# Power Quality Meter Based on FPGA and LabVIEW

## <sup>1</sup>G. Petrović, <sup>1</sup>I. Šimić, <sup>1</sup>J.A. Bosnić, <sup>2</sup>P. Mostarac

<sup>1</sup>FESB, University of Split, Split, Croatia <sup>2</sup>FER, University of Zagreb, Zagreb, Croatia Email: petrovic@fesb.hr

Abstract. This paper proposes a relatively simple program for measuring currents and voltages in electrical power system. Real-Time processing is achieved by using CompactRIO programmable automation controller which combines embedded Real-Time and FPGA modules. The device also enables users to quickly develop program code via LabVIEW graphical programming language. The process of data exchange between FPGA and RT modules using FIFO registers is described in details. In the end, some Power Quality algorithms are presented, according to IEC standard and attractive Graphical User Interface for PQ monitoring is displayed.

Keywords: Power Quality, Signal Processing, FPGA Programming

## 1. Introduction

The frequency and RMS voltage are the most important parameters of the electrical power system. While the voltage variations are allowed in relatively wide range, in normal operating conditions, the permissible frequency deviation is  $50 \pm 0.05$  Hz [1]. In order to control the frequency, different algorithms have been developed for the estimation of the frequency, such as Kalman filtering, adaptive neutral network, adaptive notch filters and others [2]. Fortunately, much simpler techniques can be used for PQ monitoring [3, 4]. According to the standards for PQ measurement, all analyses are conducted on a sample length of 10 cycles, or about 200 ms. Therefore, before calculating the RMS value, or harmonic analysis, resampling process is carried out. Well-known algorithm for which LabVIEW developed a special VI is used here. The algorithm is time-dependent and must be running on a fast signal processor or Field Programmable Logic Gate (FPGA).

CompactRIO programmable controller manufactured by National Instruments combines embedded Real-Time (RT) and FPGA modules that enable users to quickly execute complex program code. CRIO controller can be programmed via graphical programming language LabVIEW [5], which is much simpler than other programming languages such as VHDL. Complete hardware setup required for measuring voltages and currents include 3-channel voltage module and 4-channel current module. Both modules have a 24-bit Delta-Sigma ADC with a sampling rate of 50 kS/s. User interface and development environment are designed on a Windows PC.

## 2. FPGA Based Measurement and Resampling

CRIO allows combined work of FPGA and Real-Time modules. A crucial part of the code is executed on the FPGA, and then host program performed less demanding processing on the Real-Time processor. Target (FPGA) and Host (RT) use DMA (Direct Memory Access) engine for communication between them which is possible in both directions. Structures that are used for data transfers are FIFO's which retain data in the same form as it's been received and allow access to the data by the First-In, First-Out principle. There are several ways to transfer data through the FIFO's. In our case, we will use two: DMA FIFO's for transferring large amounts of data and to allocate memory on FPGA and Host side, and program defined FIFO's used for data transfers within the loops on the FPGA side. In LabVIEW program

structures, loops are executed parallel, so FIFO structure is necessary for data exchange. Fig. 1. shows the FPGA program consisting of initialization structure and three loops. Error code wiring assures sequential execution and three loops parallelism.



Fig. 1. LabVIEW program for acquisition and resampling.

The task of the first While loop is data acquisition from the modules, and it's execution rate is determined by the execution rate of acquisition modules. There are 7 input nodes in this loop that represent the module inputs (3 for voltage signals and 4 for current signals). Data is transformed into an array based on interleaving method, so data from more channels are entering the FIFO. It is important to bear in mind that the selected modules 9225 and 9227 uses FXP data type accuracy <+/-, 24,10> and the different programming blocks within the program work with the <+/-, 24,1> accuracy, so it's important that conversion is carried out before entering the FIFO. FOR loop is used to send a one-dimensional array of data in FIFO. Its execution is determined by input array length. In our case, we use two program-defined FIFO's to transfer information between loops. In the first loop, FIFO is named "Time Signal" indicating that it contains information of the input waveforms from 7 channels.

The second While loop within the program is deterministic, meaning that execution rate is determined by micro-controllers clock rate. High acquisition speed enables execution of the resampling algorithm. The signals read from FIFO, sampled in the same time intervals, are converted into signals with equal angular spacing (fixed-angle sampling). Resampled data from seven channels and instantaneous frequency are grouped in cluster. This cluster of data is filled in another program-defined FIFO called "Resampled signal" which transfers data into last While loop where output data type located in FIFO changes from FXP to U32 data type. Finally, data is written to the DMA FIFO called "AI". DMA FIFO allocates memory both on the FPGA and Real-Time side, but still acts as a single FIFO. Data from the DMA FIFO's can be accessed on a host computer with the Real-Time program. The amount of data that will be transferred from the DMA FIFO's can be regulated with Invoke method on the host side that takes part of data or all data from DMA FIFO and transfer them to Host. Based on the transferring needs we chose between ''Blocking'' or ''Pooling'' method type.

#### 3. PQ Parameters Calculation on Real-Time Machine

The task of Open 'FPGA VI Reference' command is to call generated and compiled FPGA program which ensures that the measurement resampled data are transferred via DMA FIFO on the host side. Host side provides a lower speed data processing, but considerably more flexibility for demanding mathematical algorithms, as it's shown in Fig. 2.



Fig. 2. Program on Real Time host for communication with FPGA target and signal processing.

To configure the parameters of the FIFO register at the host side AI.Configure Invoke method is called. This command allows us to determine the depth of the FIFO's. We select some elements that is at least twice the number of elements in one packet of data, to prevent buffer overflow. FPGA side takes one sample at each channel at a time and after resampling, writes it to the FIFO, in the rate of about 10,000 times per second, Host side reads one packet of data from the FIFO's 5 times per second. Choosing AI.Start Invoke method, we launch a very fast DMA engine that uses memory allocated by the DMA FIFO's on both sides. Within the FPGA Interface palette, we withdraw Read/Write control. Using this command, the desired number of samples per cycle and the number of channels are defined in the FPGA VI. AI.Run command run FPGA VI on the FPGA target.

Inside While loop for the data processing we created AI.Read Invoke method which allows us to read the collected data on the FPGA side, that are transmitted using a DMA FIFO's. Within the While loop for processing, we set another While loop in which we again call AI.Read Invoke method whose task is to check whether the data is ready for transfer. We use the pooling method, which is based on the principle that the data collected from the FPGA side which acquisition rate is known, transfers to the Real-Time side that doesn't have defined data acquisition rate. If the whole packet of data is not available we wait for 10 ms, and check again. One packet of data consists of a number of channels multiplied by the number of samples in the 10 cycles. Now the AI.Read Invoke method is able to access the data available from the DMA FIFO. After reading, one dimensional array of data is transformed into two dimensional array, which means that each channel will become a column. The eighth channel that is created inside the FPGA program is instantaneous frequency.

To calculate the most important PQ parameters, we can use algorithms in according to IEC standard [5]. Harmonic analysis of voltages and currents are made by DFT algorithm (1).

$$RMS(u) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} u_k^2}; \quad DFT(u) = U_n = \frac{1}{N} \sum_{k=1}^{N} u_k e^{-j\frac{2\pi}{N}nk}; \quad THD(u) = \sqrt{\sum_{h=2}^{40} \left(\frac{U_h}{U_1}\right)^2}$$
(1)

where: *u*- voltage or current; *U*- frequency components of signal *u*; *N*- number of samples.

Resampling algorithm on the FPGA side adjusts its sampling rate to the frequency of the power system, so we have a fixed number of samples per cycle and we can easy calculate the actual frequency.

## 4. Results

Measured time signals and calculated parameters are presented in Fig. 3.



Fig. 3. Front panel of Real Time program, GUI for PQ monitoring

Voltages from three phase public power system are monitored. Time waveforms, and frequency spectrum are presented on graphs. Also, Root Mean Squared RMS, and Total Harmonic Distortion THD, for each channel is calculated.

## 5. Conclusion

On line monitoring is achieved by using FPGA technology. PQ parameters: RMS and THD of voltages or currents, frequency, harmonics, active or reactive power, and energy, are implemented in according to IEC Standard. Calibration is performed by comparing with commercial device ION 7650. Good agreement is obtained. The difference between results of measuring same voltage is less then declared error.

### Acknowledgements

This paper is fully supported by Croatian Science Foundation under the project Metrological infrastructure for smart grid IP-2014-09-8826.

### References

- [1] International standard. Testing and measurement techniques General guide on harmonics and interharmonics measurements and instrumentation, for power supply systems and equipment connected thereto. IEC 61000-4-7, 2002.
- [2] Dong J, Ma X, Djouadi SM, Li H, Kuruganti T. Real-time prediction of power system frequency in FNET: A state space approach. *Smart Grid Communications*, 2013 IEEE International Conference on Smart Grid Communications, 109 114, 2013.
- [3] Esim J, Oleagordia IJ, Loureiro S. Research and Development of a Virtual Instrument for Measurement, Analysis and Monitoring of the Power Quality. *Journal of Fundamentals of Renewable Energy and Applications*, 5(5): 185 191, 2015.
- [4] Kokolanski Z, Srbinovska M, Simevski A, Gavrovski C, Dimcev V. Power Quality Monitoring and Power Measurements by Using Virtual Instrumentation. *Electronics*, 13(1): 70 - 76. 2009.
- [5] N. Kehtarnavaz, N. Kim, Digital Signal Processing System-Level Design Using LabVIEW. Newnes, Burlington, 2005.