ISSN:0975-9646

VLSI Design Approach to Online Analog/Digital DAQ System

Dr.K Babulu[1], P Parasuram[2]

[1,2]Electronics and Communication Engineering Department, JNT University Kakinada
University College of Engineering Kakinada, Kakinada, A.P., India

***Abstract*— An accurate and fast Data Acquisition system is the need of the modern Autonomous Navigation System. A data acquisition system on a low cost FPGA (Field programmable Gate Arrays) was designed, which will do the filtering in the hardware while more complex operation like control and navigation can be performed on the soft core of a processor sitting on the FPGA. This will provide the design flexibility to implement new filtering techniques, which is crucial during the testing and the simulation phase.**

***Keywords*— Analog Interface, Data Acquisition System, FPGA, Real Time Interface, USB 2.0.**

I. INTRODUCTION

Small scale, lightweight Unmanned Aerial Vehicles (UAVs) have very limited payload capacity. Yet the demands for greater mission complexity as well as increased flight performance [2] tend towards increased onboard processing. The major part of this onboard processing is taken up by the Data Acquisition (DAQ) Process which performs the job of reading the data from Inertial Measurement Unit (IMU) and other sensors like temperature, pressure etc. and then filtering the obtained data to remove the measurement noise. Moreover when the performance is critical then the frequency of reading has to be increased.

So if the control, navigation and data acquisition process all compete for CPU we will have to frequent context switches which will lead to low frequency updates of the servos which in turn affect the mission critical performance of the UAV. FPGAs can be advantageous over using commercial off the shelf processing platforms because they are easily expandable and configurable to allow for new features or different approaches to solving a problem.

The flexibility of an FPGA allows for three abstract design philosophies; hardware implementation, software implementation on a soft processor core running on FPGA or a combination of hardware and software implementation. We chose the third strategy for our implementation and will discuss the process and considerations of this approach.

The DAQ system consist of interface with the sensors through a Analog to Digital Converter (ADC) [1] and then applying known filtering techniques like Kalman filter to obtain the correct reading and then it deliver to the control law for further action. Here to carry out implementation we first designed an ADC card which can take 16 multiplexed inputs and provide it to the ADC chip for conversion into digital signal. This digital signal is read by the FPGA and is then converted.

II. PROPOSED DAQ SYSTEM ARCHITECTURE

The proposed system architecture for the developed system is as outlined below,

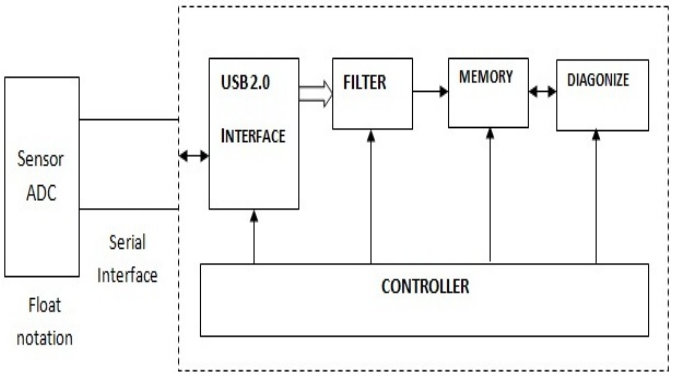

Fig.1 Generic system Architecture for proposed DAQ System

Fig.1 illustrates the developed architecture towards suggested DAQ system. The developed architecture shows the interface of sensor device with the suggested DAQ unit with their internal units. The proposed DAQ architecture consists of a USB 2.0 interface with adaptive filtration logic incorporating sampling, quantizing, and adaptive filtration, to minimize the distortion level based on the scale factor. The obtained filer coefficients are stored into the memory element. The memory input and output operations are controlled via centralized controller unit through host request the device communicated via serial unit protocol using USB interface. To achieve dynamic filtration, multi level filtration logic is being developed.

A. *Design Philosophies*

As we mentioned in the introduction there are three abstract design philosophies that we can follow. We here give a comparative argument for all the three designs principle and argues that hardware-software approach is best for this job in term of performance. A total hardware solution will give the best performance in term of fast response to change in input. But it will consume large resources on FPGAs if it is not optimized. Thus the design process with full hardware implementation will be slow. Moreover it will not be easy to implement change in algorithm, which is usually the case in a test bench, as we have to test different algorithms for their performance. The second approach is that of the program running on a soft-core processor like NIOS II on FPGAs. This will provide the much needed design flexibility but we will lose in term of performance. Moreover the soft core will consume large resources and will also not provide the design environment of a PC which makes the design process slow.

The more important drawback is however that such system perform many tasks in parallel, like at any given time we need to perform four tasks continuously i.e. Data

Acquisition from sensors, Control loop, Navigation loop and command interface to servos and engine. Leaving the last thread all the other threads are compute intensive. Thus if all the three threads run on the same core they will often get context-switched out.

Thus it may happen that the navigation loop operates on old data, which is not important for taking future navigation decision. So the other choice is to use multi soft cores on the FPGAs. Thus each thread can run on one core and there will be no need to contend with other thread for CPU. They can easily exchange data without large latencies as they are on the same FPGA. But if the gates on FPGAs are not enough to support multi core on single FPGA then we can use array of FPGAs that can communicate through serial port or other I/O ports. In [7] the author purposes a technique in which the input values are directly transferred to the registers of the soft-core processor. Thus CPU doesn't have to waste its cycle on waiting for data or reading it. We further extend this technique and implement the entire filtering outside the CPU in hardware. Thus the filtered data is directly provided to control thread. This has advantage over the above design that since it is implanted in directly by the hardware logic on FPGA, it is fast and thus input data can be collected at higher frequency. Moreover the number of gates required is not high and hence it can be easily extended. Secondly the DAQ system and control system can now work in parallel. The problem is that now we will have to interface the though thread so they can transfer the data.

B. *USB 2.0 Interface*

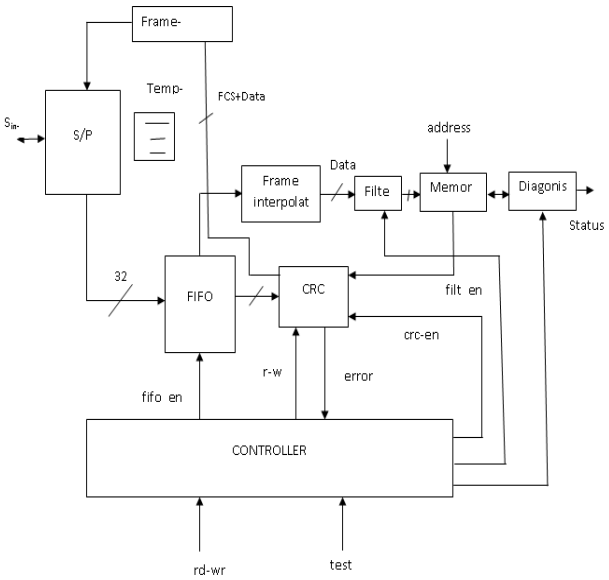

Fig. 2 Architecture for USB 2.0 interface Unit

For the communication of serial data USB controller is being developed. The USB 2.0 interface unit architecture is shown in Fig 2. The communication with the host is interfaced via a serial input line converted to parallel lines for processing internally the operation is carried out using a serial to parallel converter. The parallel data is buffered in a FIFO logic and interpolated using frame interpolator. The error checking bits is performed using CRC16 operation. On no error condition the interpolated data is passed to the filter logic to storage.

C. Filter Design

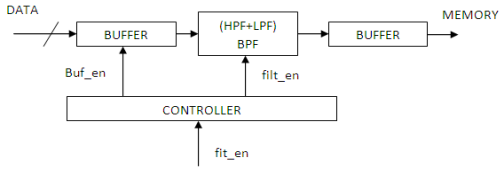

Fig 3(a)

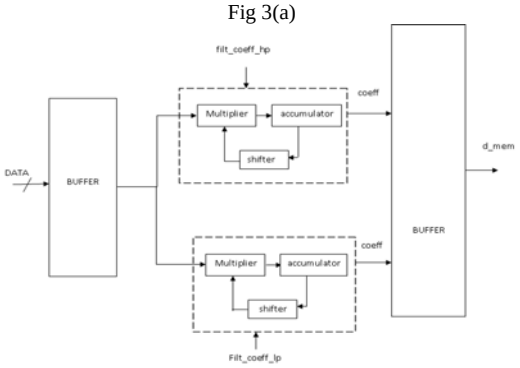

Fig 3(b)

Fig. 3 (a) & (b) Architectural Views for Filter Design

Coefficients obtained for LPF and HPF are of length 4 using db2 signal and scaling functions derived from MATLAB tool is as follows:

```
constant lpcof0: real_single:=('1',"0100","00001001000");
constant lpcof1: real_single:=('0',"0100","11001010111");
constant lpcof2:real_single:=('0',"0110","10101100010");
constant lpcof3:real_single:= ('0',"0101","11101110100");
constant hpcof0: real_single:= ('1',"0101","11101110100");
constant hpcof1:real_single:=('0',"0110","10101100010");
constant hpcof2:real_single:=('1',"0100","11001010111");
constant hpcof3:real_single:=('1',"0100","00001001000");
type ram_b is array( 0 to 3) of real_single;
constant lpf:ram_b:=(lpcof0,lpcof1,lpcof2,lpcof3);
constant hpf: ram_b:=(hpcof0,hpcof1,lpcof2,lpcof3);
```

D. *Recursive MAC Operation*

The filter logics are realized using MAC (Multiply And Accumulate) operation where a recursive addition, shifting and multiplication operations are performed to evaluate the output coefficients. The recursive operation logic is shown in the following Figure 4.7. Before passing the data to filter bank the FIFO logic realized stores the data in asynchronous mode of operation, operating on the control signals generated by the controller unit. On a read signal the off-centred data is passed to the buffer logic.

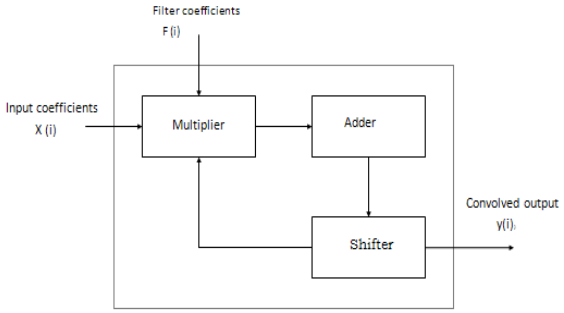

Fig. 4 Realization of Recursive MAC operation

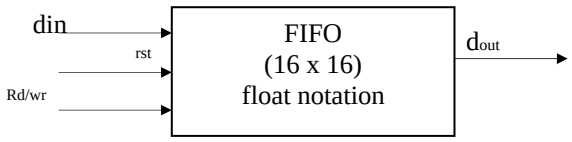

Fig. 5 Realization of FIFO logic for Coefficient Interface

The FIFO logic realized as shown in Fig5. The obtained detail coefficients are down sampled by a factor of two to reduce the number of computation intern resulting in faster operation. To realize the decimator operation comparator logic with a feedback memory element is designed as shown in Fig 6.

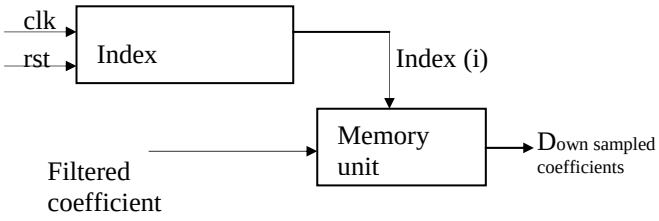

Fig. 6 Architecture for Decimation by 2 logic

E. Sensor ADC Design

In generally, they are usually low rate sensors that need a lot of data processing. A rather simple way to give information at a high frequency, 100-500Hz is to use inertial sensors mounted on or inside the vehicle frame. Normally these sensors are placed in one Inertial Measuring Unit (IMU). The IMU provide readouts about acceleration and rotation rates using accelerometers and rate gyros as sensors. The sensors were interfaced with a 12-bit ADC card with a sixteen input multiplexed channel. The multiplexer provide the selected channel to the input of the ADC chip. The ADC chip we used operates in standalone mode with a low pulse.

A state machine shown in Fig 7 controls and read the value from ADC. The control signal which triggers conversion is kept high until we want to start the conversion. Then it is made low for a period of minimum 250 ns i.e. at least 1 clock cycle of 4 MHz clock or less. The status line from ADC becomes high which is then polled by the FPGA. After approximately 35 cycles the status line becomes low and the digital data can be read from the output pins of the ADC.

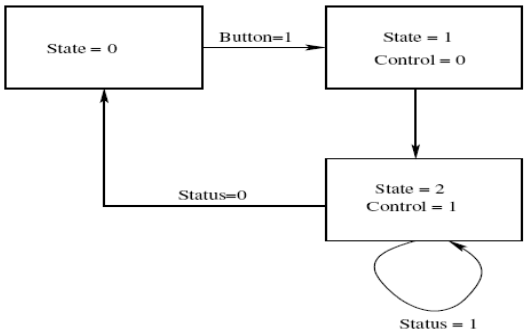

Fig. 7 State diagram for ADC

One of the major problem we faced is that the ADC chip we used doesn't have built in sample and hold amplifier (SHA). This implies that input data has to be kept constant till the conversion in ADC is taking place. If this is not the case then we will not get the correct readings. Since the maximum conversion time of ADC is 35cycle. But in the absence of sample and hold amplifier we can't records for frequency greater than 1.5Hz.

III. SIMULATION RESULTS

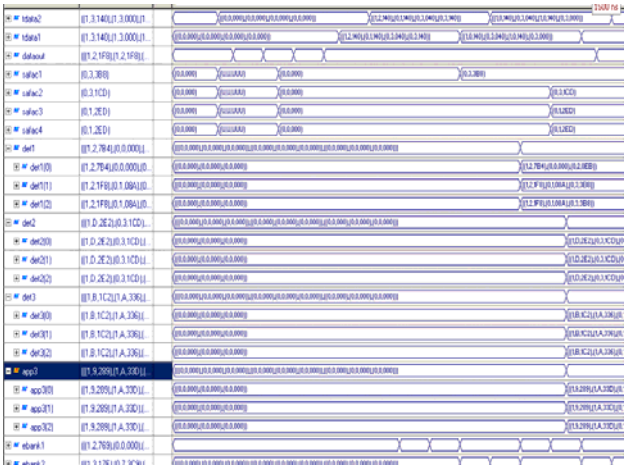

Fig. 8 Functional Simulation result for designed DAQ System

The above Fig 8 gives the functional simulation results for designed DAQ System using Aldec's simulator, which shows the detail and approximate coefficient obtained after filtration. The Fig 8 also gives the functional results obtained after the convolution operation carried out for the implemented filter design, which in turn, shows the timing simulation results for the acquisition system developed.

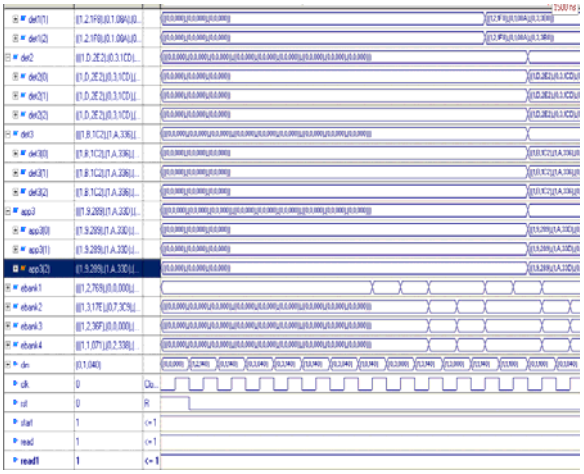

Fig. 9 Functional Simulation result for designed DAQ System with Control Signal

The coefficients are generated from the digitized samples passed from the test-bench interface where the stimulus is taken from the MATLAB generated binary coefficients of the sensor side. The coefficients are compared with resolution coefficients obtained from the MATLAB results and are almost found equal with 0.01 variations, resulting in high accuracy in computation taking about 15 cycles of system clock for performing the operation.

This time is comparatively 85-90 % less as compared to the time taken for performing filtration operation in MATLAB simulation. The test vectors are passed through test bench for simulation as illustrated in Fig 9, which also results timing simulation for DAQ Operation with necessary control signal.

A. *FPGA Realization*

The designed system is targeted onto Xilinx xc2vpx70-7-ff1704 FPGA device belonging to virtex2p family with a speed grade of –7. The logical routing can be observed from the obtained Place and route result from the FPGA Editor option in Xilinx synthesizer. It is observed that about 40% area for the targeted FPGA is covered for the implementation of DAQ System. The CLB's are connected in cascade manner to obtain the functionality for the designed system.

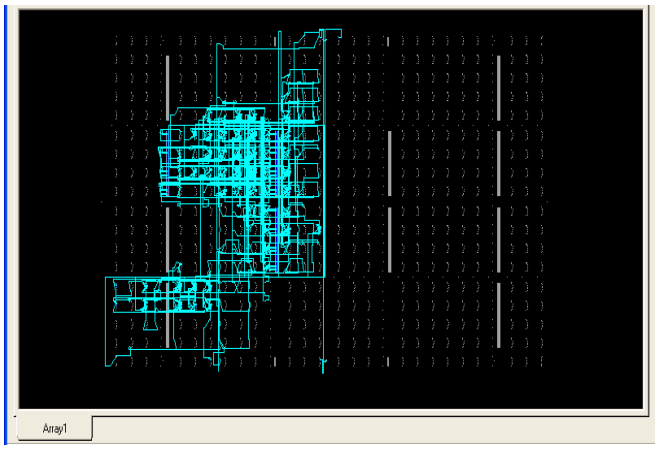

Fig. 10 Routing of Logical Placement in targeted FPGA

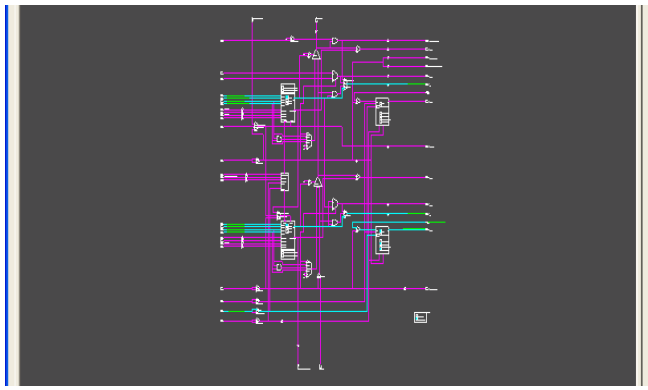

Fig. 11 Logical Utilization of CLB in targeted FPGA

The synthesis result for the designed DAQ System is presented

Macro Statistics

# Registers	: 49
# Multiplexers	: 25
# Tristates	: 74
# Adders/Subtractors	: 618
# Multipliers	: 29
# Comparators	: 128

Design Statistics

# IOs	: 26

Cell Usage :

# BELS	: 181
Minimum period	:5.220ns (Maximum Frequency: 191.571MHz)

From the result it is observed that logical counts of 181 Basic Element Logic (BEL) is required for the realization of DST processor. The real time Maximum operating frequency obtained is 191.571 MHz and this operation frequency is considerably higher than the current sample frequency and makes it more suitable for real time current analysis.

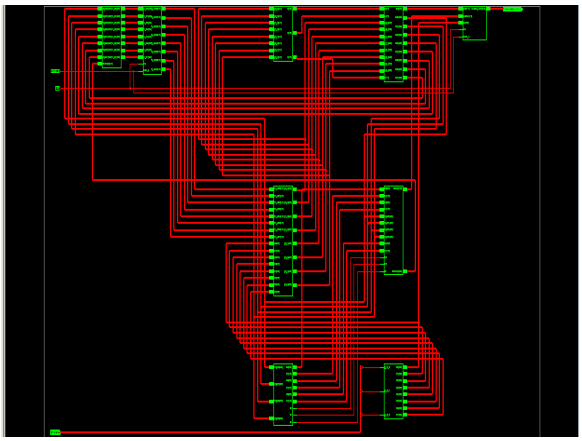

Fig. 12 RTL View for the Designed DAQ System.

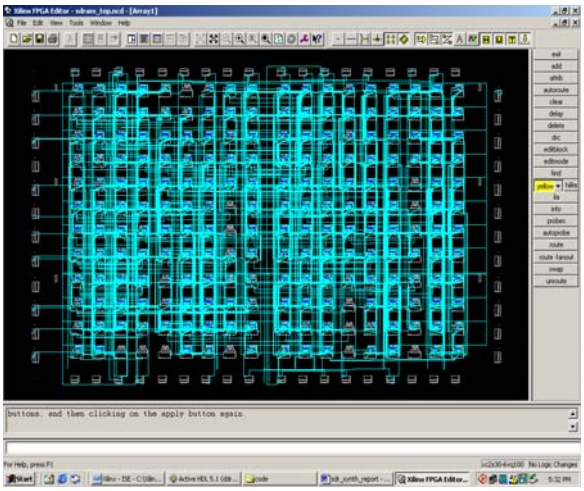

Fig. 13 FPGA Placement of the Targeted Logic onto Virtex2p device

B. *Implementation Observations*

The implementation of deigned DAQ System is illustrated in various obtained figures during the process of realization i.e., from Fig.10 to Fig.13. Fig.10 shows the routing of logical placement in targeted FPGA, Fig.11 gives the logical utilization in each Configurable Logical Block (CLB) in the implemented FPGA. The Register Transfer Logic (RTL) implementation for the designed system is shown in Fig.12 and Fig.13 presents the placement of the targeted logic onto Virtex2p device.

IV. CONCLUSION

An analog data acquisition system used for the real time acquisition of the industrial data is developed. The HDL designing and the practical realization of the developed system is found targeting on Xilinx vertex 2p device. The observation of the data acquisition over analog data set is made. The timing simulations were observed and the synthesis using Xilinx tool was performed and the relevant reports were presented. The real time interfacing is obtained to be 291.37MHz.

REFERENCES

[1] Robert L. Trapp, “Automated Intrapulse RF Data Acquisition,” Southeastcon '81, conference Proceedings, pp 508 -512.

[2] T.Ojo, and M.C. Sterling, Jr. et al, “Multi-parameter Instrument Array and Control System (MPIACS): A Software Interface Implementation of Real-time Data Acquisition and Visualization for Environmental Monitoring”.

[3] C. M. Laymon, R. S. Miyaoka, B. K. Park, and T.K. Lewellen, “Simplified FPGA-based Data Acquisition System for PET”, IEEE Proceedings, 2003, pp 704-707.

[4] G. Verni et al. “Ausili e ortesi in medicina”, Vol. 6, No.3, 1998, pp 101-133.

[5] Youichi Igarashi, et al, "A Common Data Acquisition System for High Intensity Beam Experiments” IEEE Proceedings, 2004, pp 1122-1126.

[6] The Joint Project Team of JAERI and KEK, “The Joint Project for High-Intensity Proton Accelerators”, KEK Report 99-4, July 1999.

[7] G. Boström, M. Fiocco, D. Puig, A. Rossini, J. G.M. Gonçalves, V. Sequeira, "Acquisition, Modelling and Rendering of Very Large Urban Environments", Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission , IEEE Proceedings, 2004.

[8] W. Ribarsky and H. Rushmeier, “3D reconstruction and visualization of large scale environments, “Special Issue of IEEEComputer Graphics & Applications, November/December, 2003.

[9] V. Sequeira and J.G.M. Gonçalves, “3D Reality Modelling: Photo-Realistic 3D Models of Real World Scenes”, In Proceedings, First International Symposium on 3D Data Processing Visualization and Transmission, IEEE CS Press, 2002, pp 776-783.

[10] Huijing Zhao and Ryosuke Shibasaki, “A Vehicle- Borne Urban 3-D Acquisition System Using Single-Row Laser Range Scanners”, IEEE Transactions on systems, man, and cybernetics, August 2003.

[11] Adam Lewis, A. Abdelgawad, and N.-F. Tzeng “Wireless Sensor Data Acquisition Design for Engineering Applications”, Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504, U.S.A.

[12] Vlad Popescu, Iuliu Szekely Transilvania “Wireless Data Acquisition System Using Bluetooth”, Transilvania University of Brasov, Department of Electronics and Computers Bdul Eroilor 29, 50036 BRASOV – România.

[13] Haron, N.Z.,1, Aziz, A.S.A., Ibrahim, M.M., and Salim, S.I.M,“8-Channel Logic Analyzer Controller Design Using FPGA” Journal Advanced Manufacturing Technology, Vol. No: 1. no.1, 2007, pp 87-96.

[14] Costin Cepisca “Remote Data Acquisition System for Hydro Power Plants” Proceedings of the 6th WSEAS International Conference on Power Systems, Lisbon, Portugal, September 22-24, 2006, pp 59-64.

[15] Daniel Marr, “ECG Application Featuring Data Transmission by Bluetooth”, PhD Thesis, University of Queensland, Australia, October, 2001.

[16] Ziqiang Cuil et al, “A Specific Data Acquisition Scheme for Electrical Tomography”, IEEE International Instrumentation and Measurement Technology Conference, Canada, May 12-15, 2008.

[17] W.Q. Yang, T.A. York, “New ac-based capacitance tomography system”, IEEE Proceedings, Vol. 146, no. 1,PP 47-53. January 1999.

[18] Mi Wang, Yixin Ma, Nigel Holliday, “A High-Performance EIT System”, IEEE Sensor Journal, Vol. 5, no. 2, April 2005, pp 289-299.