March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

Design and Implementation of Reconfigurable Virtual Instruments using Raspberry Pi Core

[1]R.Sundaramurthy - [1]Assistant Professor, [1]Department of Electronics and Instrumentation Engineering, [1]Pondicherry Engineering College, Pondicherry
[2]V.Nagarajan - [2]Professor, [2,3]Department of Electronics and Communication Engineering, [2]Adhiparasakthi Engineering College, Melmaruvathur
[3]K.Sakthidasan @ Sankaran - [3]Associate Professor, [2,3]Department of Electronics and Communication Engineering, [3]Hindustan Institute of Technology and Science, Chennai

Article Info
Volume 83
Page Number: 12623 - 12630
Publication Issue:
March - April 2020

Article History
ArticleReceived: 24 July 2019
Revised: 12 September 2019
Accepted: 15 February 2020
Publication: 19 April 2020

Abstract:

Virtual Instrument is a combination of hardware and software that allows the emulation of an instrument through a custom virtual console and a graphical user interface. A virtual instrument consists of a PC equipped with powerful application software, cost-effective hardware such as plug-in boards, which together perform the functions of traditional instruments. In a virtual instrument, it is the software which performs the actual process of measurement. Using virtual instruments one can design a customized instrument and automation setup that is user-defined instead of being limited by traditional fixed-function vendor-defined instruments.Virtual instruments are not efficient systems to handle time critical instrument tasks since they are run by a PC operating system which introduced substantial timing errors into the measurement.The Reconfigurable Virtual Instrument (RVI) system can be considered as a general purpose measurement instrument designed with reconfigurable hardware like FPGA connected to PC through a standard port. By designing a high level software application, one can select any specific instrument functionality from a library of instruments. The high level software application configures the RVI system to convert it into the selected instrument(s) with its associated console. By this technique, one can emulate multiple instrument functionalities like function generator, oscilloscope, multimeter, logic analyzer in a single hardware platform. In this paper, hardware realization of RVI is done using FPGA and Raspberry Pi core.

Keywords:*Reconfigurable virtual instruments, Software Defined Instrumentation.*

I. INTRODUCTION

Measurement is a process of acquiring information about the parameters and variables involved in a physical system. Measurement of a given quantity is the result of comparison between the quantity to be measured and a definite standard. Measurement generally involves using an instrument as a physical means of determining a quantity or variable. A digital measuring instrument is a measuring device in which the value of the measured physical quantity is automatically represented by a number on a digital display. Such an instrument employ a transducer to sense changes in a physical parameter, such as pressure or temperature, and to convert the sensed information into electrical signals, such as voltage or frequency variations. Conventionally a specific measurement device is used for a particular measurement. Each measuring instrument is associated with unique measurement functionality and there is one to one correspondence between the instruments and the measurements made. In a typical

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

test setup involving only very few measurements, manual way of using the instrument works fine. In a complex test and measurement system involving multiple instruments, the measurement process quickly becomes cumbersome when it is done manually.The conventional design approach for instrumentation is to design and optimize the hardware in order to meet all the measurement requirements. Specifications for measurement systems reflect this optimized hardware orientation. In a conventional instrument while hardware is performing the fundamental role of measurement, software is limited to the subordinate task of collecting and managing the experimental results. In a virtual instrument, the responsibility of meeting the fundamental measurement requirement belongs to software. In a virtual instrument, it is the software which decides the functionality of the instrument. With powerful software support, the user can realize customized instruments in a PC depending upon the measurement requirement. Conventional instruments like oscilloscopes, waveform generators perform one or more tasks defined by the vendors whereas the functionality of the virtual instruments is defined by the user. In a conventional instrument, the knobs and front panel controls are associated with a specific functionality. The user cannot extend or customize these functionalities because they are defined by the manufacturer. The traditional instruments contain application specific hardware which is not flexible when the instrument functionality is to be redefined. Any upgradation in the setup requires replacement of the hardware which is expensive and ineffective.

II. RECONFIGURABLE VIRTUAL INSTRUMENTS

The Reconfigurable Virtual Instrument (RVI) system can be considered as a general purpose measurement instrument[1] designed with reconfigurable hardware like FPGA connected to PC through a standard port. By designing a high level software application, one can select any specific instrument functionality from a library of instruments. The high level software application configures the RVI system to convert it into the selected instrument(s) with its associated console. By this technique, one can emulate multiple instrument functionalities like function generator, oscilloscope, multimeter, logic analyzer in a single hardware platform. The main challenges involved in the design of RVI are the hardware platform, which should be flexible and adoptable to a wide variety of requirements. It should also be easily upgradable to take advantage of latest electronic devices and facilities. The hardware architecture of RVI is depicted in Figure 1

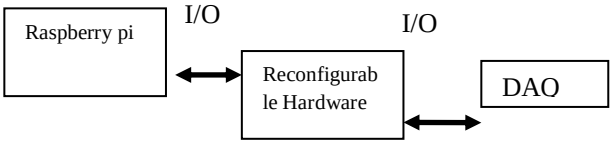

Figure 1 Reconfigurable virtual instruments

By introducing a hardware kernel in between the data acquisition stage and PC, the measurement functionality which was performed by a PC software in virtual instruments is now realized in a reconfigurable hardware[2]. RVIs are implemented on generic hardware and the underlying hardware is not explicitly designed to do any specific measurement. The entire measurement functionality is encapsulated in software written in the form a program in the reconfigurable hardware. Unlike in a traditional instrument where the measurement is performed in specific hardwired hardware logic or in a virtual instrument where the measurement is done by pure software, in a reconfigurable virtual instrument the measurement functionality is software defined in a reconfigurable hardware. The advantage of this technique is that measurement system becomes user configurable and at the same time offers real time performance as that of the conventional instrument.

III. SYSTEM IMPLEMENTATION

The hardware portion of the RVI includes three parts - a reconfigurable hardware core which is the heart of the system, instrument specific I/Os and a

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

physical interface that connects the hardware interface to Raspberry pi. The user can reconfigure the hardware kernel depending upon the measurement requirement thereby reusing the same hardware kernel for a variety of measurements. The software portion of the FPGA includes two parts - the code necessary for realizing instrument functionality in the FPGA core and the communication module to connect with a controller. It is also possible to realize multiple instrument functionalities in a single hardware kernel by integrating only the necessary measurement modules of a specific instrument. This not only reduces the form factor and power consumption of the instrument but also minimizes the overall cost of the measurement system to a great extent. Figure 2 shows the generic block diagram of hardware architecture of RVI. The FPGA unit acts as the reconfigurable core , Memory extension offers flexibility in handling real time data , instrument specific module is the minimal hardware necessary for realizing instrument specific functionalities, communication module offers the physical interface between the hardware and Raspberry pi. The RVI kernel offers the possibility of realizing multiple instruments in the same core. In a conventional virtual instrumentation technique, there is one unique underlying hardware required for each instrument used in the measurement system. If the user of the RVI is furnished with library of reconfiguration files necessary for realizing multiple instruments, it opens up the possibility of dynamically reconfiguring the hardware on the fly within the same control panel used for controlling the hardware.

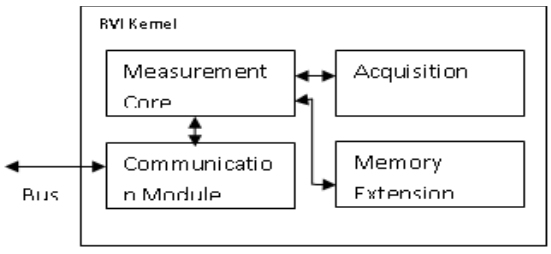

Figure 2 RVI hardware kernel

IV. HARDWARE DESIGN

A customized hardware kernel is designed for reconfigurable virtual instrument using Spartan3E FPGA ICs with the necessary support logic needed for reconfiguring the FPGA on the board and to implement a specific instrument core. It also serves the users of the RVI to prototype multiple instrument cores in a single hardware. It is designed with provisions to accept and run a user-supplied instrument core through a reconfiguring console. Reconfigurability of FPGA permits design upgrades in the field with no hardware replacement necessary. Considering the possibility to reconfigure the hardware logic, the dynamic reconfiguration of the hardware could bring new opportunities to realize multiple instruments in the same hardware kernel. The hardware of the RVI core essentially contains Xilinx Spartan-3E FPGA and Xilinx PROM Flash IC with external IO lines for interfacing instrument specific hardware. The hardware kernel is designed with onboard dedicated debugging hardware. Option for RS232 interface is provided for communicating to an external hardware or to a Raspberry pi considering its wide spread popularity. Reconfiguring the FPGA is done using master serial mode. The hardware kernel is designed with four different supply voltages using a dedicated on-board power supply - 1.2V, 2.5V, 3.3V for the FPGA's core logic and IO interface respectively, +5V for powering external add-on hardware. The identified XC3S250E FPGA core offers 250K system gates which can accommodate complex instrument cores.

4.1 Power Supply Design

The RVI hardware kernel requires multiple supply voltage inputs which is shown in Figure 3. There are two supply inputs for internal logic functions of the FPGA core namely VCCINT and VCCAUX, which operates at 1.2V and 2.5V respectively. VCCINT is the internal core supply voltage of 1.2V, which supplies all internal logic functions such as CLBs, block RAM, and multipliers. This voltage is the input to Power On Reset (POR) circuit. VCCAUX is

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

the auxiliary supply voltage of 2.5V which supplies Digital Clock Managers (DCMs), differential drivers, dedicated configuration pins and Joint Test Action Group (JTAG) interface.

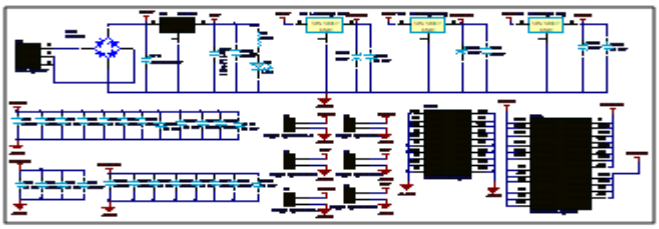

Figure 3Schematic for power module of RVI

Each of the four I/O banks has a separate VCCO supply input that powers the output buffers within the associated I/O bank. All of the VCCO connections to a specific I/O bank must be connected and must connect to the same voltage. The FPGA core has a built-in POR circuit that monitors the three power rails required to successfully configure the FPGA.

4.2 Configuration Unit Design

The functionality of a FPGA core is defined by loading application-specific configuration data into the FPGA's internal, reprogrammable CMOS configuration latches (CCLs), similar to the way a microprocessor's function is defined by its application program. The FPGA's configuration data is stored externally in a PROM or some other non-volatile medium. After applying power, the configuration data is written to the FPGA using any of seven different modes: Master Serial from a flash PROM, Serial Peripheral Interface (SPI) from a SPI serial flash, Byte Peripheral Interface (BPI) from a parallel NOR Flash, Slave Serial downloaded from a processor, Slave Parallel downloaded from a processor, Boundary Scan (JTAG)-typically downloaded from a processor or system tester.

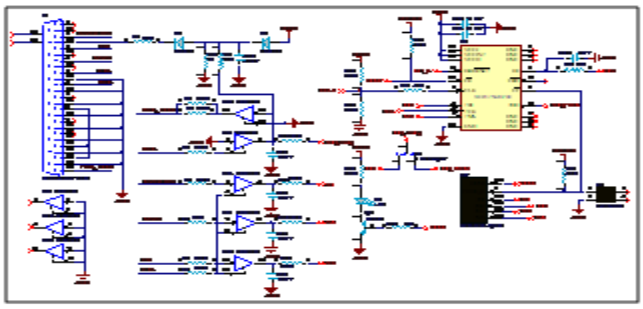

Figure 4Schematic for configuration unit of RVI

Three FPGA pins M2, M1, M0 select the desired configuration mode. The mode pin values are sampled during the start of configuration when the FPGA's INIT_B output goes high. After the FPGA completes configuration, the mode pins are available as user I/Os. Figure 4 shows the schematic connections for JTAG interface.

4.3 Design of Communication and I/O Unit

A standard 25 pin parallel port connector is essential for configuring the on board FPGA. The parallel port hardware is driven by a HC244 driver at both sides. A standard 9 pin RS-232 cable is used in the design for communicating with external hardware. CMOS RS232 transceiver MAX3232 is used as an intermediate IC for converting CMOS logic inputs into RS232 voltage levels and vice versa. RS232 is opted because of its wide spread popularity for instrument control applications. The communication unit internal to the RVI kernel should be configured to handle this communication for a specific data rate. Option for USB interfacing and its design details are furnished in the next chapter when the configuration strategies of hardware kernel are discussed.

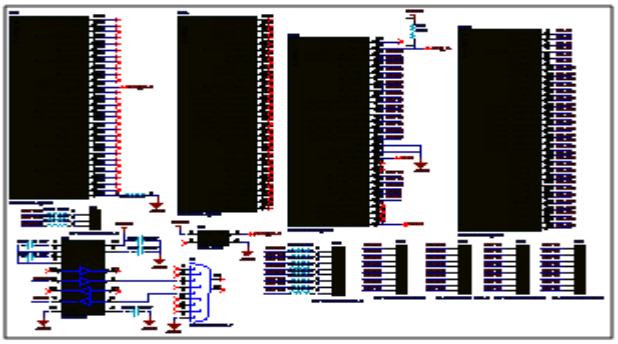

Figure 5Schematic for communication module & IO

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

The identified XC3S250E FPGA architecture offers four I/O banks-bank 0 to 3. Each bank maintains separate VCCO and VREF supplies. The separate supplies allow each bank to independently set VCCO. Similarly, the VREF supplies can be set for each bank. The following design rules were followed in assigning I/Os to banks. All VCCO pins on the FPGA are connected even if a bank is unused. All VCCO lines associated within a bank are set to the same voltage level. Figure 5 shows the I/O connections of bank0 and bank 1. Pin 178 of bank0 is meant for FPGA clock source which is fed by an external crystal oscillator of frequency 24MHz. Pin 206 is HSWAP pin which is pulled down with a 470R resistor. This pin defines whether FPGA I/O pins have a pull-up resistor connected to their associated VCCO supply pin during configuration or not. Remaining pins of bank0 and bank1 are terminated to external connector for interfacing with instrument specific hardware.Pin56 of bank 2 is INIT_B which initialize the FPGA configuration which is pulled up with 4.7K with VCCIO. Pins 84 and 86 of bank2 are the mode selection pins M1, M0. These pins must be connected to ground since master serial mode of configuration is used in the design. Pin 87 of bank2 is Data In (DIN) and the flash PROM supplies bit-serial data to this pin. The FPGA accepts bit-serial data on each rising edge of Configuration Clock (CCLK). Pin 103 of bank2 is CCLK and the FPGA configuration process is controlled by this clk. Debug LEDs are driven by an octal driver 74HC245 derived from 8 lines of bank3.

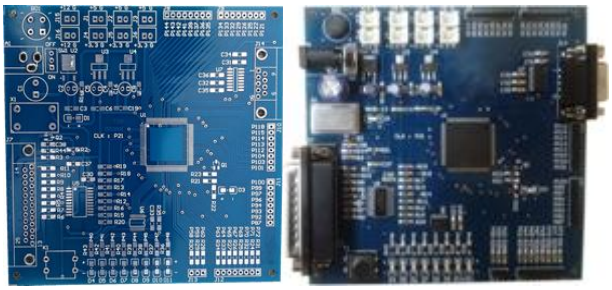

Figure 6PCB and Populated RVI Kernel

Remaining I/Os are terminated to Berge RMC connectors for interfacing with external instrument specific hardware. Pin 78 and pin 79 of bank0 are used for serial communication from FPGA to CPU. Figure 6 shows the PCB and the populated hardware RVI kernel using XC3S250E.

4.4 Design of instrument interface hardware

This section describes a general purpose analog interface board for RVI design. The analog interface card acts as the instrument specific module of the RVI. The hardware can be used to interface the input signal to the vertical amplifiers and the sampling circuits of a dual channel digital oscilloscope and logic analyzer as well as two analog outputs for a waveform generator. The key features of the hardware include an input voltage range up to ±25 V, an input sensitivity of 12 mV, an analog bandwidth of 2 MHz, and a maximum sampling rate of 500 kHz. The hardware is designed with an onboard signal conditioning circuit to meet the following design criteria - a standard input impedance, a programmable attenuation and gain, a direct-current decoupling, a wide bandwidth with low distortion, and a simple design with few components. The design goals are not easy to accomplish considering the real parameters of the components like amplifiers, passive devices and analog switches. Some of the critical parameters of the analog switches include ON-resistance, the leakage current, the voltage range, the OFF-resistance. In the case of amplifiers, the input impedance, the gain bandwidth product, the input bias current, and the phase and gain margins are also critical. Taking into account the above mentioned aspects, a general purpose instrument interface card is designed in this work. AC/DC Coupling is provided by the use of a solid-state relay of PhotoMOS technology with low-level leakage currents and low closed-circuit offset voltage. This enables a large dynamic range of analog input signals with a minimum distortion. Attenuation is provided by an RC network with two analog

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

switches providing the two attenuation ratios. The first ratio is 1:2 and the second ratio is 1:10, accordingly a pair of analog switches is used. Amplifier with selectable gain consists of a set of precision-voltage-dividers resistors array, digitally selectable by means of a set of analog switches and an operational amplifier. This board has two 8-bit ADCs and one 8-bit DAC operating up to 500 kHz.

4.5 Firmware implementation for RVI

The core measurement process is done in the RVI FPGA kernel once the data is acquired by the instrument interface card. Modular implementation methodology is followed in realizing a new instrument in the RVI kernel. This structural methodology includes a top level implementation followed by intermediate and low level implementation. Figure 7 shows the firmware implementation for RVI FPGA core. The top level implementation is used to define the description of the measuring instrument hardware. Each instrument is associated with a communication bus controller which acts as the instrumentation bus arbiter and bridge with the external bus interface.

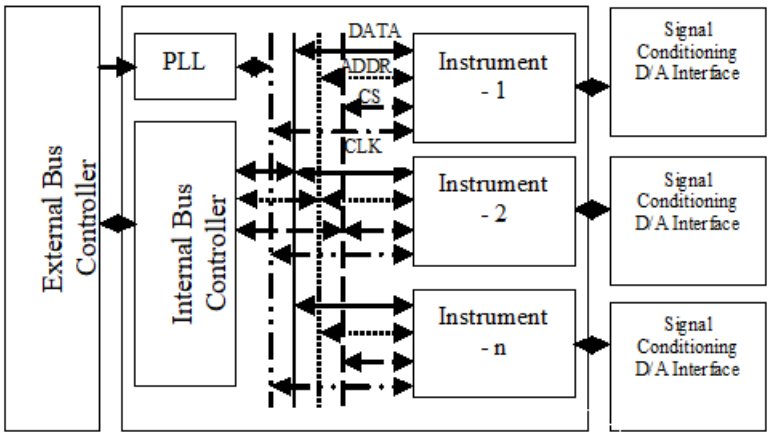

Figure 7 Firmware implementation for RVI kernel

The intermediate level implementation describes the characteristics of a particular instrument, including the control and functional blocks. Instrument controller block is the key component for intermediate level implementation and they are derived based on data sampling, data generation or a combination of both.The functional modules of each instrument are realized as individual modules. Each module performs a unique functionality like digital data filtering and analog waveform synthesizer. A library of fully tested configurable functional modules are developed in this work, which can be used by the user to quickly configure an instrument from the scratch. All the functional modules in this work are realized using Very high speed IC Hardware Description Language (VHDL). The user can construct any instrument by programming the RVI kernel using this library of functional modules. The dynamic reconfiguration feature of the FPGA allows the user to modify the functional modules according to the measurement requirement. At the low level, the functional module of each instrument is defined completely depending upon the functional requirement of the instrument. For example for an analog interface controller module generates the required signals to control the ADC and synchronize the data sampling during data acquisition. A memory module is used to manage the transference of the acquired data and to indicate when it is full. A communication controller is used configure the data frames and establish the handshake between the instrument and the bus controller.

V. RESULT ANALYSIS

The hardware kernel designed in the previous section is used for implementing multiple instrument

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

functionalities in a common hardware platform. The process of reconfiguration was done from the Xilinx ISE which can be done through customized consoles if the configuration mode is changed. The following are the various instrument functionalities realized using a single hardware kernel in this work - data logger, arbitrary waveform generator, digital storage oscilloscope, logic analyzer, function generator and digital multimeter. The instrument control firmware for the above mentioned instruments were developed based on Standard Commands for Programmable Instruments (SCPI) standard, which specifies a common syntax that is aimed at being used for controlling test and measurement instruments. SCPI provides syntax, command structure and data interchange format, which are recognized by the instruments to enable specific actions to be taken. Application software for the RVI are done using python language in Raspberry pi. Figure 8 shows the software Panel of RVI.

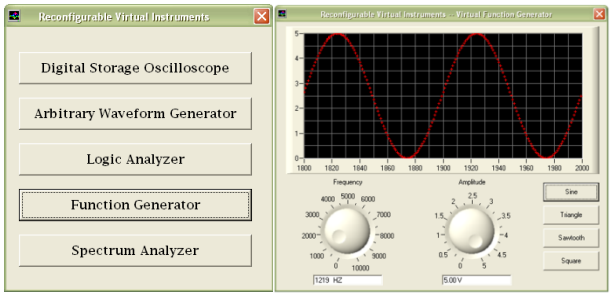

Figure 8Software Front Panel for RVI

Performance analysis of the RVI hardware kernel is done with a conventional virtual instrument for a typical oscilloscope instrument for the following parameters – measurement accuracy, repeatability and cost. Figure 9 shows the comparative analysis of the reconfigurable virtual instrument with a conventional VI normalized for accuracy, cost and precision. It is seen that for similar measurement accuracy and precision, the reconfigurable virtual instrument provides significant cost advantage over the conventional VI by a factor of 58% because of its ability to minimize hardware redundancy.

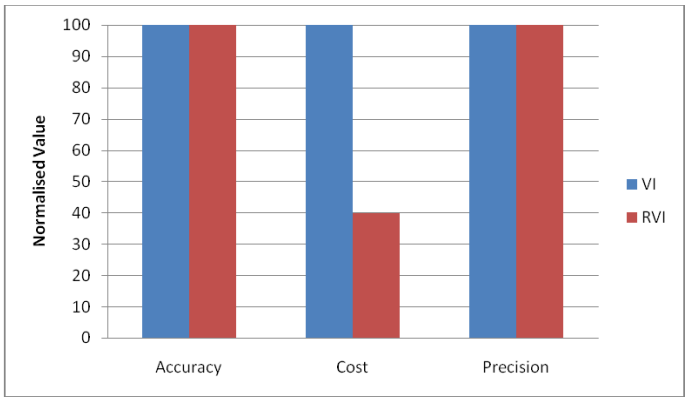

Figure 9Comparative result analysis of RVI and VI

The photograph of the hardware setup is shown in Figure 2.22. The hardware kernel used is Xilinx XC3S250E and the waveform generation with data acquisition is done by separate daughter board. The digital storage oscilloscope captures the real time waveform generated by the RVI when it is reconfigured as function generator instrument. Application software form the Raspberry pi controls the hardware in real time based on the SCPI commands. The user of the instrument can also control the hardware from terminal emulation software by using the command set.

REFERENCES

1. R.Sundaramurthy,P.Dananjayan," Control Strategies for Reconfigurable Virtual Instruments using FPGA" European Journal of Scientific Research, Volume 93 ,Issue 1, December– 2012
2. R.Sundaramurthy,P.Dananjayan," Design and Realization of Reconfigurable Virtual Instruments using FPGA" European Journal of Scientific Research, Volume 90 ,Issue 1, November – 2012
3. R.Sundaramurthy,P.Dananjayan,"Service provider architecture for dynamically reconfigurable virtual instruments in networked environments" International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, December 2010.
4. R.Sundaramurthy,P.Dananjayan,"Speech Controlled Reconfigurable Virtual Instruments Using In-System Programmable Core" International Conference on Mechanical and

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 12623 - 12630

Industrial Engineering (ICMIE), Chennai, December-2011.

5. Piotr Bilski and WieslawWiniecki, "Methodology of the Real-Time Multicore Virtual Instrumentation Design" in I2MTC 2009 - International Instrumentation and Measurement Technology Conference Singapore, 5-7 May 2009
6. Mark Meldrum ,Aeroflex, "Mapping Multiple Legacy Instruments and Associated Measurements into a Single Synthetic Test Environment" in Proceedings of IEEE International Conference, AUTOTESTCON2008, Salt Lake City, UT, 8-11 September 2008.
7. Guo-Ruey Tsai andMin-Chuan Lin - "FPGA-Based Reconfigurable Measurement Instruments with Functionality Defined by User" EURASIP Journal on Applied Signal Processing Volume 2006, Article ID 84340, Pages 1–14 DOI 10.1155/ASP/2006/84340.
8. G.R. Tsai, M.-C. Lin, G.-S. Sun, and Y.-S. Lin, "Single chip FPGA-based reconfigurable instruments," in Proceedings of International Conference on Reconfigurable Computing and FPGAs (ReConFig '04), Colima, Mexico, September 2004.
9. G.R. Tsai, M.-C. Lin, W.-Z. Tung, K.-C. Chuang, and S.- Y. Chan, "Wide-band and precisely measurement method of phase detector based on FPGA with embedded processor," in Proceedings of International Conference on Informatics, Cybernetics and Systems (ICICS '03),
10. I-SHOU University, Kaohsiung, Taiwan, December 2003.
11. J.W. Hsieh, G.-R. Tsai, andM.-C. Lin, "Using FPGA to implement a n-channel arbitrary wave form generator with various add-on functions," in Proceedings of 2nd IEEE International Conference on Field-Programmable Technology (FPT '03), pp. 296–298, University of Tokyo, Tokyo, Japan, December 2003.

Published by: The Mattingley Publishing Co., Inc.