RTU Hardware Design of SCADA Systems Using FPGA

Hamid Reza Naji and Soroush Shirali

Abstract—The Remote Terminal Units (RTUs) are crucial elements in SCADA systems which are responsible for data acquisition in substations, and accomplishment of central station commands. In this work we present a new hardware design for an RTU that performs basic functions of the RTU. This design is made by FPGA and offers high speed of operation and ample variety of configurations and high reliability.

Keywords—RTU, SCADA, FPGA

1 Introduction

In automation of power systems having real time characteristic and high reliability are required [1,2,6]. So RTUs as significant elements of SCADA systems in power applications should be improved in real time characteristics and reliability[3,4,5].

In this paper we tried to embed all digital aspects of the RTU in one FPGA and produce a RTU as an embedded system. Our approach to design such a system is to avoid using of any processor, and having redound having an RTU with parallel hardware structure which can executes all tasks with the same priority at the same time. It is obvious that such parallel structure will improve RTU real time characteristics as we call it non delay RTU. Non delay RTU will be suitable for applying in closed loop control systems.

The capability of having parallel modules on FPGA to make functionality of RTU instead of using several separate devices offers reliability advantage, because communication between devices may be affected by noise of environment which has high amplitude in power systems.
In some cases because of shortage in I/O ports, two or even more RTUs have to be used to communicate with SCADA from one station [7,8,9]. Therefore reconfigurability of FPGA based RTU offers this capability to increase I/O ports whenever it's necessary.

2. Design of System

RTUs are the brains of SCADA systems [10,11]. They are responsible for acquisition data in substations and sending them to SCADA also they execute commands of SCADA system [12,13]. Figure1 shows the block diagram of one RTU which we have synthesized on FPGA. This FPGA based RTU with four principle modules performs basic functionality of RTU and communicates with external modem via RS232 port. AI(Analog Input) and DI(Digital Input) modules monitor analog and digital quantities in substation and report changes in these values and deliver new values to CI(communication interface) module simultaneously. CI module incorporates new values with the date and time of variation as a provide data packet and will save it in its memory. When SCADA requests data from RTU, CI module sends these packets serially via RS232 port and through modem to Station. Receiving commands via RS232 port and decoding them is the other crucial duty of CI module.

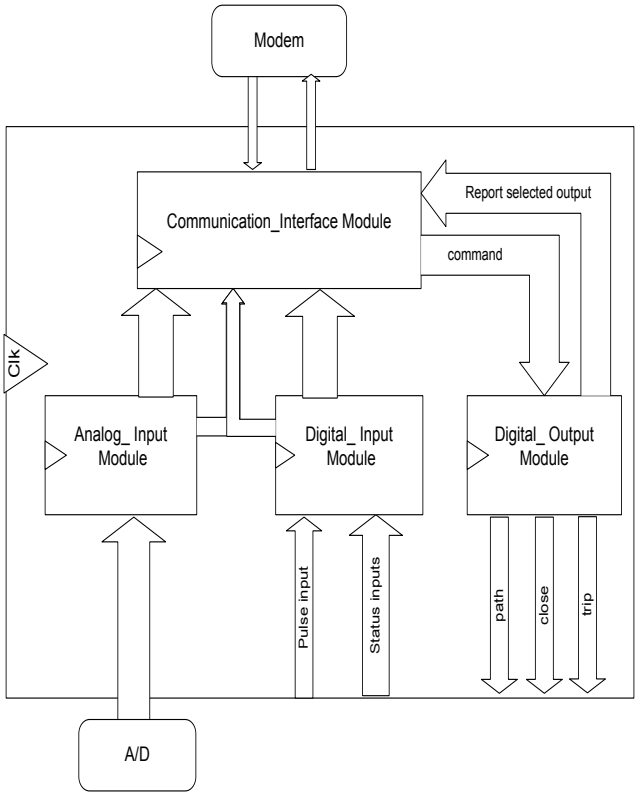

Figure 1. block diagram of RTU

- *Dr. Hamid Reza Naji is with the College of Electricl and Computer Engineering, Kerman Graduate University of Technology , Kerman, Iran.*
- *Soroush Shirali is with Dept. of Electricl and Computer Engineering, Shahid Beheshti University, Tehran, Iran.*

For instance when CI module recognizes received command as exerting commands to digital status outputs of RTU , CI module delivers decoded command to DO module and DO exert it to its outputs. It must be notified that all defined modules work simultaneously without any interruption and as will see in description of each module's architecture, processes in modules execute simultaneously as well.

3. Analog Input Module

Analog quantities must first be converted into suitable range by appropriate transducers and then be converted to binary values by ADCs. AI module always monitors these binary values on FPGA inputs and when there is any change in these values, AI sends a pulse to communication interface module and also delivers new values through the bus to this module.

The architecture of AI module is simple and effective. It consists of a register and a comparator. The combinational comparator always compares register's inputs with its outputs and if there is any difference it makes a pulse by a flip flop at its output. Thus right at the moment that register loads new values on its outputs, a pulse will be generated to inform the CI module to pick up the output of register and AI module as new value.

As discussed, AI module monitors inputs at each clock pulse rising edge and there is not any interruption in this. Therefore if we choose FPGA's clock as 100MHZ, analog quantities would be monitored each 10 ns.

4. Digital Input Module

Input signals to this module are divided into two categories, status signals and pulse signals. Status signals stand for state quantities which describe status of relays and other devices. Pulse signal indicates a quantity with its frequency. For example a power meter in a substation may modulate power consumption value on its output pulse. All signals will enter to FPGA modules without any peripheral circuit.

Figure2 shows structure of DI module. As shown in figure2 there is the same register and comparator components that had been described in section 3, to monitor values in this module. Status signals connected to register and comparator component directly while pulse signal first enters to a component nominated as Pulse-counter. Pulse-counter counts pulses in specific time period and infer a binary value, and delivers this value on its output to register and comparator input.

In both DI and AI modules, we can generate adequate inputs. Also it is possible to reconfigure FPGA in order to increase inputs of DI and AI, or synthesis more than one of these modules whenever needed.

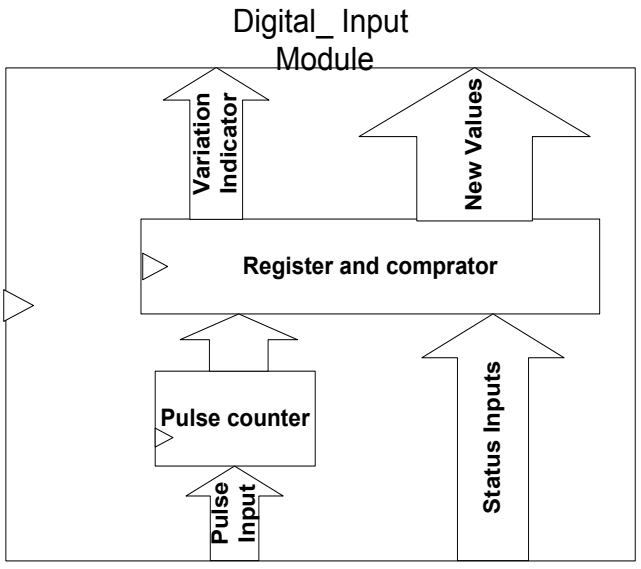

Figure 2. Architecture of Digital Input module

5. Communication Interface Module

As shown in figure 3 this module is composed of parallel components which work simultaneously. These components are serial receiver, decoder, Inner-timer, DTL (Data and Time logger) and coding and transmitter component. We will describe the structure of each component in the following sections.

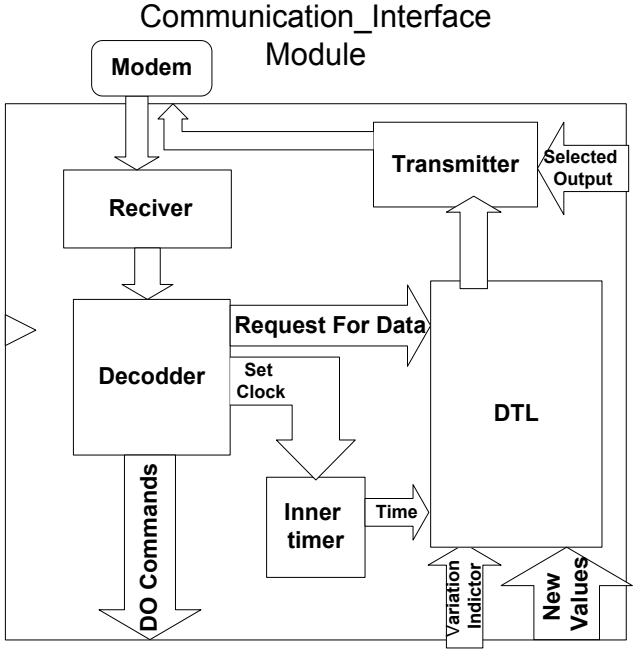

Figure 3. Architecture of Communication Interface Module

5.1 Serial receiver component

This component could be nominated as physical layer interface between modem and RTU. it receives data bits serially from modem and delivers data bytes to decoder component. Its architecture is similar to UART core in microcontrollers.

5.2 Decoder component

Usual protocol in SCADA systems is DNP3(Distributed Network Protocol Version3) so in this work we designed a decoder for RTU to decode DNP3 and recognize command types.

Station sends to RTU three types of command in DNP3 format. These types are: setting clock , exert output commands and request for registered data.

As it is shown in figure 3, decoder gets data bytes from serial receiver component and when it recognizes type of received command then it sends a pulse to relative component to inform it and simultaneously delivers appended information on private bus to that component. For instance whenever a command arrives from station, decoder sends a pulse to inner-timer component and synchronizes itself with rising edge of pulse and it delivers equivalent binary value of accurate data and date which is appended to the command.

Architecture of decoder has shown in figure 4. After receiving 2 Sync bytes of frame header bytes, decoding process would begin. At first eight bytes of header will be loaded and surveyed by header analyzer unit. These 8 bytes include bytes which indicate length of frame, destination address and source address.

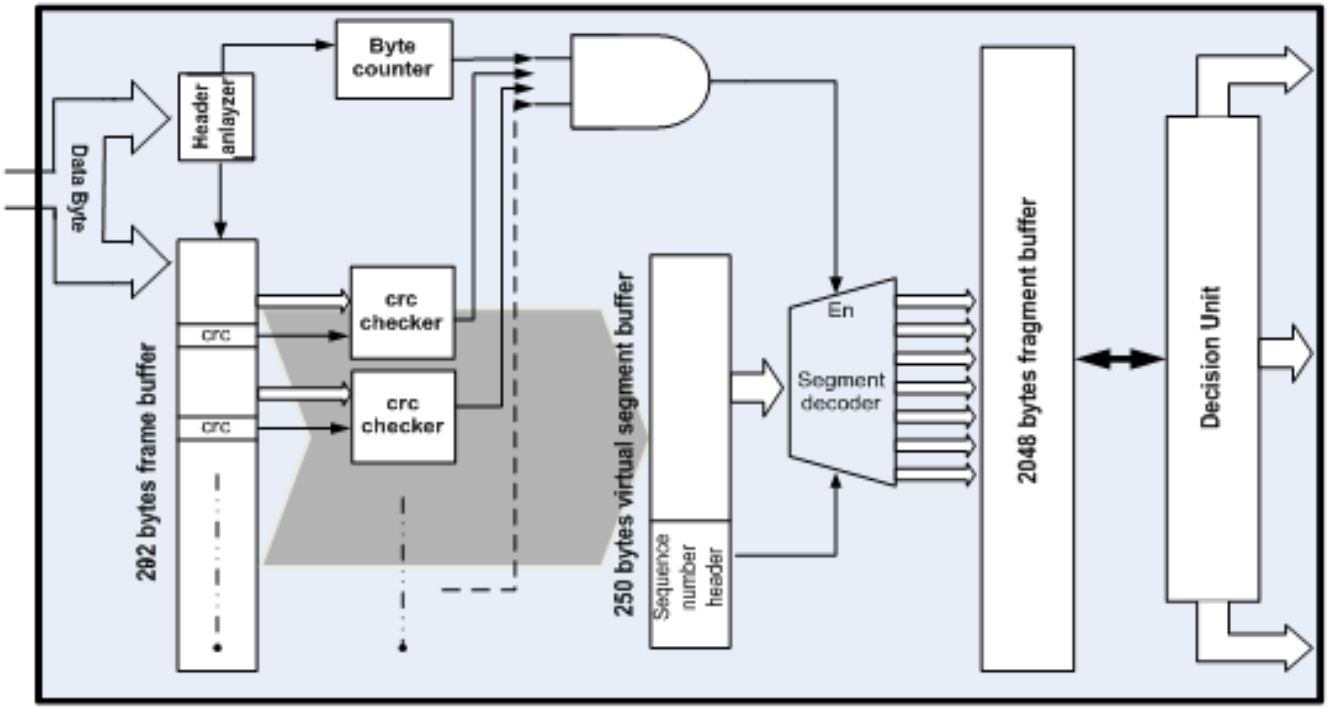

Figure 4. Architecture of decoder

If destination and source addresses were correct, then the same number of bytes as frame length shows would shift into a 282 byte shift register.

Comparing to the length of frame, when byte counter unit detects arrival of the last byte, if there was no problem in CRCs, 249 bytes of segment in respect to its header (250^{th} byte) in transport protocol layer will transfer to a fragment buffer with 2048 bytes size via demultiplexer. As it is described before, at the same time with receiving last byte of each frame, all stages of decoding operation from data link layer to application layer will proceed.

Header byte of each segment in transport layer has a bit as FIN bit which determines if that segment is the first one in fragment sequences or not. This bit will be checked in the decoder in order to specify that 2048 bytes of a fragment has accomplished and then fragment will be analyzed using database and proper decisions will take place. As we said this decoder has designed in such way that in the same time with receiving data, data will be decoded and command with its appended information will be delivered to other components without delay. In addition because separate hardware is assigned to decoding task, it will not interrupt other tasks in RTU.

5.3 Inner-Timer component

This component always delivers accurate time and date on its outputs to DTL. Architecture of this component depicted in figure5. In its architecture an accurate frequency divider is used to produce clock pulse with real second intervals from RTU clock pulse. Another aspect of this component is, it is composed of several counters which count second pulses and hold time and date.

As saw in section 5.2 setting time and date settings for

this component will be done by station commands. Inner-timer component loads all binary data and date values into its outputs at a single moment in parallel. So there is no interruption in its ordinary operation, hence there will be no tribulation in its time accuracy. Hardware design of RTU makes it possible to have buses large enough to deliver parallel data between components and modules. Obviously this property speeds up the RTU operation.

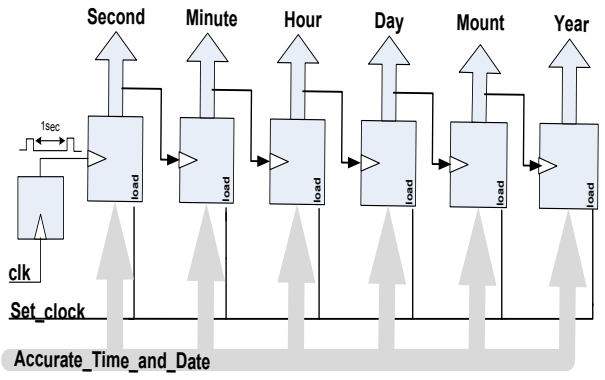

Figure 5. Architecture of Inner-Timer Module

5.4 Data and Date Logger component

As explained in section 2, AI and DI modules report their input variations to CI module. These varied data with their time and date of occurrence are saved in data packets. Actually the Data and Time Logger (DTL) component is responsible for this data saving. This component permanently gets the accurate date and time from inner-timer. When the AI and DI modules deliver their data on a rise-time of a pulse which indicates the variation, DTL component latches these data with accurate time and date and saves it in a packet form.

The backbone of DTL architecture is an 88×2048 RAM which is made by FPGA RAM.

When the request of receiving data is sent to RTU, decoder component informs the DTL by activating a digital pin. Then in order to deliver it to coding and transmitting component DTL starts to read packets from RAM in suitable time intervals. Time interval attains by a timer and it is correspond to a period of coding and transmitting a packet via coding and transmitter component. After all packets have been sent to station, by activating a digital pin, DTL notifies coding and transmitter component to declare completion of operation to station by proper message.

5.5 coding and transmitter component

As demonstrated in figure3 coding and transmitter component (CT) has two packets of inputs from DTL component and DO module. Meanwhile appealing data from station, if DTL component deliver a new packet, CT component would code and send this packet in DNP3 format, otherwise if DTL notify completion of packets, CT would send a proper message to declare accomplishment. Also when DO module reports a feedback to CT, it would send decoded corresponding message.

Coding function in CT is performed inversely of whatever explained in section 5.2 about decoding. After coding messages in CT is accomplished, it would shift frames serially on Tx line to modem and adding stop and start bits between each 8 bits.

We consider that all CI module components and their parts work in parallel and independently. In Figure 6 a part of RTU simulation is shown as an instance. In this figure RXD, SDATA and Path are serial receiver pins, serial transmitter pin and a bus for selecting path. As we see in Figure6 , after RTU receives command to select a path , it selects the path immediately. Then while RTU receives other commands from station it begins to send the report of selecting that path simultaneously.

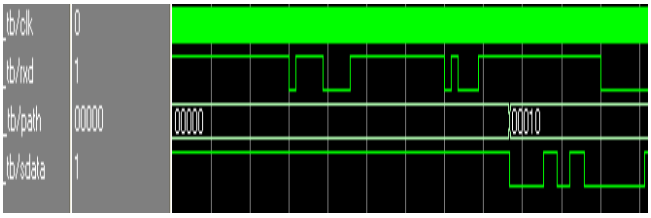

Figure 6. Part of Simulation Result

6. DIGITAL OUTPUT MODULE

As explained before, RTU receives commands from station to turn instruments on or off in substation. RTU implements these commands with Select Before Operate (SBO) method. SBO provides more security. In this method, at first station sends a command to RTU to select a path of a specific instrument. RTU selects that path then reports to station that path has selected done successfully. Station verifies RTU report and if it is a right path, station would sends to RTU an order to on or off the instrument on that path.

DO module of RTU is responsible to execute station commands in SBO method. In section 5.2 we described that Decoder delivers Decoded command in rising edge of a pulse to notify arrival of output related to DO module. In figure7 architecture of DO module is shown. As we can see in figure 6 a combinational circuit as a decoder exerts decoded commands to DO outputs. DO has three categories of outputs. These outputs are path, trip and close which are used to select path, off and on instruments on a path.

Similar to the structure we have seen in section 3 (Register and Comparator) exists in this module. Here inputs to Register and Comparator are conditions of relays which are bonded to output path. In this way Register and Comparator would report selecting a new path directly to CT component in CI Module to be transformed to a proper message and transmitted to station immediately.

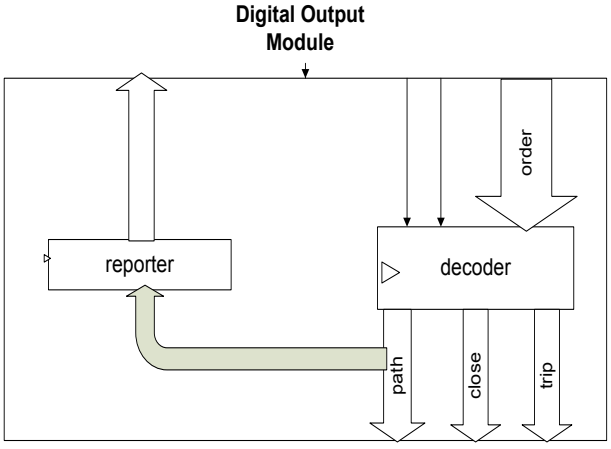

Figure 7. Architecture of Digital Output Module

At last we mention again the simple architecture of this module makes it possible and easy to reconfigure FPGA in order to increase RTU outputs whenever needed.

7. IMPLEMENTATION ON FPGA AND RESULTS

The hardware validation performed using Xilinx's Spartan III FPGA with 200000 logic gate. Implementation of RTU consumed %51 of the FPGA consists of 997 slices, 1633 LUTs and 11 blocks of RAM. Statistics shows ample possibility to reconfigure and develop established RTU in number of inputs and outputs when it is needed.

8. CONCLUSIONS

We obtained a hardware design for a basic RTU that all aspects work parallel and don't interrupt each other. This RTU can monitor its inputs with highest possible rate and be a reliable and accurate data acquisitive. On the other hand such a structure in RTU causes the processes based on orders received from station, executing them and transmitting reports to station all independently and at least possible time. So this RTU will not make any delay in SCADA systems. Also simple structure of this RTU especially in input and output modules enables use of reconfigurability to develop its hardware to satisfy our requirements. It is obvious that this RTU as an embedded system is more reliable than RTUs composed of several microprocessors.

9. REFRENCES

[1] H. Lee Smith and Wayne R. Block, "RTUs Slave for Supervisory Systems," In IEEE lSSN 089.50156/93/$3.000 January 1993 .

[2] A. Ramirez and M. Torre, "Distributed Network Protocol (DNP)". II Jornadas de Sistemas de Instrumentaci6n y Control, Caracas, Venezuela, May, 1994.

[3] Ren Yanming, Qin Lijun , and Yang Qixun , "A New RTU Based on LonWorks Technique Used in The Integrated Automation Substation System," In IEEE 0-7ao3-47~-4/9aitio.00 o 1998

[4] Jun Ge, Luyuan Tong,Juncheng Geng,Quanshi Chen,Guang Han, and Zhi Tang," Unmanned Substations Employ Multimedia Network RTUs," In ISSN 0895-0156/02/$17.00©2002 IEEE.

[5] Lu Guang, Zhang Bomiing, Sun Hangbiin, "The Embedded Realtime LINUX and ILs Application in the Automation System of the Power Nenvork," Automation of Elcclric Power System, 2002, 10, (7). pp. 62-69.

[6] Lu Guang, Zhang Bomiing, Sun Hangbiin, "The Embedded Realtime LINUX and ILs Application in the Automation System of the Power Nenvork," Automation of Elcclric Power System, 2002, 10,*(7).* pp. 62-69.

[7] L.S. Jeong, P.J. Kim, D.S. Ku, S.D. Kim, S.D. Yun, and J.B. Kim, " Implementation of DNP RTU in the Electric Power SCADA System SICE Annual Conference in Fukui, August 4-6,2003, Fukui University, Japan.

[8] Yanbin Qu, Jianyong Su, and Liguo Feng ," Design and Implementation of RTU based on the Embedded Operation System uC/OS-I1," In 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies (DRF'T2004) April 2004 Hong Kong.

[9] Srisuwan, K.; Sangmalee, V.; Thunyaphirak, V.; Skunpong, **A, Remote Terminal Air-conditioner Unit for Power Management,** SICE-ICASE, 2006. International Joint Conference, 18-21 Oct. 2006 Page(s):2752 - 2755 .

[10] M. J. Madera, and E. A. Cafiizales, "The GPRS Communication Platform and DNP Protocol as the Best Choices to Communicate the SCADA with IEDs in the EDC Distribution Network",1-4244-0288-3/06/$20.00 ©2006 IEEE.

[11] Edward Chikuni, and Maxwell Dondo,"Investigating the Security of Electrical Power Systems SCADA", 1-4244-0987-X/07/$25.00 ©2007 IEEE.

[12] PENG Dao-gang, ZHANG Hao, YANG Li, and LI Hui," Design and Realization of Modbus Protocol Based on Embedded Linux System,"In The 2008 International Conference on Embedded Software and Systems Symposia (ICESS2008).

[13] Daogang Peng; Hao Zhang; Kai Zhang; Hui Li; Fei Xia**, Research and Development of the Remote I/O Data Acquisition System Based on Embedded ARM Platform,** Electronic Computer Technology, 2009 International Conference on, 20-22 Feb. 2009 Page(s):341 – 344.

Dr. HamidReza Naji is the Dean of College of Electrical and Computer Engineering in Graduate University of Technology, Kerman, and IT consultant of the International Center for Science and High Technology & Environmental Sciences, Iran. His research interests include embedded, reconfigurable, and multi-agent systems, networks, and security. Dr. Naji has a PhD in computer engineering from the University of Alabama in Huntsville, USA. He is a professional member of the IEEE. Contact him at hamidnaji@ieee.org

Soroush Shirali has a master degree from the department of electrical and computer engineering, Shahid Beheshti University, Tehran, Iran. He has several years industrial expreinces in the area of electronic and digital systems design and implementation.