Speed up Verification of Long Transaction Sequences with MicroBlaze Soft Processors

Embedding soft processors and peripherals inside Xilinx Virtex-II Platform FPGAs has created a new set of challenges for designers. Here's one solution.

by Warren Miller
VP of Marketing, Avnet Design Services
warren.miller@avnet.com

Michael Scurry
Chief Engineer, Experience First
mscurry@expfirst.com

Ravi Pragasam
Marketing Manager, Processor Solutions
ravi.pragasam@xilinx.com

The Xilinx MicroBlaze™ soft processor core and the IBM CoreConnect™ bus architecture that the MicroBlaze processor uses can provide the hardware horsepower for you to implement an entire Internet appliance in a single Xilinx Platform FPGA.

The ability to run long sequences of processor code inside Virtex-II FPGAs, however, has made it more difficult to verify functionality and to test designs that connect to the processor within the FPGA.

Verification of a complex start-up initialization sequence or an Internet transaction – perhaps to download code to update the processor code or even to reconfigure the FPGA – would just take too long to verify in software. A software-only verification scheme could take hours or maybe even days to run, and it would involve hundreds of thousands of system cycles to accomplish verification.

In this article, we will show you an example of how you can use hardware techniques to augment software verification. This example reduces the time required to perform multiple test and modification cycles to verify a system reconfiguration from initialization and set-up to an actual transaction over the Internet.

MicroBlaze Applications

The ability to connect all sorts of electronics equipment to the Internet continues to create new applications and business models. With the availability of the MicroBlaze soft processor targeted to the Virtex-II family of Platform FPGAs, you can now include Xilinx Internet Reconfigurable Logic (IRL™) connectivity in equipment like vending machines, motors, heating and cooling equipment, industrial process controllers, and remote monitoring and communications appliances. This technology allows already deployed devices and appliances to be remotely reprogrammed, reconfigured, or upgraded without a visit from a technician.

One low-cost approach to implementing this type of equipment is to use a simple communications protocol to access the Internet. This complete device-networking solution requires only a processor with 4 Kbytes of memory and a UART to connect the embedded application to a gateway, and thus, to the Internet. In this application example, we will use a Virtex-II FPGA with an embedded MicroBlaze core, UART, and embedded memory (BRAM and distributed RAM) to provide all the resources required to connect to the Internet.

The rest of the Virtex-II FPGA, which has upwards of hundreds of thousands logic gates, is available for controlling the rest of the application. For instance, a remote control system for a modular factory floor could be upgraded and changed by reprogramming the FPGA over the Internet. An architectural level diagram of such an application is shown in Figure 1.

The MicroBlaze core can be configured in one of six combinations of busses, each with a configuration providing a different combination of performance and functionality. In particular, the amount of memory required by the application will determine which of the bus combinations is required. In our example application, we're assuming that all the instruction memory is located on-chip and that data memory may be required off-chip, perhaps as memory-mapped I/O.

The main on-chip peripherals needed for this application are a UART, a timer/counter, and an interrupt controller. The UART is a critical component in implementing the Internet protocol application. The UART is attached to the CoreConnect on-chip peripheral bus (OPB) and provides the serial communication link to the outside world. Once the UART is parameterized, you can select the base address for the internal registers, the number of data bits per character, and the type (if any) of parity supported. The registers accessible in the UART from the OPB are the read data register, write data register, read status register, and write status register. The UART must be initialized on startup to select the appropriate speed and data format. Once initial-

ized, it can be used to provide connectivity for the networking protocol.

MIcroBlaze Transaction Sequence

Now that we've set the stage with a description of the application and key system components, it's time to delve into an example problem that you might face when creating an embedded processor design. A good example would be to determine the length of time it would take for the system to reconfigure, initialize, and transact the first series of packets to establish communication over the Internet. If changes must be made in the code or in the Virtex-II FPGA design, it's necessary to establish timing for each of the main transaction sequences over the OPB.

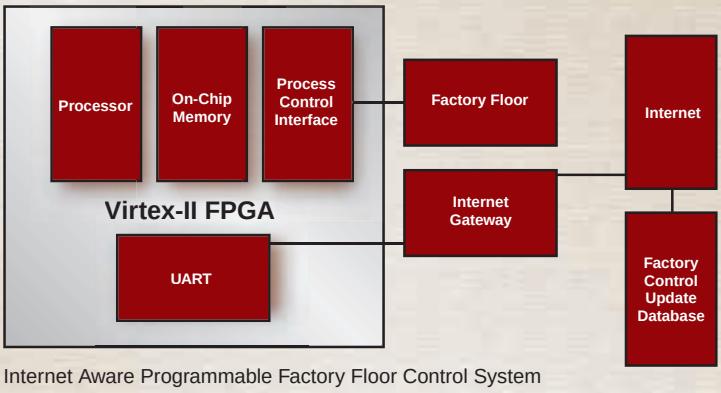

Figure 1 - Application block diagram

ChipScope Integrated Logic Analyzer

One method of timing the transactions would be to use the on-chip resources of the Xilinx ChipScope™ Integrated Logic Analyzer (ILA) to observe bus traffic in an FPGA on a prototype or development board. ChipScope ILA embeds logic analyzer cores into your design. These logic cores allow you to view all the internal signals and nodes within an FPGA. ChipScope ILA supports user selectable data channels from 1 to 256. The depth of the sample buffer ranges from 512 to 16,384 words in Virtex-II devices.

In our example, event triggers are changeable in real-time without affecting the logic or requiring recompilation of the design. ChipScope can capture OPB transactions and store them in on-chip memory. Bus transactions can be monitored, and with the trigger capability, specific transactions can be used to begin a capture event.

For instance, a specific address can be used to begin capture when a UART register is read or written. A status register read could indicate the beginning or end of a transaction sequence. You can use an ILA feature that counts the cycles between trigger events to determine the amount of time required for specific transactions – which is just what we need for our example application.

The ChipScope ILA provides the capability to observe signals on the OPB, but it does not have the ability to drive internal signals. Additionally, ChipScope ILA uses internal memory blocks to capture data on the OPB. If the application requires the use of internal memory, however, there may not be any memory left for use by ChipScope. In such a case, the ability to use off-chip memory would help solve this problem and provide an integrated software and firmware tool environment to accelerate the development, integration, and test of MicroBlaze-based Virtex-II FPGA designs.

The Raptor Solution

The Raptor Development Kit from Avnet automatically inserts logic around the MicroBlaze core to route inputs and outputs to real-time input signals, output signals, and buffer memory available on the development board. This allows the core to be exercised at "hardware speeds." Core output signals that are stored in the buffer memory can be read out over the USB port to a host computer and displayed on the waveform viewer, just as if the core output signals were software simulation results.

This "hardware speed" approach to verification reduces the design/debug cycle time dramatically. The Raptor Kit makes it easy to perform design or test set-up changes – and see the results immediately. Used in conjunction with a hardware-based input stimulus, verification of very complex and robust test suites are orders of magnitude faster than pure software simulation-based approaches.

The Raptor Kit provides the input signals from the UART to simulate the start-up transactions. The entire transaction sequence can be run at hardware speeds (25 MHz for our example design). The transactions on the OPB can be observed, and detailed measurements can be made, to pinpoint the key time delays for each portion of the start-up, initialization, and UART communications code phase. Once the code is optimized to remove unnecessary waits and loops, a total of only 236,000 clock cycles are required for initialization and first UART transaction with the gateway. The design now has enough time to reconfigure the FPGA without overflowing the FIFO at the application layer of the protocol.

Conclusion

Implementing embedded soft processors like the MicroBlaze core on Xilinx Virtex-II FPGAs presents designers with a new set of verification challenges. For long transaction cycles on the on-chip peripheral bus (which is part of the IBM CoreConnect architecture specification), software simulation alone may not be the best answer when time is of the essence. If observation of transactions is required, the ChipScope ILA capability of the Virtex-II FPGAs can provide a hardware-based assist that is much faster than software-only techniques. If observation and stimulus, on-chip or at the system level, is required, a MicroBlaze Development System like Raptor offers a hardware-based solution that you need to complete system verification on schedule.

For more information on the Raptor Development Environment, contact Bob Read at Experience First at (408) 985-9683 or *bread@expfirst.com*.