Portland State University
PDXScholar

Physics Faculty Publications and Presentations | Physics

3-2018

A Virtual Lock-In Amplifier, Spectrum Analyzer, Impedance Meter and Semiconductor Analyzer Implemented on an SR7265 Hardware Target

Jianghua Bai
Portland State University

John L. Freeouf
Portland State University, sfrenkel@pdx.edu

Andres H. La Rosa
Portland State University, andres@pdx.edu

Follow this and additional works at: https://pdxscholar.library.pdx.edu/phy_fac

 Part of the Physics Commons
Let us know how access to this document benefits you.

Citation Details
Bai, J., Freeouf, J. L., La Rosa, A. (2018). A Virtual Lock-in amplifier, spectrum analyzer, impedance meter and semiconductor analyzer Implemented on an SR7265 Hardware Target. Journal of Measurement Science & Instrumentation. Vol. 9 Issue 1, p50-58. 9p.

This Article is brought to you for free and open access. It has been accepted for inclusion in Physics Faculty Publications and Presentations by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

A Virtual Lock-in amplifier, spectrum analyzer, impedance meter and semiconductor analyzer Implemented on an SR7265 Hardware Target

Jianghua Bai, John L. Freeouf, Andres La Rosa

Portland State University, the Physics Department

1719 SW 10th AVE, Portland, OR 97201, USA

Abstract:

Lock-in amplifiers are used to detect and measure very small AC signals down to the range of nVs. Accurate measurements may be made even when the small signals are buried by noise sources thousands of times larger. With the digital signal processing (DSP) technology involved in modern instrumentation, a lock-in amplifier is more versatile than sensing and recover small signals. Combining the virtual instrumentation technology, we reorganize the functional blocks of a programmable lock-in amplifier and build it as a virtual spectrum analyzer, virtual impedance meter, virtual network analyzer, virtual semiconductor parameter analyzer, signal generator, etc. A 4 layer model is used to implement these virtual instruments. The same virtual instrument can also be implemented on a general purpose FPGA developing board.

Keywords: Virtual Lock-in amplifier, virtual spectrum analyzer, virtual impedance meter, virtual semiconductor analyzer, 4 layer model

1. Introduction:

In this paper Signal Recovery 7265 lock-in amplifiers will be used as the hardware target. Lock-in amplifiers from other manufactures or a home-made lock-in amplifier can be easily integrated into the platform following the steps shown in this article.

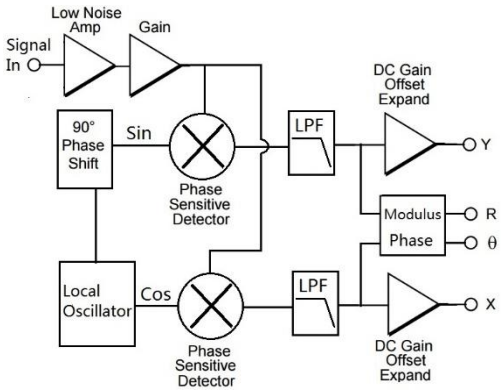

Fig 1, the functional block of a dual phase lock-in amplifier

2. How does a typical dual phase lock-in amplifier work?

A lock-in amplifier multiplies the input signal by a pure cosine or sine wave at the reference frequency. All components of the input signal are multiplied by the reference simultaneously. Theoretically, cosine and sine waves of differing frequencies are orthogonal, i.e. the average of the product of two sine or cosine waves is zero unless the frequencies are exactly same. The product of this multiplication yields a DC output signal proportional to the power of the signal component whose frequency is exactly locked to the reference frequency. The low pass filter which follows the multiplier provides the averaging which removes the AC output signals, which are the AC part of the frequency addition and the products of the reference with components at all other frequencies. [1] [2]

The mathematical details are shown below.

Suppose that the amplified input signal flows into the PSD is modeled by,

$$\sum_{i=0}^{\infty}[V_i \cos(\omega_i t + \theta_i)]$$

Where V_i and θ_i are the amplitude and phase respectively of the signal component at frequency ω_i . The PSDs multiply the reference signal with the input signal. Along the in phase signal flow path, the multiplication results

$$\mathrm{I} = \mathrm{V}_r \cos(\omega_r t + \theta_r) \sum_{i=0}^{\infty}[V_i \cos(\omega_i t + \theta_i)]$$

Following the trigonometrical identities, one has

$$\mathrm{I} = \frac{1}{2}\mathrm{V}_r \sum_{i=0}^{\infty} V_i[\cos(\omega_i t + \omega_r t + \theta_r + \theta_i) + \cos(\omega_i t - \omega_r t + \theta_i - \theta_r)]$$

Then the signal flows into the low pass filter, which averages the AC signal to zero, then one can get

$$\mathrm{I} = \frac{1}{2}\mathrm{V}_r \sum_{i=0}^{\infty} V_i \cos(\omega_i t - \omega_r t + \theta_i - \theta_r)$$

As long as $\omega_i \neq \omega_r$, $\cos(\omega_i t - \omega_r t + \theta_i - \theta_r)$ is also an AC signal, which will be averaged to zero by the low pass filter. Indeed the final result shows at the terminal X is

$$\mathrm{X} = \frac{1}{2}\mathrm{V}_r V_i \cos(\theta_i - \theta_r)$$

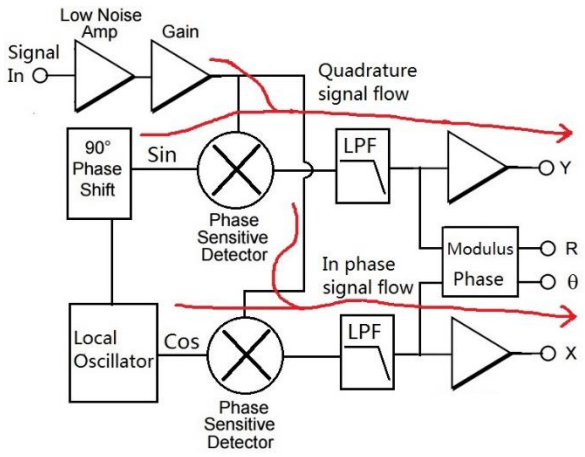

Fig 2, signal flows of a dual phase lock-in amplifier

Similarly, along the quadrature signal flow path, the PSD produces

$$\mathrm{Q} = \mathrm{V}_r \sin(\omega_r t + \theta_r) \sum_{i=0}^{\infty} [V_i \cos(\omega_i t + \theta_i)]$$

With the trigonometrical identities, one has

$$\mathrm{Q} = \frac{1}{2} \mathrm{V}_r \sum_{i=0}^{\infty} V_i [\sin(\omega_r t - \omega_i t + \theta_r - \theta_i) + \sin(\omega_i t + \omega_r t + \theta_r + \theta_i)]$$

After the low pass filter, one has

$$\mathrm{Q} = \frac{1}{2} \mathrm{V}_r \sum_{i=0}^{\infty} V_i \sin(\omega_r t - \omega_i t + \theta_r - \theta_i)$$

By properly setting the time constant, The DC signal shows at the Y terminal is

$$\mathrm{Y} = \frac{1}{2} \mathrm{V}_r V_i \sin(\theta_r - \theta_i)$$

From the calculations above, one can see the input consists of signal plus noise. Noise is represented as varying signals at all frequencies. The ideal lock-in only responds to noise at the reference frequency. Noise at other frequencies is removed by the low pass filter following the multiplier. Bandwidth narrowing is the primary advantage that a lock-in amplifier provides. Only signal components at the reference frequency result in an output. Thus this signal is singled out, even if it is buried by noise thousands of times larger at other frequencies. [1]- [6]

If the result is represented in a polar form, one has

$$\mathrm{R} = \frac{1}{2} \mathrm{V}_r V_i$$

$$\theta = \theta_r - \theta_i$$

3. The principle of a typical virtual lock-in amplifier.

With the digital signal processing (DSP) technology employed in modern instrumentation, one can implement the major signal flow part into a digital signal processor (DSP). The functional block will be something like the one shown in fig 3.

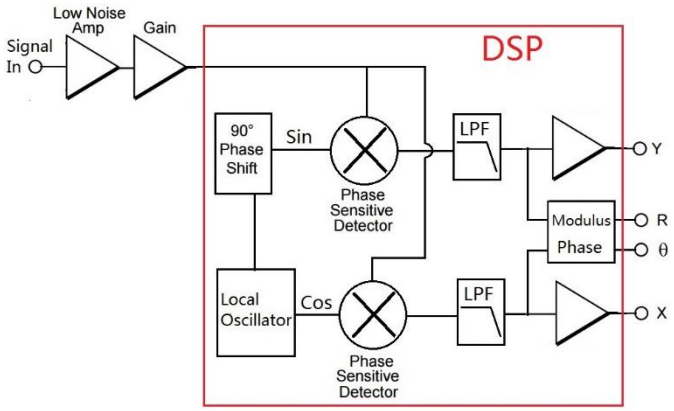

Fig3, the functional block of a DSP lock-in amplifier.

In virtual instrumentation, everything is controlled by a computer. One does not need to output the X, Y, R, θ real signals, then measure them and input them to the controller again. Instead, one can store the X, Y, R, θ information into the DSP FIFO, from which the controller can read directly. This way, one just save the DAC, signal regulation and amplifier parts and the future sensing parts. One can easily see the advantage of virtual instrumentation, by saving extra electric components and avoiding the extra errors associated with the analog parts and sensors. Basically, the functional block of a virtual lock-in amplifier will be something like the one shown in fig4. From Fig4, one can see, with virtual instrumentation, the digital lock-in amplifier is fully controlled by the computer, which makes the whole system cheaper, faster and more accurate. Furtherly, other instruments depending on the lock-in outputs can be controlled by the same computer. From the view of system integration, virtual instrumentation makes the system flexible, scalable and easy to use.

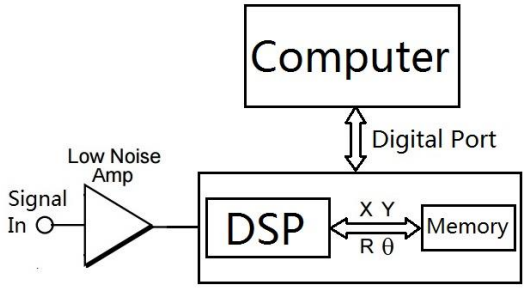

Fig4, the functional block of a virtual lock-in amplifier.

4. Implementation of the virtual lock-in amplifier on an SR7265 target

One of the easiest ways to build virtual lock-in amplifiers is to combine the virtual instrumentation technology with a standalone lock-in amplifier with computer interfaces. One can implement the virtual lock-in amplifier algorisms with LabVIEW, then control the target hardware by RS232, GPIB or USB interfaces. Fig5 shows the basic idea of a virtual lock-in amplifier implementation. The benefit of this approach is that a unified user interface and functional block can be established. After all the functional blocks are fully verified and tested, the same virtual lock-in amplifier can be implemented in other commercial products, analog products or a house-made lock-in amplifier with various technologies, such as FPGA, ARM and DSP.

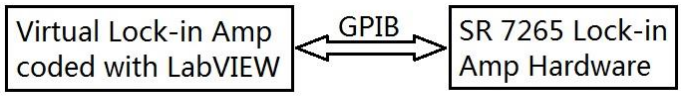

Fig5, direct implementation of the virtual lock-in amplifier to an SR 7265 target

5. The system architecture of the virtual instrument

This virtual lock-in amplifier is implemented with a 4 layer model. The 4th layer implements the various applications, such as a lock-in amplifier, spectral analyzer, network analyzer, impedance analyzer, noise meter, semiconductor parameter analyzer, etc. The 3rd layer implements the necessary functional blocks, such as frequency sweeps, FFT block, complex voltage and current measurement block, etc. The 2nd layer is the commands offered by the lock-in manufacture. The 1st layer is the physical equipment. See fig 6.

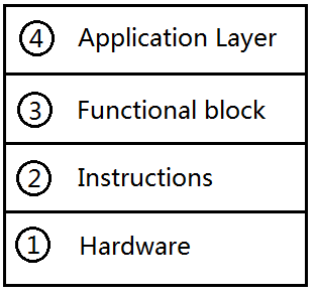

Fig 6, a structural model of a virtual lock-in amplifier.

An interesting question is why it is best to organize the virtual lock-in amplifier into a 4 layer model [7]. Suppose the instrument is organized by a 3 layer model. Then, we merge the 3rd and 4th layers together. There will be several problems. One of the problems is that different lock-in amplifiers may have different command sets. In order to fully apply the commands, the application layer may be

programmed differently for different lock-in amplifiers. Finally, this may lead to a slightly different user interface and operational procedure. But we want the different hardware shares the same user interface. The other problem is that LabVIEW is graphic programming. It is nice to put everything into one screen. If one layer has too many functions to implement, one would program the codes in several screens. This makes the reading and maintenance of the codes difficult.

6. Implementation of the spectrum analyzer, impedance meter, and semiconductor analyzer

After the virtual lock-in amplifier is successfully implemented. Its resources can be reorganized to build other virtual instruments. Calling the frequency sweep subroutine, and measuring the responses, one can build a spectrum analyzer. Controlling the oscillator to power the device under test (DUT), and measuring the complex current through it, one can build an impedance meter. Programming the auxiliary inputs and outputs and lock-in core, one can build a sophisticated semiconductor analyzer with the same hardware. Due to the space limitation, we will not explain the detailed algorithm here. One can check Ref [2], [4], [8] and [9] for the impedance meter. One needs some extra knowledge about semiconductor physics to build the semiconductor analyzer. Some useful ideas can be got from Ref [10] and [11].

7. Implement the virtual instrument on an FPGA board

With the LabVIEW FPGA module and MATLAB, it is much easier to develop a complex DSP project on an FPGA board than with traditional Verilog programming. We employ the NI7831R FPGA board to carry out the Layer 1 and Layer 2 functions. An analog interface is also employed to amplify the signals inputting to the FPGA board, and output the FPGA signals to the DUT. The system block is shown in fig 7. The system structure is shown in fig 8. Again, due to space, we will not cover the details of How to program the FPGA board and how to design the diff amp. Interested readers can check Ref [12] for more clues.

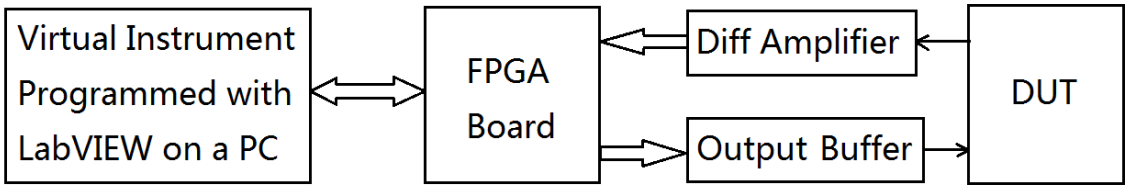

Fig 7, block diagram of the virtual instruments targeted on an FPGA board

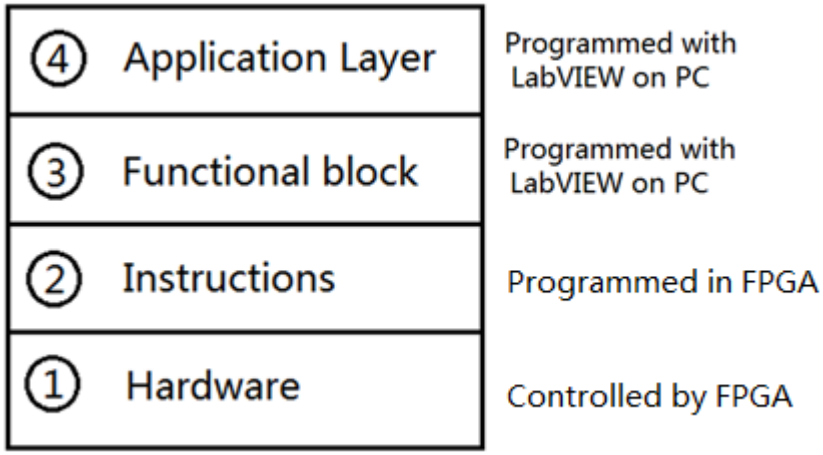

Fig 8, structural model of the virtual instrument implemented on an FPGA board

8. The User Interface

The software is important for any virtual instrument. A good virtual instrument should be strong, fault tolerant and intuitive and intelligent. Some of the users may not have sufficient computer science background. So, here, we briefly describe the software with the layered model in mind. One may get some clues from this section and develop his own complicated application with virtual instrumentation technology.

The virtual lock-in amplifier tab offers a wide frequency range sweep, a fine sweep, and basic lock-in setup and data displays. The top windows display the amplitude responses and the bottom windows display the phase responses. During the operation, users first take a rough frequency sweep to find the interest frequency range, then click the button STOP1. The interest frequency range will be passed to the right area. One can change the frequency spans by DF1 and DF2 controls. After the fine sweep, the lock-in amplifier will lock to the peak frequency. The default mode is Auto Lock-In. By click the Auto Lock-In button, one can switch between auto mode and manual mode. Under the manual mode, when one set the frequency at the ManualFrq control. Then the lock-in amplifier will lock to the frequency as one set. The virtual lock-in amplifier communicates with the hardware through GPIB or USB. The BusDelay control set the bus update time in milliseconds. OscAmp sets the amplitude of the excitation oscillator. TimeCons and Sensitivity controls are pulldown menus for Time constant and sensitivity setups. X,Y, R, θ are displays of the results. See fig 3 for what they represent.

The spectrum analyzer tab has a similar style as the virtual lock-in amplifier. The top windows display the amplitude response. The bottom window displays the phase response. An extra feature is added to change the background color of the display. The little black square can change the color of the windows by a click. If users may view the spectrum on a monitor, they can set the background color to black. If users may print the spectrum on a printer, they can set the background color to white.

The impedance analyzer has a similar style too. It can measure the complex impedance around a frequency range. For detailed hardware connection to measure impedance with a lock-in amplifier, please see Ref [2].

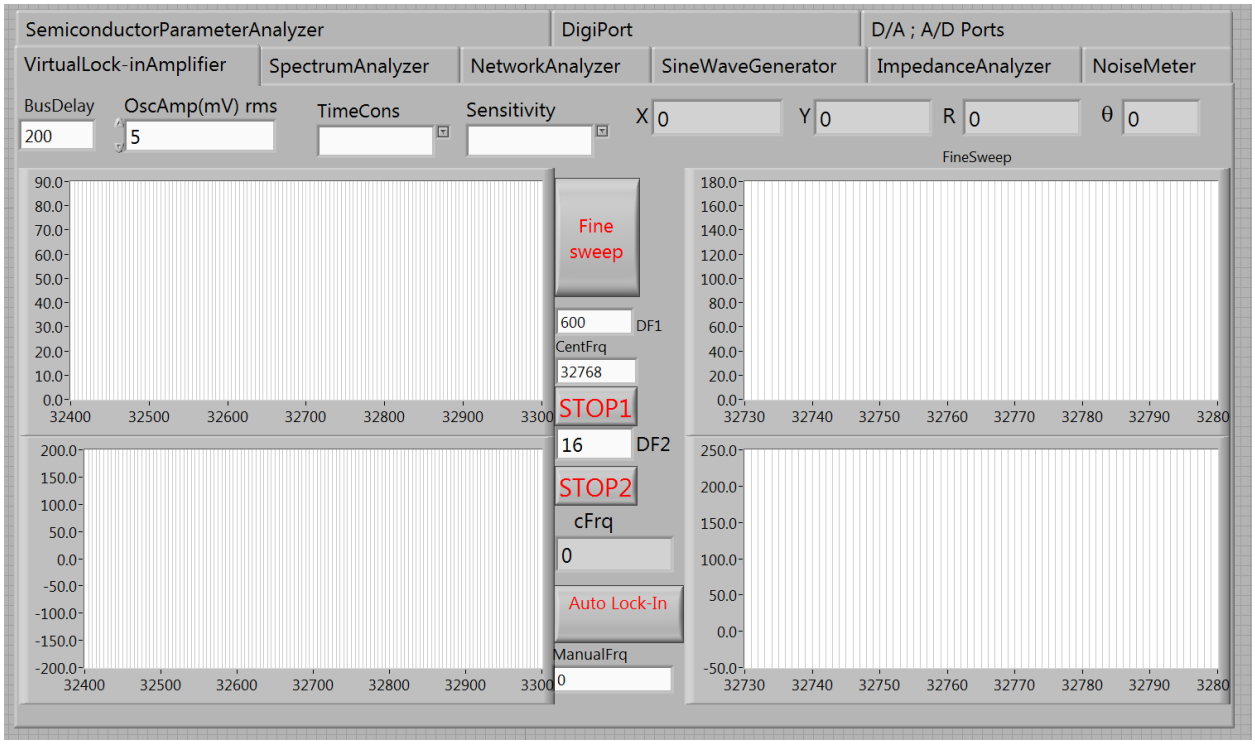

Fig 7, the user interface of the virtual lock-in amplifier.

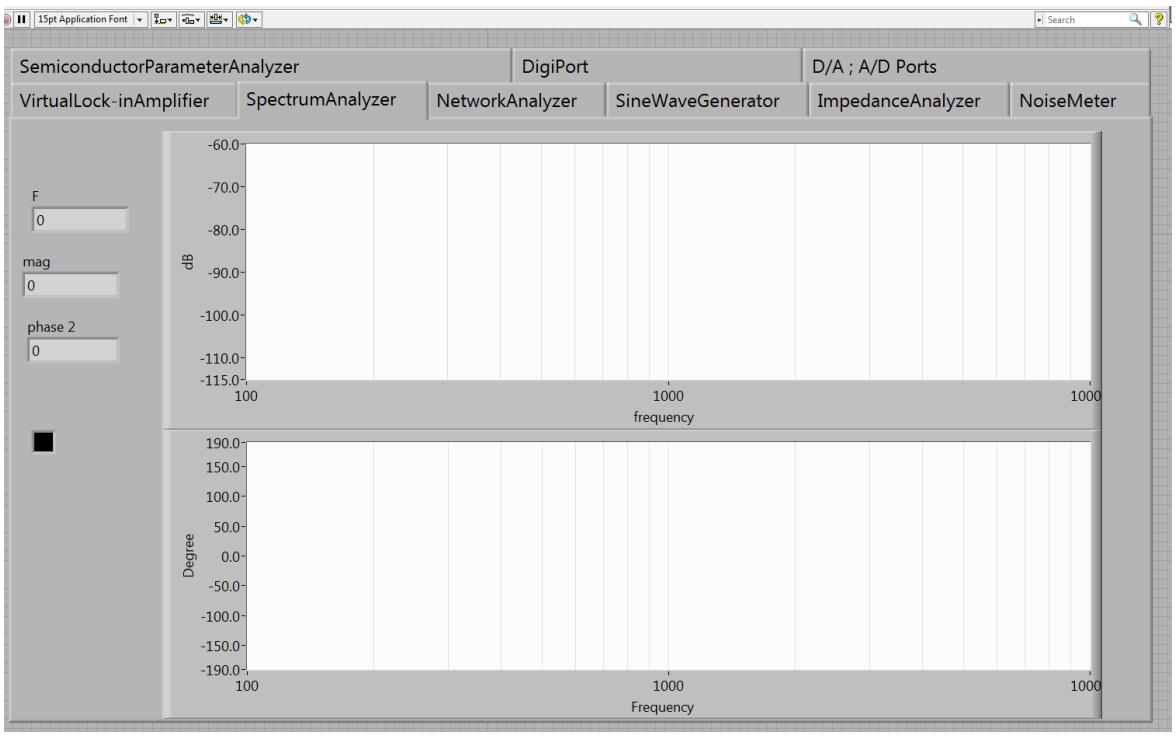

Fig 8, the user interface of the spectrum analyzer.

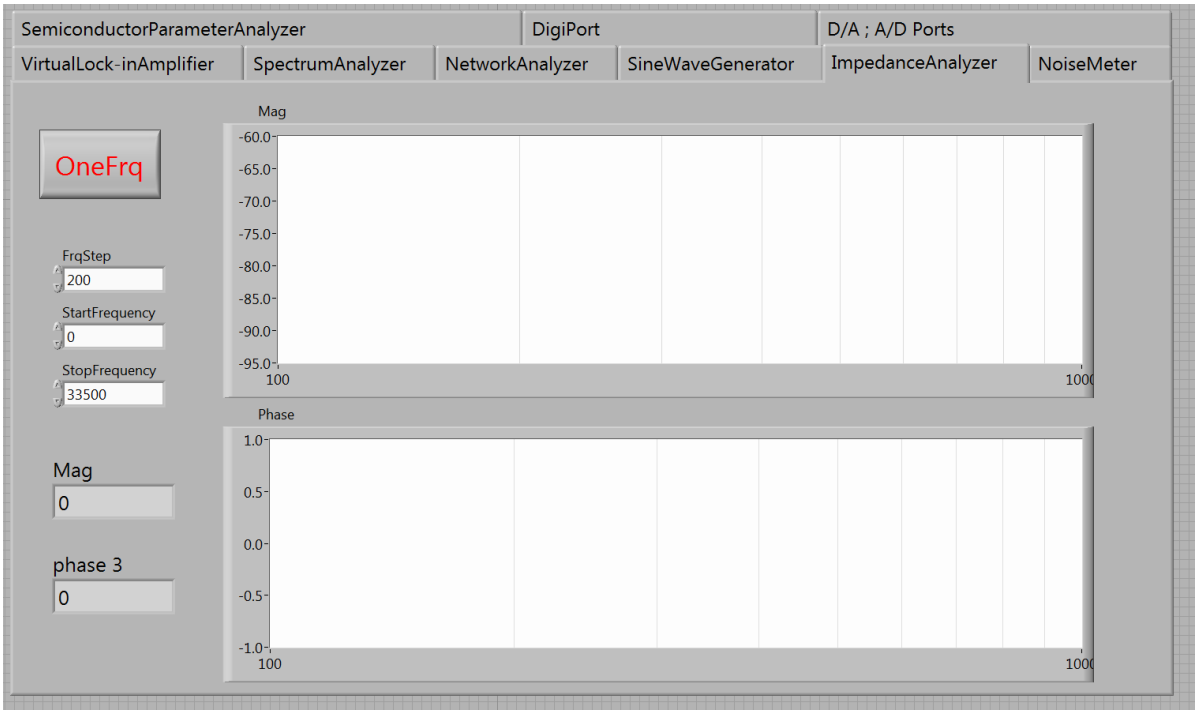

Fig 9, the user interface of the spectrum analyzer.

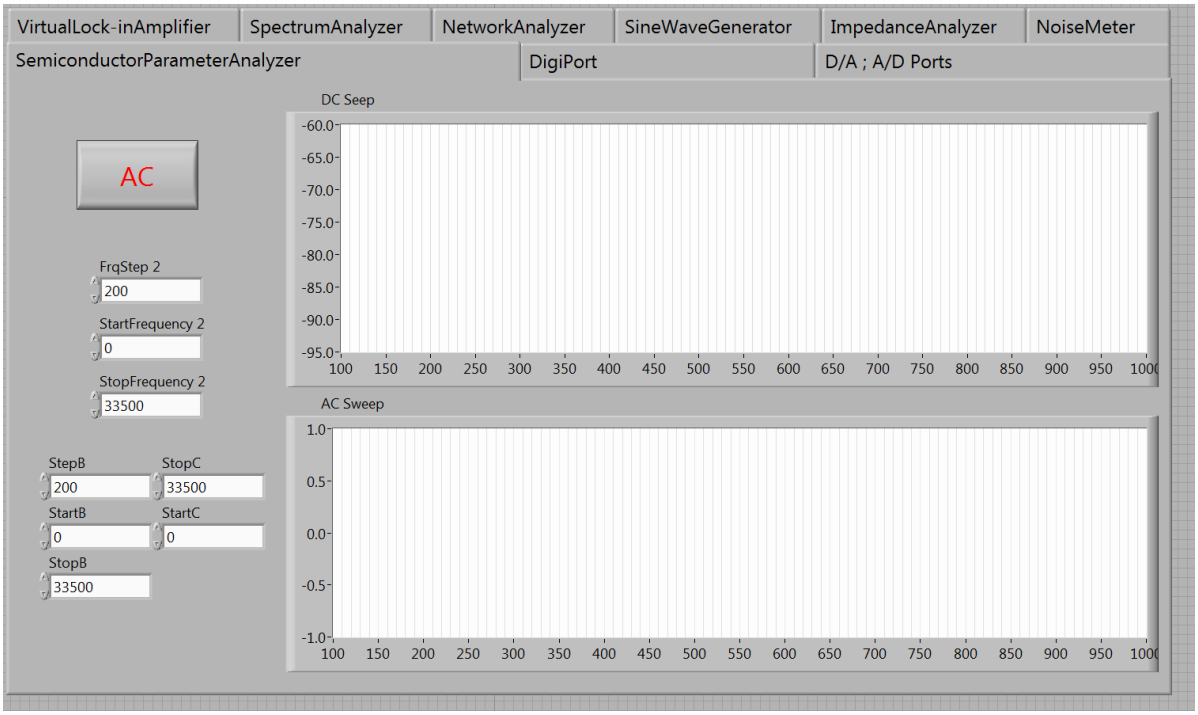

Fig 10, the user interface of the semiconductor parameter analyzer.

The semiconductor analyzer tab implements PN junction test, bipolar transistor test, and FET test. By clicking the AC button, one can switch between high-frequency AC mode and low-frequency DC mode. By DC sweeps, one can get bias, threshold, load characteristics, etc. By high-frequency sweeps, one can measure junction capacitance, diffusion time, depletion curves, and the device ac equivalent models, etc. For the physics behind these measurements, please read Ref [10].

9. Implementation of the virtual instruments

A. Layer 4

All the software is implemented with a 4 Layer model. Fig 11 shows the Layer 4 of the virtual lock-in amplifier. It is coded as a sequence. The first step sets the sensitivity, time constant, and oscillator amplitude. The second step takes the rough sweep. The third step takes the fine sweep. The final step takes the actual measurements at the locked frequency.

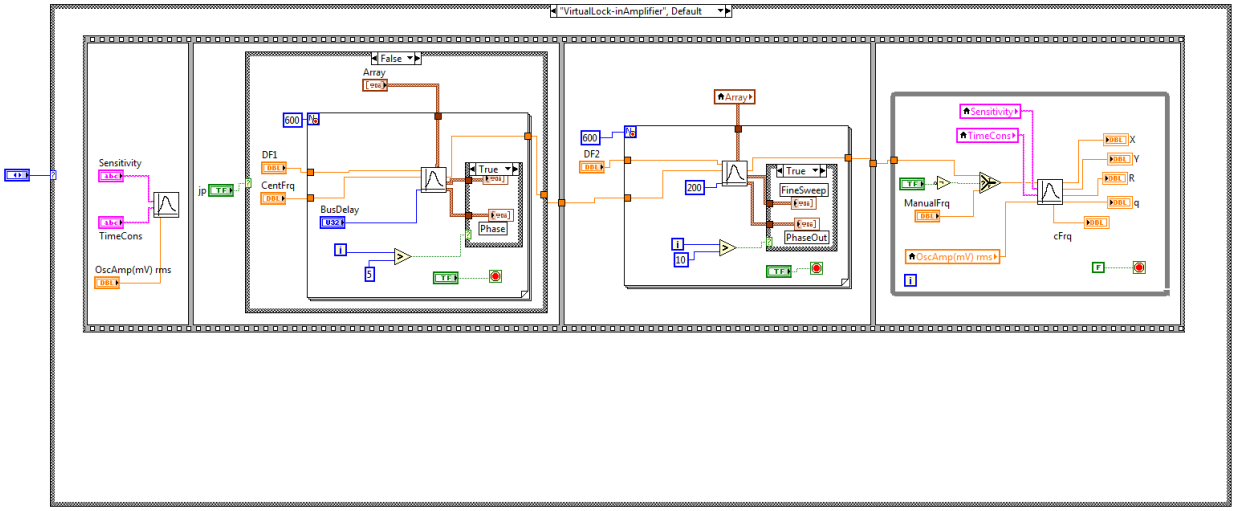

Fig 11, the Layer 4 codes of the virtual lock-in amplifier.

For an illustration purpose, the spectrum analyzer is coded with a mixture of Layer 4 and Layer 3. See fig 12. The detailed procedure to take a spectrum analysis is implemented in the while loop. One can put the procedure into a subroutine.

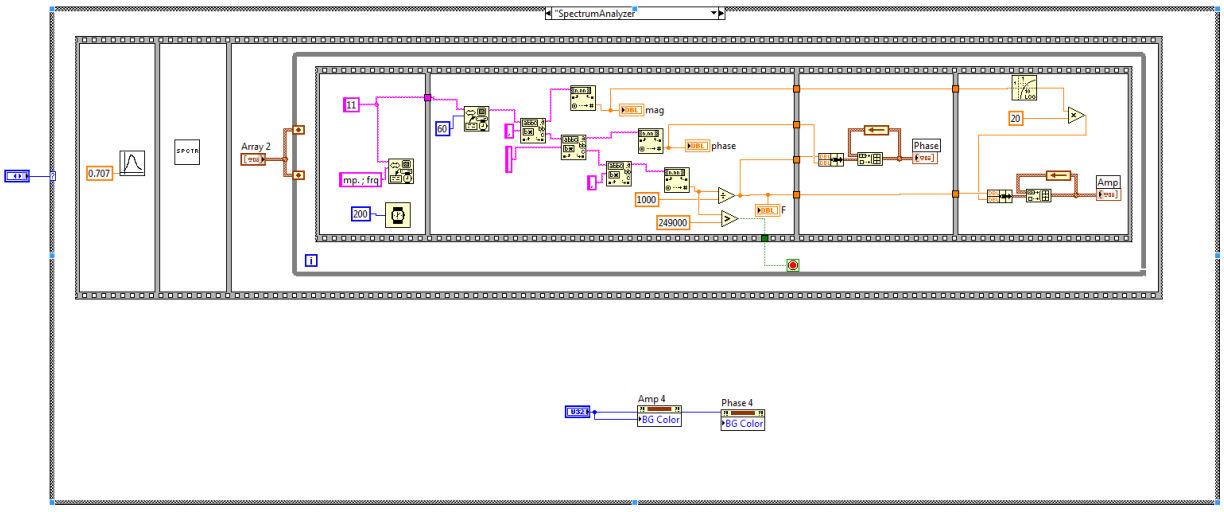

Fig 12, codes of the virtual spectrum analyzer.

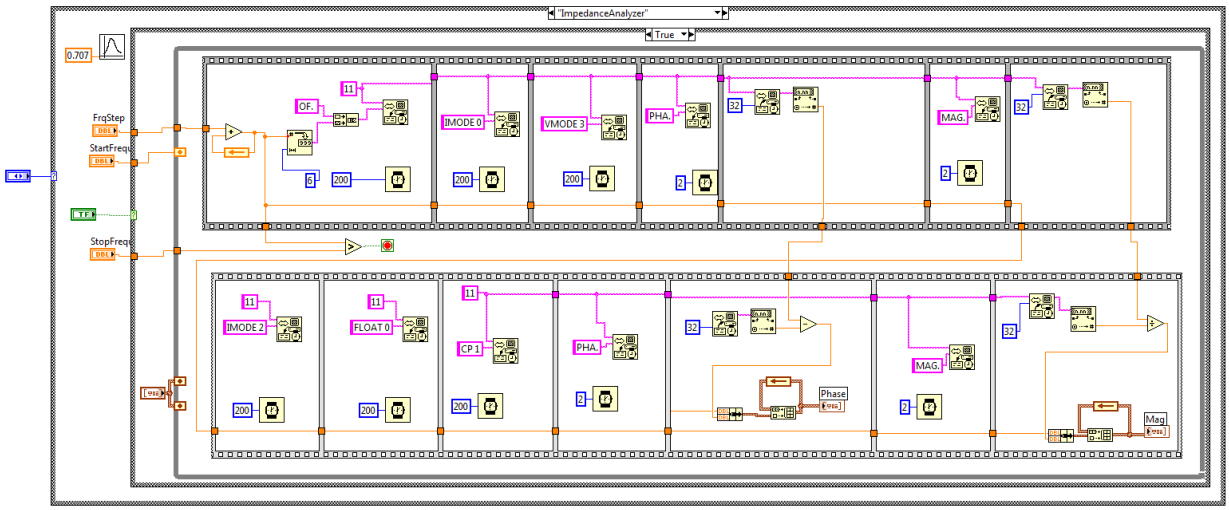

Fig 13, codes of the virtual impedance analyzer for variant frequencies.

The impedance analyzer is also purposely coded at Layer 3 to show the detail of the operation of the instrument. For an impedance analyzer, one needs to power the device under test (DUT) and measure its reaction continuously. At first, the instrument outputs a sinusoidal signal to the DUT, the swiftly the instrument switches to measure the voltage across the DUT and current through the DUT. By dividing the complex voltage and current, one can get the impedance of the DUT. The DUT needs to be wired to the differential input. The hardware connections can be found in Ref [2].

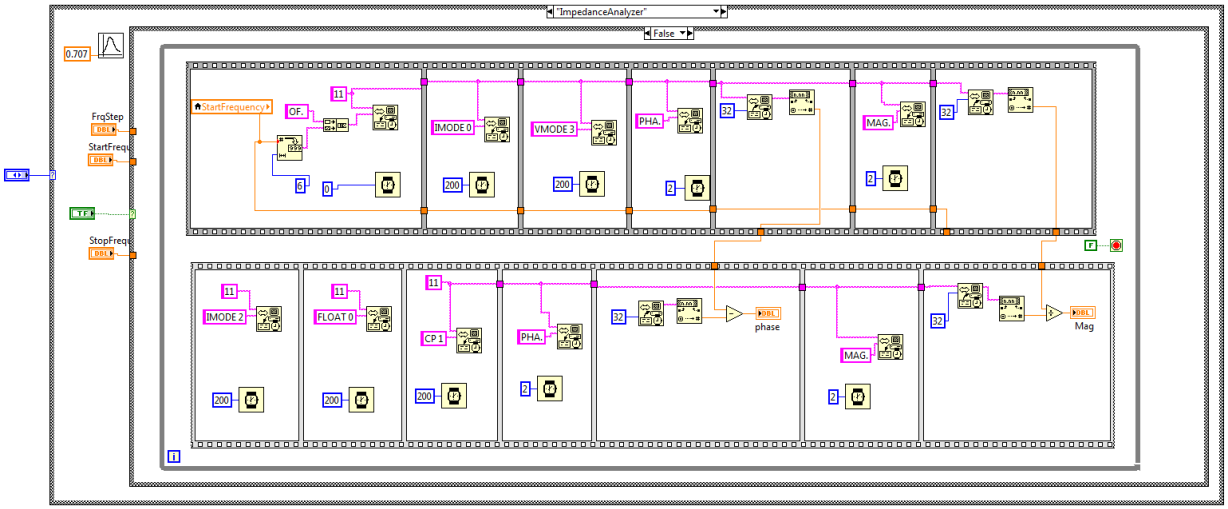

Fig 14, codes of the virtual impedance analyzer for a specific frequency.

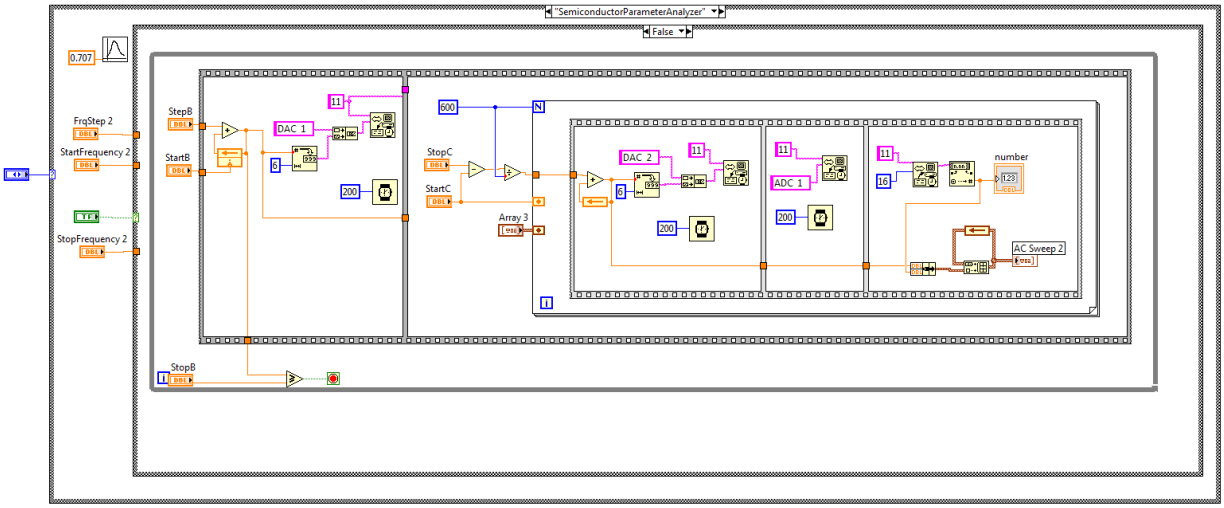

Fig 15, a section of the code of the virtual semiconductor parameter analyzer.

The code for the semiconductor analyzer is even more complicated. Here we only briefly describe the ideas. The auxiliary outputs set the DC bias of the DUT. Then an AC signal is generated by the instrument and injected to the DUT. Finally, the instrument will measure the voltage and current at the DUT terminals. Then the results will be either displayed on the monitor or saved in a file. For different purses, the instrument should be programmed specifically. Here we do not have enough space to cover the details.

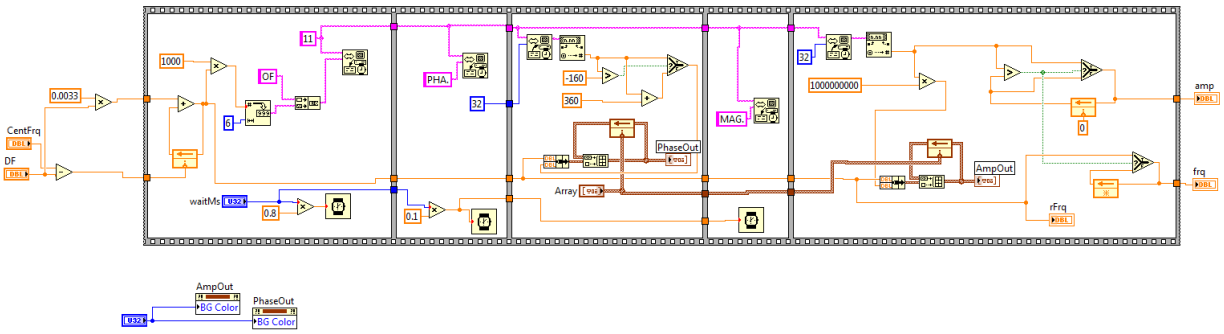

Fig 16, the frequency sweep function at Layer 3

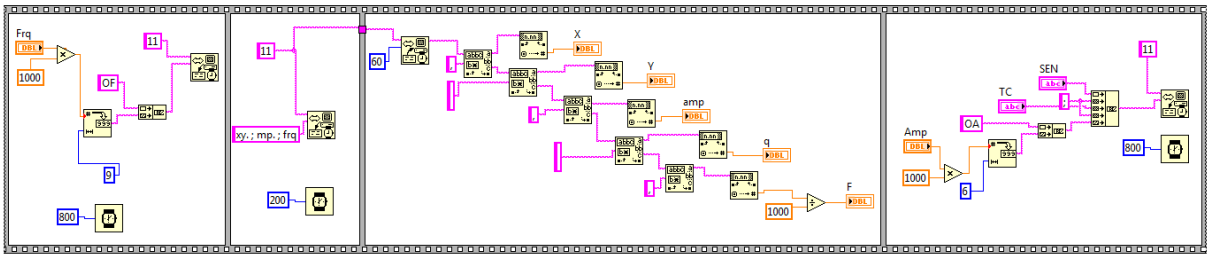

Fig 16, the oscillator and signal channel setup function at Layer 3

B. Layer 3

Layer 3 subroutines carry out the basic functional blocks like frequency sweep, hardware condition setup, and signal generating and measuring, etc.

C. Layer 2

Here, we use a commercial lock-in amplifier as our hardware base. The Layer 2 of the virtual instruments are the instruction set offered by the manufacture of the lock-in amplifier. If one wants to build his own hardware with a DSP or FPGA developing board. The layer 2 functions should be implemented in the boards directly. An interface should be built in the board to communicate with the Layer 3 subroutines programmed by LabVIEW.

D. Layer 1

Lock-in amplifiers are high-speed instruments. A commercial lock-in amplifier with computer interface will be a good option. Beside of this, one can choose a DSP board or FPGA board to implement the basic logic shown in fig 1 to fig 4. Usually, an analog interface is also needed to connect the real world to the boards. For different applications, some specifications of the DSP boards should be taken care of, such as sampling rate, memory size, and signal update rate, etc.

10. Conclusions.

1) This platform can enhance the applications of a lock-in amplifier. In addition of traditional lock-in amplifier applications, one can use the lock-in as a network analyzer, impedance analyzer, and semiconductor parameter analyzer, etc.

2) It is easier to deploy the virtual instrument on a commercial hardware. After this, one can deploy the instruments on DSP or FPGA developing boards, to build fully house-made virtual instruments through hardware and software. Besides of functional IP cores, one also needs to build an efficient interface on the DSP or FPGA boards, to communicate with the LabVIEW programs running on a PC.

References:

[1] Stanford Research Systems, DSP lock-in amplifier model SR830 user manual [M], Nov 1999. P3-1 – P3-21.

[2] Zurich Instruments, MFLI User Manual [M], Jul 14, 2016. P71-80

[3] AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC, Model 7265 DSP Lock-in Amplifier Instruction Manual [M], 2002. P6-1 – P6-38.

[4] Zurich Instruments, MFIA User Manual [M], Jul 14, 2016. P200-205,

[5] AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC, Model 181 Low-Noise Current Preamplifier instruction manual [M], 2002.

[6] AMETEK ADVANCED MEASUREMENT TECHNOLOGY, INC, Models 5105 & 5106 Dual Phase Lock-in Amplifiers Instruction Manual [M], 2009. P3-1—P3-17.

[7] JH Bai，JW Chen，J Freeouf，AL Rosa, A 4-layer method of developing integrated sensor systems with LabVIEW, Journal of Measurement Science & Instrumentation [J], 2013.

[8] Alexandre Rius Rueda, Measurement of low impedances with a lock-in amplifier, http://diposit.ub.edu/dspace/bitstream/2445/59951/1/TFG_Alexandre_Rius_Rueda.pdf

[9] T. Kouh， U. Kemiktarak ， O. Basarir, C. Lissandrello, and K. L. Ekincia Measuring Gaussian noise using a lock-in amplifier, American Journal of Physics [J], 2014. Volume 82, Issue 8, 10.1119/1.4873694

[10] Robert Pierret，Semiconductor Device Fundamentals [M], Addison Wesley, 1996.

[11] S.M.Sze， Physics of Semiconductor Devices 3rd Edition [M], Wiley-Interscience， 2006.

[12] NI Community, Lock-in Amplifier on LabVIEW FPGA, https://forums.ni.com/t5/Example-Programs/Lock-in-Amplifier-on-LabVIEW-FPGA/ta-p/3500412

View publication stats