Embedded System Web Server

Uroš Legat
Computer Systems Department, Jožef Stefan Institute
Ljubljana, Slovenia
E-mail: uros.legat@ijs.si

***Abstract*- This paper describes a development of a lightweight web server that runs on limited hardware resources of an embedded system. The web server hosts static web pages, enables HTTP file transfer and supports input/output communication. The embedded system is based on the Microblaze soft-core processor implemented on a FPGA platform. The web server application runs on the uClinux operating system.**

I. INTRODUCTION

Embedded systems are specialized computer systems designed and optimized to perform a particular task. Usually they are a part of a larger system or a machine [1]. In today's world, embedded systems are everywhere, homes, offices, cars, factories, hospitals, plains and consumer electronics. They span all aspects of modern life and examples of their use are numerous. Modern embedded systems are able to connect to the internet and can be remotely maintained and diagnosed [2]. M2M (Machine to machine) communication is growing with a considerable rate. The possibility to connect two or more embedded systems enables developers to build more powerful distributed systems such as networked embedded systems.

Remote maintenance is performed by different communication protocols. The most common communication protocol is HTTP which enables remote system control and monitoring. A web server is a computer program that implements HTTP protocol. It accepts HTTP requests from clients like web browsers and serves HTTP responses which are usually HTML pages with linked objects. There are many web servers available, and a number of them are free, like Apache [3], AOL [4], Roxen [5]. Internet Information Services [6], Sun Java System web Server [7] are some of the most common commercial web servers. Some web servers can run on almost any operating system while others are platform specific.

The general purpose web servers are intended to run on powerful server computers, workstations or personal computers and support a number of advanced features. On the other hand, web servers for embedded systems have limited resources and offer only a set of required features. Requirements of an embedded system web server are:

- small RAM and ROM footprint,
- low CPU consumption,
- easy integration with existing application, including static link with the operating system and application,
- serving pages from the RAM if there is no hard drive.

Embedded system web servers are used in different applications. In our case, we developed a generic embedded system web server for remote monitoring in integrated ICT solutions for aging society. In the following, we describe our approach and the main features of the developed system.

II. IMPLEMENTATION DILEMMAS

From the hardware point of view the embedded system web server must be comprised of an embedded processor, embedded memory and required peripherals like Ethernet controller and input/output interface. In order to assure flexibility for different possible applications the embedded system was implemented on a FPGA platform. Soft-core processors like Xilinx Microblaze, Altera Nios as well as any processor written in HDL languages can be implemented in FPGA devices. This avoids the need of a separate processor chip. The soft-core processor configuration can be additionally tailored for specific application. Required peripheral devices are connected via general purpose bus. The embedded system must have sufficient memory to contain an operating system with a server application. We decided to use the open source operating system. uClinux.

III. HARDWARE PLATFORM

The designed embedded system is based on Xilinx ML401 development board with Virtex 4 [8] shown in Fig. 1. A Microblaze embedded processor is used [9]. The Microblaze soft processor core is a reduced instruction set computer (RISC) optimized for implementation in Xilinx FPGAs. The core is highly configurable, allowing one to select a specific set of features required by your design. Fixed processor features are:

- thirty-two 32-bit general purpose registers,
- 32-bit instruction word with three operands and two addressing modes,
- 32-bit address bus,
- single issue pipeline.

Figure 1: Xilinx ML401 development board

The processor is adjusted and implemented in the FPGA. The peripherals are connected to the processor via IBM OPB (Open Peripheral Bus) [10] The embedded system architecture is depicted in Fig 2. An external dynamic RAM of the size 64 MB is used as main memory. The internet connection is established by an on board Ethernet controller. Platform flash is used as a non-volatile memory to program the FPGA and boot the server on power up. For the purpose of outside interface the server uses two digital expansion connectors with 64 digital input/output signals. It also has some human interface devices such as LCD display, buttons, LEDs, VGA output and a sound card.

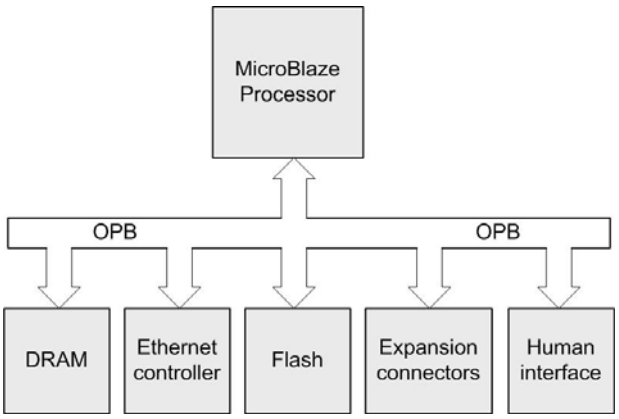

Figure 2: Hardware connection inside Xilinx ML401 development board

IV. SOFTWARE DESIGN

The web server runs on uClinux (microcontroller Linux version) operating system [11]. The uClinux is a port of Linux to systems without a Memory Management Unit (MMU) and therefore designed for dedicated small computer systems like microcontrollers and small microprocessors which are suitable for implementation in FPGA. Today's uClinux releases are based on current versions of linux kernel (2.6, 2.4 as well as 2.2) and include matching tool-chains and collection of different libraries and applications.

uClinux kernel has to be adjusted to the target system and compiled. Drivers are selected and some custom made drivers are added as shown in Fig. 3.

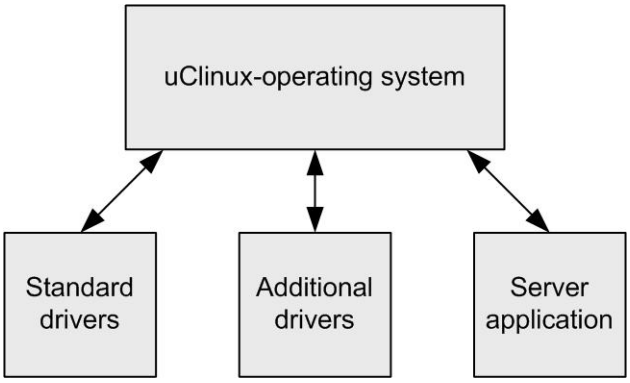

Figure 3: Software parts

Embedded web server application was programmed in Linux environment. The differences between our embedded web server and web servers like *Apache* are limited configuration settings and limited number of simultaneous connections. Major general purpose servers immediately spawn a dedicated process for every incoming request of a client. The process then continues the communication with the client. This requires a lot of memory resources during run time and of course a MMU. Many web browsers can access such a web server simultaneously. On the other hand, embedded web server works only with one single process. If two users need to get access to an embedded web server simultaneously, one of them has to wait until the other finishes. This may be justifiable in specific applications of embedded systems (i.e., remote maintenance, remote configuration, etc.) where not many simultaneous requests are expected.

V. WEB SERVER PROGRAM

This simple web server provides access to an embedded system via web browser (see [12], [13]). The server delivers the desired HTML pages and pictures over the internet or a local intranet network to the web browser. The web content is built by individual files. The base is built by static files with HTML pages. Within such HTML files there are references to other embedded files. These files are typically pictures in GIF or JPEG format and are also transmitted to the web clients. The communication protocol is HTTP described in RFC 1945 [14] and RFC 2616 [15].

A web server in a simplest form can be viewed as a special kind of a file server. Its operation can be summarized as follows:

- A web browser requests a file from a web server by issuing the HTTP GET request.

- The web server receives the HTTP GET request and accesses its file system for the requested file.
- The web server generates the HTTP response which is comprised of a http header and the requested file and transmits it to the browser.
- If the requested file is not accessible a special error response is generated and transmitted to the browser.

The process of client-server file transfer is illustrated in Fig. 4.

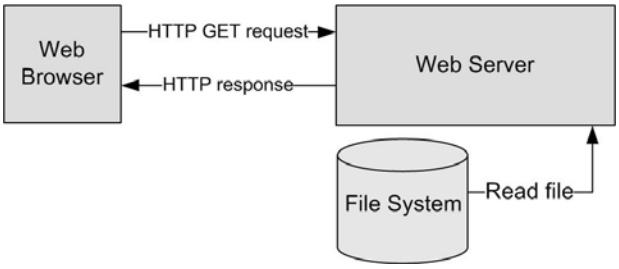

Figure 4: Client-server file transfer

Apart from transferring HTML and other files, this server application can communicate with the input/output devices (different kinds of sensors and actuators, LCD display, buttons, LEDs, VGA display) of the embedded system:

- The web browser request comprises of a command and parameters.
- The server recognizes the command and writes on the requested input/output device or reads from it and transmits the data in a form of HTML page to the web browser.

The process of client-server communication and server access to the file system and input/output devices is illustrated in Fig. 5.

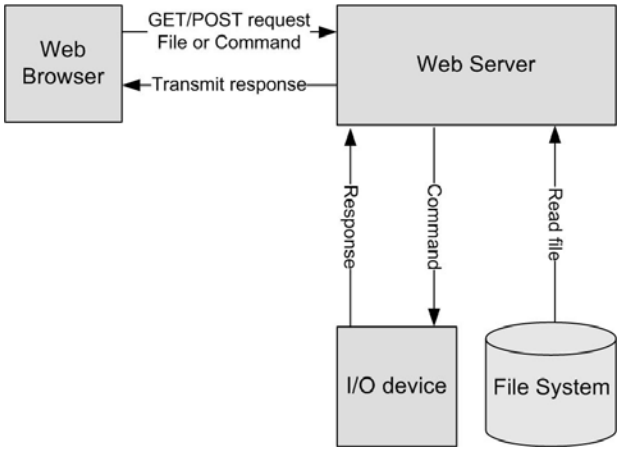

Figure 5: Client-server communication

VI. IMPLEMENTATION ISSUES

The web server was implemented on Virtex 4 FPGA board. A Xilinx Embedded Development Kit (EDK) was used to select and construct the required hardware. The Microblaze processor was adjusted to minimize FPGA resources and peripherals were connected via OPB bus. The hardware drivers of peripherals were selected from Xilinx IP cores. The uClinux libraries were added to the EDK to include the uClinux operating system in the software platform settings. Then the hardware was compiled and downloaded into the FPGA device. Occupied space and FPGA resources are shown in Tab. 1.

Table I
Implementation facts

	Used capacity	Whole capacity	Percent of capacity
Slices	7,112	10,752	66 %
Lookup tables	10,014	21,504	46 %
Flip Flops	6,382	21,504	29 %
I/O signals	176	448	39 %
BRAM	54	72	75 %
DRAM	7MB	64 MB	11%

The software development of the embedded system is divided into two parts:

- Configuration and compilation of uClinux operating system
- Programming of server application.

As regards operation system, the latest distribution of uClinux was downloaded. The uClinux kernel was adjusted to the given embedded system. The appropriate drivers were selected and custom drivers for LCD display and general purpose input/output devices were developed. The server program was first programmed and debugged on a personal computer and then ported to uClinux. The uClinux operating system and developed web server were packed to a software image. This image is downloaded to an embedded system memory during the boot-up process. The image was built on a personal computer in a Linux environment with a Microblaze toolchain.

To initialize the server at power-up a platform flash was used. On the first location of the platform flash, a hardware configuration was stored along with a small bootloader program. On the second location of the flash, uClinux operating system with the web server were stored. The bootloader program is used to transfer the software from second part of the flash into DRAMs and to boot the uClinux operating system.

VII. FUNCTIONALITY

The developed embedded system has the following hardware specifications:

- Xilinx Virtex 4 XC4VLX25 FPGA.
- Xilinx Microblaze soft core embedded processor.
- 100 MHz computer clock.
- 16 kB data cache and 16 kB instruction cache.
- 64kB block RAM inside FPGA.
- 64MB DDR SDRAM as main memory.
- 10/100/1000 tri-speed Ethernet PHY transceiver
- Xilinx XCF32P Platform Flash configuration storage device.
- Input/output devices.

The web server program has the following features:

- HTTP file transfer,
- Static HTML page hosting,
- up to 100 simultaneous connections,
- custom input/output communication support.

Various input/output devices are connected to the embedded system, such as:

- A/D converter,
- temperature and pressure sensors,
- LCD display ,
- AC97 audio sound card,
- push buttons and position switches,
- LEDs.

VIII. PRACTICAL EXPERIENCE

For initial system test in practice, we developed a simple embedded application. A web page was made and selected input/output devices were remotely accessed. The application includes:

- parallel input/output testing using push buttons and LEDs,
- LCD display to show messages from web,
- A/D converter to interface the analog sensors,
- temperature sensor to measure room temperature,
- pressure sensor to measure air pressure,
- AC97 audio sound card to play and record sound.

IX. CONCLUSION

We developed an embedded system web server that provides a simple network interface for custom FPGA embedded designs. The system is based on the Microblaze soft-core processor and the application runs on the uClinux operating system.

In the development of the embedded web server we dealt with the following issues:

1. definition of user requirement specification,
2. selection of software environment dilemma,
3. selection of supporting hardware platform and availability of associated cores,
4. provision of development tools,
5. programming of target application.

Limited space does not allow us to describe individual issues in details. Nevertheless, the described solution may be helpful to potential designers facing similar problems in practice. In addition, given references may broaden the perspective and suggest possible alternatives.

REFERENCES

[1] Tammy Noergaard, *Embedded Systems Architecture*, Elsevier inc., 2005.
[2] James Kurien,Xsenofon Koutsoukos, Feng Zhao, *Distributed Diagnosis of Networked, Embedded Systems*, proceedings of the 13th International Workshop on Principles of Diagnosis, 2002.
[3] http://www.apache.org/ (access date 10.1.2008).
[4] http://www.aolserver.com/ (access date 10.1.2008).
[5] http://www.roxen.com/products/cms/webserver/ (access date 10.1.2008).
[6] http://www.microsoft.com/windowsserver2003/iis/default.mspx/ (access date 10.1.2008).
[7] http://www.sun.com/software/products/web_srvr/home_web_srvr.xml/ (access date 10.1.2008).
[8] Xilinx, *ML401/ML402/ML403 Evaluation Platform User Guide*, version 2.5, 2006.
[9] Xilinx, *MicroBlaze Processor Reference Guide UG081*, version 8.0, 2007.
[10] IBM, *On-Chip Peripheral Bus*, Architecture Specifications, version 2.1, 2001.
[11] Xilinx, *Getting Started with uClinux on MicroBlaze Processor*, version 1.2, 2007.
[12] Xilinx, *Embedded System Example, XAPP433*, version 2.2, 2006.
[13] W. Richard Stevens, *UNIX Network Programming*, Prentice Hall, London, 1990.
[14] Berners-Lee, T., Fielding, R. and H. Frystyk, *Hypertext Transfer Protocol-HTTP/1.0*, RFC 1945, 1996.
[15] Berners-Lee, T., Fielding, R. and H. Frystyk, *Hypertext Transfer Protocol-HTTP/1.1*, RFC 2616, 1999.