Third Prize

Embedded Electric Power Network Monitoring System

Institution: **Jiangsu University**

Participants: **Xu Leijun, Guo Wenbin, and Sun Zhiquan**

Instructor: **Zhao Buhui**

Design Introduction

Electric power is the mainstay of a nation's economy and the lifeline of industrial production and social life. All parameters of an electric grid, especially harmonic parameters, are related to the quality of electricity generated, which guarantees safe operation of electric equipment. Therefore, this design combines a web-based electric grid parameter measurement system with a video-monitoring system to arrive at an integrated monitoring system of embedded electric grid/station. In the measurement of electric grid parameters, we check the voltage, current, harmonics, etc. in the target areas. The video section monitors the key instruments and environment and transfers these results to a web interface in real time. Therefore, no matter where the monitoring personnel are, they can observe the electric grid parameters and inspect both equipment and environment in real time, as long as they are connected to the Internet. Our design applies to electric stations at all levels, enterprises with requirements of a reliable network, as well as applications requiring remote monitoring.

The Altera® system-on-a-programmable-chip (SOPC) solution is a flexible, efficient solution, integrating the function modules necessary to system design such as CPU, memory, and I/O interface into an FPGA. The SOPC design approach makes for a flexible system design allowing you to modify, expand, and upgrade system modules using Altera-supplied software and hardware tools. We chose the Nios® II soft core embedded processor because of its low cost and abundant FPGA logic resources, which can cater to the demands of different applications. In addition, Altera provides a complete solution based on the Nios II processor, which includes the Quartus® II and integrated development environment (IDE) tools, further reducing product development cost.

Function Description

The system design consists of two parts: the measurement and delivery of electric grid parameters and video monitoring**.**

Measurement & Delivery of Electric Grid Parameters

Our measurement circuit uses an application-specific DSP chip to collect, store, and send the following data:

- A measurement of valid values
- The harmonic component of three-phase voltage/current and neutral current
- The voltage imbalance factor

The DSP chip also measures real power, reactive power, power factor, power supply frequency, and power cut times, calibrated against the national standard precision.

During measurement, the three-phase voltage and current circuitry was isolated using voltage and current sensors for data collection and attenuation, and then sent to A/D converter with 128 sample points in every phase. The DSP chip performed FIR filtering, fast Fourier transform (FFT) operation, storage, and display of sample data. The communication between the circuit being measured and the Nios II processor was conducted through a serial port, based on the Modbus protocol.

Remote measurement of electric grid parameters can be divided into two parts: measurement of real-time electric grid data and monitoring power cut data. Measuring the real-time electric grid data is handled through a web interface. When the user clicks the **Measure Now** button in a web browser, a measurement command is sent to the system based on the Nios II processor, which in turn instructs the DSP chip to start real-time measurement. Then, the DSP chip returns the measurement results back to the web browser for users to verify. For example, the application software displays history of the most recent 25 blackouts.

The measurement functions of the electric grid parameters are realized in a dynamic web page display, using the Boa web server under uClinux and common gateway interface (CGI) technology. When the user sends a measure/check data command through the web browser, CGI scripts compiled with C language send the user command to the web server (Boa), which interprets and executes the CGI script and sends the results back to the browser.

Video Monitoring

Video monitoring is needed to guarantee safe, normal operation of important measuring instruments in real time. To implement video monitoring, our system has two critical components: video data collection and video data transmission.

Video data collection is realized using both hardware and software. The hardware part comprises a web camera and USB adaptor. Today, web cameras are a popular low-cost network video capture devices that are convenient for use in embedded applications. Because there is no support for USB devices on the Cyclone™ EP1C20 board included in the Altera Nios II development kit, we designed a USB interface to read video data from a web camera. We developed the software modules by modifying open source software like vgrabbj and xawtv, whose main function is to capture video from a web camera and convert it to the JPEG format. Next, this video data is sent to the network in real time using CGI.

We implemented video transmission using CGI and the web server. Working with the available electric grid parameters, the application software combines video information with electric grid information in an easy format that remote users can view through a web browser.

Performance Parameters

The performance is based on electric grid parameters and video image parameters.

- *Electric Grid Parameters*—The measurement accuracy of the electric frequency is 0.01 Hz. For the three-phase voltage and current measurement, the accuracy is 0.5%. For the three-phase voltage imbalance factor, the accuracy is 0.2%. For the three-phase current imbalance factor, the accuracy is 1%. The harmonic measurement accuracy meets the GB/T 14549-93 B standard. The shortest measurement interval is 3 seconds.

- *Image Parameters*—The screen-capture function adopts a 320[x]240 JPEG format with a 1-Hz refresh frequency. This performance meets the requirement of normal video monitoring.

Design Methodology

Our design comprises the following parts: software/hardware modules for measuring electric-grid parameters, uClinux OS kernel and file system configuration, web server configuration, CGI program development, USB interface board design, and development of an image capture program.

Hardware Design

The hardware circuitry is based on Texas Instrument's TMS320LF2407A DSP chip as the computing engine. The TMS320LF2407A features a 40-MHz clock and a single-cycle instruction execution time of 25 ns. Because the DSP chip operates at 3.3 V, and because all other chips in the system operate at 5 V, we needed to perform voltage-level conversion. Therefore, we chose the Altera EPM7128S device to implement level conversion and logic control. We also took advantage of the device's compatibility with 3.3-V and 5-V levels for use in communication control and address decoding functions between the DSP chip and other chips in the system.

To measure the three-phase voltage and current, we designed a protective circuit that deploys voltage sensors and current transformers for safe isolation from high voltage and currents. This circuit performs AC attenuation, acts as an anti-aliasing low-pass filter, performs A/D conversion, and passes data to the DSP chip for computation. The results are displayed and then stored for further analysis. At this point, all operations, including synchronous sampling by software, measuring signal frequency before sampling, and calculating sampling frequency based on the signal frequency, must be handled carefully to avoid FFT errors caused by frequency fluctuations.

Software Design

We can code DSP programs in C or Assembly language; C programs are easily readable, changeable, and are good for porting. However, their executable code has low efficiency. In contrast, Assembly language routines yield highly efficient executable code. To improve code efficiency and meet the requirements of a real-time system, we deployed the C2xxAssembly language routines for each software module and interrupt program. For example, we used Assembly language to take advantage of DSP special instructions in the FFT subprogram of the DSP data processing software module: bit-reversed indirect addressing, which is designed for real-time implementation of FFT arithmetic.

Configuration of uClinux OS Kernel & File System

We configured the uClinux OS kernel and file system as described in the following section.

- *uClinux Kernel Configuration*—The embedded uClinux utilizes a customized Linux kernel with high flexibility, and is an open-source code that is stable and reducible. In this design, we adopted the uClinux version 1.3 edition, which was developed by Microtronix for the Nios/Nios II processor. Further, we reconfigured the kernel according to the requirements of the contest, utilizing the least system resources and keeping the function execution in mind.

- *File System Configuration*—We chose the minimal install option in the interface configuration, and then added agetty, boa, dhcpcd, ftpd, inetd, init, ping, route, and telnetd.

- *Modification of Kernel Source Code*—We modified the USB driver source code, USB controller SL811HS driver, web camera OV511 driver, and system files such as Kconfig and Makefile.

Web Server Configuration

The web server used in the project belongs to the uClinux file system. uClinux's own file system includes two web server programs, httpd and Boa. In our design, we chose the Boa web server because of its support for CGI. Boa configuration includes the following steps:

1. Open the **boa.conf** file in the **/target/etc/config** folder in the established file system.

2. Change **ChRoot** to **/mnt/ide0/www** to make the **/www** file on the compact flash card the main folder of the web server. When you type `IP system`, the server automatically analyzes and searches the web page named index.htm in the main directory, which is also the homepage of this design.

3. Add the command **ScriptAlias /mnt/ide0/www/cgi-bin/ /cgi-bin/** in the **ScriptAlias** section to map the forefront folder address with a complete path. This setting saves time for entering an address in the address column and enhances the security of the system. Defaults can be selected for other options.

4. Save the configured files. After downloading the file system, create a **cgi-bin** directory in the **/mnt/ide0/www** file for storing CGI scripts.

CGI Program Design

The CGI program enables monitoring from the client, who can issue measurement commands to the system through the web browser. Upon receiving commands, the CGI program performs the task of parameter monitoring and image supervision of the electric grid. Because the CGI program is written in C and embedded in an HTML page, when it is executed, the system can perform measurement operations on specific ports display the result on a web page for further action.

First, the CGI script called GET receives the analytical QUERY_STRING passed to the web page from web server, which acts as the customer's monitoring command center. After receiving the command, the CGI program decodes it and sends the collected command to the electric grid parameter data collecting module. Next, it receives the returned data and immediately passes it to the web interface, which is browsed by supervisors. It is important to note that after the CGI script is programmed and executed, it should be moved to the relevant directory on the compact flash card. It should also have its suffix changed to **cgi**, and attribute changed to executable for the web server to recognize and execute the CGI program correctly.

USB Interface Board Design

The SL811HS is a widely used USB controller in embedded systems supporting the USB1.1 protocol. Based on this controller, we designed a USB interface board and used the J16 pin on the Cyclone development board as the connecting interface to SL811HS hardware. Simultaneously, we programmed the timing conversion using HDL to conform to Avalon® bus timing and SL811HS timing specifications, and added it to the user-defined logic in the SOPC Builder tool.

Image Capture Program Design

Our image capture program in uClinux mainly uses video4linux application programming interfaces (APIs) provided by the kernel, which can capture images using image codecs. The APIs also support the USB interface and web camera's driver programs. The main functions of the program include an image collection device (web camera) initialization, mapping and capture of the current video frame, converting an image into JPEG format, and saving it to a file. We access this file through CGI, and display it on the supervisor's web page, which denotes the completion of the image capture and display process. Our software program uses several video collection programs, which observe the GPL license and run on Linux. These programs include vgrabbj, xawtv, and webcam server programs.

Design Features

Our design combines both electric grid parameter measurement and video monitoring tasks, which is perhaps a first of its kind in the market. We anticipate many application possibilities for our system with a huge market potential.

Our design is very economical because it uses Altera's FPGA as the core of the whole system. We adopted the Nios II processor, a JTAG-based hardware-debugging module that supported JTAG debugging online, and omitted an external emulator. By using a web camera to perform video capture, we saved a lot of money. Furthermore, our design is flexible enough to accommodate changes based on the customer's actual requirements, and can cater to different applications.

We ported the uClinux real-time operating system (RTOS) to the Nios II processor to take full advantage of RTOS resources and to facilitate faster system development.

Video capture and compression tasks use complicated computing. Thanks to the Nios II processor's powerful data-processing capability, we could accomplish all of the video capture and compression tasks of our system. Also, the Nios II processor has a superior price/performance ratio, and is a good choice for most embedded developers.

We can add peripherals to our system without making any modifications to system hardware. For example, we could use the Cypress SL811HS device as the main USB controller, and can easily interface the SL811HS chip onto the Avalon bus through user-defined peripherals in the SOPC Builder tool. We cannot make these types of modifications using other embedded systems based on the ARM MCU, etc.

By expanding the compact flash card memory as external storage for our embedded system, we can store dynamic web pages and update data at any time. Again, the enhancement and interconnection of the compact flash card module can be made quickly using SOPC Builder.

Conclusion

We learned a lot from this three-month design contest. We gradually developed a through understanding of the Nios II processor, which enhanced our confidence in it. We also gained a lot of knowledge about embedded design techniques. In our opinion, Altera's SOPC design methodology is a highly flexible, efficient solution. Using SOPC Builder, we could customize the Nios II CPU for our design requirements and use system resources efficiently. Compared to other 32-bit processors with the same functions, the Nios II processor offers a very good price/performance ratio. Additionally, the Nios II IDE development environment has a user-friendly interface, facilitating fast system design and program debugging. Furthermore, Altera provided many reference designs, some of which we used with some modifications.

As Altera has improved the Nios II processor, other intellectual (IP) and software companies have perfected their support for it. For example, we used the Microtronix's uClinux RTOS, which is an embedded OS especially tailored for the Nios II processor, in our design. Using the Nios II IDE environment, we could easily make modifications to the RTOS: we modified the USB driver code to support our USB web camera better, developed a CGI program for communication with the web server, and implemented dynamic web page technology. We also edited the programs and grafted them onto GPL in IDE.

Because of the need for real-time arithmetic computation, we used a special DSP device for the electric grid parameter measurement. However, we believe we could have implemented the same measurement circuitry totally in the FPGA if we had used the Stratix® FPGA. Using this device, we could have integrated all of the system functions, and realized the whole system as a single-chip implementation. This implementation is what we intend to do in future.

During debugging of the USB web camera function, there appeared to be a collision (in communication) between the USB controller and the web camera hardware. Although we had tried our best, we could not solve this problem because of time considerations. Although our video capture software ran very well on Red Hat Linux 9.0 and Fedora 4, and after a successful edit in IDE 1.1 and 5.0, we failed to collect images on the EP1C20 board because of this hardware collision. Although the contest has come to an end, we will keep trying to address this problem to develop a more perfect system.