

# READ - Remote Analog ASIC Design System

D.G. Zutin<sup>1</sup>, M. E. Auer<sup>2</sup>

<sup>1</sup> State University of São Paulo / Electrical Engineering Department, Bauru, Brazil

<sup>2</sup> Carinthian University of Applied Sciences / School of Electronics, Villach, Austria

**Abstract**—The scope of this work is to present a solution to implement a remote electronic laboratory for testing and designing analog ASICs (ispPAC10). The application allows users to create circuit schematics, upload the design to the device and perform measurements. The software used for designing circuits is the PAC-Designer and it runs on a Citrix server. The signals are generated and the responses are acquired by a data acquisition board controlled by LabView. The virtual instruments interact with some ActiveX controls specially designed to look like real oscilloscope and function generator devices and represent the user interface of the lab. These ActiveX give users the control over the LabView VIs and the access to its facilities in order to perform electronic exercises.

**Index Terms**—ActiveX controls, Analog ASIC, DataSocket, Remote Access, Remote Laboratories.

## I. INTRODUCTION

The Center of Competence “Online Labs and e-learning” at the Carinthian University of Applied Sciences (CUAS) carries out the REL (Remote Electronic Laboratory) project, which is part of the telelearning system of the institution and turns available some real electronics related exercises to be performed through the Internet. Due to the importance of laboratories for an engineering curriculum, remote labs can become a tool to aggregate facilities to practical lessons once this solution allows displaying and controlling real experiments through the internet. There are different ways to implement a remote laboratory, depending on the exercises to be performed.

The solution presented here consists of a remote electronic laboratory for test and design of analog ASICs (Application Specific Integrated Circuits) and it was implemented with a data acquisition board (DAQ) from National Instruments (PCI-6251) controlled by LabView. The user interface was built with Visual Basic 6.0, with some ActiveX controls provided by Measurement Studio for Visual Basic 6.0.

The idea was to create a remote lab to interact with the ispPAC10 device. The ispPAC10 is an in system programmable analog device from Lattice Semiconductor, which allows the implementation of several analog circuits like low pass and band pass filters, amplifiers, etc. The circuit programming and uploading are done by means of a windows based application called PAC-Designer, which runs in our case on a Citrix server in order to allow the remote users to access the application.

Real experiments are carried out on the server side. The user can control these experiments via Internet by setting the parameters of the circuit input signal (waveform, frequency, amplitude, offset). After the parameters have been set the user can read the circuit output and compare the output waveform signal with the input waveform signal

Only one user can access the remote lab at a time, so, the user control access system establishes a thirty minutes session for each user connected to the system.

To access the remote lab, the user is asked to insert a username and a password. If both are valid, the system enables the access to the lab.

As mentioned previously, the user interface consists of some ActiveX controls embedded in HTML files. This interface was designed to look similar to a real oscilloscope and a function generator and to have nearly the same functionalities as real devices.

## II. SYSTEM DESCRIPTION

The Remote Electronic Laboratory implementation involved the LabView VI and sub VIs programming in order to control the DAQ board, the virtual lab equipments (function generator and oscilloscope) construction using the Measurement Studio for Visual Basic 6.0 ActiveX controls, the HTML and CAB files publishing in the web server and the PAC-Designer installation in the Citrix server.

### A. The DataSocket Server

The communication between the REL server (LabView VIs) and the REL users (ActiveX controls) occurs by means of a DataSocket server, which is a programming tool from National Instruments that enables writing, reading and sharing raw data (arrays and variables) between applications and different data sources and targets. DataSocket provides a unified API (Application Programming Interface) for low level communication protocols (TCP/IP, FTP, HTTP). It allows a connection to different data sources without writing different code to support different data formats and protocols. Figure 1 shows the software communication diagram. The user sends the parameters to the REL server, which performs the proper operations with the DAQ hardware and returns the measurements to the user.

The DataSocket also has its own scheme, dstp (data socket transfer protocol), for sharing live data through DataSocket servers

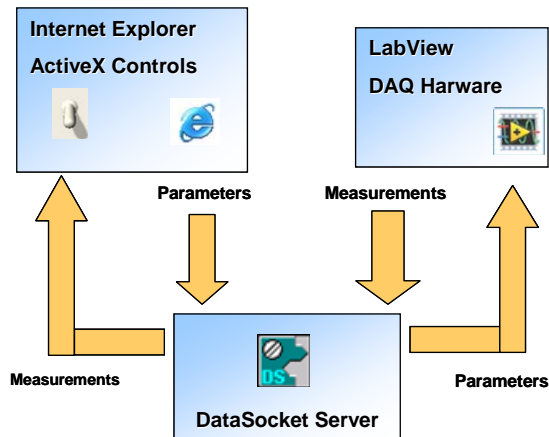


Figure 1. Basic Software Communication

Another benefit of this tool is that it is not necessary to assign a public IP address to the REL server, so its location can be changed at any time without any other changes in the system.

**B. The REL Server**

In the REL server run the LabView Virtual instruments controlling the DAQ board and also sending to and receiving data from the DataSocket server. To control the data acquisition and generation were used the NI DAQmx driver software, that can be called from LabView to program NI measurement devices. The driver software has an application programming interface (API), which is a library of VIs, functions, classes, attributes, and properties for creating applications. This is a library of polymorphic VIs that replaces large parts of a complex DAQ application with a simple configuration dialog.

The REL LabView VIs can be divided in three main parts, the DataSocket API VIs, the oscilloscope VIs and the function generator VIs, each one with well defined tasks. A simplified diagram can be seen in Figure 2.

The DataSocket API VIs are continuously communicating with the DataSocket server. It keeps monitoring the incoming parameters and publishing the measurements in the DataSocket server when requested to. When new parameters are read from the DS server, they are passed to the function generator and oscilloscope

VIs directly or through global variables. The proper actions are taken and the data containing the measurements is returned and sent to the DS server.

The oscilloscope VI should work similarly to a real oscilloscope. The user must be able to measure some properties of the signal and also to visualize the waveform signals. In order to do that, the VIs acquire data from two analog channels simultaneously. The data acquisition properties (sampling rate and number of samples) are calculated based on the parameter called TimeBase sent by the user according to the following equation.

$$TimeBase \times Sampling\ Rate = N^{\circ}\ of\ samples \quad (1)$$

This parameter specifies the time interval to be displayed on the oscilloscope screen.

The calculation is an iterative process; the sampling rate is decreased by one element on each iteration, starting in its maximum value until the number of samples reaches its maximum. The N° of samples acquired is limited, so, this calculation is necessary to prevent a misrepresentation of high frequency signals and to ensure that the number of samples will be enough to visualize the requested time interval when sampling lower frequencies signals.

The ispPAC10 device has 4 analog inputs and 4 analog outputs and for each input/output was assigned an analog input of the DAQ board, so that each one might be read. The user can choose which pair input/output will be displayed by setting the proper parameters.

The PCI-6251 has an input sampling rate of 1.25 MS/s, but the hardware has only one ADC (Analog to Digital Converter), which leads to a rate of 625 KS/s for each channel.

The oscilloscope VI also performs measurements (Function OUT) after acquiring the samples. These measurements are Vpp, Vmax, Vmin, Vrms, Vaverage, frequency and period and these tasks are accomplished by using some VIs from the LabView libraries.

The function generator VI uses the DAQmx API VIs to output an analog waveform signal. The user is able to choose the waveform shape, the amplitude, the frequency and the signal offset. The DAQ board PCI-6251 has 2 analog output channels that support a sampling rate of 2.8 MS/s. In order to control the waveform generation, important parameters are the number of samples per buffer and the sampling rate. Although, these parameters should not be of the users concern, so, LabView calculates these parameters based on the desired frequency provided

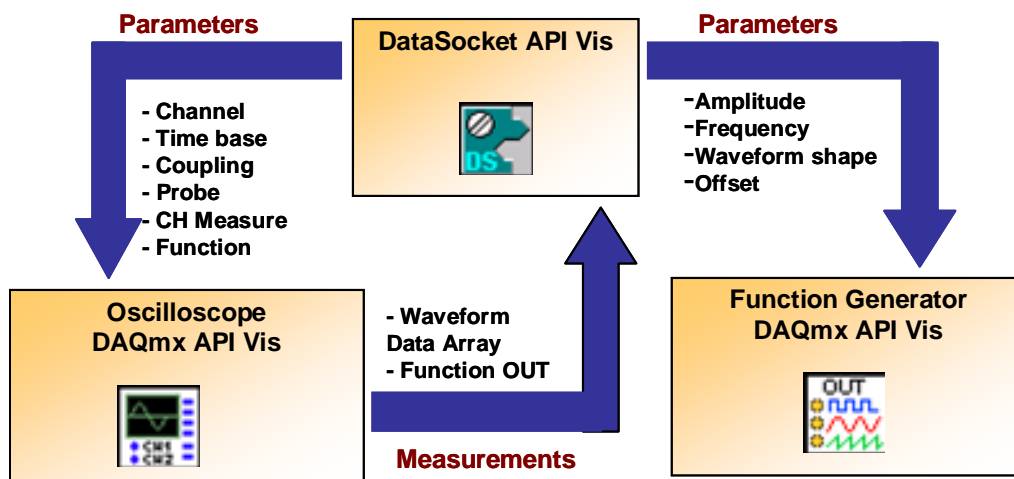


Figure 2. The Communication of LabView Vis

by the user and ensures the number of samples to be as grater as possible, thus keeping a good quality of the signal. This is done according to the following equation.

$$Freq \times N^{\circ} \text{ of samples per period} = \text{Sampling rate} \quad (2)$$

As mentioned previously, the ispPAC10 has 4 analog inputs to which signals can be applied to. In order to give the REL user the facility to choose one among these inputs, an analog multiplexer (3x8) was used. Three digital I/O lines were used to switch the MUX according to the parameters received from the user.

C. The User Interface

In order to create a user interface, the ActiveX technology was used. ActiveX control is a group of components, or software objects. Each object encapsulates and exposes a specific functionality and all of the objects work together to provide the entire functionality of the control. ActiveX is not a programming language, but rather a set of rules about how applications should share information, which can be automatically downloaded and executed by a web browser.

New activeX controls were assembled by using the Visual Basic 6.0 and Measurement Studio for VB 6.0 suit of activeX controls. Figures 3 to 5 show these activeX controls.

These activeX controls were embedded in HTML files with the “Package & Deployment Wizard” tool, which also creates an internet package (CAB file) containing the necessary windows libraries (DLL files) and the activeX controls files (OCX files) to be installed in the user’s personal computer. The HTML files and the internet package (CAB files) were published in the HTML server. When the user accesses these files with a web browser, the internet package is downloaded and installed. Once activeX are programs, administrative privileges are necessary to install and run the remote lab.

When the software runs, it connects to the DataSocket server and everytime the user presses the Enter button new parameters are published in the server.

The user access system enables the access to the remote lab. The login and password are sent to the REL server and if it matches with the login and password stored in the REL users list the function generator and oscilloscope links become reachable.

The PAC-Designer software was installed on a Citrix server. The Citrix MetaFrame is a Windows NT application for delivering Windows based applications to several desktops and supports up to 15 simultaneous connections. Once installed, the PAC-Designer becomes available to be remotely accessed. The user access system doesn’t control the access to the Citrix server, which must be done separately.

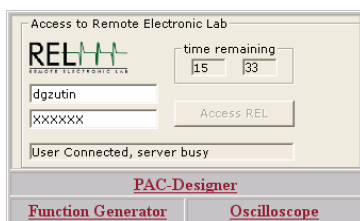


Figure 3. User Control Access Interface

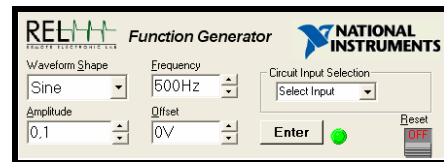


Figure 4. Function Generator Interface



Figure 5. Oscilloscope Interface

D. The Circuit Under Test

Lattice Semiconductor offers the ispPAC10 evaluation board for design and test of ASICs. The remote lab idea was to make the same functionalities of this board also available to the REL user. In order to accomplish that, a new board was assembled. Within this board, each DAQ card terminal was directly connected to its respective node in the circuit and no extra wires were needed. A simetric power supply was necessary because of the analog multiplexer. Two coaxial terminals and jumpers were also made available for testing purposes. Figure 6 shows the REL board.

The ispPAC10 is member of a family of analog circuits capable of realizing a variety of popular analog functions including precision filtering, summing/differencing, gain/attenuation and integration. The device is fully differential, but to connect the differential inputs to a single-ended signal, one of the differential inputs needs to be connected to a DC bias, preferably VREFOUT (2.5V). The input signal must either be AC coupled or have a DC bias equal to the DC level of the other input.

The ispPAC10 operates within a voltage range from 1 to 4V. Once all performed experiments were carried out in single ended mode, all the inverter analog inputs were connected to VREFOUT (2.5V).

After implemented, the remote electronic laboratory had the architecture shown in Figure 7.

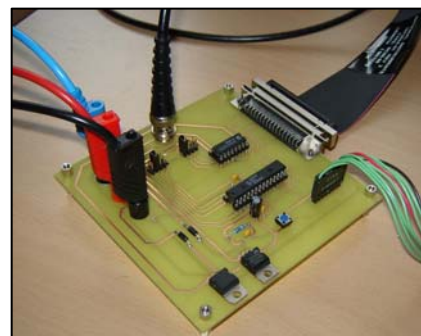


Figure 6. The Remote Laboratory Board

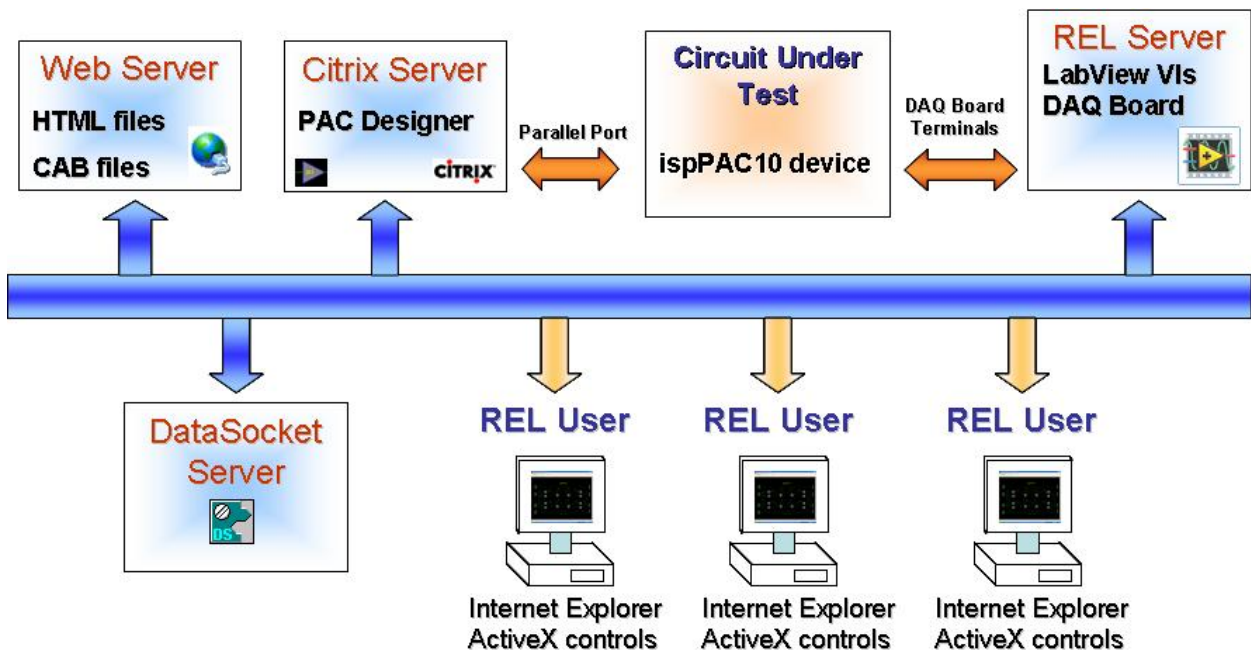


Figure 7. System Diagram

### III. CONCLUSIONS

Remote solutions have been showing to be an attractive tool for engineering. It offers a wide band of possibilities for applications in industries and for e-learning purposes.

The project goals were achieved and the Remote Electronic Laboratory is working properly. The experiments with the ispPAC device could be performed in a satisfactory manner. Due to the hardware limitations, the function generator could not generate signals within a large frequency interval, although the circuit behaviors could be well visualized for the performed exercises.

Another important aspect found after the REL board development is the noise level that affects the measurements performed in the circuit. For signals with low amplitude and high frequencies the noise effects could be well noticed. For higher amplitudes this effect could badly be noticed. Although it is an important consideration because the experiments with the ispPAC10 device can only be performed within a voltage range from 1 to 4V, which leads to the constant use of low amplitude signals when doing experiments.

Another important concern of this project is the ActiveX controls distribution and visualization. ActiveX are programs downloaded and installed on the user's computer, so administrator's privileges are required. Internet browsers, like internet explorer offer some features for protection against ActiveX controls that could be arbitrarily installed and cause some damages. So, some features must be disabled in the internet browser prior to run the remote lab.

The ActiveX controls are packaged in CAB files in order to be distributed via the internet. These CAB files contains

some files (activeX ocx files, DLL files) for running on the user's computer.

### REFERENCES

- [1] M. E. Auer, "Virtual Lab versus Remote Lab," *Villach, Austria*.
- [2] M. E. Auer, W. Gallent, "The Remote Electronic Lab as a part of the Telelearning Concept at the Carinthia Tech Institute," *Proceedings of the ICL 2000, 2000, Villach, Austria*.
- [3] H. Edwards, "Building an Interactive Web Page with DataSocket," *National Instruments Corporation, Austin, Tx, 1999, Application note 127*.
- [4] G. Perry, "Visual Basic 6.0 in 21 days," *Indianapolis, Indiana, 1998*.
- [5] "LabView Basics I Course Manual", *National Instruments Corporation, Austin, Tx, 1998*.
- [6] "Getting Started with Measurement Studio for Visual Basic," *National Instruments Corporation, Austin, Tx, 2001*.
- [7] W. Gallent, "Remote Electronic Lab," *Diploma Theses, Villach, Austria: Carinthia Tech Institute, 2000*.

### AUTHORS

**D. G. Zutin** is with the Electrical Engineering Department, São Paulo State University, Bauru, SP Brazil. (e-mail: [dgzutin@gmail.com](mailto:dgzutin@gmail.com)).

**M. E. Auer** is with the School of Electronics, Carinthia University of Applied Sciences, Villach, Austria, (e-mail: [M.Auer@IEEE.org](mailto:M.Auer@IEEE.org)).

Manuscript received October 29, 2006. This work was supported in part by National Instruments.