

Implementing a Lightweight Web Server Using PowerPC and Tri-Mode Ethernet MAC in Virtex-4 FX FPGAs

Using Ethernet, you can connect your device to the Internet, monitoring and controlling it remotely.

by Jue Sun
Systems Engineering Intern
Xilinx, Inc.
juebejue@gmail.com

Peter Ryser
Sr. Manager, Systems Engineering
Xilinx, Inc.
peter.ryser@xilinx.com

The Tri-Mode Ethernet MAC (TEMAC) UltraController-II Module (UCM) is a minimal footprint, embedded network processing engine based on the PowerPC™ 405 (PPC405) processor core and the TEMAC core embedded within a Xilinx® Virtex™-4 FX Platform FPGA. It allows you to interact with your Virtex-4-based system through an Ethernet connection and to control or monitor your system remotely using TCP/IP from miles away. Our design uses minimal resources and ensures that you will have enough logic area for your application.

In this article, we will explain the advantage of our design and how it is implemented on an ML403 board. We will also introduce a few applications that you can build on top of this implementation.

Implementation

The TEMAC UCM leverages key innovations in Virtex-4 FPGAs to implement TCP/IP applications with minimal resource usage, as shown in Figure 1. The whole implementation takes one embedded PPC405, one integrated TEMAC, two Virtex-4 FIFO16s, 20 slice flip-flops, and 18 look-up tables (LUTs).

On the ML403 board, the TEMAC UCM connects to an external PHY through a gigabit media independent interface (GMII) and a management data input/output (MDIO) interface, and auto-negotiates tri-mode (10/100/1000 Mbps) Ethernet speeds. Other physical interfaces like RGMII and SGMII are possible and take a minimum amount of change in the reference design that is provided as source code.

The software is ported from the open-source μ IP TCP/IP stack and runs completely out of the PPC405's 16 KB instruction and 16 KB data cache. It accesses Ethernet frames in two FIFO16s through the PPC405 on-chip memory (OCM) interface. One FIFO buffers inbound while the other buffers outbound Ethernet frames. The FIFOs also act as entities to synchronize clock domains between the PPC405 OCM and the TEMAC. The PPC405 is capable of running the software at maximum frequency, for example, 350 MHz in a -10 speed grade FPGA.

The application implemented in Xilinx Application Note XAPP807, "Minimal Footprint Tri-Mode Ethernet MAC Processing Engine," (www.xilinx.com/bvdocs/appnotes/xapp807.pdf) runs a Web server on top of the μ IP TCP/IP stack. The Web server serves a Web form to connecting clients. You simply enter a string in a text field and submit it to the server, which interprets the data and displays the string on a two-line character LCD. See Figure 2 for a picture of the process.

The μ IP TCP/IP stack has a well-defined API and comes with other demonstration applications like telnet. Based on the well-written μ IP documentation, you can implement your own application without much effort. The stack is optimized for minimal footprint at the cost of performance. The restriction on TCP/IP perform-

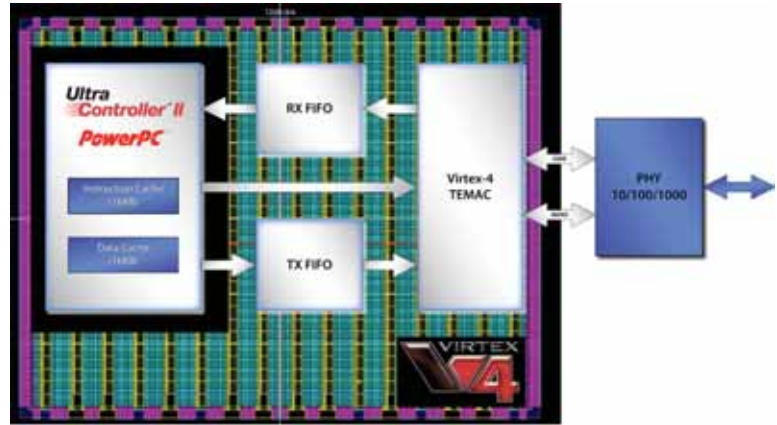


Figure 1 – General datapath of TEMAC UCM



Figure 2 – Example Web server allowing remote control of ML403

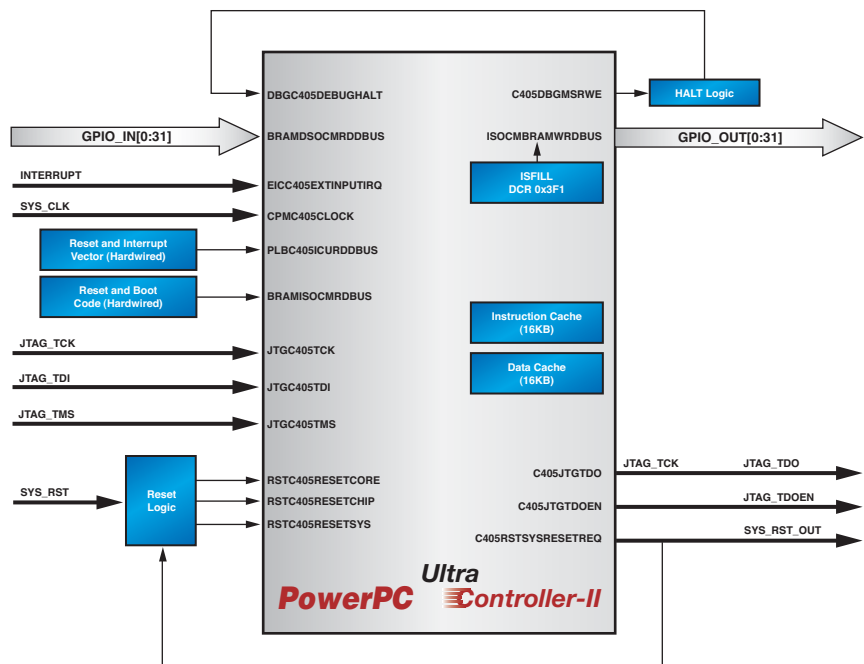


Figure 3 – UltraController-II ports

Other configuration solutions besides System ACE CF can load code and data into the PPC405 caches. Such solutions include Xilinx Platform Flash as well as external processors and methods...

ance is not a limiting issue for most monitoring and control applications.

Implementation Flow

Implementing the TEMAC UCM is straightforward. The implementation flow comprises three main parts: one flow in Project Navigator for hardware bit file generation; one flow in EDK for software ELF file generation; and another flow to combine the software and hardware file into a single bitstream or PROM file to program the Virtex-4 FX12 FPGA or the Platform Flash on the ML403 board, respectively. Project files and scripts for all of these flows are included in XAPP807. The hardware source code for the TEMAC UCM is available in Verilog and VHDL. The software source code is written in C.

The TEMAC UCM is based on UltraController-II, as shown in Figure 3 and documented in XAPP575, "UltraController-II: Minimal Footprint Embedded Processing Engine" (www.xilinx.com/bvdocs/appnotes/xapp575.pdf). UltraController-II is a black-box processing engine that includes 32 bits of user-defined general-purpose input and output, as well as interrupt handling capability.

Loading the Software into Caches

Loading software into the instruction and data caches of the PowerPC has been possible for some time through System ACE™ CF technology or other JTAG-based methods. However, for the first time, the TEMAC UCM makes this also possible through all configuration modes, including JTAG mode, slave and master serial modes, and slave and master SelectMap modes. Other configuration solutions besides System ACE CF can load code and data into the PPC405 caches. Such solutions include Xilinx Platform Flash as well as external processors and methods described in application notes such as XAPP058, "Xilinx In-System Programming Using an

Embedded Microcontroller" (www.xilinx.com/bvdocs/appnotes/xapp058.pdf).

The cache loading solution emphasizes another new feature in the Virtex-4 FPGA family. The USER_ACCESS register is a 32-bit register that provides a port from the configuration block into the FPGA fabric. For loading the caches, the USER_ACCESS register is connected through a small state machine to the JTAG port of the PPC405. A script converts a software ELF file into a bitstream that you can load into the processor caches with Impact or the ChipScope™ Analyzer. If you are interested in finding out more about this solution, Xilinx Application Note XAPP719, "PowerPC Cache Configuration Using the USER_ACCESS_VIRTEX4 Register," (www.xilinx.com/bvdocs/appnotes/xapp719.pdf) provides full details.

Use Cases for the TEMAC UCM

Figure 4 shows some use cases for the TEMAC UCM. Besides monitoring and control application, other applications include statistics gathering, system diag-

nostics, display control, or math and data manipulation operations.

A system application that deals with MPEG transport streams gathers statistical information about the video and audio streams. The TEMAC UCM makes this information available through the Web interface. On the control side, the content provider dynamically loads decryption keys for encrypted streams into the system through the same Web interface.

In another example, an FPGA controlled machine gathers information about erroneous conditions that the machine manufacturer accesses locally or remotely for diagnosis. In return, the manufacturer loads new machine parameters into the system to adjust undesired behavior.

In a third example, some networking equipment reports its operational status on a regular basis through e-mail to a control center where the information is processed. The control center equipment or personnel takes action if the equipment reports a malfunction or fails to report at all.

In all of these cases, the TEMAC UCM fits into a design for which it was

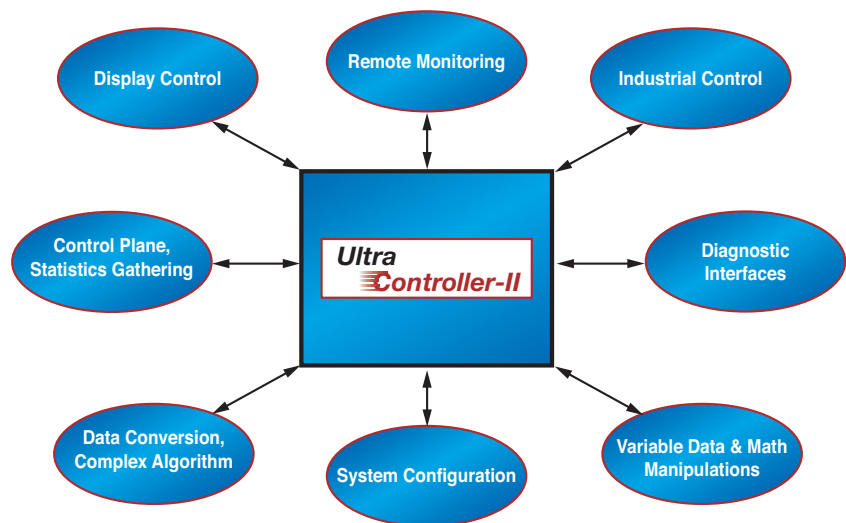


Figure 4 – UltraController-II's possible applications

not originally designed. Its small hardware and software footprint makes it suitable for designs that already use all of the available FPGA resources.

Network-Attached Co-Processor

In a recent experiment, we combined the TEMAC UCM with the auxiliary processor unit (APU) on the PPC405. The APU provides a direct way to access hardware accelerators by the way of user-defined

instructions (UDI). Typically, a software application running on the PPC405 uses the APU to accelerate the execution speed. In this case, however, the application software runs on a regular PC that distributes the workload to multiple ML403 boards with integrated TEMAC UCM and APU, as shown in Figure 5. The ML403 acts as a network-attached co-processor, speeding up the application running on the PC.

For example, we compute pictures gener-

ated by the Mandelbrot function. The software running on the PC divides the whole picture into smaller areas and distributes the parameters for these smaller areas to the ML403 boards through TCP connections. The PPC405 on each ML403 board reads the parameters from the TCP socket, loops through the area, and calculates every pixel using a MandelPoint co-processor connected to the APU interface. The PPC405 sends the results of this calculation back to the PC through the same TCP connection. Finally, the PC collects all the results from the network-attached co-processors and displays the final picture on the screen.

Figure 6 shows the result of the acceleration. A single ML403 connected to the PC computes a 1024 x 768 picture in about three seconds, including communication overhead. Adding more ML403 boards brings the overall computation time down below 1.5 seconds for three boards. Four boards do not show a significant improvement, as communication overhead becomes the main part of the computation.

Solving the same problem natively on a high-end Linux workstation takes about three seconds, about the same amount of time as one network-attached ML403 with TEMAC UCM and APU acceleration.

Conclusion

The TEMAC UCM provides a convenient and inexpensive way of adding Ethernet functionality to any design. It uses a minimal amount of FPGA resources and the software runs from the embedded PPC405 caches. As a result of these simple interfaces, development time is reduced. The Web server application is used to demonstrate the design, but you can use it for other solutions, as shown with the example of the network-attached co-processors.

For more information about the TEMAC UltraController Module, download XAPP807, including the reference design for the Xilinx ML403 board, and see the Xilinx Webcast, "Learn about the UltraController-II Reference Design with PowerPC and Tri-Mode EMAC in Virtex-4 FPGAs," at www.xilinx.com/events/webcasts/110105_ultracontroller2.htm.

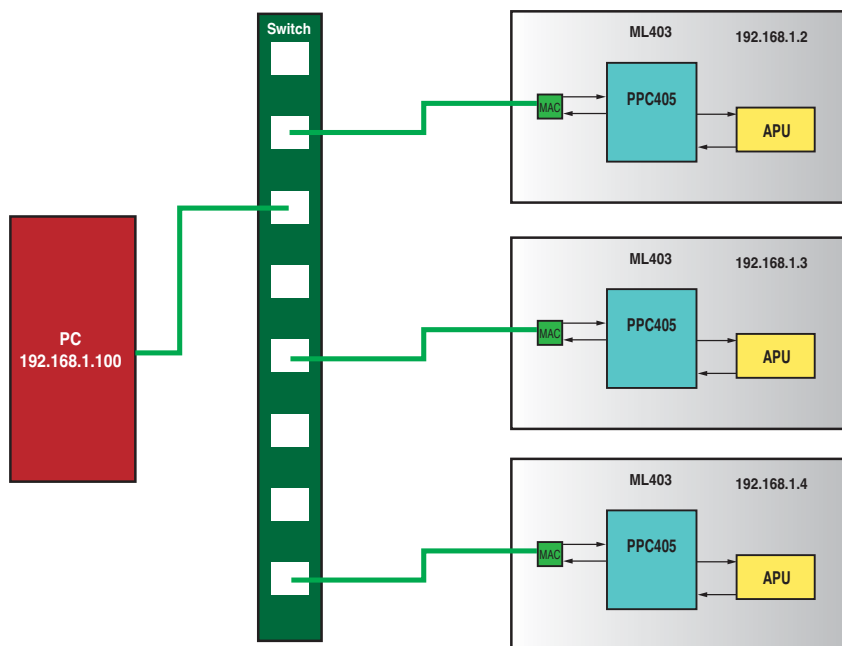


Figure 5 – Several ML403s with APU activated to accelerate computation

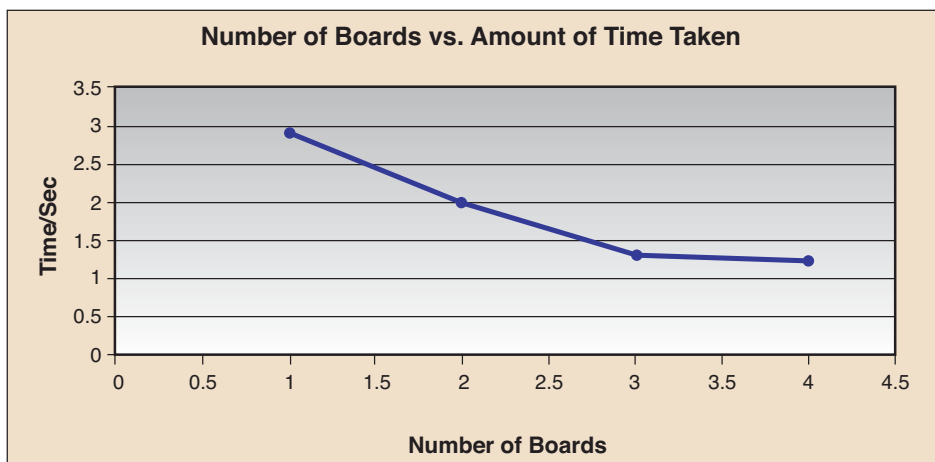


Figure 6 – Use multiple boards to shorten computation time