An Alternate Control Scheme for Reconfigurable Virtual Instruments

R.Sundaramurthy
Departmant of EIE,
Pondicherry Engineering College,
Puducherry,India.
e-mail: sundar@pec.edu

P.Dananjayan
Departmant of ECE,
Pondicherry Engineering College,
Puducherry,India.
e-mail: pdananjayan@pec.edu

Abstract— The widespread usage of personal computers in many scientific and technological fields makes them an ideal hardware and software platform for the implementation of measurement instruments. Reconfigurable virtual instruments are implemented using a universal general purpose reconfigurable hardware whose functionality is defined by the measurement requirement. It is a versatile hardware device that can be reconfigured into different electronic instruments using a software tool. A high-level software application runs on the PC and provides a user interface to the operator who can select a virtual instrument (e.g. digital oscilloscope, arbitrary waveform generator, logic analyzer, digital filter…) from a library of instruments and configures the RVI system to convert it into the selected instrument with its associated console. The speech recognition interface enhances the ability of the operator to control various system components without manually navigating the graphical user interface (GUI). Several options were considered during the analysis but only one option proved to be optimal. The solution described in this paper uses the NXP semiconductors ARMLPC2148 microcontroller to handle all speech recognition calculations. The GUI control system runs on the main PC processor and the controlled instruments are attached to the system through RS232 interface. Speech recognition performance analysis is done for both PC based approach and dedicated hardware based approach in terms of realizing and controlling the VIs and the results are compared.

Keywords- *Reconfigurable Virtual Instruments (RVI), Speech Recognition*

*****

I. INTRODUCTION

A virtual instrument results from the combination of a general purpose computer with a generic hardware in order to emulate a traditional measurement instrument .Many a time, for a specific Test and measurement requirement we have to use different virtual instruments. Moreover we normally don’t use all the functionality of a given VI hardware. So it is desired to have a single hardware platform, where multiple virtual instruments are realized with the required functionality by reconfiguring the hardware. A high-level reconfigurable virtual instrument (RVI) system architecture comprises both hardware and software sub-systems[1]. The hardware includes a standard personal computer and the reconfigurable instrument (RI) connected to it through a physical connection.

The connection allows the reconfiguration of the RI as well as the exchange of information between the PC and the RI. The information exchanged can be data, commands, error messages, or acknowledgement messages. Interest in reconfigurable devices today is encouraged by the fact that manufacturers promote some of their products as dynamically reconfigurable. The specification and design of such a system is done at the highest level of abstraction, register transfer level (RTL) using hardware description language (HDL).

Through a reconfigurable virtual instrumentation (RVI) platform, it is possible to emulate many standard general purpose instruments. It is also possible to implement sophisticated instrumentation for custom specific applications that cannot be accomplished by standard instruments. A single hardware kernel is used for performing different instrument functionalities. Reconfiguration in the present day RVI systems are done by manually controlling the GUI, which adds lot of latency when the measurement system requires lot of VI.

In this paper, we present a virtual instrument system based on reconfigurable hardware that improves the features of virtual instruments preserving their versatility and low cost along with speech control feature for the virtual instrument for dynamic reconfiguration.

Speech recognition technology is ubiquitous, and there are varieties of products available in today’s markets that incorporate speech recognition to facilitate hands-free or convenient operation of the devices in our environments [2]. Speech controlled reconfigurable virtual instruments enhances the ability of the operator to control the virtual instruments by voice commands. for developing speech controlled virtual instrument, two vital components are needed - An instrument control software running in the PC to control the instrument and a reconfigurable instrument hardware which is to be controlled.

The voice command is picked up by the microphone which is digitized by voice acquisition hardware. The voice recognition engine embedded in the application software will parse the user commands and control the virtual instruments. As a comparative approach , the voice recognition engine is also realized in the hardware to test the performance of the system. Two distinct classes of command sets are provided. Once class of commands will reconfigure the hardware dynamically depending upon the measurement requirement and the other class of commands controls the operation of the instrument itself.

The application software which is the virtual instrument will recognize the voice command and send the corresponding data to the instrument kernel. Communication is established between instrument and the PC by RS232 serial communication port. The hardware instrument kernel will receive this data and take the necessary action depending upon the voice command from the user.

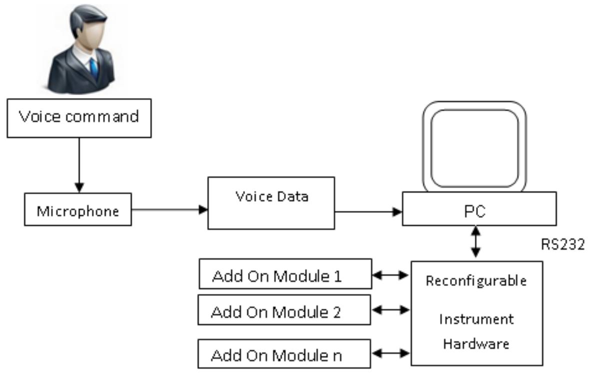

Figure 1. Block diagram of RVI

II. CONCEPTION OF VIRTUAL INSTRUMENT

2.1 Traditional Instruments

Stand-alone traditional instruments such as oscilloscopes and waveform generators are very powerful, expensive, and designed to perform one or more specific tasks defined by the manufacturer. But , the user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, and the functions available to the user, are all specific to the nature of the instrument and are not customizable. In addition, special technology and costly components must be developed to build these instruments, making them very expensive.

2.2 Virtual Instruments

Virtual instrumentation is the use of customizable software and modular measurement hardware to create user-defined measurement systems, called virtual instruments.

Traditional hardware instrumentation systems are made up of pre-defined hardware components, such as digital multimeters and oscilloscopes that are completely specific to their stimulus, analysis, or measurement function. Because of their hard-coded function, these systems are more limited in their versatility than virtual instrumentation systems. The primary difference between hardware instrumentation and virtual instrumentation is that software is used to replace a large amount of hardware. The software enables complex and expensive hardware to be replaced by already purchased computer hardware;

e. g. analog to digital converter can act as a hardware complement of a virtual oscilloscope.

2.3 Reconfigurable Virtual Instruments

Reconfigurable virtual instruments use a single hardware kernel and the user can reconfigure the same kernel for a variety of instrument requirement by making using of a software[3].

2.4 Speech Controlled Reconfigurable Virtual Instruments

Speech controlled virtual Instruments enhances the ability of the operator to control the virtual Instruments by Voice commands. For developing Speech controlled Virtual Instrument, two vital components are needed - An Instrument control software running in the PC to control the instrument and an Instrument hardware [RVI] which is to be controlled. Voice data acquisition is done by using two methods in this paper – In the first approach the voice command is picked up by the microphone which is connected to the dedicated data acquisition hardware used for realizing RVI itself. In the second approach PC performs the voice data acquisition. In the first approach , the hardware would acquire the voice data , analyze and recognize the voice data to perform instrument specific functionalities. In the second approach the application software which is the Instrument control software will recognize the voice command and send the corresponding data to the hardware instrument. Communication is established between Hardware Instrument and the PC by RS232 serial communication port. The Hardware Instrument will receive this data and take the necessary action based on the voice commands by the user. The difference between the two approach is discussed in the following figure.

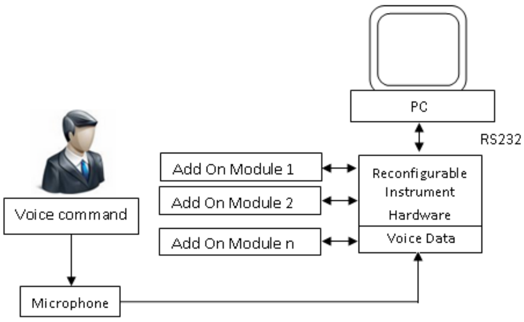

Figure 2. Block diagram of Speech controlled RVI

III. SPEECH RECOGNITION

Although various techniques are available for Speech recognition, the most widely used algorithm is Hidden Markov Model (HMM) for speech recognition. The HMM is a probabilistic pattern matching technique in which the observations are considered to be the output of stochastic process. If the probabilities between the spoken word model and the trained word model is minimum then the word would be recognized. But the only practical issue with the HMM model is, it requires high speed processor. Another Speech recognition algorithm is Tor's algorithm which calculates the Speech templates using FFT or band pass filter method. If the Euclidean distance is minimum between the already stored templates and the current templates for the particular word then the word would be recognized. But the recognition accuracy is very less. From the PC side, speech processing SDKs are being used which performs accurate speech recognition. In this paper , HMM is used in the hardware side and SDK is used in the PC side toevaluate the performance of both.

3.1 Speech Recognition Using Hmm

Among several types of speech recognizers, the Hidden Markov Model is one of the most dominant algorithms and has proven to be an effective method of dealing with large units of speech. HMM computes the probable match between the input it receives and phonemes contained in a database of hundreds of native speaker recordings. That is, a speech recognizer based on HMM computes how close the phonemes of a spoken input are to a corresponding model, based on probability theory.

The speech-recognition algorithm, contains two fundamental parts, which are the acoustic front end and the search algorithm itself. The acoustic front end is the process of converting sequences of raw-speech data to observation vectors, which represent events existing in a probability space. The search algorithm then finds the most probable sequence of these events while operating under a set of syntactic constraints.

Speech recognition in HMM is implemented by using the following steps.

1) For each Normalized wave file the beginning and end of each word is found , following these steps:
   (a) Division of the signal in 512-point frames, with 200-point overlaps.
   (b) Calculation of the signal's energies and number of zero crossings.
   (c) Based on those values the word is isolated.
   (d) Calculation of the signal's noise energy.
2) In the second phase, the Mel-cepstral coefficients and delta energies are calculated.
3) The vector codification of the frames using Kmeans algorithms is realized.
4) Next step is the recognition or system training using Viterbi algorithms.
5) Finally, the HMMs are used to obtain a final vector of the characteristics of the whole word.

This vector is compared with the initially trained table of word vectors. Speech is converted to a digital representation by sampling speech at the rate of 8 kHz/second.

The algorithm was coded in real time using embedded C-language program which was then ported to the ARM LPC 2148 platform. Extensive effort was made to place the organization of the code and the implementation of the memory overlays in a manner that is tailored to the architecture and memory constraints of the ARM LPC2148.

3.2 Speech Recognition Using Sdk

In the second approach , Microsoft Speech SDK- which is a software development kit for building speech engines and applications for Microsoft Windows is used for speech recognition. The SDK contains the Microsoft Win32-compatible speech application programming interface (SAPI), the Microsoft continuous speech recognition engine and Microsoft concatenated speech synthesis engine, a collection of speech-oriented development tools for compiling source code and executing commands.

The SAPI provides a high-level interface between an application and speech engines. SAPI implements all the low-level details needed to control and manage the real-time operations of various speech engines. The two basic types of SAPI engines are text-to-speech (TTS) systems and speech recognizers. TTS systems synthesize text strings and files into spoken audio using synthetic voices. Speech recognizers convert human spoken audio into readable text strings and files. Speech application programming interface[SAPI] is integrated along with the visual studio GUI for the purpose of recognizing and reconfiguring the VI as per user commands

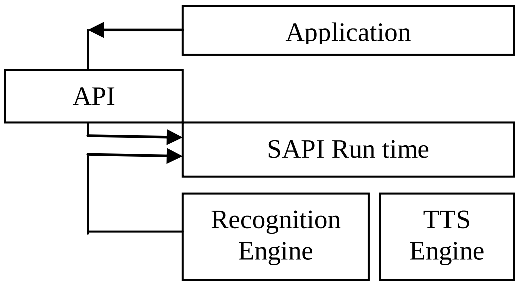

Figure 3. Block diagram of Speech application programming interface (SAPI)

IV. RVI REALISED IN THIS WORK

With the complete hardware kernel architecture, we can configure the ARM microcontroller to match the necessaryfunction specifications for various measurement environments and requirements. Here for this work the followinginstruments are considered for dynamic reconfiguring

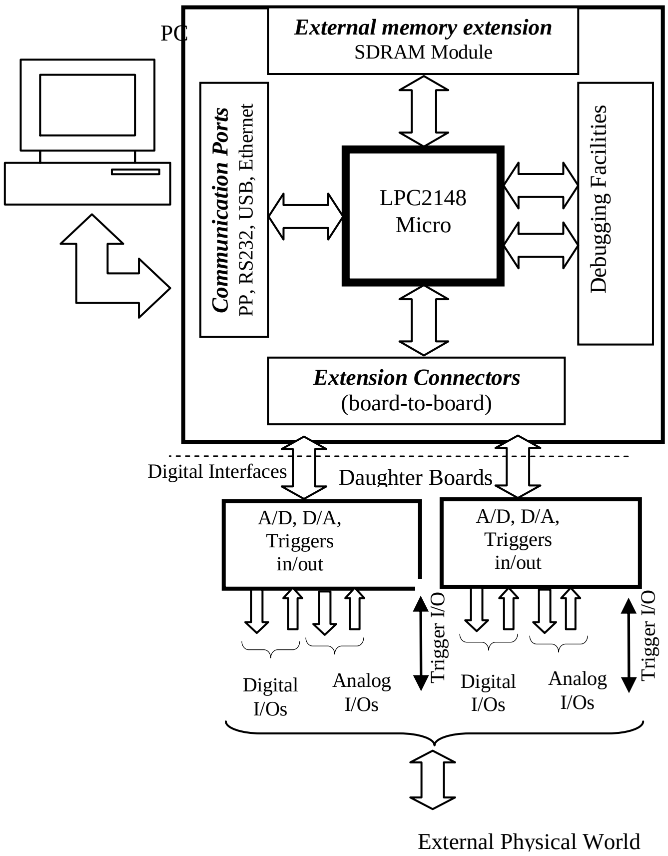

Figure 4. Block diagram Hardware Kernel RVI Main Board

Arbitrary Waveform Generator, Oscilloscope, Logic Analyzer, High Speed Data Logger, Function Generator, Spectrum Analyzer. The voice based control is achieved for three instruments namely Virtual Function generator , Spectrum Analyzer and Logic Analyzer. Virtual Function Generator Utilizing direct digital synthesizer (DDS) algorithm[4], we can generate any periodic function with arbitrary frequency, amplitude, and waveform. The function waveforms are preloaded into flash RAM, and will be directly loaded into the built-in RAM in the Microcontroller when powered up. The waveform frequency can be set up to half the system clock. Using 32-bit phase accumulator one can achieve considerable frequency resolution. The embedded processor is in charge of the control of HIDs and setting calculation of frequency and amplitude. Reorganizing the DDS data processing path , single channel analog output is derived for standard and arbitrary waveforms.

Logic Analyzer.

A logic analyzer is realized in this work , that displays signals in a digital circuit that are too fast to be observed and presents it to a user so that the user can more easily check correct operation of the digital system. They are typically used for capturing data in systems that have too many channels to be examined with an oscilloscope.

Spectrum Analyzer

A real-time sweep spectrum analyzer is realized using a fixed IF filter and a sweeping local oscillator (LO).The mixer output contains the input signal, the LO signal, the sum and difference between these two signals, and various other frequency components. If we know the LO frequency exactly, then by sending these frequency components through a narrow IF filter, we can identify both the amplitude and the frequency of the unknown input signal. Whenever any of these components falls within the IF filter bandwidth, an AC voltage, which is related to the input signal's amplitude, is produced. This AC voltage is converted to a DC voltage by an envelope detector, and the result is displayed on the y-axis of the screen[5].

V. HARDWARE IMPLEMENTATION

The figure4 illustrates the proposed hardware kernel architecture for voice Based RVI. Besides Microcontroller LPC2148, other ASIC chips are needed to process analog signals. For data acquisition the system uses the onboard10 bit ADC of the microcontroller. The original stimulus signal generated by the microcontroller is in digital form. For generating analog waveforms the built in DAC feature of the microcontroller is used. For analog exciting signal requirement, it must be converted by digital-to-analog converter, filtered and shaped by low-pass-filter and amplified or attenuated by amplifier or DC offset. The other accessory systems include precision voltage regulators for power supply, ISP circuit for reconfiguring the system on the fly , serial interface system for PC communication, Brown-out detection for the microcontroller and an optional SD-MMC card for data storage. The entire hardware kernel is controlled by application software which reconfigures all the instruments virtually into a single instrument, controlled by the virtual core. Altogether, a reconfigurable virtual instrument is designed and implemented by the programmable core. The functionality of the hardware core is controlled by the voice commands from the user.

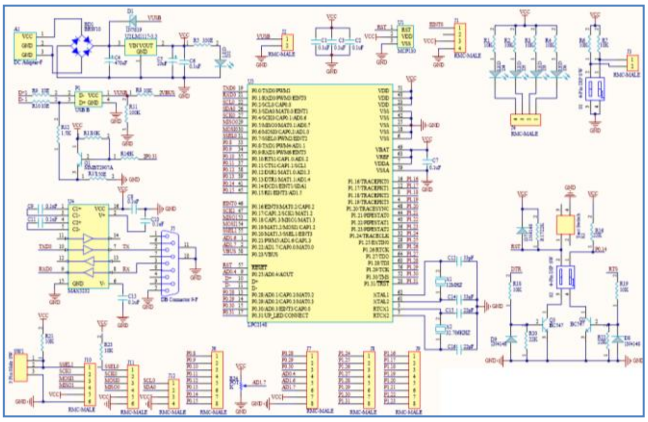

Figure 5. Schematic of hardware kernel

VI. APPLICATION SOFTWARE DEVELOPMENT

The application software for this work has been developed using visual studio GUI. The voice commands given by the user is acquired by the system in two different methods which act as the input for the system. This input is processed by the application software that invokes the appropriate application to transform the hardware kernel to a specific instrument. The following are the sample commands provided for processing function generator virtual instrument. The user commands are parsed by the system in two layers.

- Layer1 –Reconfiguration commands
- Layer2 – Instrument control commands.

Reconfiguration commands are used to reconfigure the hardware kernel for a specific instrument requirement like function generator , oscilloscope tc.

Instrument control commands are used to control a specific instrument and its functionalities.

The system waits for a valid command from the user. Upon receiving a valid command , the system first reconfigure the hardware kernel by using ISP capability of the microcontroller at Run time.

Later the system moves to layer2 where in it expects the user for instrument control commands for controlling that particular instrument. At any given point of time , command 'EXIT' would exit the hardware kernel from the current functionality.The screen shot of the login window for entering the RVI is given below. For security purpose a unique user ID and login password is assigned to the system. The com port selection option is also provided when more than one port is used. The below window allows the access to the RVI when proper login details are provided

ISSN: 2321-8169
26 – 31

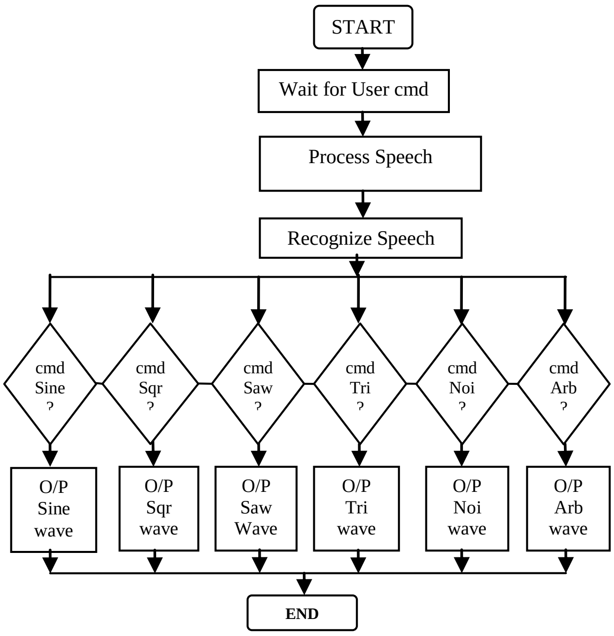

Figure 6. Figure6. Algorithm for function generator VI control

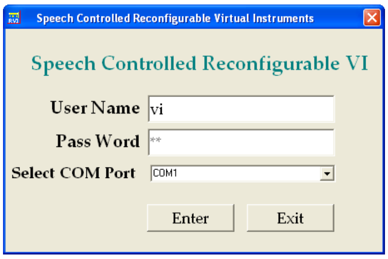

Figure 7. Screen shot of software Panel

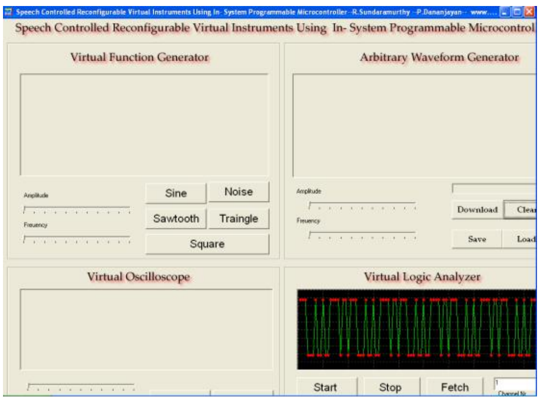

Figure 8. Screen shot of main panel

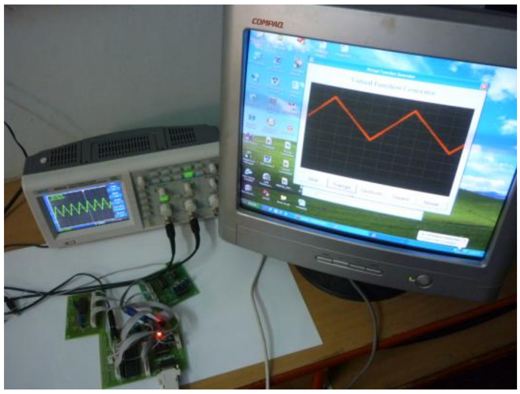

Figure 9. Photograph of speech RVI hardware setup

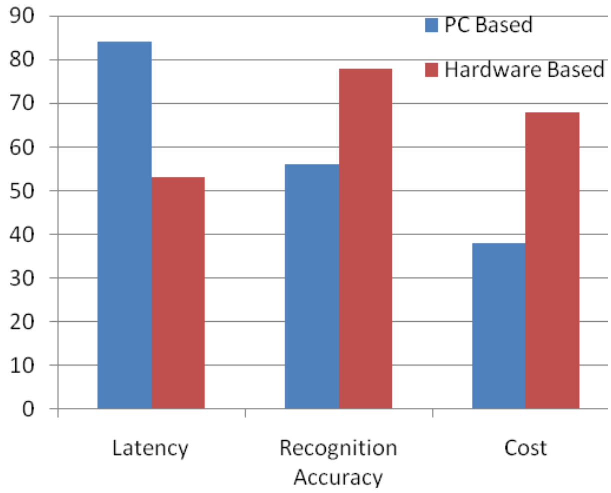

Figure 10. Comparative analysis of PC based and Dedicated hardware based speech recognition.

VII. HARDWARE IMPLEMENTATION

The rapid development in measurement and test technology has transferred functions from manufacturer's choice to user's choice. This ability makes the device reconfigurable, robust, and rapidly prototyped will be the main criteria in measurement and in test market. This versatile hardware device can be reconfigured into different electronic instruments using a software tool. The speech recognition interfaced application runs on the PC and provides a user interface to select a virtual instrument from a library of instruments and configures the Speech Controlled RVI

system to convert it into the selected instrument with its associated console.The above shows the comparative analysis of PC based speech recognition versus the speech recognition done by the dedicated arm core. It is found that the dedicated hardware core offers better accuracy and latency figures, but at the expense of higher hardware cost. This paper demonstrates all these in Single Embedded Hardware using in system programmable microcontroller ARMLPC2148.

REFERENCES

[1] 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.

[2] Sergiu Nedevschi, Rabin K. Patra, Eric A. Brewer" Hardware Speech Recognition for User Interfaces in Low Cost, Low Power Devices" DAC 2005, June 13.17, 2005, Anaheim, California, USA

[3] G.R. Tsai, M.-C. Lin, G.-S. Sun, and Y.-S. Lin, "Single chip FPGA-based reconfigurable instruments," Proceedings of International Conference on Reconfigurable Computing and FPGAs (ReConFig '04), Colima, Mexico, September 2004.

[4] 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," Proceedings of 2nd IEEE International Conference on Field-Programmable Technology (FPT '03), pp. 296–298, University of Tokyo, Tokyo, Japan, December 2003.

[5] 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," Proceedings of International Conference on Informatics, Cybernetics and Systems (ICICS '03), I-SHOU University, Kaohsiung, Taiwan, December 2003.

[6] ReguArchana, Mr. J.V.Rao, " Implementation of I2C Master Bus Protocol on FPGA", Int. Journal of Engineering Research and Applications 2248-9622, Vol. 4, Issue 10, October 2014.

[7] RajuPatil, PanditNad, Smt.SujathaHiremath, " FPGA Implementation of Smart Multi-Protocol Translator", International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013.

[8] Jianlong Zhang, Chunyu Wu Wenjing Zhang, Jiwei Wang, "The design and realization of a comprehensive SPIinterface controller", 2011 IEEE

[9] G.Sai Ram, G.Ravi Chandra, K.mohanrao," Design and Implementation of SPI with Built-In-Self-Test Capability over SPARTAN 2", International Journal of Advanced Research in Computer and Communication EngineeringVol. 4, Issue 4, April 2015

[10] K.S.Gowthaman, L.Jubaired, S.Janani, "Effective Communication Protocols For Verification On SocUsing FPGA", International Journal Of Advanced Engineering Research And Technology (IJAERT)Volume 2 Issue 7, October 2014.

[11] Bibin M C, Premananda B S, "Implementation of UART with BIST Technique inFPGA", International Journal of Inventive Engineering and Sciences Volume-1, Issue-8, July 2013.