INTEGRATING DESIGN AND SIMULATION TOOLS WITH LABVIEW TESTS

Georgi Todorov Nikolov, Boyanka Marinova Nikolova

FETT, Technical University of Sofia, Studenstki Grad, TU-Sofia, block 1, room 1366, 1797, Sofia, Bulgaria, e-mail: gnikolov@tu-sofia.bg, : bnikol@tu-sofia.bg

Keywords: LabVIEW, Virtual Instrument, EDA Tools, Integrated Environments, Engineering Education

In creating a new product, the designer needs to iterate between modeling, simulation, prototyping, and testing before he can arrive at a completed design. There are a number of tools, solutions and user code created that perform such integration. Unfortunately for many of products the solutions are partially which make difficulties for electronic circuits designers. For example the simulation results by Cadence's PSpice can be used only in LabVIEW environment by users accustomed with this product. The designers familiar with EDA graphical environments like PROBE can't use measurement results completely. The objectives of the presented paper are dedicated just for solving mentioned upper problems. As introduction a survey of existing build-in tools and third-party solutions is done. The detailed considerations of popular but less treated CSDF (Common Simulation Data Format) are performed. Using plug-in modular instruments or traditional GPIB instruments controlled by LabVIEW, engineers can easily generate stimulus signals, and acquire and analyze circuit response. In the paper a number of LabVIEW virtual instruments (VIs) for CSDF interfacing are created and explained. With such achieved integration, users can import measured data from LabVIEW and compare them on the same graph displays (PROBE) in PSpice simulators. As conclusion in the end of the paper some examples are appended in order to illustrate usefulness of created VIs.

1. INTRODUCTION

Integration of design and simulation software and test instrumentation provides new measurement-based modeling, verification, debugging and advanced engineering education capabilities that are not possible using EDA software or test instruments alone. Recently, reducing time-to-market in the design process is a major theme among EDA developers. Because designers use a wide range of tools for designing products, better integration between these design tools can make it much easier to move designs through the development cycle. As electronic designs move from software tools to hardware prototypes, the process of characterizing the performance can become disconnected. The traditional approach to design verification and characterization is to perform a series of manual measurements using benchtop instruments in a design lab. The result is a growing gap in productivity between the automated EDA tools and the manual measurement and data collection processes currently used in design verification and characterization.

To solve this problem a number of EDA software and measurement manufacturer leaders offer to the market a number of closely specific software environments such as Agilents's EESoft (IC-CAP, ADS) [1, 2, 3], Teradyne's LASAR Test-Simulation Software [4] or Credence's IMS (Integrated Measurement Systems). These

environments are distinguished by high productivity and high cost. For this reason they are inaccessible for low and medium staff factories as well as for educational institution. The DesignSoft's Tina implements some integration but functionality of measurement equipment used is more than restricted [10].

In this context, the very rapid rate of change in the fields of technology poses special problems for academic institutions, specifically for the engineering disciplines. There is of course a continual need to update and augment the content of lecture courses to keep pace with this change, but it is in the area of engineering education and experimental work where major concerns arise. Many academic courses that teach engineering subjects have already begun incorporating computer-based educational tools for student use, either in the lectures or in the laboratory practices or both. Computer-based design, simulation and measurement tools are as equally valued and commonly used as the traditional benchtop lab. Growing demand from industry to apply these modern tools in the undergraduate curriculum has also resulted in corresponding changes in pedagogy. Industry is looking for graduates who not only understand electronics and engineering concepts but must also be able to design circuits, understand layout intricacies, and plan for efficient testing of circuits. As students are building and experimenting with complex circuits, academia must provide the latest design, development, and testing tools and technologies.

EDA software tools provide powerful and user-friendly platforms for teaching, learning and research to students and educators. Most electronics curricula use simulation tools to present their electronics curriculum in both classroom and lab as well as to prepare students for employment in a professional workplace.

On the other hand LabVIEW's Virtual Instrumentation paradigm provides an ideal platform to develop instructional curriculum and scientific research. In an instructional laboratory course, students perform various experiments that combine measurements, automation and control. In a research environment, virtual instrumentation provides the flexibility that a researcher must have to modify the system to unpredictable needs. As a result LabVIEW has become an industry standard for virtual instrumentation in both education and industry.

While the use of simulation software in classroom and laboratory has been widely adopted, an emphasis on incorporating and comparing measurements from the actual circuits has lagged behind. Most laboratories use the traditional bench top approaches including bread boarding with measurement using oscilloscopes, function generators and DMMs. In some cases these instruments are controlled using a PC based, IEEE-488 (GPIB) interface board. As a result a number of tools, solutions and user code for importing simulated data from EDA to LabVIEW and vice versa was created recently. As can be seen from figure 1 for some EDA software (The Electronic Workbench's Multisim 7, Anadigm Designer 2) the integration is fully accomplished, whereas for other products the solutions are partially which make difficulties for electronic circuits designers [5, 6, 7]. For example the simulation results by Cadence's PSpice can be used only in LabVIEW environment by users accustomed with this product. The designers familiar with EDA graphical environments like

PROBE can't use measurement results completely and do not provide flexibility to compare measurements from actual circuits with simulations.

The subject of the presented paper is dedicated just for mentioned upper problems with practical example for importing measured data from LabVIEW based hardware platform to PSpice compatible EDA.

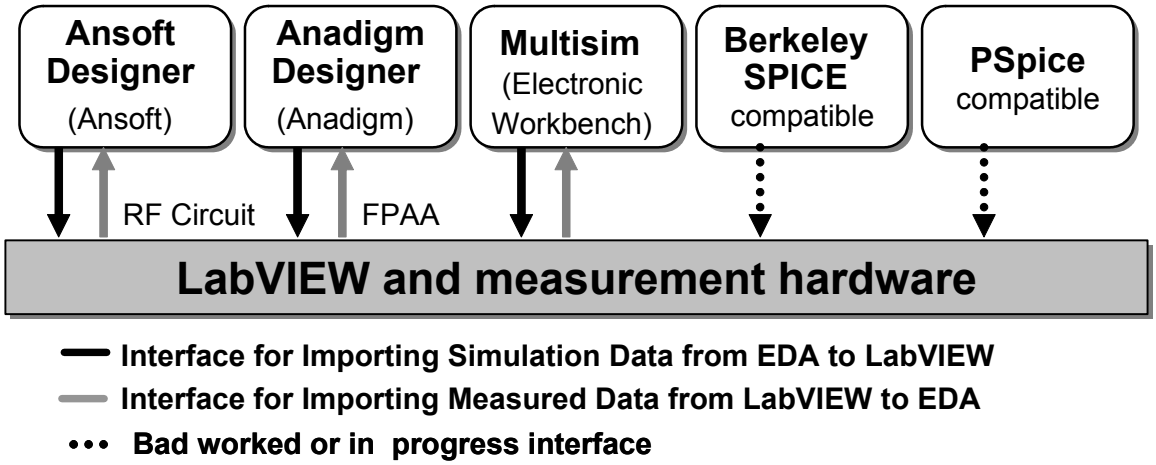

Fig. 1. How nowadays LabVIEW integrates with EDA environments

2. EDA-LABVIEW INTEGRATION GUIDELINES

The base steps for creating interface importing measured data from LabVIEW to EDA are more or less as follows:

1. Examine of EDA data file formats

Different EDA software tools used different output data. Some of them are in binary format (.dat file for PSpice) and are inaccessible by the third party software products. The other use text output format witch is large in size but is most common and most portable.

In the case of PSpice the default waveform data file format is binary. However, the user can save the waveform data file in the ASCII format or so called Common Simulation Data Format (CSDF) instead. The disadvantage of this popular data format is absents of documentation describing it. For purpose of generating source code converting measured data to CSD Format it is useful to divide last mentioned in tree divisions:

- Header: marked with #H and describing service information such as program version, title, date and time, analysis type, sweep type etc.
- Nodes: marked with #N and referred to node names
- Data: marked with #C, that could be presented with real or complex values

2. Choosing a LabView Data File Format

As is well-known text files is the easiest format to use and to share. Almost any computer can read from or write to a text file. A variety of text-based programs can read text-based files. Most instrument control applications use text strings. For these reasons spreadsheets files are most suitable.

3. Development of supplemental software modules for instrument drivers.

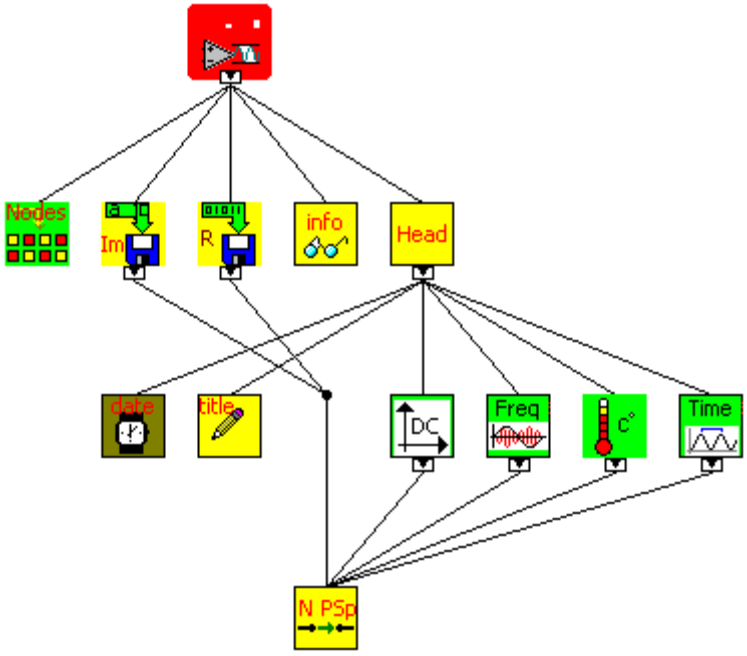

Fig. 2 VI's Hierarchy

The output data file format is not treated in requirements and recommendations for creating a certified LabVIEW Plug & Play instrument driver [8]. For this reason the development of software module (Virtual Instrument-VI) supplementing instrument driver is necessary. This module must generate output spreadsheet file containing data measured.

4. Creating VI's Hierarchy according to EDA output file format. It is useful to follow the concepts of top-down design [9]. The basic idea is to divide the task into manageable pieces at logical places. According to conception the hierarchical tree is created by take into consideration divided CSDF in first step (Fig. 2). The explanation for each module or subVI follows:

Export_Nodes.vi This software module generate dialog box asking the user for names and number of measured nodes.

Export_Header.vi The dialog box generated by this VI asks for measurement type and sweep type.

Get_Data_file_info.vi This module allows obtaining the sweep range and the measured points from the spreadsheet file.

Export_Real_Data.vi The software module manipulates with measured data and convert them into CSDF real values format**.**

Export_Complex_Data.vi This VI manipulates with measured data and converts them into CSDF complex values.

Frequency_Analyzis.vi This module is SubVI of Export_Header.vi and is responsible for establishing the appropriate AC sweep parameters into the header section of CSDF.

Time_Domain_Analyzis.vi This module is SubVI of Export_Header.vi and establishes the appropriate Transient Analysis parameters into the header section of CSDF.

DC_Analyzis.vi This VI is SubVI of Export_Header.vi and establishes the appropriate DC sweep parameters into the header section of CSDF.

Temperature_Analyzis.vi This module is SubVI of Export_Header.vi and is responsible for establish the temperature parameters appropriately.

Time_Date.vi The module is SubVI of Export_Header.vi and gets the current date and time from OS of the computer's platform.

Source_Title.vi This SubVI of Export_Header.vi generates the information for current file name.

5. Source code development Once the test engineer begin programming, the hierarchy fills out naturally, one subVI at a time. It is useful to add lower level support VIs as required, such as the Time_Date.vi, Analysis type VIs, a routine to parse header messages, or an error handler. The created source code is illustrated on Fig. 3.

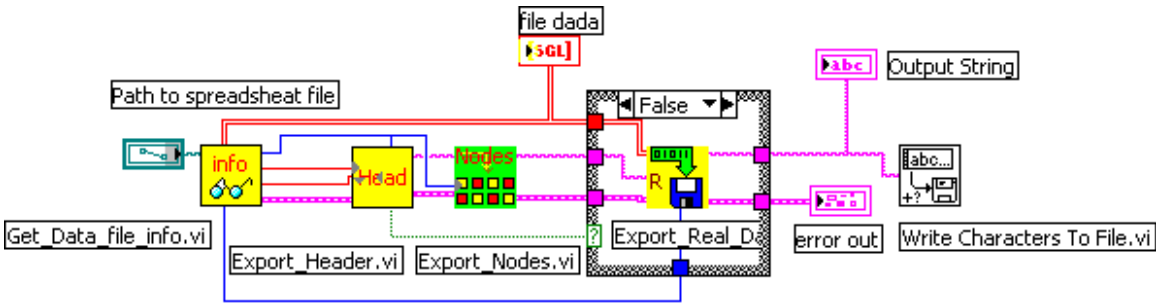

Fig. 3 The block diagram (source code) of software module.

6. Design of Front Panels and planning their appearance (User Interface)

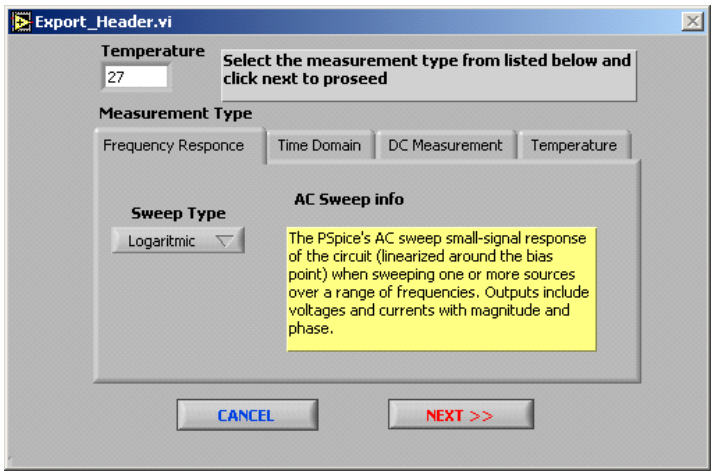

Fig. 4 Example of user interface

Front panel prototypes can provide insight into the organization of the program. Assuming the program is user-interface intensive, the program developer can attempt to create a mock interface that represents what the user sees – Fig. 4.

7. Creating Documentation

The two main categories for needed documentation are:

• Design-related documentation

Requirements, specifications, detailed design plans, test plans, and change history documents are examples of the kinds of design-related documents.

• User documentation explains how to use the software.

3. EXAMPLES

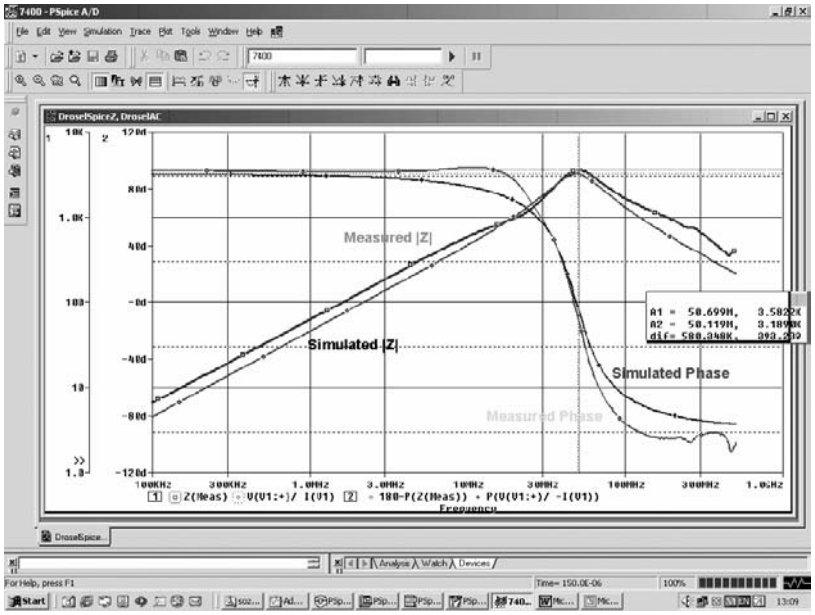

Fig. 5 Frequency response measurement and simulation comparisons in Probe

Following the suggested in topic 2 approaches the software module converting spreadsheet file with measured data to CSD file was developed. To illustrate usefulness of created module two examples are appended in the presentation. The first one concerns frequency response measurement of the coreless coil. The comparison between measured and simulated data can be seen in fig. 5. In the second one Time domain behavior of TTL 7400 is measured –

fig. 6. To achieve accurate results, should be noted that the test and simulation conditions must be equal.

4. CONCLUSION

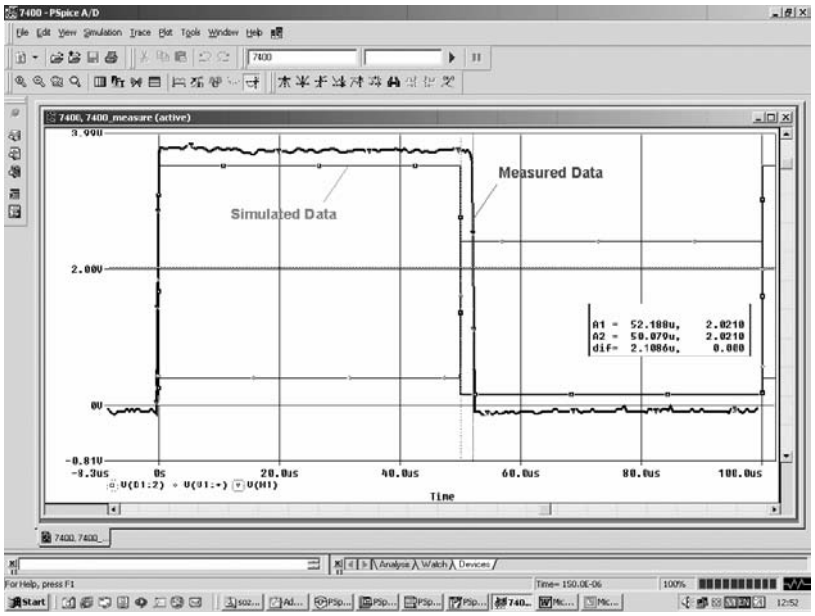

Fig. 6 Time domain behavior of TTL 7400 measurement and simulation comparisons

The design and development of software module converting measured data stored in spreadsheet format file to CSD format commonly used by PSpice compatible EDAs is presented. The exposed material can be useful for test and design engineers that going to involve the virtual measurement technologies into theirs practice. The applications of introduced software module cover - Optimizing behavioral models for top-down design and model generation; Optimizing a model to match a device's measured curves; parametric extraction; Device characterization and optimization; Engineering education; etc. It is relevant to remark that LabVIEW and PSpice have provided the educator with excellent opportunities to deliver curricula: in the hardware laboratory, to provide student self study resources, or for research purposes.

Possible utilizations are in the field of distance education and training and for the remote exchange of measured and simulated data in realm of industrial processes.

5. REFERENCES

[1] Agilent Technologies, "IC-CAP Device Modeling Software. Complete and Accurate parameter Extraction and Statistical Analysis, Release 2002

[2] Agilent Technologies, "RF Design Environment - World Class RF/Mixed-Signal IC Design Technology", November 2002.

[3] Agilent Technologies, "Advanced Design System", December 2003

[4] Teradyne, "LASAR V6.60 Simulation System - The defense industry standard for digital TPS development", 1999 (www.teradyne.com)

[5] Ansoft Corporation, "Enhancing productivity through EDA and T&M interoperability", 2003

[6] ANADIGM, "Anadigmvortex Webcast Seminar", 2003

[7] Marawar R., Edelmon G., Nybro C., Duncan S., Britton C., "Combining the use of Multisim and LabVIEW for Circuit Design and Analysis in Undergraduate Electronics Curriculum", National Instruments/ Workbench, White Paper, 2003

[8] National Instruments "LabVIEW Instrument Driver Checklist and Submittal Form", Austin, Mart 2003.

[9] National Instruments "LabVIEW Development Guidelines", Austin, 2000.

[10] DesignSoft "TinaPro Advanced Topic", User Manual, 2004.