[n - 1]), \quad (4.2)$$

where $V_{out}^{hp}[n]$, $V_{out}^{lp}[n]$ and $V_{in}[n]$ are the output of the high-pass filter, the output of the low-pass filter, and the input to the filters, respectively. We have also used $\alpha \equiv T_s/(\tau + T_s)$, where T_s is the time interval between samples and $\tau = RC$ is the filter time constant. The low-level implementation of these filters can be found in figure 7c. The gain-filter complex is divided into four stages: the input stage (S1), the gain stage (S2), the floating-point multiplying and summing stage (S3), and the output stage (S4). The pipeline is implemented such that when data from one channel enters S4, data from the next channel simultaneously enters S1, as shown in figure 7c. The pipelined design and availability of up-to-date values of prior filters to subsequent filters acts to minimize the overall signal processing latency in the case of several filters applied sequentially. When implementing equation (4.2) for the high-pass filter, the multiplication by the factor $(1 - \alpha)$ was ignored for computational efficiency reasons; this factor is very close to 1 for all reasonable values of T_s and τ .

A global filter reset button allows recovery from any integrator windup.

4.3 Dynamic output bandwidth allocation

Sampling bandwidth is a limiting factor of signal fidelity and control latency and to a great extent determines the performance of the servo-lock provided by this system. On this FPGA board, the slow DAC shares its $B = 1.5$ MS/s bandwidth among four output channels. Because some signals

Table 2. Shared DAC bandwidth allocation. In configuration 2, one of the fast DAC channels is used for the modulation output.

Channel	Function	Frequency	Configuration 1	Configuration 2
Channel 1	Modulation output	1–200 kHz	1.02 MS/s (68%)	0 (0%)
Channel 2	Not used	N/A	0 (0%)	0 (0%)
Channel 3	Slow servo (piezo voltage)	1 kHz	30 kS/s (2%)	30 kS/s (2%)
Channel 4	Fast servo (diode current)	20 kHz	450 kS/s (30%)	1.47 MS/s (98%)

naturally require faster updating than others, a method to arbitrarily allocate bandwidth among these four channels is implemented by the Dynamic Bandwidth Allocation Unit (DBAU).

The DBAU is implemented as follows: each channel i is assigned a 10-bit integer n_i and receives total bandwidth $n_i B / 1024$, with the condition that $n_1 + n_2 + n_3 + n_4 = 1024$. Each channel i is also assigned a counter whose value is denoted by N_i . Following each DAC update, each counter value N_i is increased by n_i . Next, the new values N_i are examined sequentially. If the new value N_i of the given channel is greater than 1024, then that channel is chosen to be updated; simultaneously its N_i is reduced by 1024, and the other channels receive no update. On the other hand, if the new value is less than 1024, then the above algorithm repeats on the next channel. If after examining all four channels no channel is selected, the most recently updated channel will again be updated. In the absence of a channel being updated, the output remains at the previous value. Since there are 32 FPGA clock cycles for every one DAC update, there is ample time for the DBAU to perform the above calculations between DAC updates.

The current FPGA configuration file allows for use of the DBAU in two different configurations. In configuration 2, the modulation signal is output on the fast DAC (with 8-bit precision), leaving the DBAU to allocate the slow DAC bandwidth between only the slow servo control (piezo voltage) and the fast servo control (diode current). For applications requiring the greater precision of the slow DAC (12 bits) for the modulation signal, configuration 1 allocates the necessary bandwidth to output the modulation on the slow DAC as well. Table 2 details the exact allocation and resulting bandwidths for configurations 1 and 2. While switching between configurations 1 and 2 may be done with a slide switch on the development board, additional custom bandwidths allocations are possible by editing the source code.

4.4 FPGA usage

After the final design was implemented, we checked the resource usage of our design using the Xplorer tool in the Xilinx ISE software (included in the Spartan-3AN starter kit). With the implementation described here, the design occupies 35% of the 5888 available slices, 3 of the 20 block RAMs, and 9 of the 20 high-speed multipliers in the FPGA. Specifically, the IIR filter module occupies 4% of the lookup tables and 4 of the multiplier cores. The overall power consumption is 0.026 W.

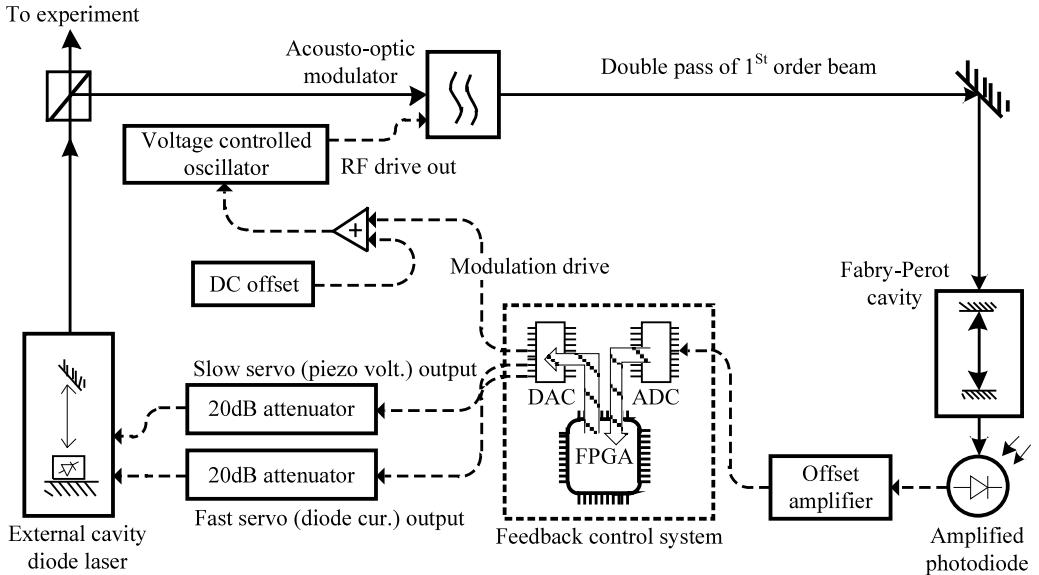


Figure 8. Test setup to demonstrate laser frequency servo locking. The fast and slow outputs are connected to the diode injection current and piezo drive inputs, respectively, of the laser controller. The input signal is drawn from the amplified photodiode and offset (using a stand-alone amplifier) by 400 mV to place the unipolar photodiode signal entirely within the 0.4–2.9 V input range of the ADC. The input to the VCO is provided by summing the modulation waveform output drive with a DC signal in a separate amplifier.

5 Performance

5.1 Test setup

To test the performance of this system in a typical application, we used it to lock the frequency of a commercial external-cavity diode laser (Toptica Photonics, DL100)² to a resonant optical transmission peak of a confocal Fabry-Perot cavity. The schematic of the test setup is shown in figure 8. The laser controller includes two voltage inputs to change the laser frequency, one via a piezo that changes the external cavity length (slow but with a large dynamic range) and one via the laser diode current (fast but with a small dynamic range). On the output end, a 20 dB Mini-Circuits attenuator is used between both outputs and the laser controller inputs; this improves the signal-to-noise ratio by taking advantage of the DAC's full dynamic range.

Modulation of the laser frequency for the purpose of generating the lock-in error signal is accomplished with an AOM external to the laser. The AOM drive frequency is generated by a voltage-controlled oscillator (VCO) whose input voltage is dithered by summing the modulation drive signal from our system with a DC offset in a home-built circuit. The Fabry-Perot cavity (Toptica Photonics, FPI 100) has transmission peaks with spectral width of ~2 MHz. An amplified photodiode (Thorlabs, PDA36A) transforms the optical intensity transmitted through the Fabry-Perot cavity into a voltage signal. The frequency response of the photodiode extends well beyond our loop bandwidth. In our test setup, the modulation waveform is provided via channel 1 of the

²The DL100 was chosen as the test ECDL because it is representative of typical ECDLs used in atomic physics experiments. The DL100 is a commercial laser similar in design to the original Littrow-Hänsch ECDL [16], the most widespread ECDL design to date.

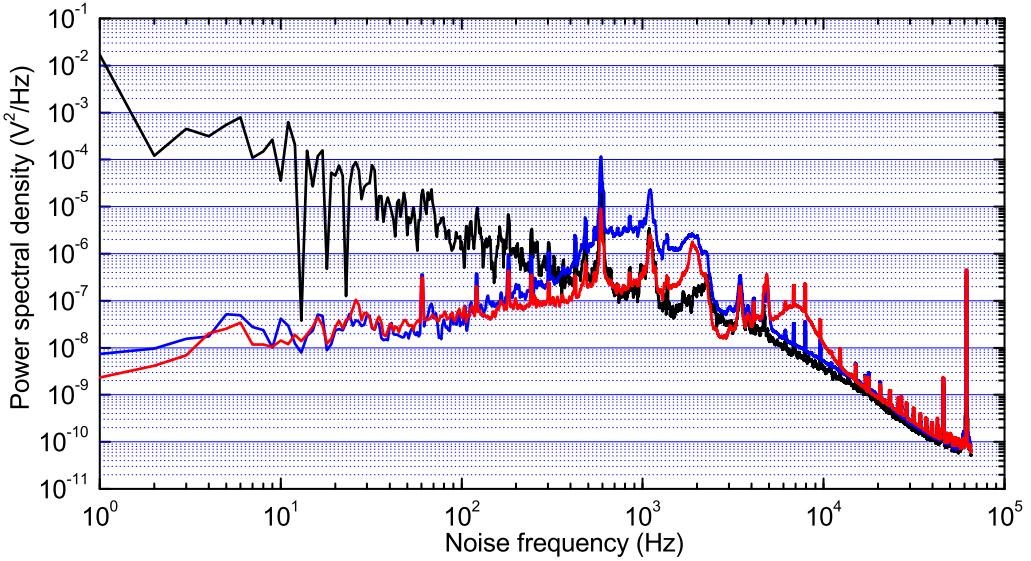


Figure 9. Performance of the system as a laser frequency servo lock. The plot shows the power spectral density of frequency fluctuations of the laser under several conditions. Black depicts the free running laser, blue the laser locked only with the slow (piezo) servo loop, and red the laser locked with both the fast (current) and slow servo loops. Over the band from 1 Hz to the 62 kHz modulation frequency, the measured FWHM laser linewidths are 733 kHz, 411 kHz, and 188 kHz, respectively. The data are binned such that the point at frequency f is the average over a window of width $f/50$ centered at f .

slow DAC (configuration 1). We used 62 kHz as our modulation frequency and set the lock-in low-pass filter cutoff at 9 kHz. The effective loop bandwidth of the system is therefore limited to around 7 kHz, much lower than the sampling rate of the ADC circuit.

The laser linewidth (relative to the cavity) is determined from the integrated power spectrum of the error signal [17]. In particular, the RMS frequency deviation $\Delta\nu_{\text{RMS}}$ within the frequency band from $f_1 = 1$ Hz to $f_2 = 62$ kHz is given by

$$\Delta\nu_{\text{RMS}} = \sqrt{\eta^2 \int_{f_1}^{f_2} P df},$$

where P is the power spectral density of the error signal (in V^2/Hz) and $\eta = 2 \text{ MHz/V}$ is the voltage-to-frequency conversion factor for our system (determined by ramping the laser frequency across the known free-spectral range of the cavity at a known rate and, with the same ramp rate, measuring the slope of the error signal in the vicinity of the locking peak). Assuming a Gaussian lineshape, the FWHM linewidth $\Delta\nu_{\text{FWHM}}$ is given by

$$\Delta\nu_{\text{FWHM}} = 2\sqrt{2\ln 2}\Delta\nu_{\text{RMS}}.$$

As shown in figure 9, with the laser frequency locked using both the fast and slow servos, we measured a $\Delta\nu_{\text{FWHM}} \approx 200$ kHz in our test setup.

6 Conclusions and possible future extensions

In this work, we demonstrate a laser frequency locking circuit that incorporates a waveform synthesizer, a lock-in amplifier, a ramp generator, and two channels of servo control with widely tunable filters and gains, using an inexpensive, off-the-shelf FPGA prototyping board. This simple-to-replicate system reproduces all the functionality of many standard analog laser locking circuits and provides an easy-to-use interface. The throughput of the signal processor is 1.5 MS/s, with less than 200 ns processing latency. Our test setup achieved a loop bandwidth of 7 kHz using the integrated lock-in amplifier and the two channels of servo control. Hundreds of kilohertz of loop bandwidth should be possible by using an external lock-in amplifier and selecting the lock-in bypass setting on the FPGA or by using the fast resistor-ladder DAC to output the modulation; however, we have not yet tested the system for use in these configurations.

Several natural improvements seem possible in the near future. Faster ADCs and DACs could easily be incorporated into the system, opening the possibility to use the system at much higher loop bandwidths. With our current design, we believe it would be possible to implement systems that operate with MHz loop bandwidths. Even faster performance might be possible with newer and more powerful FPGAs. Such improvements would open the door to using Pound-Drever-Hall-type locking schemes [3] and hence attaining much tighter locks.

Another possible improvement is the addition of a remote control option to allow a central computer to communicate with one or more locking circuits. The particular FPGA board used here includes an onboard Ethernet port, and the FPGA source code could be modified to support querying and modifying the values of control parameters by a networked computer. Remote control could be especially useful for a user setting up several locking circuits at once. However, since control parameters are typically optimized once for a given laser locking setup and varied rarely thereafter, we did not consider this option critical at this stage of development of the system, and so remote control has not yet been implemented.

Acknowledgments

This research was supported by NSF, AFOSR, and ARO. We thank Lin Sapack and Andrew Mack for assistance in the early stages of development of the system.

A Assembly, mounting, and connectorization

The FPGA board is assembled, mounted, and connectorized into a usable instrument box as shown in figure 10. The FPGA board is mounted to the box via glue on the four corners of the FPGA board. Power and programming cables (USB) are passed through holes in the box to the native connectors onboard. Since the FPGA board's native physical inputs and outputs are DIP header pins, we use home-made adapter cables with a BNC jack on one end and a DIP socket on the other end to allow for easy access to signals. The interface board allows connection to the four resistor-ladder DACs. Additional slide switches (Digilent, PmodSWT) are connected directly to the FPGA board.

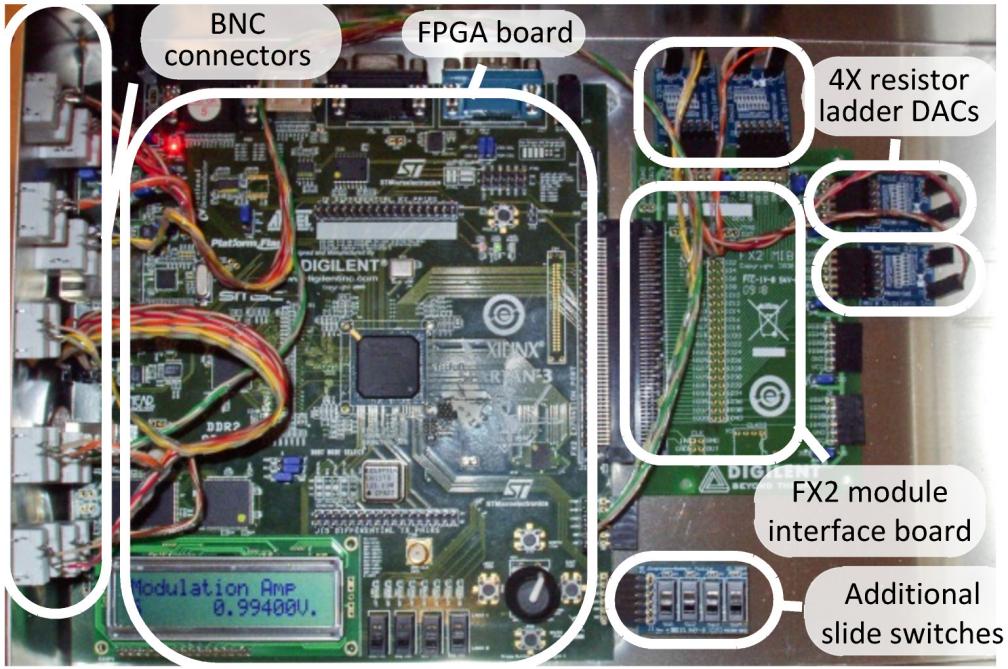


Figure 10. Photo of the final assembled system. Note the interface board plugged in to the FX2 connector on the FPGA board and the four resistor-ladder DACs attached to the interface board. An additional slide switch card is also added to accommodate various features.

B Electronic characteristics

All relevant electronic characteristics of the FPGA board and its accessory modules are available from the manufacturer. For convenience, we point out a few of the salient features that determine what devices can be connected to the input and output signals without difficulty.

B.1 Slow DAC

The Linear Technology LTC2624 DAC used on the Digilent FPGA board provides rail-to-rail voltage outputs of 0 V to 3.3 V, with a slew rate of $1 \text{ V}/\mu\text{s}$, and is capable of driving loads with input impedance $> 150 \Omega$.

B.2 Fast DAC

The fast DAC outputs are provided by the Digilent PmodR2R resistor-ladder DAC, which provides 8-bit digital to analog conversion at up to 25 MHz. From the design schematics we estimate that this DAC has an output impedance greater than $10 \text{ k}\Omega$ and therefore should be used only with an oscilloscope or other high input-impedance device, such as a buffer amplifier. The high output impedance of the Pmod R2R DAC also means that looking at fast signals with this device is impossible if connected via coaxial cables of any significant capacitance.

B.3 Analog capture circuit

The analog capture circuit on the Digilent board consists of an LTC1407 serial, dual-channel, 14-bit ADC and an LTC6912 dual-channel, programmable-gain pre-amplifier. The LTC1407 ADC has two 1.5 MS/s, simultaneously-sampled inputs with a total range of 2.5 V. They are connected to the dual programmable amplifier, which is set to have a unity gain in this design. The input impedance is 10 k Ω when the programmable preamplifier is set to gain 1 (and lower at higher gains). The input has an idle potential of 1.65 V when not connected.

C RC filter equations in discrete time

The equation for an RC low-pass filter in continuous time is given by

$$\frac{d}{dt}V_{\text{out}}^{\text{lp}}(t) = \frac{1}{RC} \left(V_{\text{in}}(t) - V_{\text{out}}^{\text{lp}}(t) \right), \quad (\text{C.1})$$

where $V_{\text{in}}(t)$ and $V_{\text{out}}^{\text{lp}}(t)$ are the input and output voltages of the filter respectively as a function of time t . Defining $\tau \equiv RC$, equation (C.1) can be expressed in discrete time by the difference equation

$$V_{\text{out}}^{\text{lp}}[n] = V_{\text{out}}[n-1] + \frac{T_s}{\tau} \left(V_{\text{in}}[n] - V_{\text{out}}^{\text{lp}}[n] \right), \quad (\text{C.2})$$

where T_s is the time interval between samples. Substituting $\alpha \equiv T_s/(\tau + T_s)$ yields

$$V_{\text{out}}^{\text{lp}}[n] = \alpha V_{\text{in}}[n] + (1 - \alpha) V_{\text{out}}^{\text{lp}}[n-1]. \quad (\text{C.3})$$

Now consider the equation for an RC high-pass filter in continuous time,

$$V_{\text{out}}^{\text{hp}}(t) = RC \frac{d}{dt} \left(V_{\text{in}}(t) - V_{\text{out}}^{\text{hp}}(t) \right), \quad (\text{C.4})$$

where $V_{\text{out}}^{\text{hp}}(t)$ is the output voltage of the high-pass filter as a function of time. The analogous difference equation in discrete time is given by

$$V_{\text{out}}^{\text{hp}}[n] = \frac{\tau}{T_s} \left[(V_{\text{in}}[n] - V_{\text{in}}[n-1]) - \left(V_{\text{out}}^{\text{hp}}[n] - V_{\text{out}}^{\text{hp}}[n-1] \right) \right], \quad (\text{C.5})$$

where we have again made the substitution $\tau \equiv RC$ and introduced the time interval between samples T_s . Further manipulation yields

$$V_{\text{out}}^{\text{hp}}[n] = \left(\frac{\tau}{\tau + T_s} \right) \left(V_{\text{in}}[n] - V_{\text{in}}[n-1] + V_{\text{out}}^{\text{hp}}[n-1] \right). \quad (\text{C.6})$$

We now invoke the property that $V_{\text{in}}[n] = V_{\text{out}}^{\text{hp}}[n] + V_{\text{out}}^{\text{lp}}[n]$ for high-pass and low-pass filters with the same RC time constants and rewrite equation (C.6) as

$$V_{\text{out}}^{\text{hp}}[n] = (1 - \alpha) \left(V_{\text{in}}[n] - V_{\text{out}}^{\text{lp}}[n-1] \right). \quad (\text{C.7})$$

References

- [1] C.E. Wieman and L. Hollberg, *Using diode lasers for atomic physics*, *Rev. Sci. Instrum.* **62** (1991) 1.
- [2] H. Patrick and C.E. Wieman, *Frequency stabilization of a diode laser using simultaneous optical feedback from a diffraction grating and a narrowband fabry-perot cavity*, *Rev. Sci. Instrum.* **62** (1991) 2593.
- [3] R.W.P. Drever et al., *Laser phase and frequency stabilization using an optical resonator*, *Appl. Phys. B* **31** (1983) 97.
- [4] J.E.D. Morris and H.S. Johnston, *Digital phase sensitive detector*, *Rev. Sci. Instrum.* **39** (1968) 620.
- [5] L.G. Rubin, *The new(?) wave: digital lock-in amplifiers*, *Rev. Sci. Instrum.* **59** (1988) 514.
- [6] E. Iacopini, B. Smith, G. Stefanini and S. Carusotto, *Digital techniques applied to phase-sensitive detection*, *J. Phys. E* **16** (1983) 844.
- [7] P.K. Dixon and L. Wu, *Broadband digital lock-in amplifier techniques*, *Rev. Sci. Instrum.* **60** (1989) 3329.
- [8] R.W.M. Smith, I.L. Freeston, B.H. Brown and A.M. Sinton, *Design of a phase-sensitive detector to maximize signal-to-noise ratio in the presence of gaussian wideband noise*, *Meas. Sci. Tech.* **3** (1992) 1054.
- [9] L.A. Barragan, J.I. Artigas, R. Alonso and F. Villuendas, *A modular, low-cost, digital signal processor-based lock-in card for measuring optical attenuation*, *Rev. Sci. Instrum.* **72** (2001) 247.
- [10] M.O. Sonnallion and F.J. Bonetto, *A low-cost, high-performance, digital signal processor-based lock-in amplifier capable of measuring multiple frequency sweeps simultaneously*, *Rev. Sci. Instrum.* **76** (2005) 024703.
- [11] A. Restelli, R. Abbiati and A. Geraci, *Digital field programmable gate array-based lock-in amplifier for high-performance photon counting applications*, *Rev. Sci. Instrum.* **76** (2005) 093112.
- [12] S.J. Lascos and D.T. Cassidy, *Multichannel digital phase sensitive detection using a field programmable gate array development platform*, *Rev. Sci. Instrum.* **79** (2008) 074702.
- [13] J. Labaziewicz, P. Richerme, K.R. Brown, I.L. Chuang and K. Hayasaka, *Compact, filtered diode laser system for precision spectroscopy*, *Opt. Lett.* **32** (2007) 572.
- [14] A. Schwettmann, J. Sedlacek and J.P. Shaffer, *Field-programmable gate array based locking circuit for external cavity diode laser frequency stabilization*, *Rev. Sci. Instrum.* **82** (2011) 103103.
- [15] R.G. Lyons, *Understanding digital signal processing*, second edition, Prentice Hall, Upper Saddle River U.S.A. (2004).
- [16] L. Ricci et al., *A compact grating-stabilized diode laser system for atomic physics*, *Optic. Commun.* **117** (1995) 541.
- [17] E.C. Cook, P.J. Martin, T.L. Brown-Heft, J.C. Garman and D.A. Steck, *High passive-stability diode-laser design for use in atomic-physics experiments*, *Rev. Sci. Instrum.* **83** (2012) 043101.